To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas 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 Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “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.

   “High Quality”: 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; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

RENESAS

User's Manual

μPD780862 Subseries

8-Bit Single-Chip Microcontrollers

μPD780861
μPD780862
μPD78F0862
μPD78F0862A
μPD780861(A)
μPD780862(A)
μPD78F0862(A)
μPD78F0862A(A)
μPD780861(A1)
μPD780862(A1)
μPD78F0862A(A1)
μPD780861(A2)
μPD780862(A2)
μPD78F0862A(A2)

Document No. U16418EJ3V0UD00 (3rd edition)
Date Published July 2006 NS 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.

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT is a trademark of International Business Machines Corporation.
HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Company.
SPARCstation is a trademark of SPARC International, Inc.
Solaris and SunOS are trademarks of Sun Microsystems, Inc.
SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

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

- **The information in this document is current as of December, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. 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 product 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).

M8E 02. 11-1

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD780862 Subseries and design and develop application systems and programs for these devices.
The target products are as follows.

μPD780862 Subseries: μPD780861, 780862, 78F0862, 78F0862A, 780861(A), 780862(A), 78F0862(A), 78F0862A(A), 780861(A1), 780862(A1), 78F0862A(A1), 780861(A2) , 780862(A2), 78F0862A(A2)

Purpose

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

Organization

The μPD780862 Subseries manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD780862 Subseries User's Manual (This Manual)	**78K/0 Series Instructions User's Manual**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- When using this manual as the manual for (A) grade, (A1) grade, and (A2) grade products:
  → Only the quality grade differs between standard products and (A) grade, (A1) grade, and (A2) grade products. Read the part number as follows.
  - μPD780861 → μPD780861(A), 780861(A1), 780861(A2)
  - μPD780862 → μPD780862(A), 780862(A1), 780862(A2)
  - μPD78F0862 → μPD78F0862(A)
  - μPD78F0862A → μPD78F0862A(A), 78F0862A(A1), 78F0862A(A2)
- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.

- 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)**.

Caution Examples in this manual employ the "standard" quality grade for general electronics. When using examples in this manual for the "special" quality grade, review the quality grade of each part and/or circuit actually used.

Conventions

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

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

Documents Related to Devices

Document Name	Document No.
μPD780862 Subseries User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E

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

Document Name		Document No.
RA78K0 Ver.3.80 Assembler Package	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
CC78K0 Ver.3.70 C Compiler	Operation	U17201E
	Language	U17200E
SM+ System Simulator	Operation	U17246E
	User Open Interface	U17247E
ID78K0-QB Ver.2.90 Integrated Debugger	Operation	U17437E
PM plus Ver.5.20		U16934E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when 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-780862-NS-EM1 Emulation Board	U16810E

Documents Related to Flash Memory Programming

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

Other 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 when designing.

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.2 μs: @ 10 MHz operation with high-speed system clock) to low speed (3.2 μs: @ 10 MHz operation with high-speed system clock)
- General-purpose registers: 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)
μPD780861	Mask ROM	8 KB	512 bytes
μPD780862		16 KB	768 bytes
μPD78F0862, 78F0862A Note 1	Flash memory	16 KB Note 2	

<R> **Notes 1.** μPD78F0862 and 78F0862A differ only in the characteristics of a flash memory. For details, refer to "Flash Memory Programming Characteristics" in the chapter of electrical specifications.

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

- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- Short startup is possible via the CPU default start using the internal low-speed oscillator
- On-chip clock monitor function using the internal low-speed oscillator
- On-chip watchdog timer (operable with low-speed oscillation clock)
- I/O ports: 14
- Timer: 5 channels
- Serial interface
  UART (LIN (Local Interconnect Network)-bus supported): 1 channel
  CSI1: 1 channel
- On-chip Manchester code generator
- 10-bit resolution A/D converter: 4 channels
- Supply voltage: V_{DD} = 2.7 to 5.5 V Note 1
- Operating ambient temperature: T_A = –40 to +85°C (standard products, (A) grade products) Note 2
  T_A = –40 to +110°C ((A1) grade products)
  T_A = –40 to +125°C ((A2) grade products)

Notes 1. Use the product in a voltage range of 3.0 to 5.5 V because the detection voltage (V_{POC}) of the power-on-clear (POC) circuit is 2.85 V ±0.15 V.

<R> **2.** Only the standard product and (A) grade product are available in μ PD78F0862.

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

(1) Mask ROM versions

	Part Number	Package	Quality Grade
	μPD780861MC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Standard
<R>	μPD780861MC-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Standard
	μPD780862MC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Standard
<R>	μPD780862MC-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Standard
	μPD780861MC(A)-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD780861MC(A)-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
	μPD780862MC(A)-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD780862MC(A)-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
	μPD780861MC(A1)-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD780861MC(A1)-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
	μPD780862MC(A1)-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD780862MC(A1)-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
	μPD780861MC(A2)-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD780861MC(A2)-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
	μPD780862MC(A2)-×××-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD780862MC(A2)-×××-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special

Remarks 1. ××× indicates ROM code suffix.

2. Products with -A at the end of the part number are lead-free products.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

(2) Flash memory versions

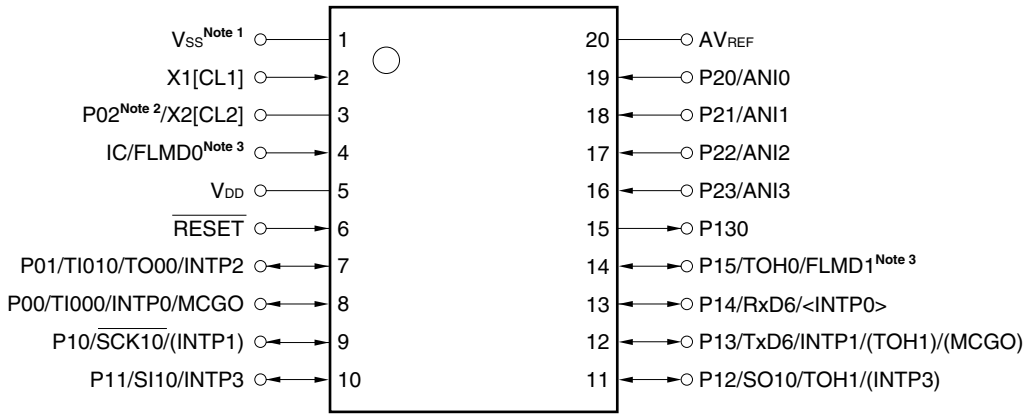
	Part Number	Package	Quality Grade
	μPD78F0862MC-5A4	20-pin plastic SSOP (7.62 mm (300))	Standard
<R>	μPD78F0862MC-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Standard
<R>	μPD78F0862AMC-5A4	20-pin plastic SSOP (7.62 mm (300))	Standard
<R>	μPD78F0862AMC-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Standard
	μPD78F0862MC(A)-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD78F0862MC(A)-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD78F0862AMC(A)-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD78F0862AMC(A)-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD78F0862AMC(A1)-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD78F0862AMC(A1)-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD78F0862AMC(A2)-5A4	20-pin plastic SSOP (7.62 mm (300))	Special
<R>	μPD78F0862AMC(A2)-5A4-A	20-pin plastic SSOP (7.62 mm (300))	Special

Remark Products with -A at the end of the part number are lead-free products.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

1.4 Pin Configuration (Top View)

- 20-pin plastic SSOP (7.62 mm (300))



Notes 1. VSS and AVSS are internally connected in the μPD780862 Subseries. Be sure to connect VSS to a stabilized GND (= 0 V).

2. When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

3. FLMD0 and FLMD1 are available only in the μPD78F0862 and 78F0862A.

Cautions 1. Connect the IC (Internally Connected) pin directly to VSS.

2. Connect the AVREF pin to VDD.

Remarks 1. Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

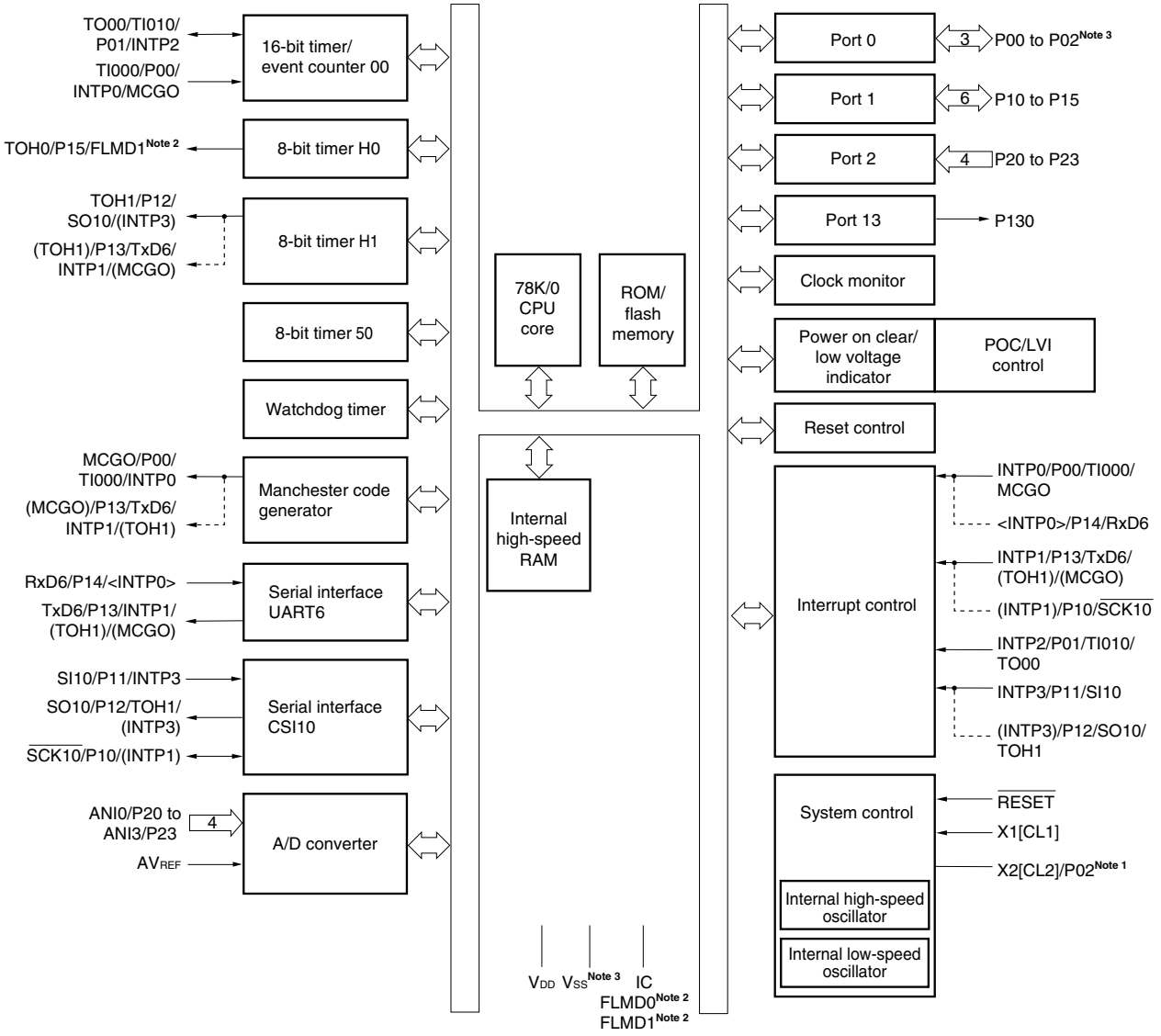
2. Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

3. Items in brackets [ ] are pin names when using external RC oscillation.

Pin Identification

ANI0 to ANI3:	Analog input
AV_{REF}:	Analog reference voltage
CL1, CL2:	RC oscillator
FLMD0, FLMD1:	Flash programming mode
IC:	Internally connected
INTP0 to INTP3:	External interrupt input
MCGO:	Manchester code output
P00 to P02:	Port 0
P10 to P15:	Port 1
P20 to P23:	Port 2
P130:	Port 13
$\overline{\text{RESET}}$:	Reset
RxD6:	Receive data
$\overline{\text{SCK10}}$:	Serial clock input/output
SI10:	Serial data input
SO10:	Serial data output
TI000, TI010:	Timer input
TO00, TOH0, TOH1:	Timer output
TxD6:	Transmit data
V_{DD}:	Power supply
V_{SS}:	Ground
X1, X2:	Crystal oscillator (X1 input clock)

1.5 Block Diagram



Notes 1. When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

2. FLMD0 and FLMD1 are available only in the μPD78F0862 and 78F0862A.

3. V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

Remarks 1. Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

2. Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

3. Items in brackets [ ] are pin names when using external RC oscillation.

1.6 Outline of Functions

(1/2)

Item		μPD780861	μPD780862	μPD78F0862, 78F0862A[Note 1]
Internal memory	ROM	8 KB	16 KB	16 KB (flash memory)
	High-speed RAM	512 bytes	768 bytes	
Memory space		64 KB		
<R> High-speed system clock (oscillation frequency)	Standard products, (A) grade products[Note 2]	• Ceramic/crystal/external clock oscillation (2 to 10 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) • External RC/external clock oscillation (3 to 4 MHz: V_{DD} = 2.7 to 5.5 V) • Internal high-speed oscillation (8 MHz (TYP.): V_{DD} = 4.0 to 5.5 V)		
	(A1) grade products	• Ceramic/crystal/external clock oscillation (2 to 10 MHz: V_{DD} = 4.0 to 5.5 V, 2 to 5 MHz: V_{DD} = 2.7 to 5.5 V) • External RC/external clock oscillation (3 to 4 MHz: V_{DD} = 2.7 to 5.5 V) • Internal high-speed oscillation (8 MHz (TYP.): V_{DD} = 4.0 to 5.5 V)		
	(A2) grade products	• Ceramic/crystal/external clock oscillation (2 to 9.2 MHz: V_{DD} = 4.0 to 5.5 V, 2 to 5 MHz: V_{DD} = 2.7 to 5.5 V) • External RC/external clock oscillation (3 to 4 MHz: V_{DD} = 2.7 to 5.5 V) • Internal high-speed oscillation (8 MHz (TYP.): V_{DD} = 4.0 to 5.5 V)		
Internal low-speed oscillation clock (oscillation frequency)		• Internal low-speed oscillation (240 kHz (TYP.): V_{DD} = 2.7 to 5.5 V)		
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time		0.2 μs/0.4 μs/0.8 μs/1.6 μs/3.2 μs (high-speed system clock: @ f_{XH} = 10 MHz operation)		
		8.3 μs/16.7 μs (TYP.) (internal low-speed oscillation clock: @ f_R = 240 kHz (TYP.) 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: 14 CMOS I/O 8 CMOS input 5 CMOS output 1		
Timers		• 16-bit timer/event counter: 1 channel • 8-bit timer: 3 channels • Watchdog timer: 1 channel		
A/D converter		10-bit resolution × 4 channels		

<R> **Notes 1.** μPD78F0862 and μPD78F0862A differ only in the characteristics of a flash memory. For details, refer to "Flash Memory Programming Characteristics" in the chapter of electrical specifications.

<R> **2.** Only the standard product and (A) grade product are available in μPD78F0862.

(2/2)

Item		μPD780861	μPD780862	μPD78F0862, 78F0862A[Note 1]
Serial interface		• UART mode supporting LIN-bus: 1 channel • 3-wire serial I/O mode: 1 channel		
Manchester code generator		1 channel		
Vectored interrupt sources	Internal	12		
	External	4		
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		
Supply voltage		V_{DD} = 2.7 to 5.5 V[Note 2]		
Operating ambient temperature		Standard products, (A) grade products[Note 3]: T_A = −40 to +85°C (A1) grade products: T_A = −40 to +110°C (A2) grade products: T_A = −40 to +125°C		
Package		20-pin plastic SSOP (7.62 mm (300))		

<R> **Notes 1.** μPD78F0862 and μPD78F0862A differ only in the characteristics of a flash memory. For details, refer to "Flash Memory Programming Characteristics" in the chapter of electrical specifications.

2. Use the product in a voltage range of 3.0 to 5.5 V because the detection voltage (V_{POC}) of the power-on-clear (POC) circuit is 2.85 V ±0.15 V.

<R> **3.** Only the standard product and (A) grade product are available in μPD78F0862.

An outline of the timer is shown below.

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

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are two types of pin I/O buffer power supplies: AV_{REF} 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 P23
V_{DD}	Pins other than P20 to P23

(1) Port pins

Pin Name	I/O	Function		After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port.	Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI000/INTP0/MCGO
P01					TI010/TO00/INTP2
P02[Note 1]	Input		Input-only	Input	X2[CL2]
P10	I/O	Port 1. 6-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	$\overline{SCK10}$/(INTP1)
P11					SI10/INTP3
P12					SO10/TOH1/(INTP3)
P13					TxD6/INTP1/(TOH1)/(MCGO)
P14					RxD6/<INTP0>
P15					TOH0/FLMD1[Note 2]
P20 to P23	Input	Port 2. 4-bit input-only port.		Input	ANI0 to ANI3
P130	Output	Port 13. 1-bit output-only port.		Output	–

Notes 1. When the internal high-speed oscillation clock is selected as the high-speed system clock, this pin can be used as a port input pin.

2. FLMD1 is available only in the μPD78F0862 and 78F0862A.

Remarks 1. Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

2. Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

3. Items in brackets [ ] are pin names when using external RC oscillation.

(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	P00/TI000/MCGO
<INTP0>				P14/RxD6
INTP1				P13/TxD6/(TOH1)/(MCGO)
(INTP1)				P10/$\overline{\text{SCK10}}$
INTP2				P01/TI010/TO00
INTP3				P11/SI10
(INTP3)				P12/SO10/TOH1
SI10	Input	Serial data input to serial interface	Input	P11/INTP3
SO10	Output	Serial data output from serial interface	Input	P12/TOH1/(INTP3)
$\overline{\text{SCK10}}$	I/O	Clock input/output for serial interface	Input	P10/(INTP1)
RxD6	Input	Serial data input to asynchronous serial interface	Input	P14/<INTP0>
TxD6	Output	Serial data output from asynchronous serial interface	Input	P13/INTP1/(TOH1)/(MCGO)
MCGO	Output	Manchester code output	Input	P00/TI000/INTP0
(MCGO)				P13/TxD6/INTP1/(TOH1)
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/INTP0/MCGO
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00/INTP2
TO00	Output	16-bit timer/event counter 00 output	Input	P01/TI010/INTP2
TOH0	Output	8-bit timer H output	Input	P15/FLMD1[Note 1]
TOH1				P12/SO10/(INTP3)
(TOH1)				P13/TxD6/INTP1/(MCGO)
ANI0 to ANI3	Input	A/D converter analog input	Input	P20 to P23
AV_{REF}	Input	A/D converter reference voltage input and positive power supply for port 2	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
X1 [CL1]	Input	Connecting resonator for high-speed system clock [Connecting RC for high-speed system clock]	–	–
X2 [CL2]	–		Input	P02
V_{DD}	–	Positive power supply	–	–
V_{SS}[Note 2]	–	Ground potential	–	–
IC	–	Internally connected. Connect directly to V_{SS}.	–	–
FLMD0[Note 1]	–	Flash memory programming mode lead-in.	–	–
FLMD1[Note 1]				P15/TOH0

Notes 1. FLMD0 and FLMD1 are available only in the μPD78F0862 and 78F0862A.

2. V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

Remarks 1. Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

2. Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

3. Items in brackets [ ] are pin names when using external RC oscillation.

2.2 Description of Pin Functions

2.2.1 P00 to P02 (port 0)

P00 to P02 function as a 3-bit I/O port. These pins also function as external interrupt request input, Manchester code output, timer I/O, and crystal/ceramic resonator connection [RC connection] for high-speed system clock oscillation.

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

Caution When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

(1) Port mode

P00 and P01 function as an I/O port, and P02 functions as an input-only port. P00 and P01 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 P02 function as external interrupt request input, Manchester code output, timer I/O, and crystal/ceramic resonator connection [RC connection] for high-speed system clock oscillation.

(a) INTP0 and INTP2

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

(b) MCGO

This is a Manchester code output pin.

(c) TI000

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

(d) TI010

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

(e) TO00

This is a timer output pin.

(f) X2 [CL2]

This is the pin for crystal/ceramic resonator connection [RC connection] for high-speed system clock oscillation.

2.2.2 P10 to P15 (port 1)

P10 to P15 function as a 6-bit I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, clock I/O, timer output, and flash memory programming mode lead-in.

P10 to P15 can be assigned as external interrupt request input, timer output, and Manchester code output by setting the alternate-function pin switch register (PSEL) and input switch control register (ISC).

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

(1) Port mode

P10 to P15 function as a 6-bit I/O port. P10 to P15 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 P15 function as external interrupt request input, serial interface data I/O, clock I/O, timer output, flash memory programming mode leading-in, and Manchester code output.

(a) SI10

This is a serial data input pin of the serial interface.

(b) SO10

This is a serial data output pin of the serial interface.

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

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

(d) INTP0, INTP1, and INTP3

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

(e) RxD6

This is a serial data input pin of the asynchronous serial interface.

(f) TxD6

This is a serial data output pin of the asynchronous serial interface.

(g) TOH0 and TOH1

These are timer output pins.

(h) MCGO

This is a Manchester code output pin.

(i) FLMD1[Note]

This is a flash memory programming mode lead-in pin.

Note FLMD1 is available only in the μPD78F0862 and 78F0862A.

2.2.3 P20 to P23 (port 2)

P20 to P23 function as a 4-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 P23 function as a 4-bit input-only port.

(2) Control mode

P20 to P23 function as A/D converter analog input pins (ANI0 to ANI3). When using these pins as analog input pins, see **(5) ANI0/P20 to ANI3/P23** in **10.6 Cautions for A/D Converter**.

2.2.4 P130 (port 13)

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

2.2.5 AV_{REF}

This is an A/D converter reference voltage input pin and a positive power supply pin.

When A/D converter is not used, connect this pin directly to V_{DD}.

2.2.6 $\overline{\text{RESET}}$

This is an active-low system reset input pin.

2.2.7 X1 and X2

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

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

Remark When the internal high-speed oscillation clock is selected as the high-speed system clock, the X2 [CL2] pin can be used as a port input pin (P02).

2.2.8 CL1 and CL2

These are the pins for connecting a resistor (R) and capacitor (C) for high-speed system clock.

When supplying an external clock, input a signal to CL1 and input the inverse signal to CL2.

Remark When the internal high-speed oscillation clock is selected as the high-speed system clock, the X2 [CL2] pin can be used as a port input pin (P02).

2.2.9 V_{DD}

This is a positive power supply pin.

2.2.10 V_{SS}

This is a ground potential pin.

Caution V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

2.2.11 FLMD0 and FLMD1 (flash memory versions only)

These are pins for flash memory programming mode lead-in.

Connect FLMD0 to V_{SS} in the normal operation mode (FLMD1 is not used in the normal operation mode).

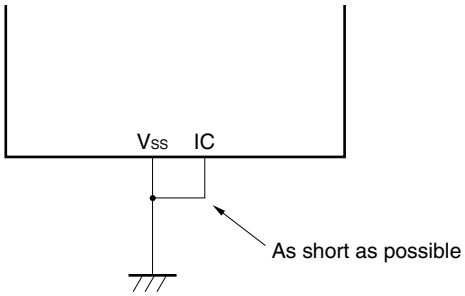
Be sure to connect these pins to the flash programmer in the flash memory programming mode.

2.2.12 IC (mask ROM versions only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780862 Subseries at shipment. Connect it directly to V_{SS} with the shortest possible wire in the normal operation mode.

When a potential difference is produced between the IC pin and the V_{SS} pin because the wiring between these two pins is too long or external noise is input to the IC pin, the user's program may not operate normally.

- **Connect the IC pin directly to V_{SS}.**



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 circuits 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/INTP0/MCGO	8-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/TI010/TO00/INTP2			
P02[Note 1]/X2 [CL2]	16	Input	Connect directly to V_{SS}.
P10/$\overline{SCK10}$/(INTP1)	8-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P11/SI10/INTP3			
P12/SO10/TOH1/(INTP3)	5-A		
P13/TxD6/INTP1/(TOH1)/(MCGO)			
P14/RxD6/<INTP0>	8-A		
P15/TOH0/FLMD1[Note 2]	5-A		
P20/ANI0 to P23/ANI3	9-C	Input	Connect directly to AV_{REF} or V_{SS}.
P130	3-C	Output	Leave open.
$\overline{RESET}$	2	Input	–
AV_{REF}	–	Input	Connect directly to V_{DD}.
X1 [CL1]	16	–	
IC	–		Connect directly to V_{SS}.
FLMD0[Note 2]			Connect to V_{SS}.

Notes 1. When the internal high-speed oscillation clock is selected as the high-speed system clock, this pin can be used as a port input pin.

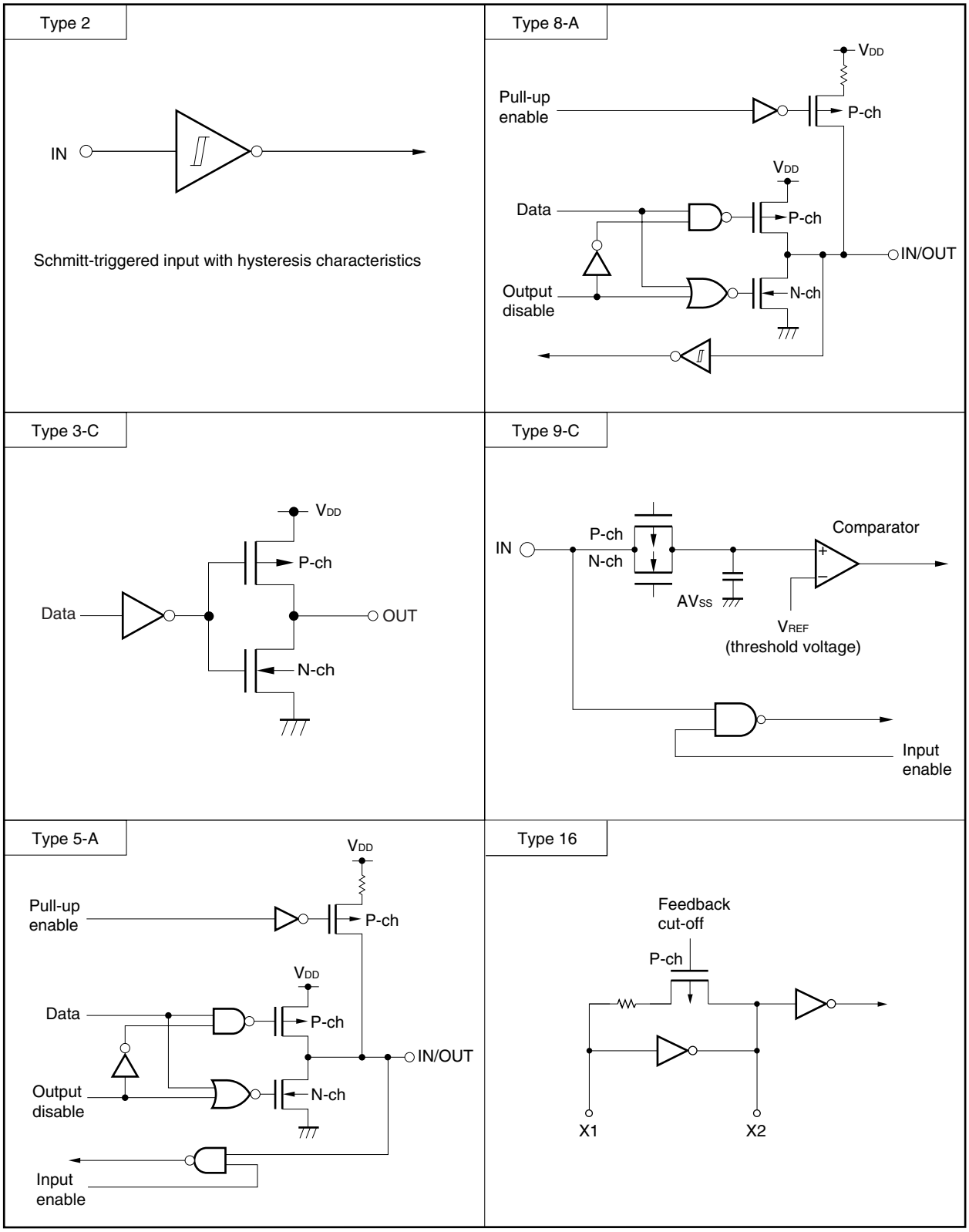
2. FLMD0 and FLMD1 are available only in the μPD78F0862 and PD78F0862A.

Remarks 1. Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

2. Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

3. Items in brackets [ ] are pin names when using external RC oscillation.

Figure 2-1. Pin I/O Circuit List



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the μPD780862 Subseries can each access a 64 KB memory space. Figures 3-1 to 3-3 show the memory maps.

Caution Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) of all products in the μPD780862 Subseries are fixed (CFH). Therefore, set the value corresponding to each product as indicated below.

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

	Internal Memory Size Switching Register (IMS)
μPD780861	42H
μPD780862	04H
μPD78F0862, 78F0862A	Value corresponding to mask ROM version

Figure 3-1. Memory Map (μPD780861)

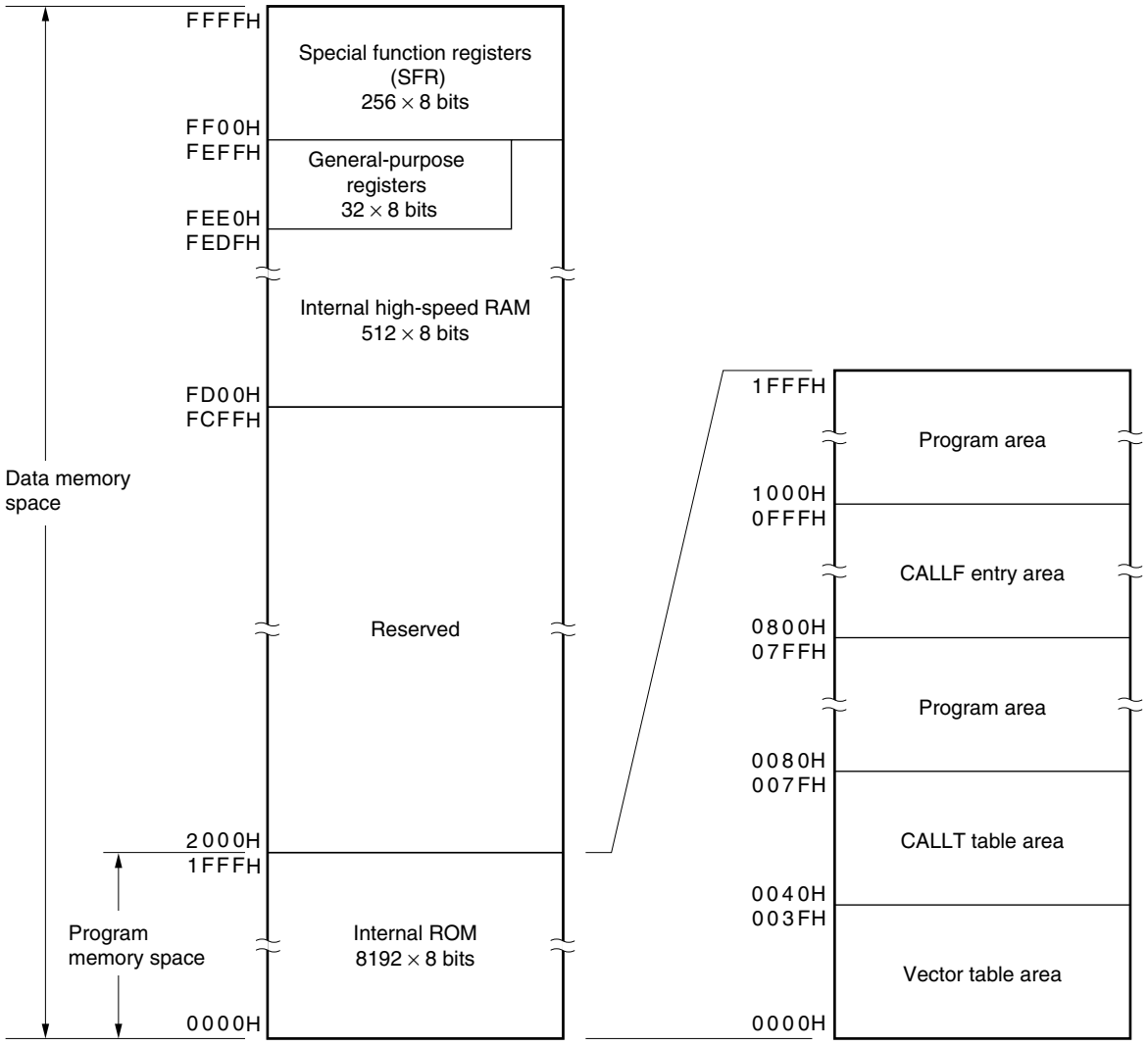


Figure 3-2. Memory Map (μPD780862)

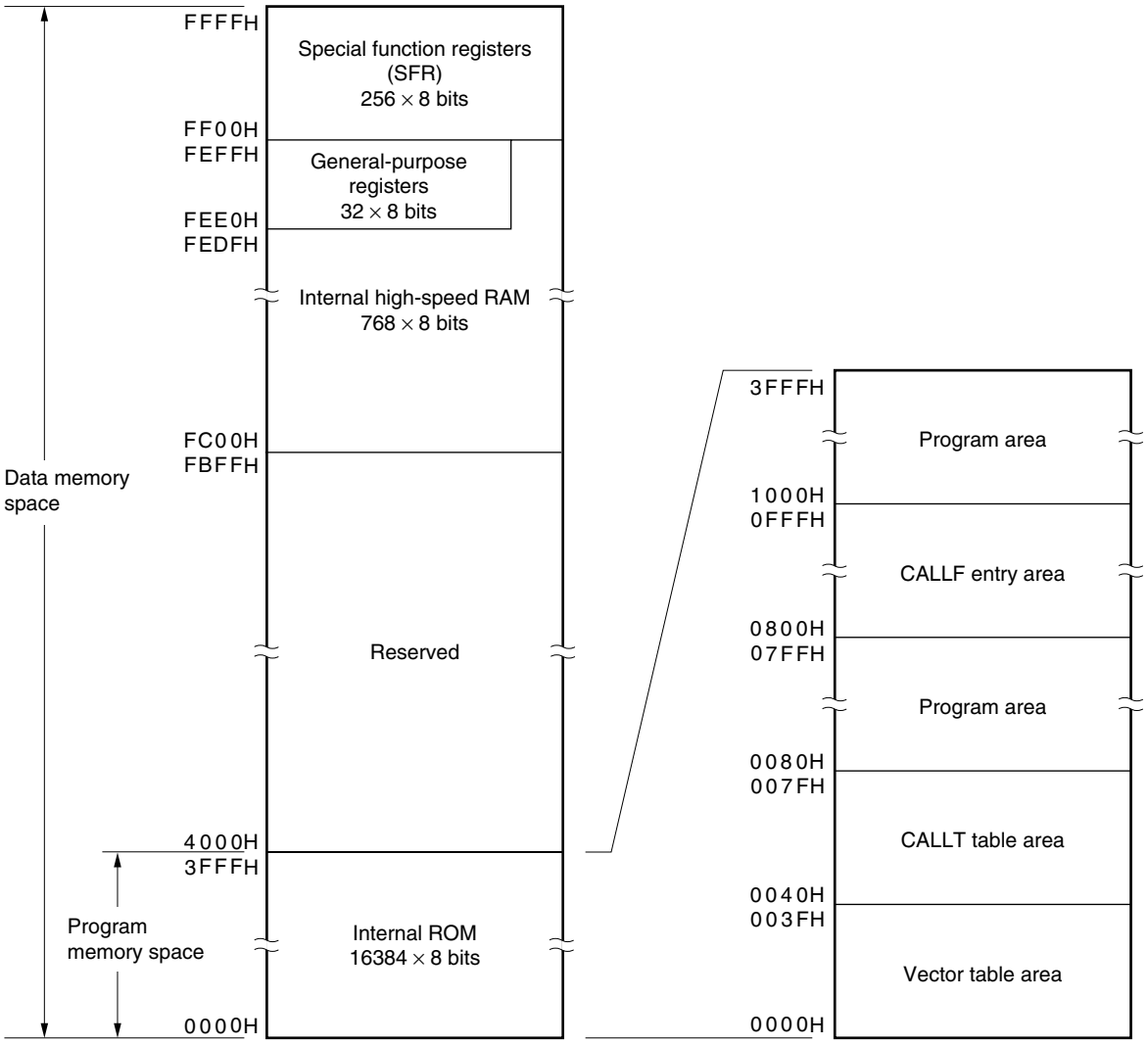
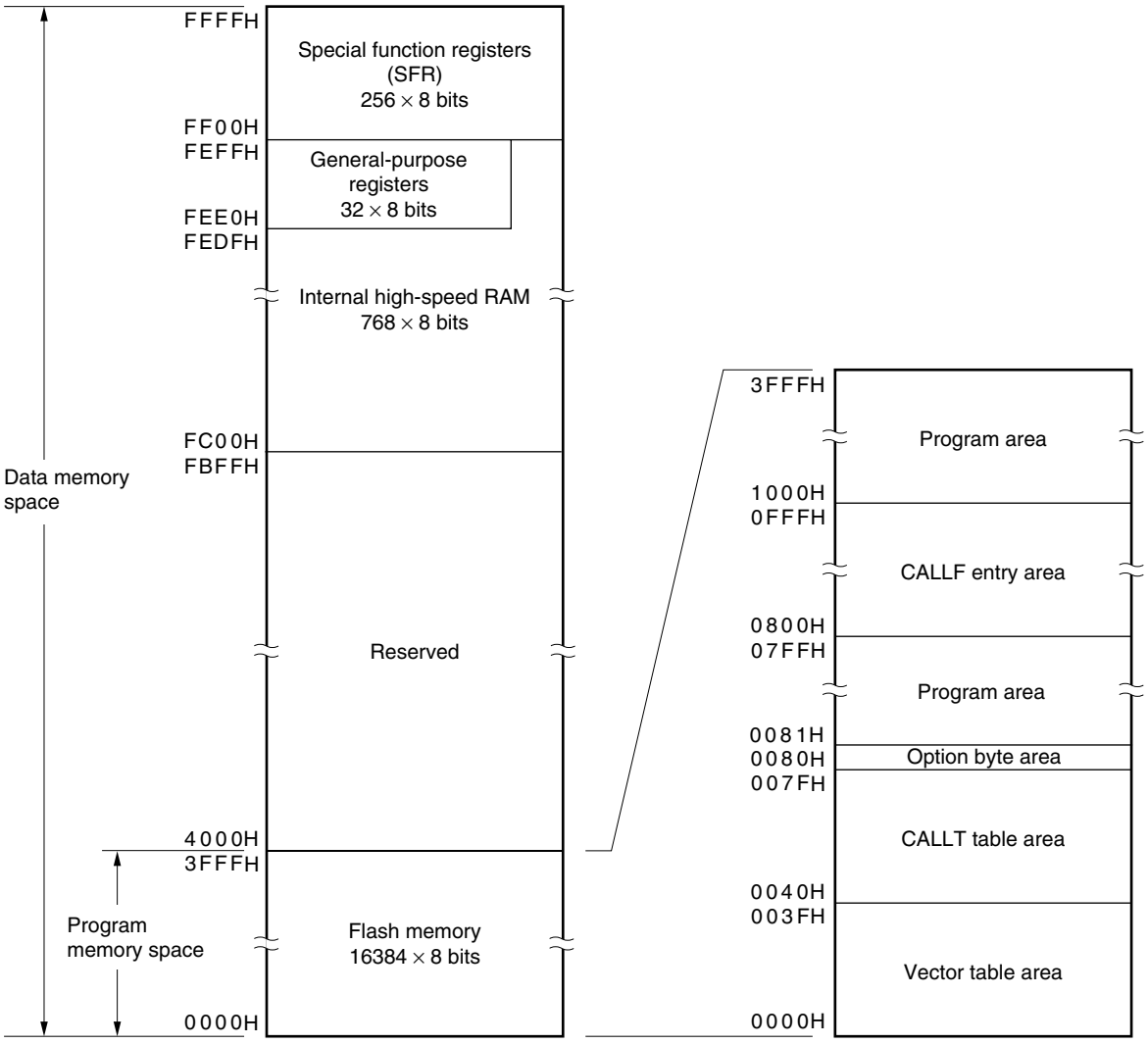


Figure 3-3. Memory Map (μPD78F0862, 78F0862A)



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).

μPD780862 Subseries products incorporate internal ROM (mask ROM or flash memory), as shown below.

Table 3-2. Internal Memory Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD780861	Mask ROM	8192 × 8 bits (0000H to 1FFFH)
μPD780862		16384 × 8 bits (0000H to 3FFFH)
μPD78F0862, 78F0862A	Flash memory	

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 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	0014H	INTSR6
		0016H	INTST6
0004H	INTLVI	0018H	INTCSI10
0006H	INTP0	001AH	INTTMH1
0008H	INTP1	001CH	INTTMH0
000AH	INTP2	001EH	INTTM50
000CH	INTP3	0020H	INTTM000
000EH	INTMCG	0022H	INTTM010
0012H	INTSRE6	0024H	INTAD

(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 (flash memory version only)

The option byte area is assigned to the 1-byte area of 0080H. For details, refer to **CHAPTER 20 MASK OPTIONS/OPTION BYTE**.

(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

μPD780862 Subseries products incorporate the following internal high-speed RAM.

Table 3-4. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD780861	512 × 8 bits (FD00H to FEFFH)
μPD780862	768 × 8 bits (FC00H to FEFFH)
μPD78F0862, 78F0862A	

The 32-byte area FEE0H to FEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per 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. The address of the instruction to be executed next is addressed by the program counter (PC) (for details, refer to **3.3 Instruction Address Addressing**).

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780862 Subseries, 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. Data memory addressing is illustrated in Figures 3-4 to 3-6. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

Figure 3-4. Data Memory Addressing (μPD780861)

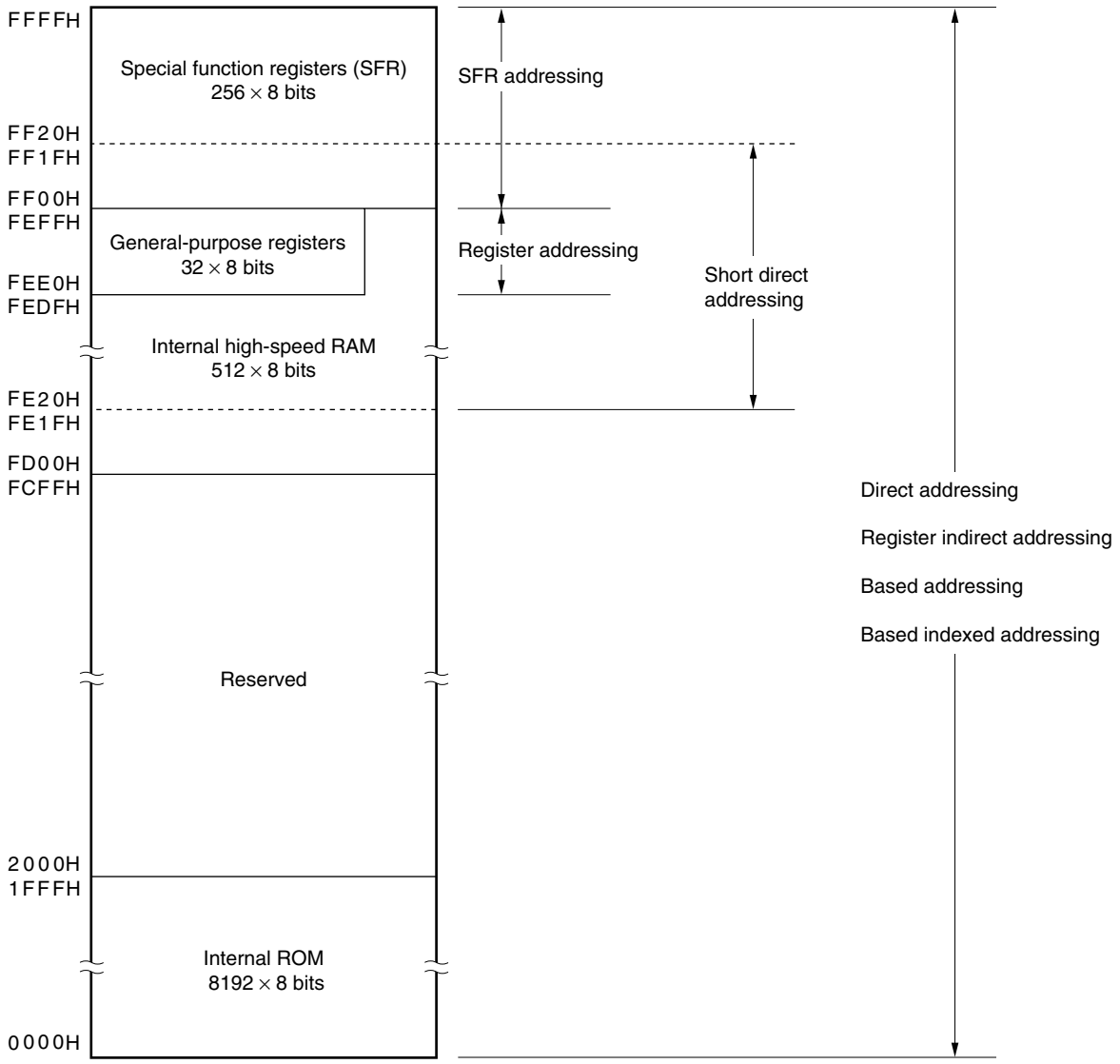


Figure 3-5. Data Memory Addressing (μPD780862)

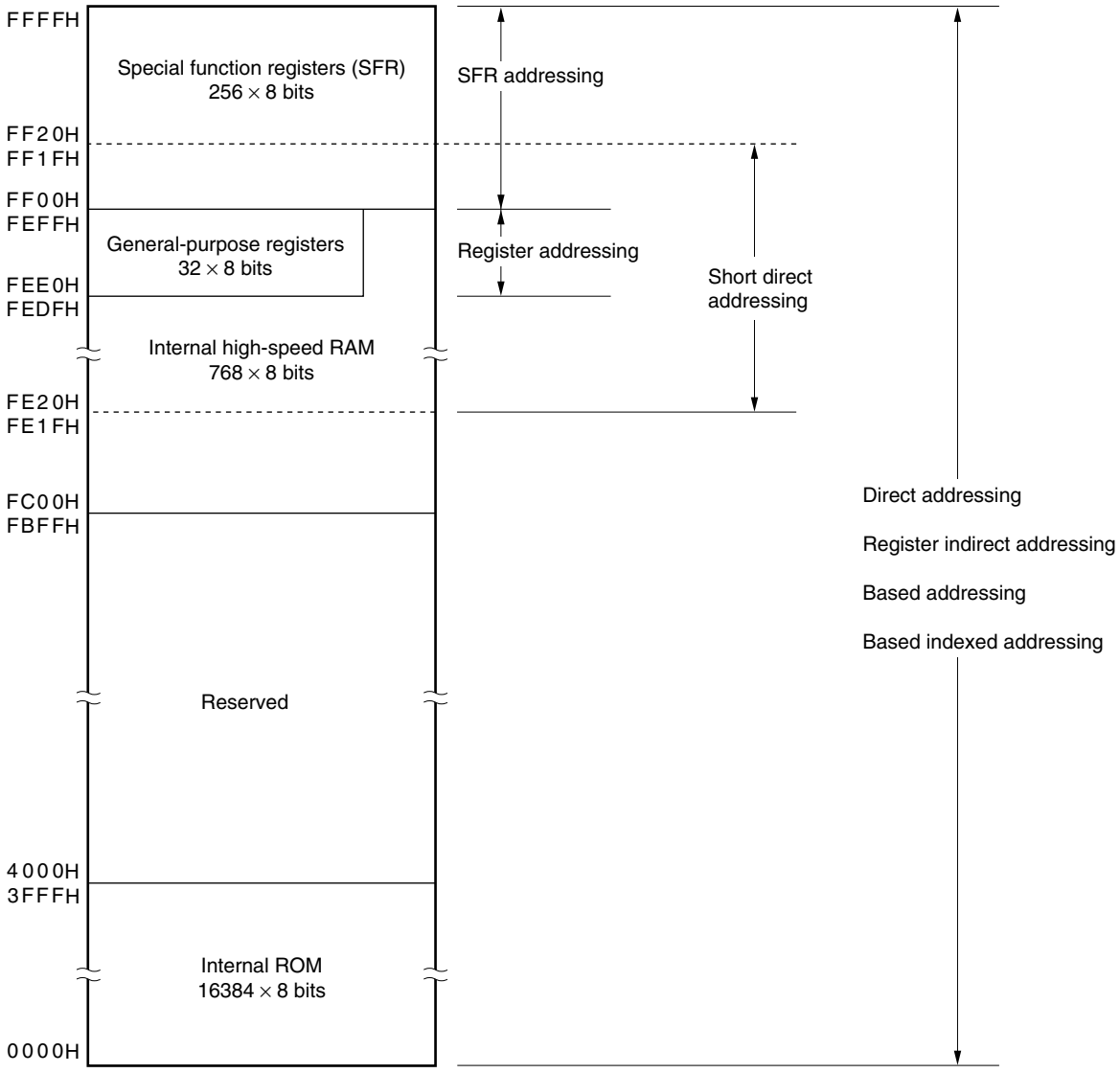
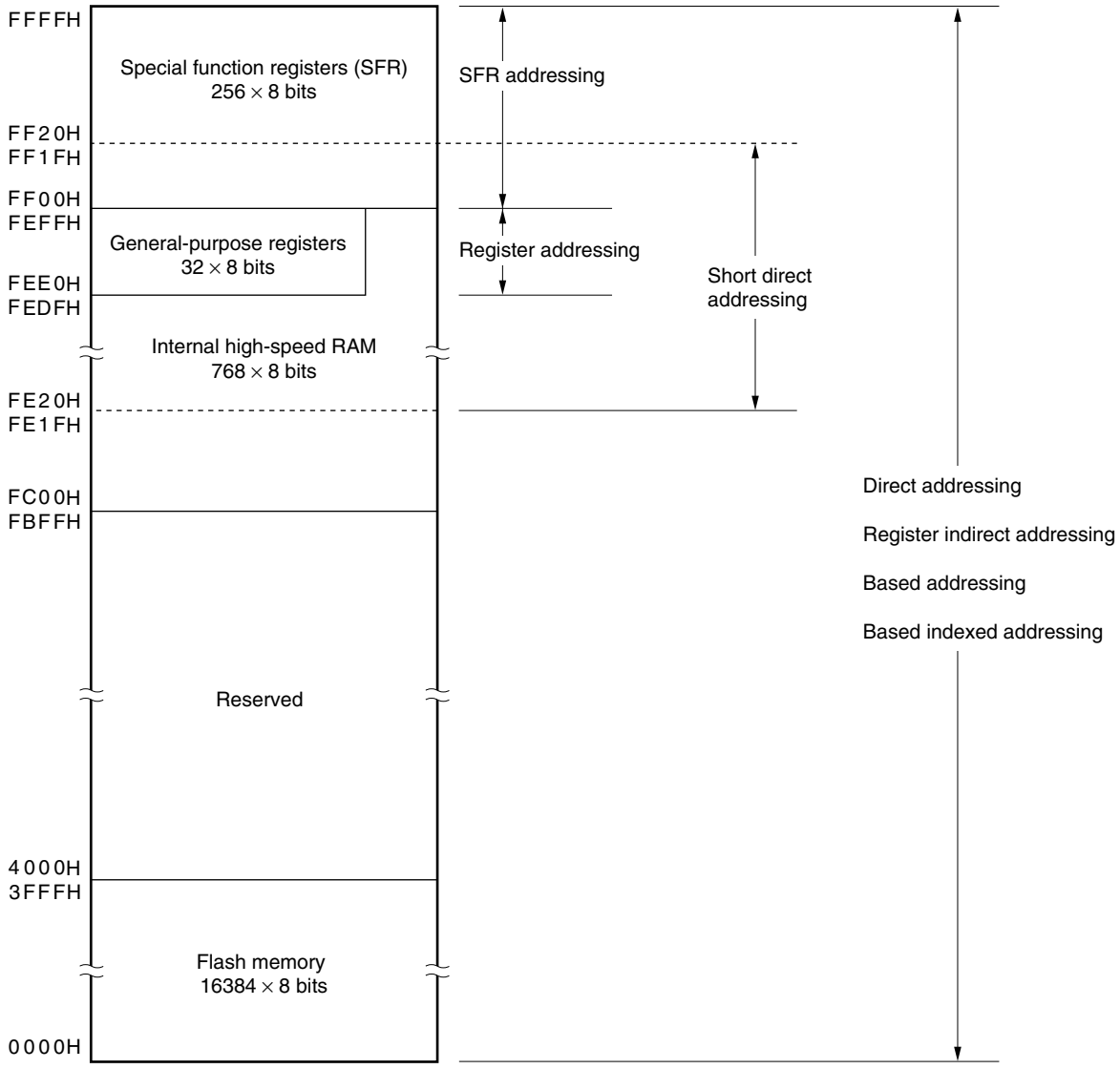


Figure 3-6. Data Memory Addressing (μPD78F0862, 78F0862A)



3.2 Processor Registers

μPD780862 Subseries 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 3-7. 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 to be 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 3-8. 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 acknowledgment operations of the CPU.

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

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

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

(b) Zero flag (Z)

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

(c) Register bank select flags (RBS0 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 with a priority specification flag register (PR0L, PR0H, PR1L) (refer to **14.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) are disabled for acknowledgment. Actual interrupt request acknowledgment is controlled with 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 manipulation 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 3-9. Format of Stack Pointer

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

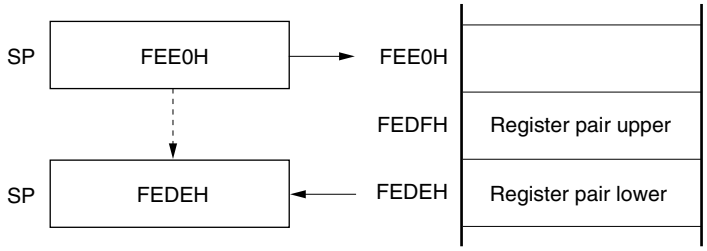
The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-10 and 3-11.

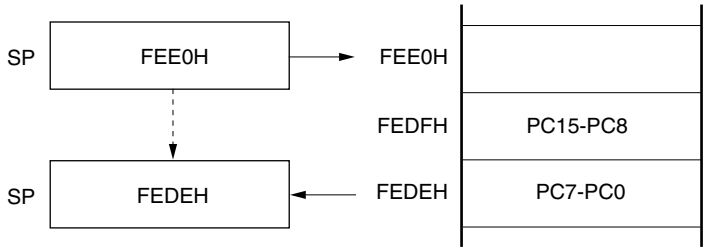
Caution Since $\overline{\text{RESET}}$ input makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-10. 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)

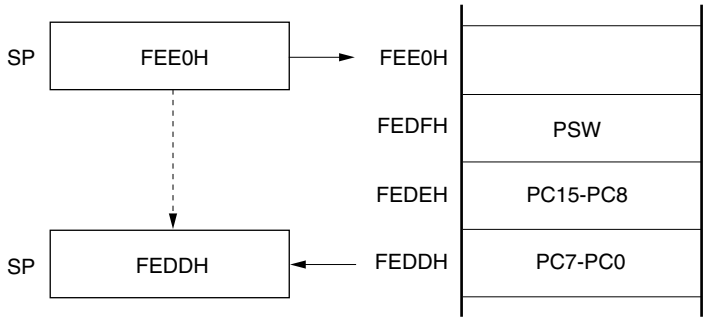
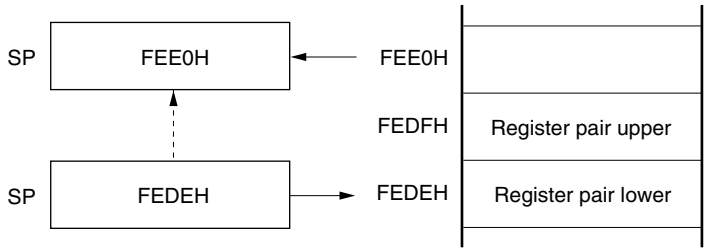
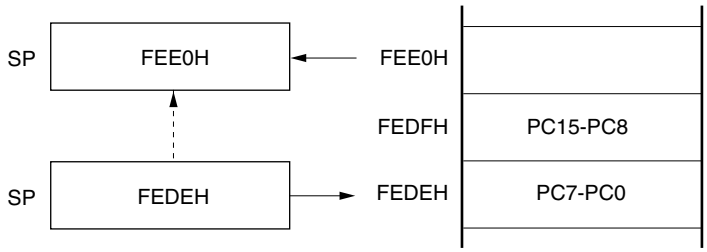


Figure 3-11. Data to Be Restored from Stack Memory

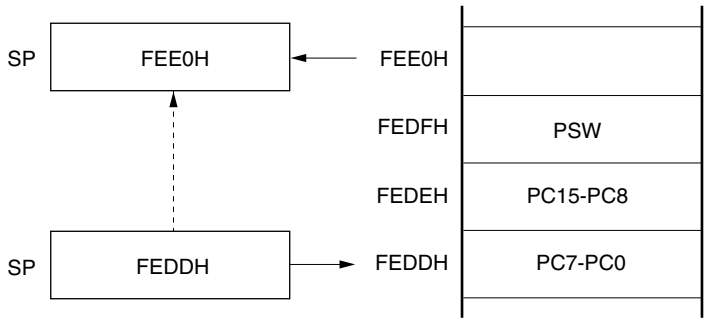
(a) POP rp instruction (when SP = FEDEH)



(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).

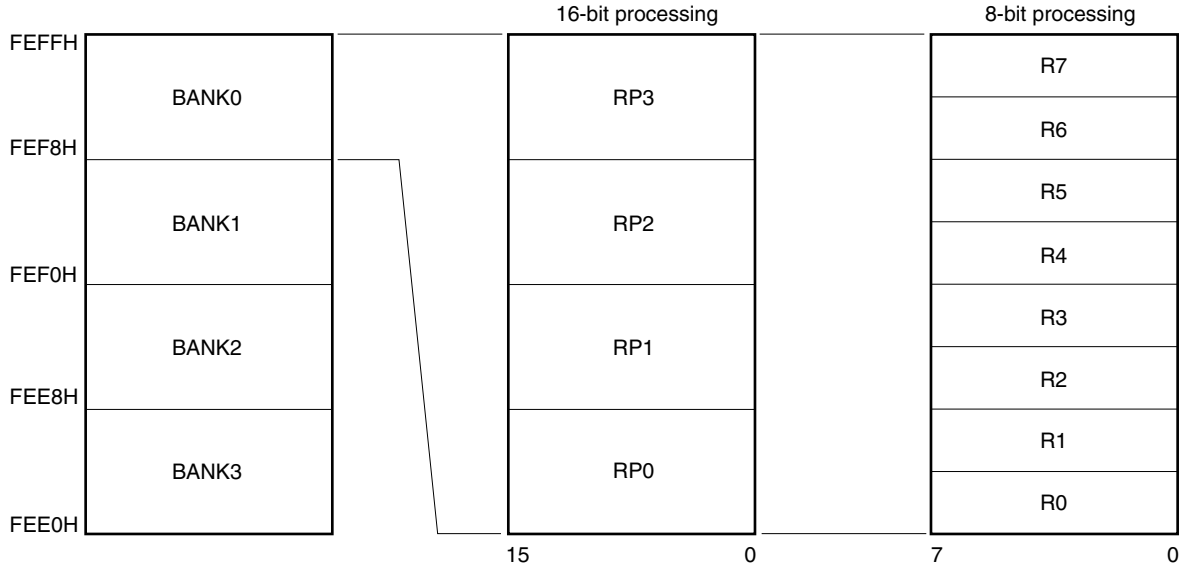
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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).

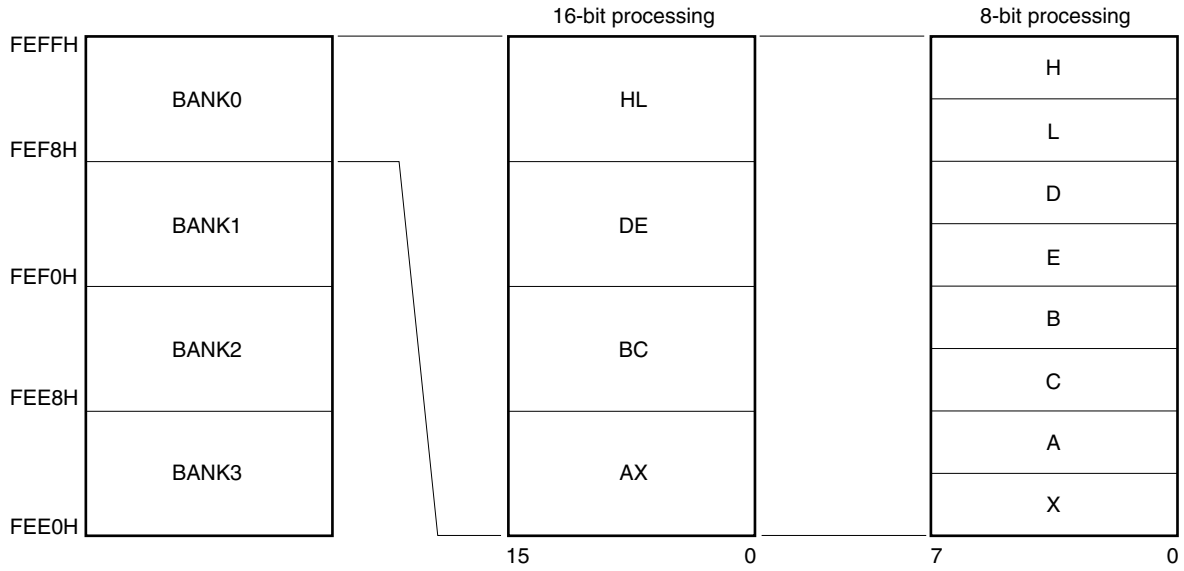
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-12. 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 in the FF00H to FFFFH area.

The special function registers can be manipulated like the 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

<R> Symbol indicating the address of a special function register. It is a reserved word in the RA78K0, and is definedas an sfr variable using the #pragma sfr directive 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	√	√	–	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
FF0DH	Port register 13	P13	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
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1	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
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
FF31H	Pull-up resistor option register 1	PU1	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

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	
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
FF60H	MCG control register 0	MC0CTL0	R/W	√	√	–	10H
FF61H	MCG control register 1	MC0CTL1	R/W	–	√	–	00H
FF62H	MCG control register 2	MC0CTL2	R/W	–	√	–	1FH
FF63H	MCG status register	MC0STR	R	√	√	–	00H
FF64H	MCG transmit buffer register	MC0TXBW, MC0TX	R/W	–	√	√	FFH
FF65H	MCG transmit bit count specification register	MC0BIT	R/W	–	√		07H
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
FF70H	Alternate-function pin switch register	PSEL	R/W	√	√	–	00H
FF71H	Timer clock switch control register	CSEL	R/W	√	√	–	00H
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
FF98H	Watchdog timer mode register	WDTM	R/W	–	√	–	67H
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	9AH
FFA0H	Internal low-speed oscillation 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
FFACH	Reset control flag register	RESF	R	–	√	–	00H[Note]
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

Note This value varies depending on the reset source.

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	
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	1F1L		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]	IMS		R/W	–	√	–	CFH
FFFBH	Processor clock control register	PCC		R/W	√	√	–	00H

Note The default value of IMS is fixed (CFH) in all products in the μPD780862 Subseries regardless of the internal memory capacity. Therefore, set the following value to each product.

	Internal Memory Size Switching Register (IMS)
μPD780861	42H
μPD780862	04H
μPD78F0862, 78F0862A	Value corresponding to mask ROM version

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

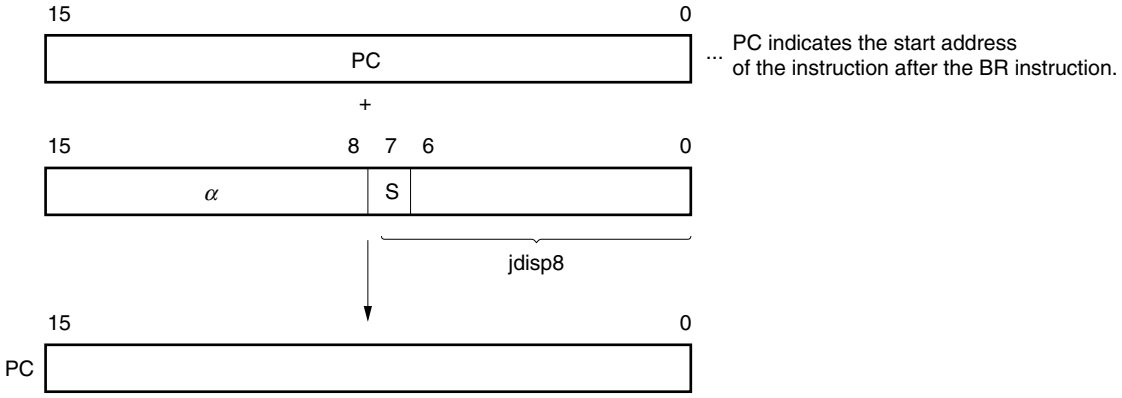
[Function]

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]



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

3.3.2 Immediate addressing

[Function]

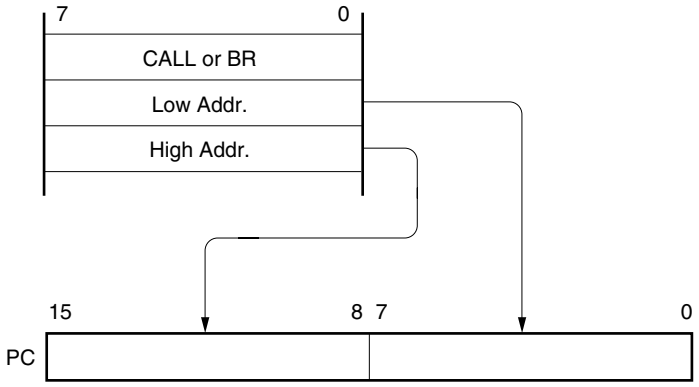
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

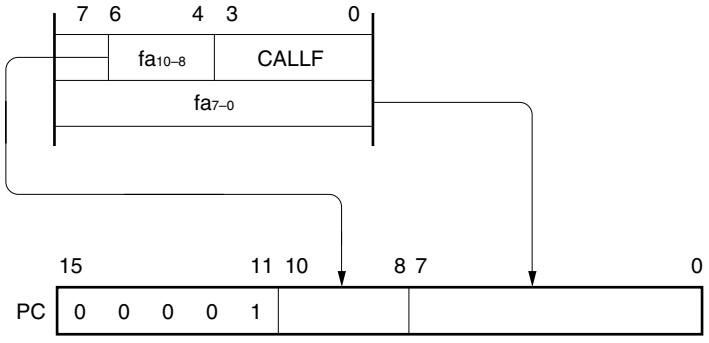
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

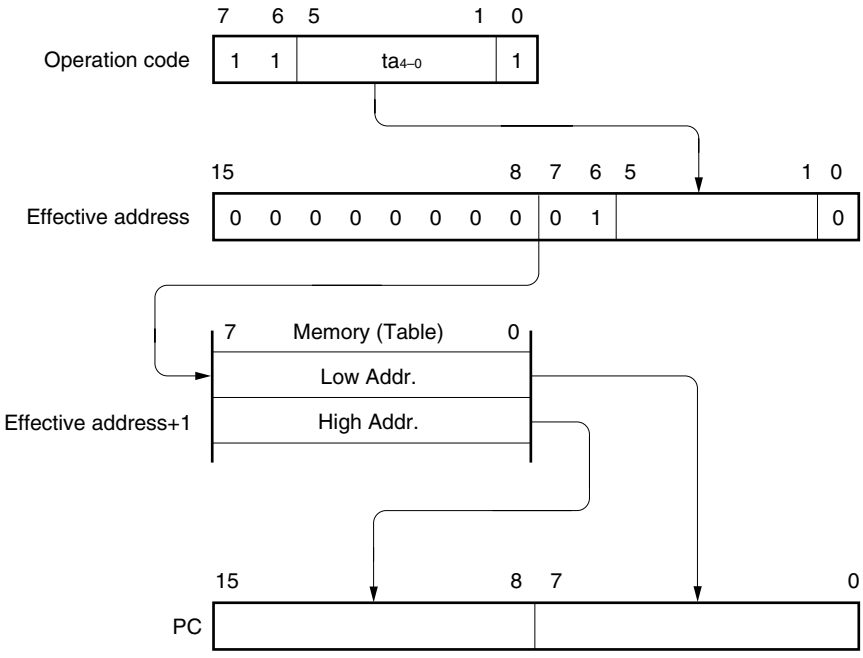
[Function]

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]



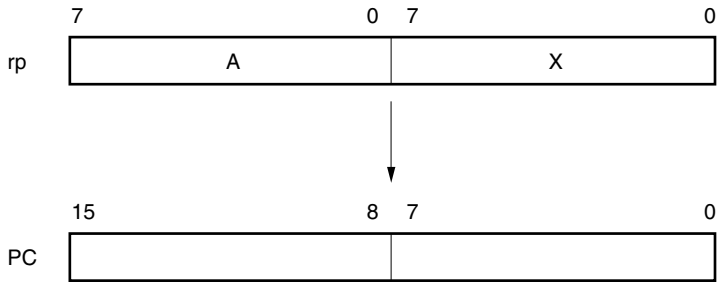
3.3.4 Register addressing

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

3.4.1 Implied addressing

[Function]

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

Of the μPD780862 Subseries 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 which become decimal correction targets
ROR4/ROL4	A register for storage of digit data which 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 and 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.

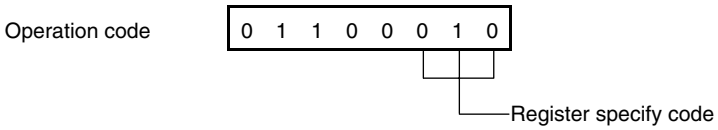
[Operand format]

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

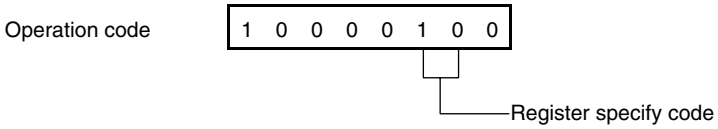
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[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]

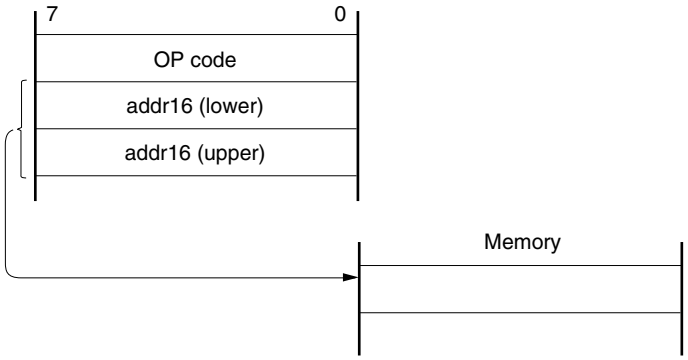
Identifier	Description
addr16	Label or 16-bit immediate data

[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 (SFR) 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.

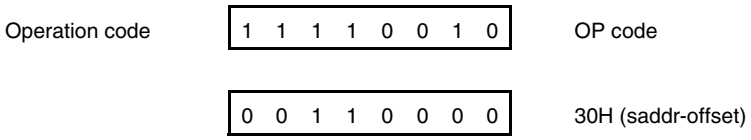
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is cleared to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to the **[Illustration]** shown below.

[Operand format]

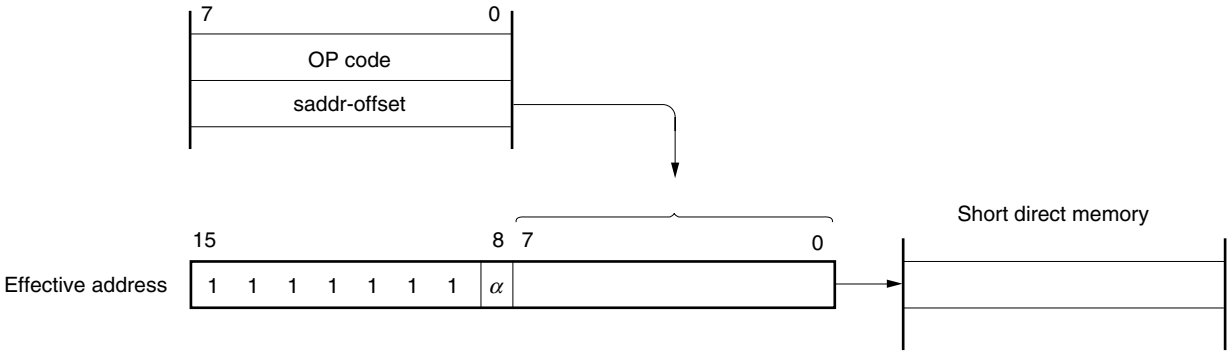
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

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



[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$

When 8-bit immediate data is 00H to 1FH, $\alpha = 1$

3.4.5 Special function register (SFR) addressing

[Function]

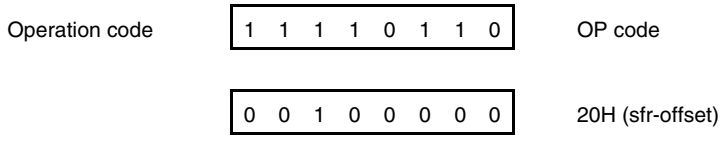
A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word. This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

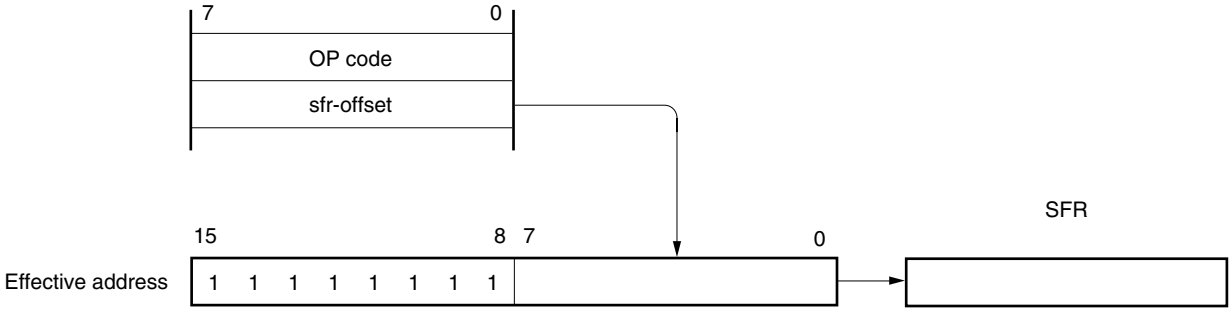
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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



[Illustration]



3.4.6 Register indirect addressing

[Function]

Register pair contents specified by a register pair specify code in an operation code and by the register bank select flags (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]

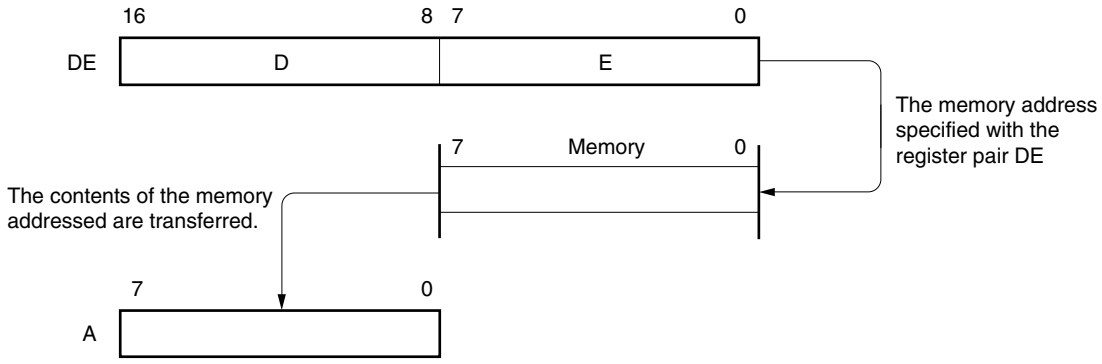
Identifier	Description
–	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

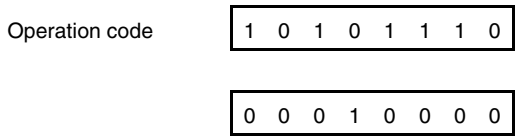
8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flags (RBS0 and RBS1) and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive 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]

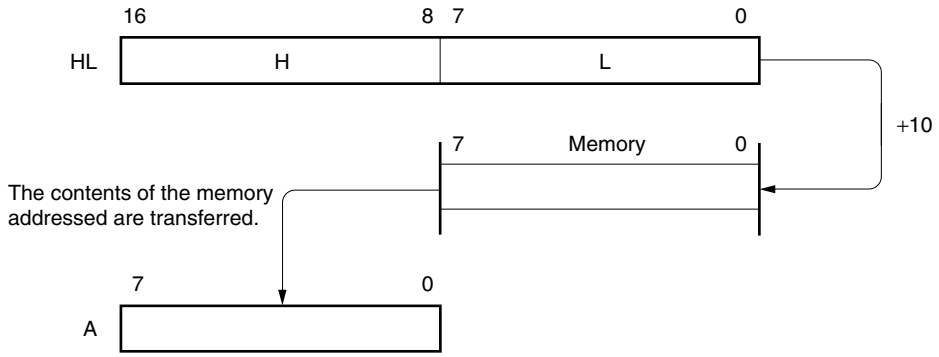
Identifier	Description
–	[HL + byte]

[Description example]

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



[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 flags (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]

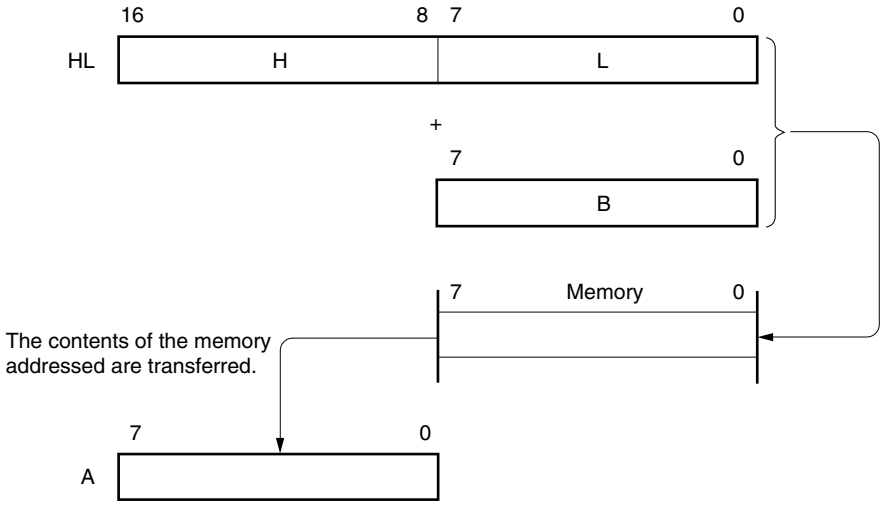
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

MOV A, [HL + B]; when 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/restored upon generation of an interrupt request.

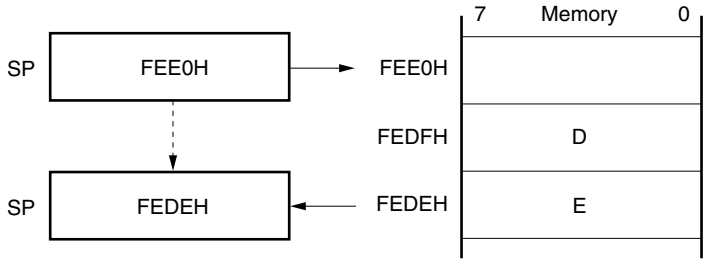
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

PUSH DE; when 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 V_{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 P23
V_{DD}	Pins other than P20 to P23

μPD780862 Subseries 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

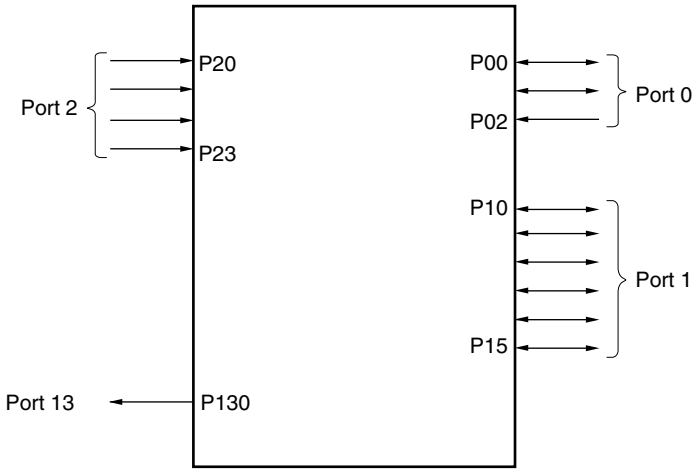


Table 4-2. Port Functions

Pin Name	I/O	Function		After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port.	Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI000/INTP0/MCGO
P01					TI010/TO00/INTP2
P02[Note 1]	Input		Input-only	Input	X2 [CL2]
P10	I/O	Port 1. 6-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	$\overline{\text{SCK10}}$/(INTP1)
P11					SI10/INTP3
P12					SO10/TOH1/(INTP3)
P13					TxD6/INTP1/(TOH1)/(MCGO)
P14					RxD6/<INTP0>
P15					TOH0/FLMD1[Note 2]
P20 to P23	Input	Port 2. 4-bit input-only port.		Input	ANI0 to ANI3
P130	Output	Port 13. 1-bit output-only port.		Output	–

Notes 1. When the internal high-speed oscillation clock is selected as the high-speed system clock, this pin can be used as a port input pin.

2. FLMD1 is available only in the μPD78F0862 and 78F0862A.

Remarks 1. Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

2. Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

3. Items in brackets [ ] are pin names when using external RC oscillation.

4.2 Port Configuration

A port includes the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode register (PM0, PM1) Port register (P0 to P2, P13) Pull-up resistor option register (PU0, PU1) Alternate-function pin switch register (PSEL) Input switch control register (ISC)
Ports	Total: 14 (CMOS I/O: 8, CMOS input: 5, CMOS output: 1)
Pull-up resistors	Total: 8 (software control only)

4.2.1 Port 0

Port 0 is a 3-bit I/O port with an output latch. The P00 and P01 pins can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). The P02 pin is input-only. When the P00 and P01 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

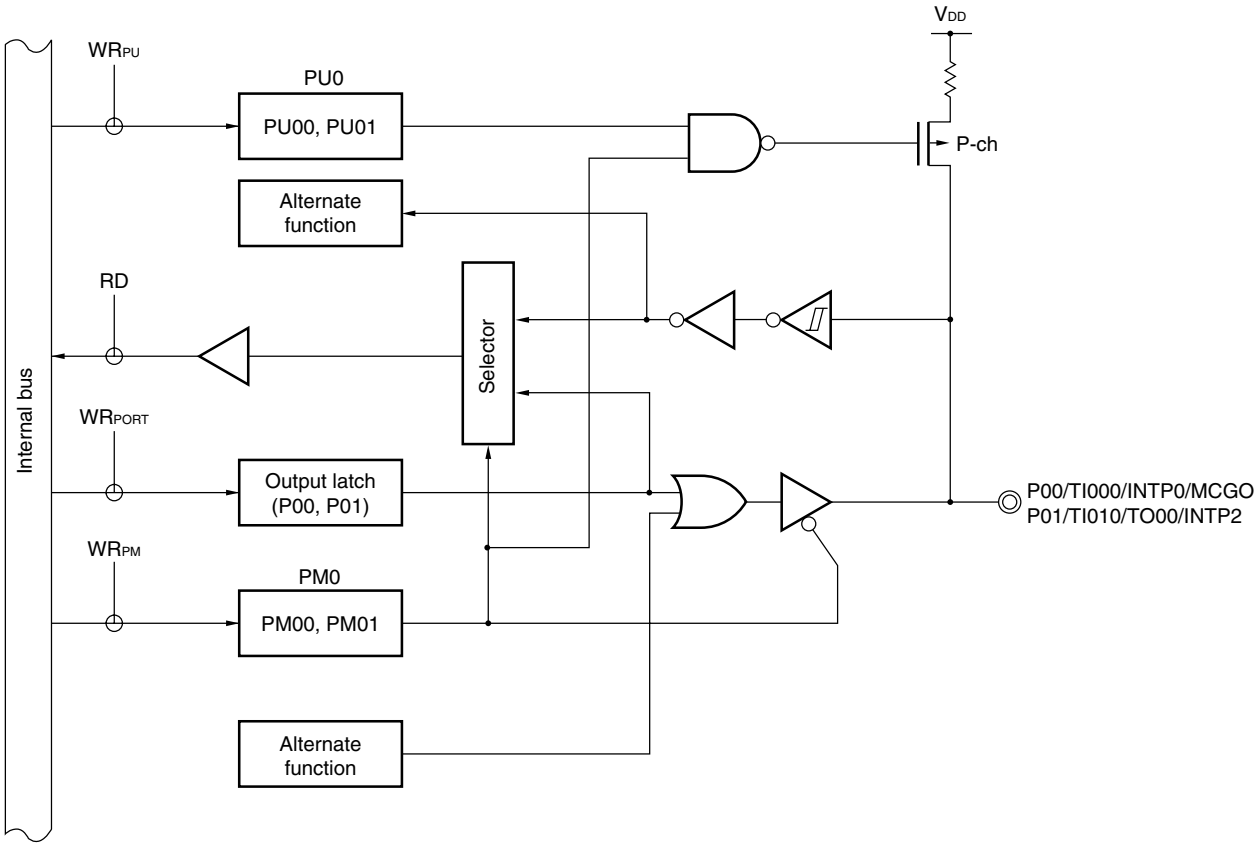
This port can also be used for external interrupt request input, Manchester code output, timer I/O, and crystal/ceramic resonator connection [RC connection] for high-speed system clock oscillation.

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

Figures 4-2 and 4-3 show block diagrams of port 0.

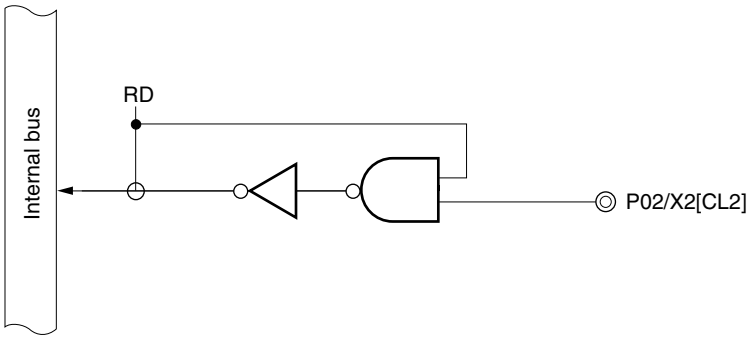
Caution When the internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version), P02 can be used as an input-only port pin (when a crystal/ceramic or external RC oscillation is selected as the high-speed system clock by a mask option, P02 becomes a resonator connection pin).

Figure 4-2. Block Diagram of P00 and P01



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

Figure 4-3. Block Diagram of P02



RD: Read signal

Caution If a read instruction is executed while this pin is being used as its alternate function (X2 [CL2]), the read data is undefined.

4.2.2 Port 1

Port 1 is a 6-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 P15 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

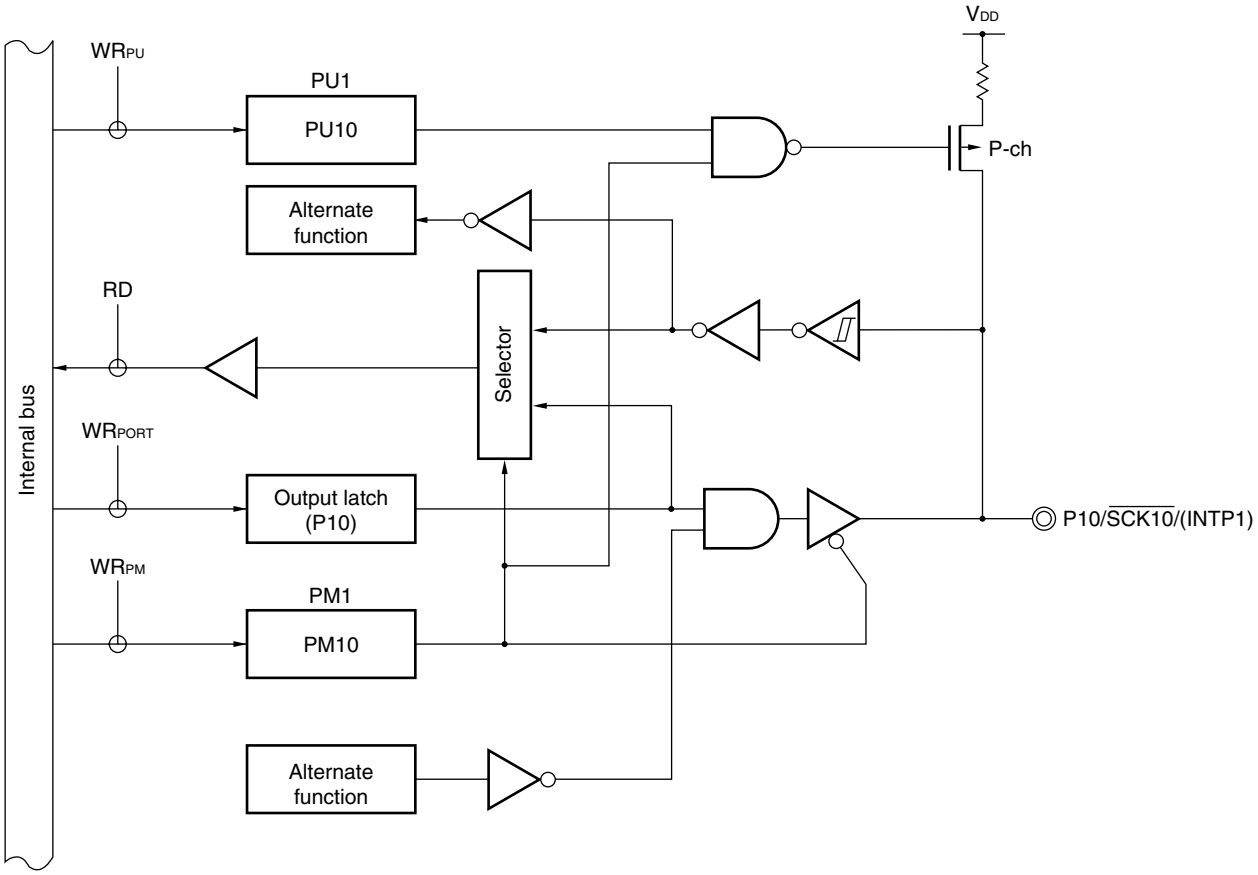
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, timer output, and flash memory programming mode lead-in. P10 to P15 can be assigned as external interrupt request input, timer output, and Manchester code output by setting the alternate-function pin switch register (PSEL) and input switch control register (ISC).

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

Figures 4-4 to 4-8 show block diagrams of port 1.

<R> **Caution To use P10/$\overline{\text{SCK10}}$/(INTP1), and P12/SO10/TOH1/(INTP3) as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection register 10 (CSIC10) to the default status (00H).**

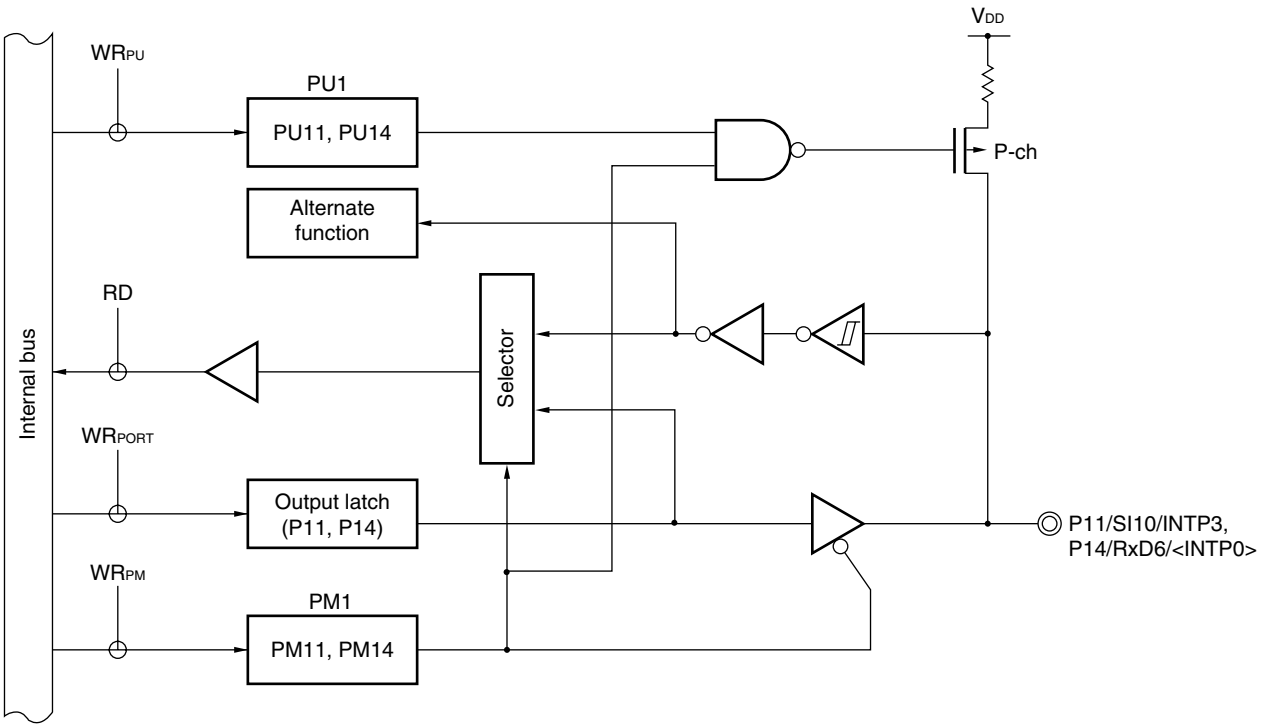
Figure 4-4. Block Diagram of P10



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

Remark Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

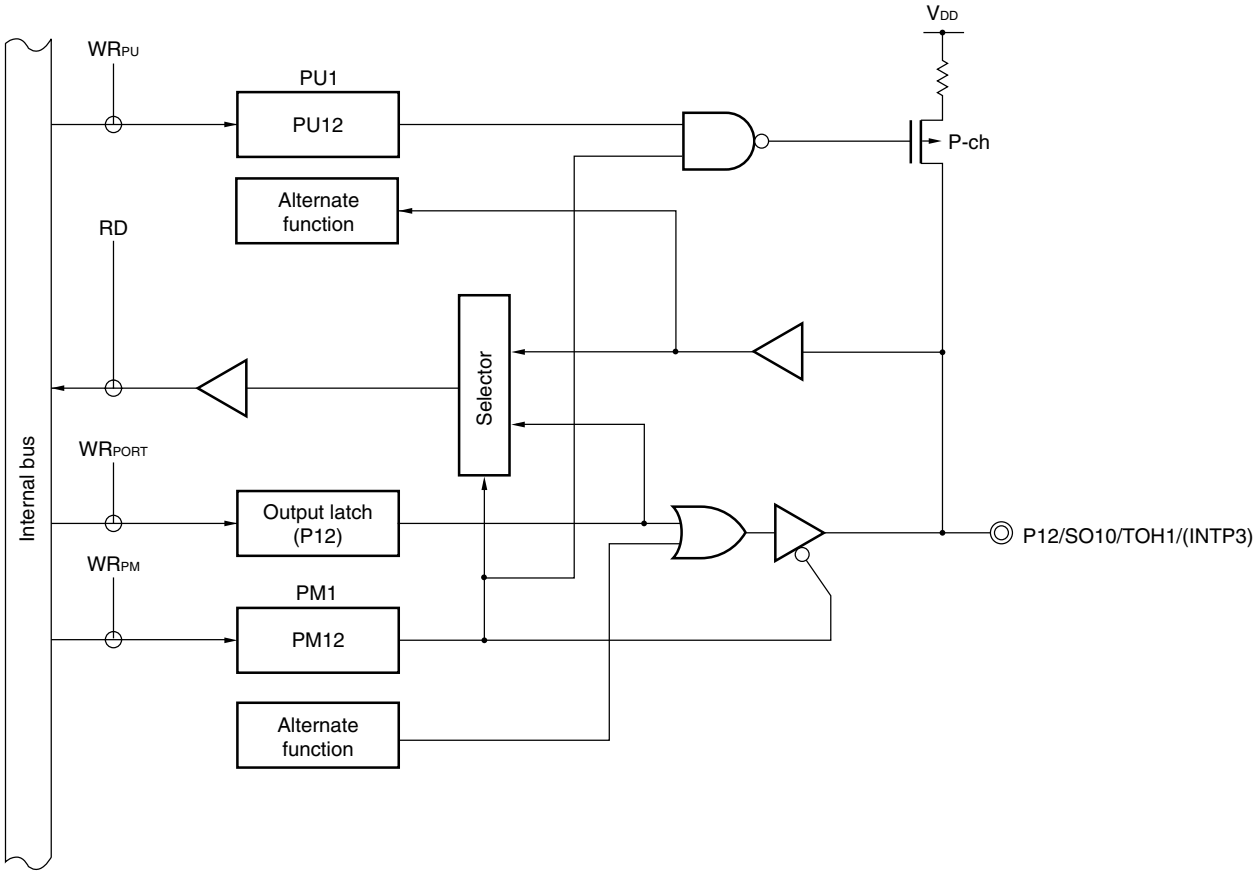
Figure 4-5. Block Diagram of P11 and P14



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

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

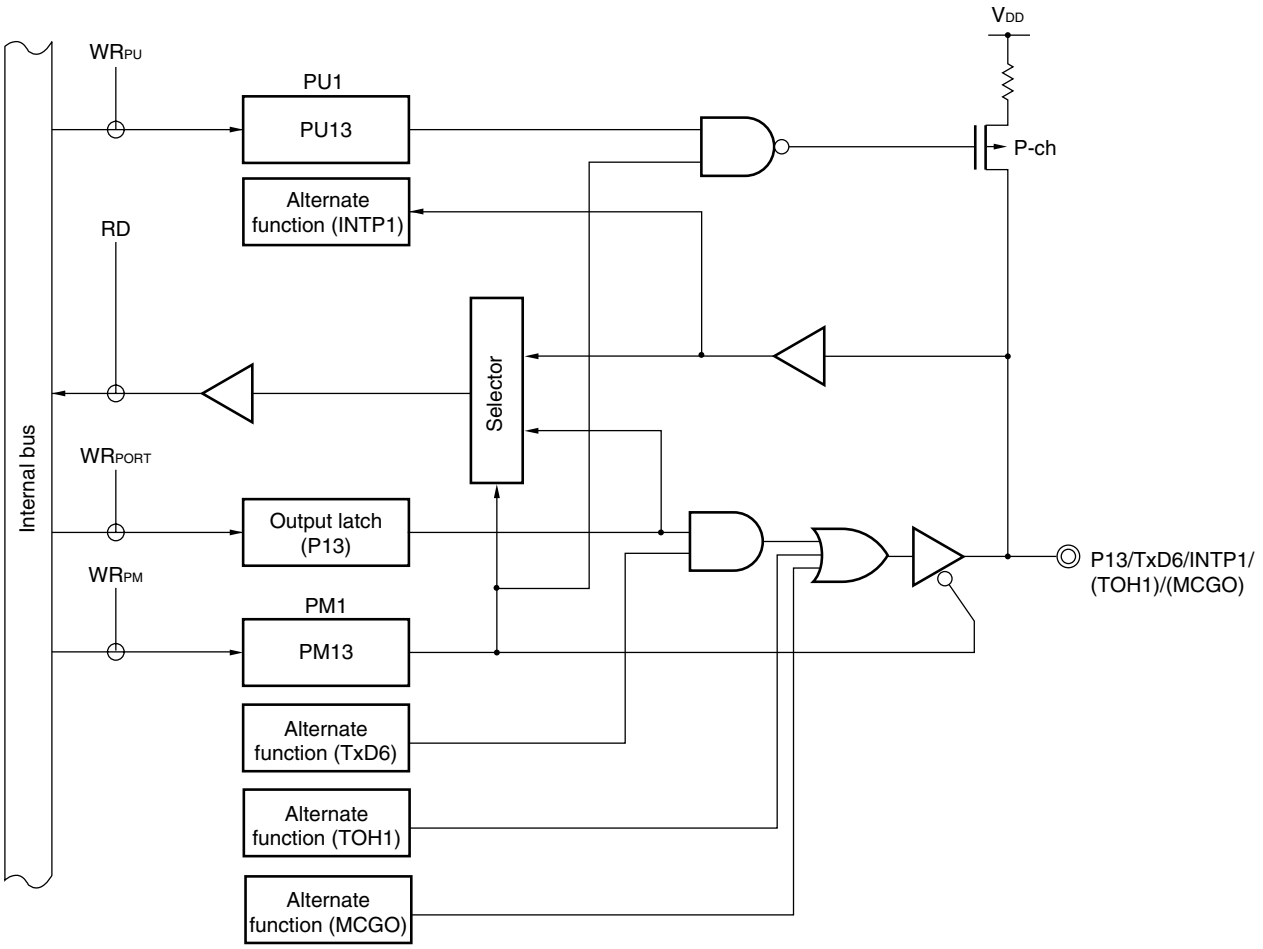
Figure 4-6. Block Diagram of P12



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

Remark Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

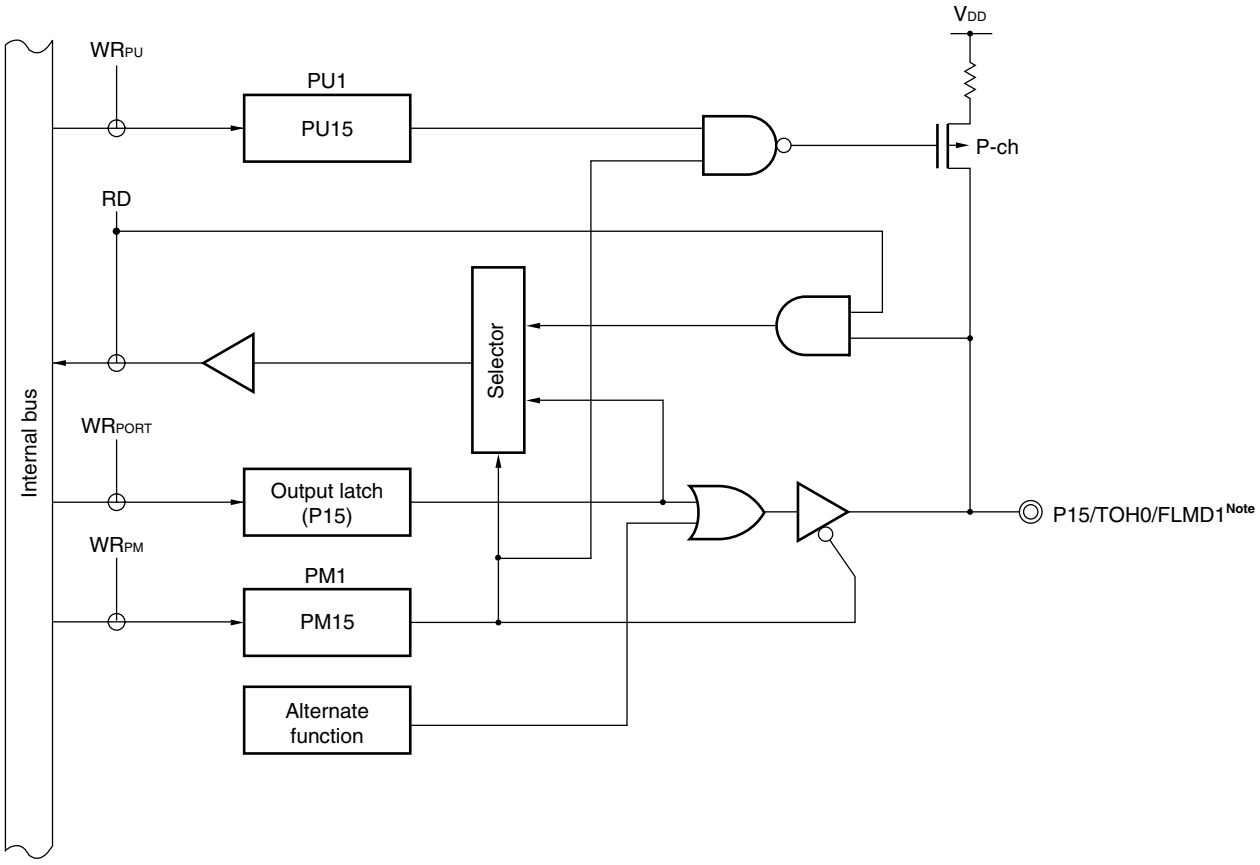
Figure 4-7. Block Diagram of P13



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

Remark Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

Figure 4-8. Block Diagram of P15



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

Note FLMD1 is available only in the μPD78F0862 and 78F0862A.

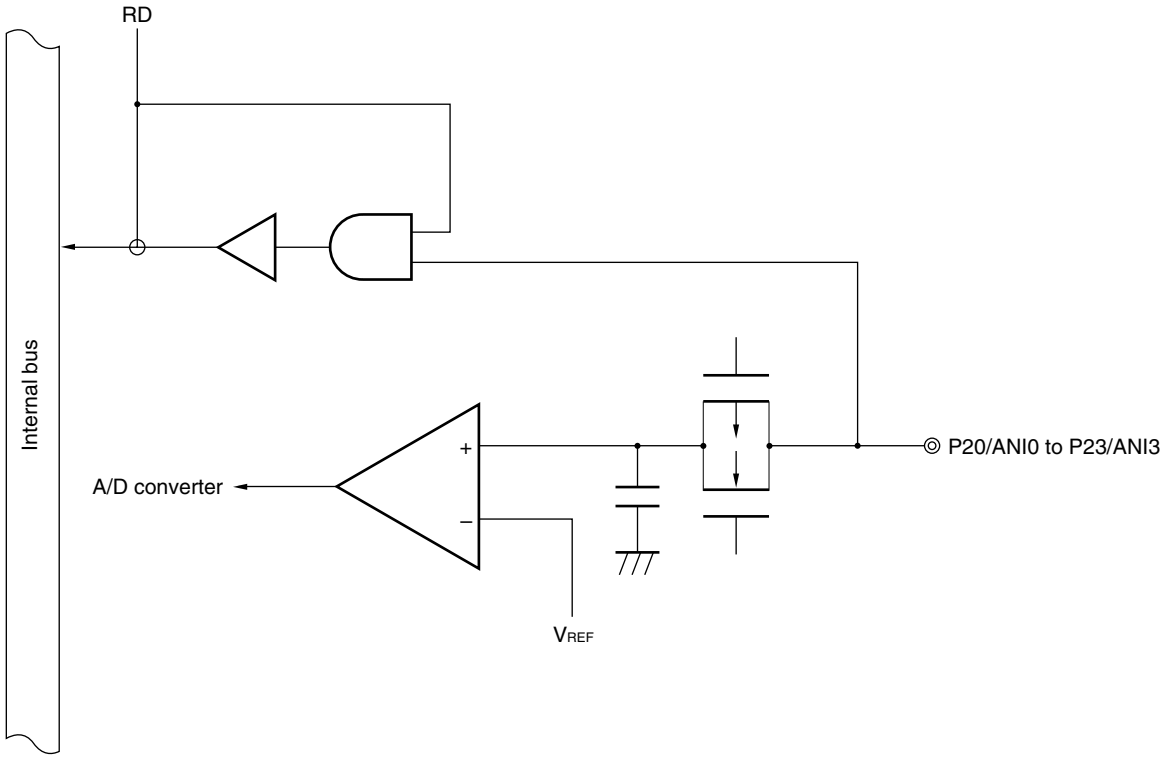
4.2.3 Port 2

Port 2 is a 4-bit input-only port.

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

Figure 4-9 shows a block diagram of port 2.

Figure 4-9. Block Diagram of P20 to P23



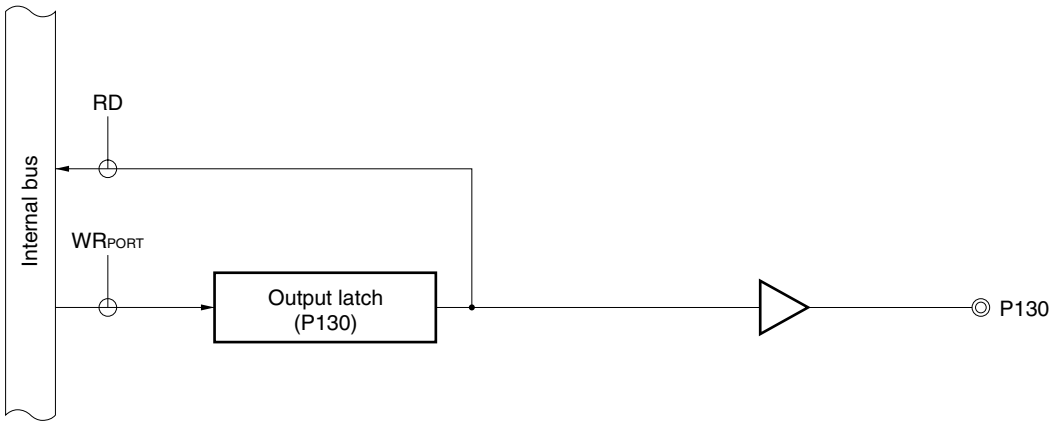
RD: Read signal

4.2.4 Port 13

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

Figure 4-10 shows a block diagram of port 13.

Figure 4-10. Block Diagram of P130



RD: Read signal

WR××: Write signal

Remark P130 outputs a low level at reset, so the output from P130 can be output as a pseudo-CPU reset signal if P130 is set to output a high level before reset is effected.

4.3 Registers Controlling Port Function

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

- Port mode registers (PM0, PM1)
- Port registers (P0 to P2, P13)
- Pull-up resistor option registers (PU0, PU1)
- Alternate-function pin switch register (PSEL)
- Input switch control register (ISC)

(1) Port mode registers (PM0 and PM1)

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.

Cautions 1. Because P00, P01, P11, and P13 can also be used as external interrupt input pins and P10, P12, and P14 can be assigned as an external interrupt input by setting the alternate-function pin switch register (PSEL), when port function output mode is specified to change the output level, the interrupt request flag is set. Therefore, when these pins are used in output mode, preset the interrupt mask flags (PMK0 to PMK3) to 1.

Cautions 2. P02 is an input-only pin. When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

3. When writing to PM0 using an 8-bit memory manipulation instruction, be sure to set bits 2 to 7 to 1.

When writing to PM1 using an 8-bit memory manipulation instruction, be sure to set bits 6 and 7 to 1.

Figure 4-11. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 1; n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

Pin Name	Alternate Function		PM××	P××
	Name	I/O		
P00	TI000	Input	1	×
	INTP0	Input	1	×
	MCGO	Output	0	0
P01	TI010	Input	1	×
	TO00	Output	0	0
	INTP2	Input	1	×
P10	$\overline{SCK10}$	Input	1	×
		Output	0	1
	(INTP1)	Input	1	×
P11	SI10	Input	1	×
	INTP3	Input	1	×
P12	SO10	Output	0	0
	TOH1	Output	0	0
	(INTP3)	Input	1	×
P13	TxD6	Output	0	1
	INTP1	Input	1	×
	(TOH1)	Output	0	0
	(MCGO)	Output	0	0
P14	RxD6	Input	1	×
	<INTP0>	Input	1	×
P15	TOH0	Output	0	0

Remarks 1. Functions in parentheses ( ) can be assigned by setting the alternate-function pin switch register (PSEL).

2. Functions in angle brackets < > can be assigned by setting the input switch control register (ISC).

3. ×: Don't care

PM××: Port mode register

P××: Port output latch

(2) Port registers (P0 to P2, P13)

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 4-12. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	P02[Note]	P01	P00	FF00H	00H (output latch)	R/W
P1	0	0	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FF02H	Undefined	R
P13	0	0	0	0	0	0	0	P130	FF0DH	00H (output latch)	R/W

Pmn	m = 0 to 2, 13; 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

Note When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

(3) Pull-up resistor option registers (PU0 and PU1)

These registers specify whether the on-chip pull-up resistors of P00, P01, or P10 to P15 are to be used or not. An on-chip pull-up resistor can be used in 1-bit units only for the bits set to input mode of the pins of PU0 or PU1 to which the use of an on-chip pull-up resistor has been specified. On-chip pull-up resistors cannot be used for bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU0 and PU1.

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

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

Caution The P02 pin does not incorporate a pull-up resistor.

Figure 4-13. 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	0	0	PU01	PU00	FF30H	00H	R/W
	7	6	5	4	3	2	1	0			
PU1	0	0	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1; n = 0 to 5)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

(4) Alternate-function pin switch register (PSEL)

This register is used to select the TOH1, INTP1, INTP3, and MCGO pins.

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

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

Figure 4-14. Format of Alternate-Function Pin Switch Register (PSEL)

Address: FF70H After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	2	<1>	<0>
PSEL	0	0	TOH1SL	MCGSL	0	0	INTP1SL	INTP3SL

TOH1SL	TOH1 pin selection
0	P12/SO10/TOH1/(INTP3)
1	P13/TxD6/INTP1/(TOH1)/(MCGO)

MCGSL	MCGO pin selection
0	P00/TI000/INTP0/MCGO
1	P13/TxD6/INTP1/(TOH1)/(MCGO)

INTP1SL	INTP1 pin selection
0	P13/TxD6/INTP1/(TOH1)/(MCGO)
1	P10/$\overline{\text{SCK10}}$/(INTP1)

INTP3SL	INTP3 pin selection
0	P11/SI10/INTP3
1	P12/SO10/TOH1/(INTP3)

Cautions 1. Set bit 7 (TMHE1) of 8-bit timer H mode register 1 (TMHMD1) to 0 before rewriting the TOH1SL bit.

2. Set bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) to 0 before rewriting the MCGSL bit.

<R> **(5) 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 this register to 00H.

Figure 4-15. 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 (P00)
1	RxD6 (P14)

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 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 two system clock oscillators are available.

- High-speed system clock oscillator
  The following three high-speed system clock oscillators are available.

  - Crystal/ceramic oscillator: Oscillates a clock of 2 to 10 MHz.
  - External RC oscillator: Oscillates a clock of 3 to 4 MHz.
  - Internal high-speed oscillator: Oscillates a clock of 8.0 MHz (TYP.).

  High-speed system clock oscillation can be selected by a mask option when using a mask ROM version or by an option byte when using a flash memory version. For details, refer to **CHAPTER 20 MASK OPTIONS/OPTION BYTE**.
  Oscillation of the high-speed system clock oscillator is stopped by executing the STOP instruction or setting the main OSC control register (MOC).
- Internal low-speed oscillator
  The Internal low-speed oscillator oscillates a clock of 240 kHz (TYP.). Oscillation can be stopped by setting the internal low-speed oscillation mode register (RCM) when “Can be stopped by software” is set by a mask option (option byte if using a flash memory version) and the high-speed system clock is used as the CPU clock.

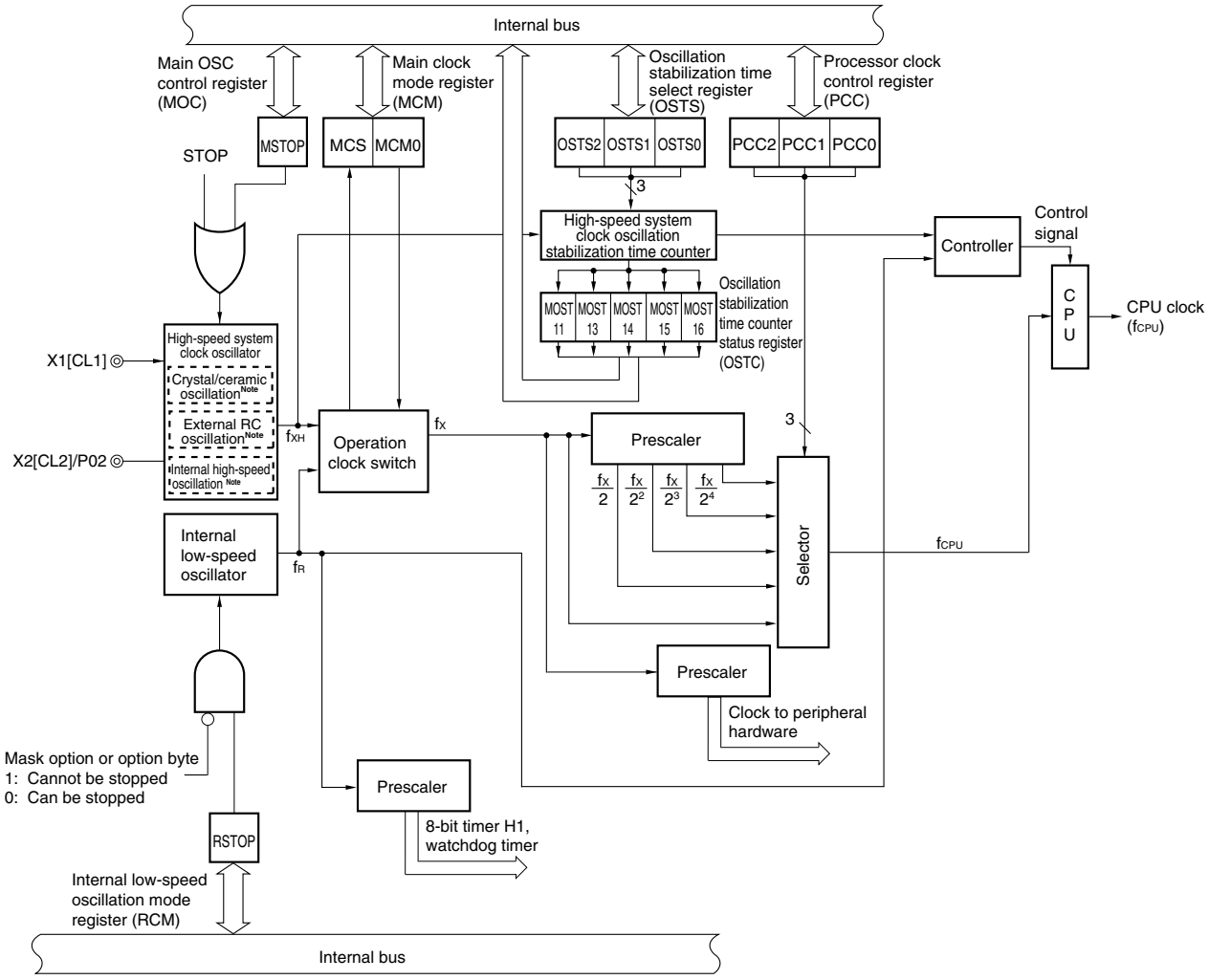
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Processor clock control register (PCC) Internal low-speed oscillation 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)
Oscillators	High-speed system clock oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



Note Select one of these as the high-speed system clock oscillation by a mask option when using a mask ROM version or by an option byte when using a flash memory version.

5.3 Registers Controlling Clock Generator

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

- Processor clock control register (PCC)
- Internal low-speed oscillation 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)

(1) Processor clock control register (PCC)

This register sets the division ratio of the CPU clock.

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

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

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

Address: FFFBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	PCC2	PCC1	PCC0

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

Note Setting is prohibited for (A1) grade products and (A2) grade products.

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 Internal low-speed oscillation frequency)

3. f_R: Internal low-speed oscillation frequency

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

The fastest instruction can be executed in 2 clocks of the CPU clock in the μPD780862 Subseries. Therefore, the relationship between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in Table 5-2.

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

CPU Clock (f_{CPU})[Note 1]	Minimum Instruction Execution Time: $2/f_{CPU}$	
	High-Speed System Clock (at 10 MHz Operation[Note 2])	Internal Low-Speed Oscillation Clock (at 240 kHz (TYP.) Operation)
f_X	0.2 μs	8.3 μs (TYP.)
$f_X/2$	0.4 μs	16.6 μs (TYP.)[Note 3]
$f_X/2^2$	0.8 μs	Setting prohibited
$f_X/2^3$	1.6 μs	Setting prohibited
$f_X/2^4$	3.2 μs	Setting prohibited

Notes 1. The main clock mode register (MCM) is used to set the CPU clock (high-speed system clock/internal low-speed oscillation clock) (see **Figure 5-4**).

2. When crystal/ceramic oscillation is used.

3. Setting is prohibited for (A1) grade products and (A2) grade products.

(2) Internal low-speed oscillation mode register (RCM)

This register sets the operation mode of the internal low-speed oscillator.

This register is valid when "Can be stopped by software" is set for the internal low-speed oscillator by a mask option, and the high-speed system clock is input as the CPU clock. If "Cannot be stopped" is selected for the internal low-speed oscillator by a mask option, 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 this register to 00H.

Figure 5-3. Format of Internal low-Speed oscillation 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	Internal low-speed oscillator oscillating/stopped
0	Internal low-speed oscillator oscillating
1	Internal low-speed oscillator stopped

Caution Make sure that 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/internal low-speed oscillation clock).

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

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

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

Address: FFA1H After reset: 00H R/W

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 internal low-speed oscillation clock
1	Operates with high-speed system clock

MCM0	Selection of clock supplied to CPU
0	Internal low-speed oscillation clock
1	High-speed system clock

Caution When the internal low-speed oscillation clock is selected as the clock to be supplied to the CPU, the divided clock of the internal low-speed oscillator output (f_X) is supplied to the peripheral hardware (f_X = 240 kHz (TYP.)).

Operation of the peripheral hardware with the internal low-speed oscillation clock cannot be guaranteed. Therefore, when the internal low-speed oscillation 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 internal low-speed oscillation clock. Note, however, that the following peripheral hardware can be used when the CPU operates on the internal low-speed oscillation 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**

(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 when the CPU is operating with the internal low-speed oscillation clock. Therefore, this register is valid only when the CPU is operating with the internal low-speed oscillation clock.

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

$\overline{\text{RESET}}$ input clears this register 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 oscillation
0	High-speed system clock oscillating
1	High-speed system clock stopped

Caution Make sure that bit 1 (MCS) of the main clock mode register (MCM) is 0 before setting MSTOP.

(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 internal low-speed oscillation 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 a reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, clock monitor, or WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Caution Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or the internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

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
1	0	0	0	0	$2^{11}/f_{XH}$ min. (204.8 μs min.)
1	1	0	0	0	$2^{13}/f_{XH}$ min. (819.2 μs min.)
1	1	1	0	0	$2^{14}/f_{XH}$ min. (1.64 ms min.)
1	1	1	1	0	$2^{15}/f_{XH}$ min. (3.28 ms min.)
1	1	1	1	1	$2^{16}/f_{XH}$ min. (6.55 ms min.)

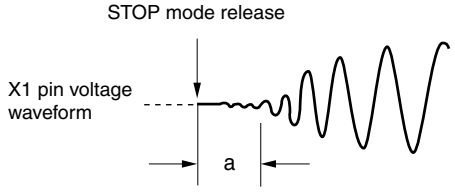
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the internal low-speed oscillation 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.



Remarks 1. Values in parentheses are reference values for operation with f_{XH} = 10 MHz.

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

(6) Oscillation stabilization time select register (OSTS)

This register is used to select the oscillation stabilization wait time of the high-speed system clock when STOP mode is released. The wait time set by OSTS is valid only after the STOP mode is released while the high-speed system clock is selected as the CPU clock. Check the oscillation stabilization time by OSTC after the STOP mode is released when the internal low-speed oscillation clock is selected as the CPU clock.

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
0	0	1	$2^{11}/f_{XH}$ (204.8 μs)
0	1	0	$2^{13}/f_{XH}$ (819.2 μs)
0	1	1	$2^{14}/f_{XH}$ (1.64 ms)
1	0	0	$2^{15}/f_{XH}$ (3.28 ms)
1	0	1	$2^{16}/f_{XH}$ (6.55 ms)
Other than above			Setting prohibited

<R> **Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set OSTS before executing the STOP instruction.**

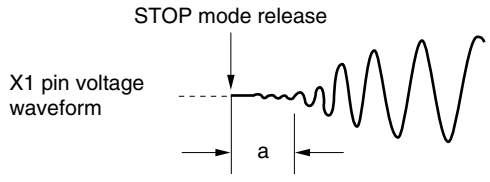
<R> **2. Execute the OSTS setting after confirming that the oscillation stabilization time has elapsed as expected in OSTC.**

3. If the STOP mode is entered and then released while the internal low-speed oscillation 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.

4. 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.



Remarks 1. Values in parentheses are reference values for operation with f_{XH} = 10 MHz.

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

5.4 System Clock Oscillator

5.4.1 High-speed system clock oscillator

The following three high-speed system clock oscillators are available.

- Crystal/ceramic oscillator: Oscillates a clock of 2 to 10 MHz.
- External RC oscillator: Oscillates a clock of 3 to 4 MHz.
- Internal high-speed oscillator: Oscillates a clock of 8.0 MHz (TYP.).

High-speed system clock oscillation can be selected by a mask option when using a mask ROM version or by an option byte when using a flash memory version. For details, refer to **CHAPTER 20 MASK OPTIONS/OPTION BYTE**.

(1) Crystal/ceramic oscillator

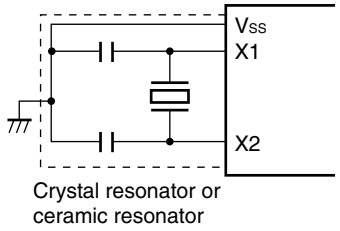
The crystal/ceramic oscillator oscillates via a crystal resonator or ceramic resonator connected to the X1 and X2 pins.

An external clock can be input to the crystal/ceramic oscillator. In this case, input the clock signal to the X1 pin and input the inverse signal to the X2 pin.

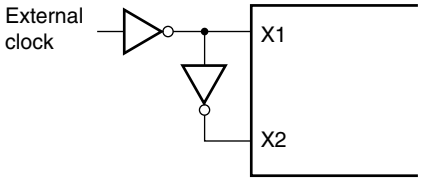
Figure 5-8 shows the external circuit of the crystal/ceramic oscillator.

Figure 5-8. External Circuit of Crystal/Ceramic Oscillator

(a) Crystal/ceramic oscillation



(b) External clock



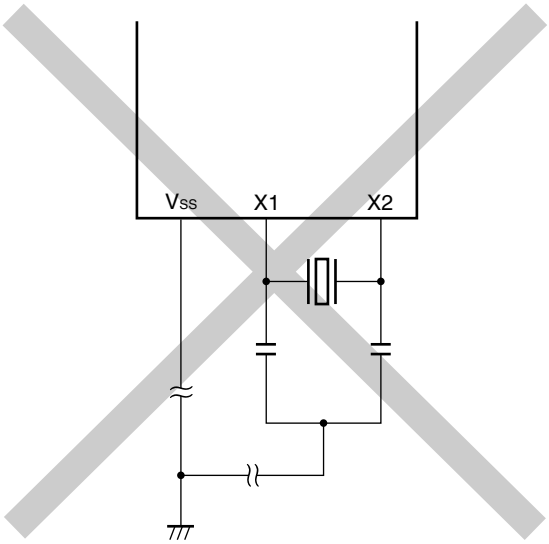
Caution When using the crystal/ceramic oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-9 to avoid an adverse effect from wiring capacitance.

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

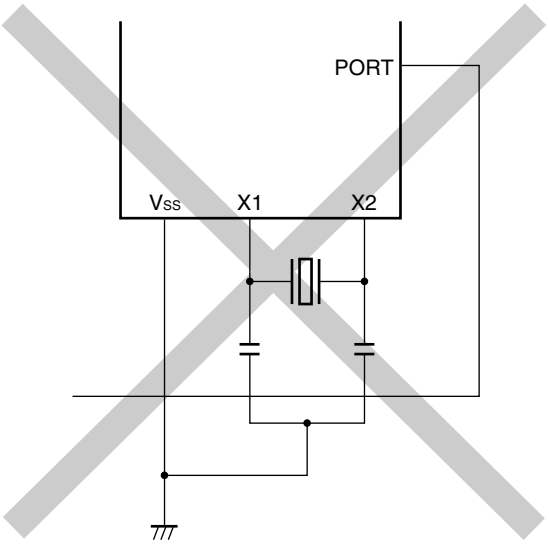
Figure 5-9 shows examples of incorrect resonator connection.

Figure 5-9. Examples of Incorrect Resonator Connection

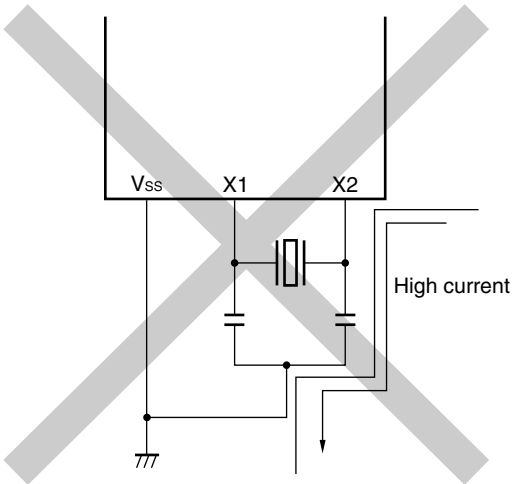
(a) Too long wiring



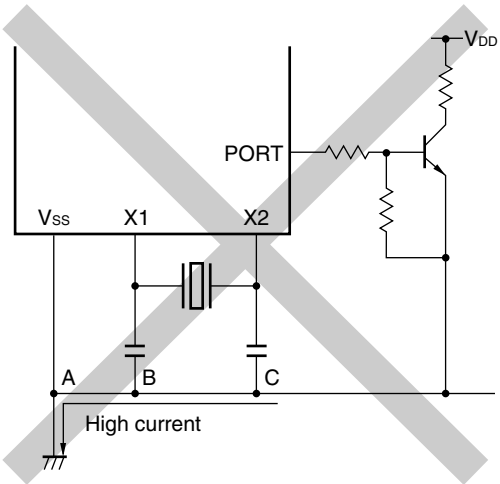
(b) Crossed signal line



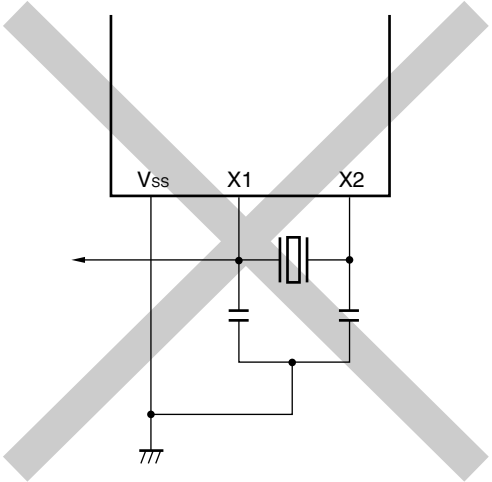
(c) Wiring near high fluctuating current



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



(e) Signals are fetched



(2) External RC oscillator

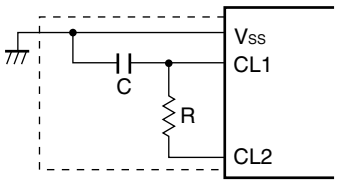
The external RC oscillator is oscillated by the resistor (R) and capacitor (C) connected across the CL1 and CL2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the CL1 pin, and input the inverted signal to the CL2 pin.

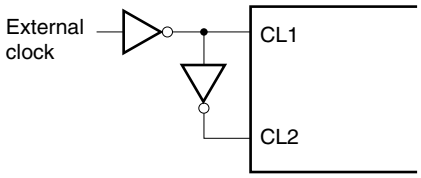
Figure 5-10 shows the external circuit of the external RC oscillator.

Figure 5-10. External Circuit of External RC Oscillator

(a) RC oscillation



(b) External clock



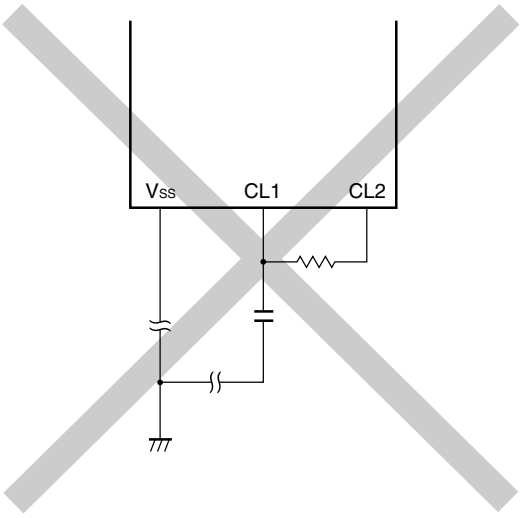
Caution When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-10 to avoid an adverse effect from wiring capacitance.

- **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.**

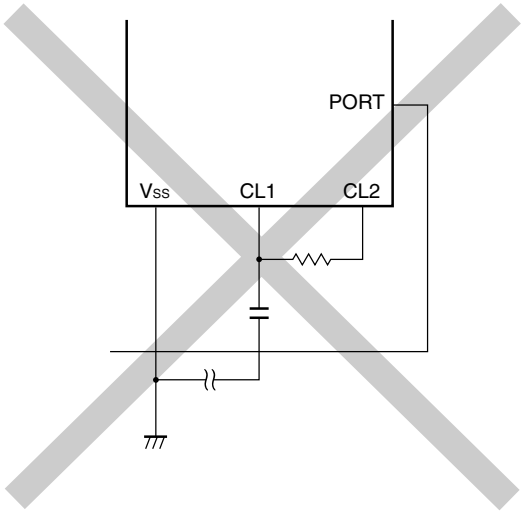
Figure 5-11 shows examples of incorrect resonator connection.

Figure 5-11. Examples of Incorrect Resonator Connection

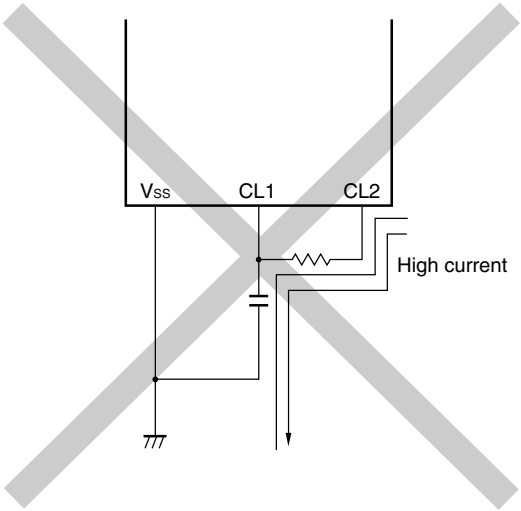
(a) Too long wiring



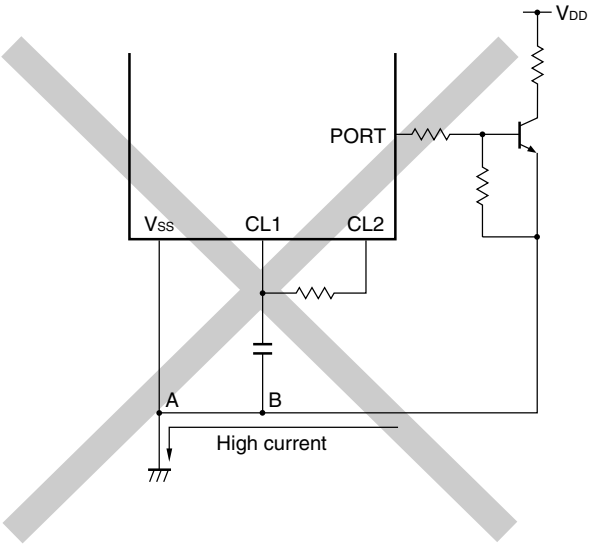
(b) Crossed signal line



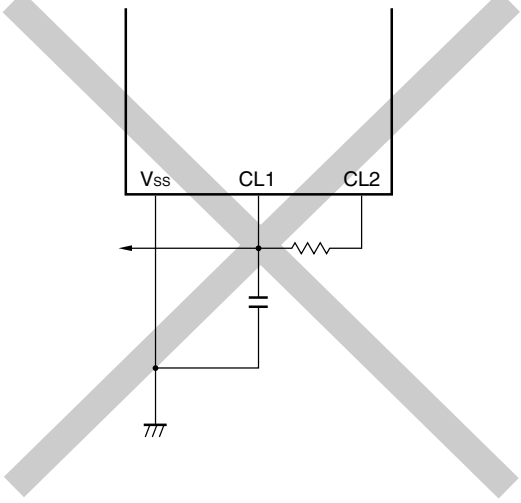
(c) Wiring near high fluctuating current



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



(e) Signal is fetched



(3) Internal high-speed oscillator

The μPD780862 Subseries incorporates an internal high-speed oscillator.

When using the internal high-speed oscillator, handle the X1[CL1] and X2[CL2] pins as follows.

X1[CL1]: Connect directly to V_{DD}.

X2[CL2]: Connect directly to V_{SS}.

Remark The X2[CL2] pin can be used as an input-only pin (P02).

5.4.2 Internal low-speed oscillator

An internal low-speed oscillator is incorporated in the μPD780862 Subseries.

"Can be stopped by software" or "Cannot be stopped" can be selected by a mask option. The internal low-speed oscillation clock always oscillates after $\overline{\text{RESET}}$ release (240 kHz (TYP.)).

5.4.3 Prescaler

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

Caution When the internal low-speed oscillation clock is selected as the clock supplied to the CPU, the prescaler generates various clocks by dividing the internal low-speed oscillator (f_X) (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_{XH}
- Internal low-speed oscillation clock f_R
- CPU clock f_{CPU}
- Clock to peripheral hardware

The internal low-speed oscillation clock via the internal low-speed oscillator is used as the CPU clock after reset release in the μPD780862 Subseries, thus enabling the following.

(1) Enhancement of security function

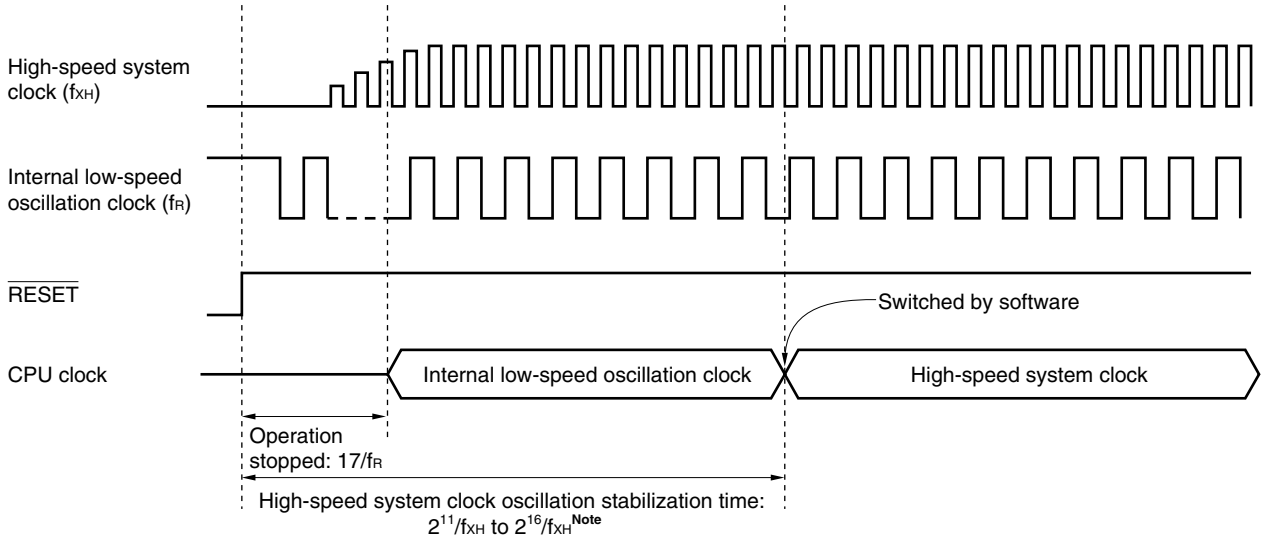
When the high-speed system clock is set as the CPU clock by the default setting, the device cannot operate if the high-speed system clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal low-speed oscillation clock, so the device can be started by the internal low-speed oscillation 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 the internal low-speed oscillation clock is shown in Figure 5-12.

Figure 5-12. Timing Diagram of CPU Default Start Using Internal Low-Speed Oscillation Clock



Note Check using the oscillation stabilization time counter status register (OSTC).
Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or the internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

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

(b) After $\overline{\text{RESET}}$ release, the CPU clock can be switched from the internal low-speed oscillation 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) Internal low-speed oscillator can be set to stopped/oscillating using the internal low-speed oscillation mode register (RCM) when “Can be stopped by software” is selected for the internal low-speed oscillator by a mask option (option byte when using a flash memory version), if the high-speed system clock is used as the CPU clock. Make sure that MCS is 1 at this time.

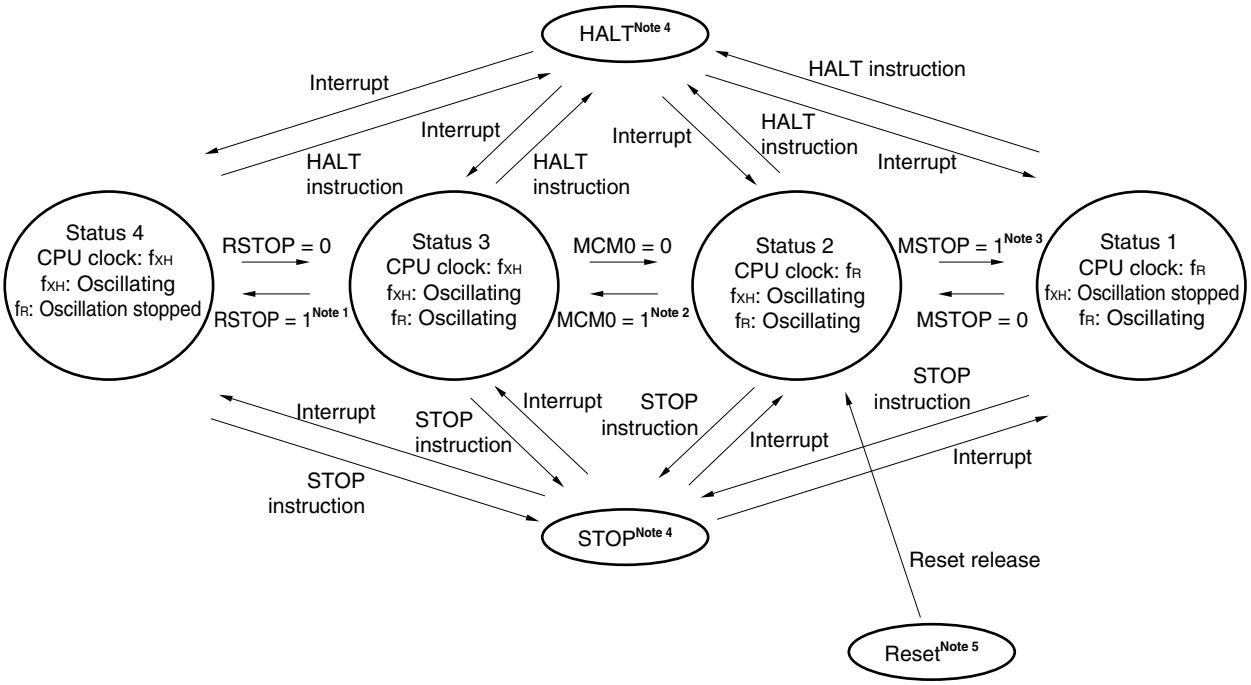
(d) When the internal low-speed oscillation clock is used as the CPU clock, the high-speed system clock can be set to stopped/oscillating using the main OSC control register (MOC). Make sure that MCS is 0 at this time.

(e) Select the high-speed system clock oscillation stabilization time ($2^{11}/f_{XH}$, $2^{13}/f_{XH}$, $2^{14}/f_{XH}$, $2^{15}/f_{XH}$, $2^{16}/f_{XH}$) using the oscillation stabilization time select register (OSTS) when releasing STOP mode while the high-speed system clock is being used as the CPU clock. In addition, when releasing STOP mode while $\overline{\text{RESET}}$ is released and the internal low-speed oscillation clock 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 5-13, 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 5-13. Status Transition Diagram (1/2)

(1) When "Internal low-speed oscillator can be stopped by software" is selected by mask option



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).

Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or the internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

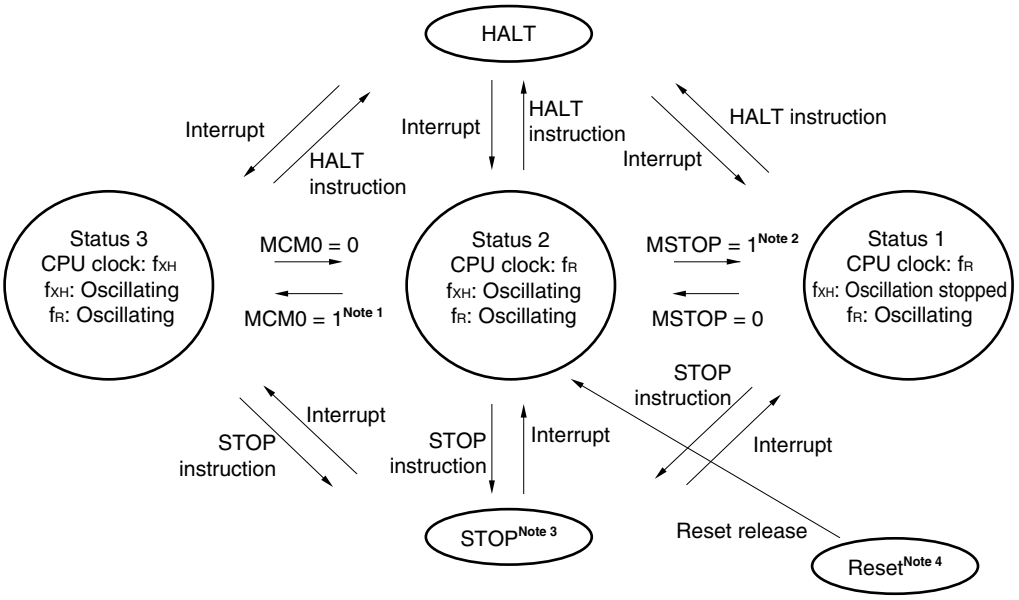
3. When shifting from status 2 to status 1, make sure that MCS is 0.

4. When "Internal low-speed oscillator can be stopped by software" is selected by a mask option (option byte when using a flash memory version), the watchdog timer stops operating in the HALT and STOP modes, regardless of the source clock of the watchdog timer. However, the internal low-speed oscillator 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 5-13. Status Transition Diagram (2/2)

(2) When "Internal low-speed oscillator cannot be stopped" is selected by mask option



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).

Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or the internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

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

3. The watchdog timer operates using the internal low-speed oscillation clock even in STOP mode if "Internal low-speed oscillator cannot be stopped" is selected by a mask option (option byte when using a flash memory version). Internal low-speed oscillation 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)

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

Status / Operation Mode	High-Speed System Clock Oscillator	Internal Low-Speed Oscillator			CPU Clock After Release	Prescaler Clock Supplied to Peripherals	
		Note 1	**Note 2**			MCM0 = 0	MCM0 = 1
			RSTOP = 0	RSTOP = 1			
Reset	Stopped	Stopped			Internal Low-speed oscillation clock	Stopped	
STOP		Oscillating	Oscillating	Stopped	**Note 3**	Stopped	
HALT	Oscillating				**Note 4**	Internal Low-speed Oscillation clock	High-speed system clock

Notes 1. When “Cannot be stopped” is selected for the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

2. When “Can be stopped by software” is selected for the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

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 the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

Remark RSTOP: Bit 0 of the internal low-speed oscillation 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	Internal Low-Speed Oscillation Clock
MSTOP = 1	RSTOP = 0	Stopped	Oscillating
	RSTOP = 1	Setting prohibited	
MSTOP = 0	RSTOP = 0	Oscillating	Oscillating
	RSTOP = 1		Stopped

Caution The RSTOP setting is valid only when “Can be stopped by software” is set for the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
RSTOP: Bit 0 of the internal low-speed oscillation mode register (RCM)

5.6 Time Required to Switch Between Internal Low-Speed Oscillation Clock and High-Speed System Clock

Bit 0 (MCM0) of the main clock mode register (MCM) is used to switch between the Internal low-speed oscillation 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 (see **Table 5-5**).

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

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

Table 5-5. Maximum Time Required to Switch Between Internal Low-Speed Oscillation Clock and High-Speed System Clock

PCC			Maximum Time Required for Switching	
PCC2	PCC1	PCC0	High-Speed System Clock → Internal Low-Speed Oscillation Clock	Internal Low-Speed Oscillation Clock → High-Speed System Clock
0	0	0	f_{XH}/f_R + 1 clock	2 clocks
0	0	1	$f_{XH}/2f_R$ + 1 clock[Note]	2 clocks[Note]

Note When the internal low-speed oscillation clock is used, setting is prohibited for (A1) grade products and (A2) grade products.

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

Remarks 1. PCC: Processor clock control register

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

3. f_R: Internal low-speed 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) 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 (see **Table 5-6**).

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

Set Value Before Switchover			Set Value After Switchover														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 clocks			16 clocks			16 clocks			16 clocks		
0	0	1	8 clocks						8 clocks			8 clocks			8 clocks		
0	1	0	4 clocks			4 clocks						4 clocks			4 clocks		
0	1	1	2 clocks			2 clocks			2 clocks						2 clocks		
1	0	0	1 clock			1 clock			1 clock			1 clock					

Caution When the CPU is operating on the internal low-speed oscillation clock, setting the following values is prohibited.

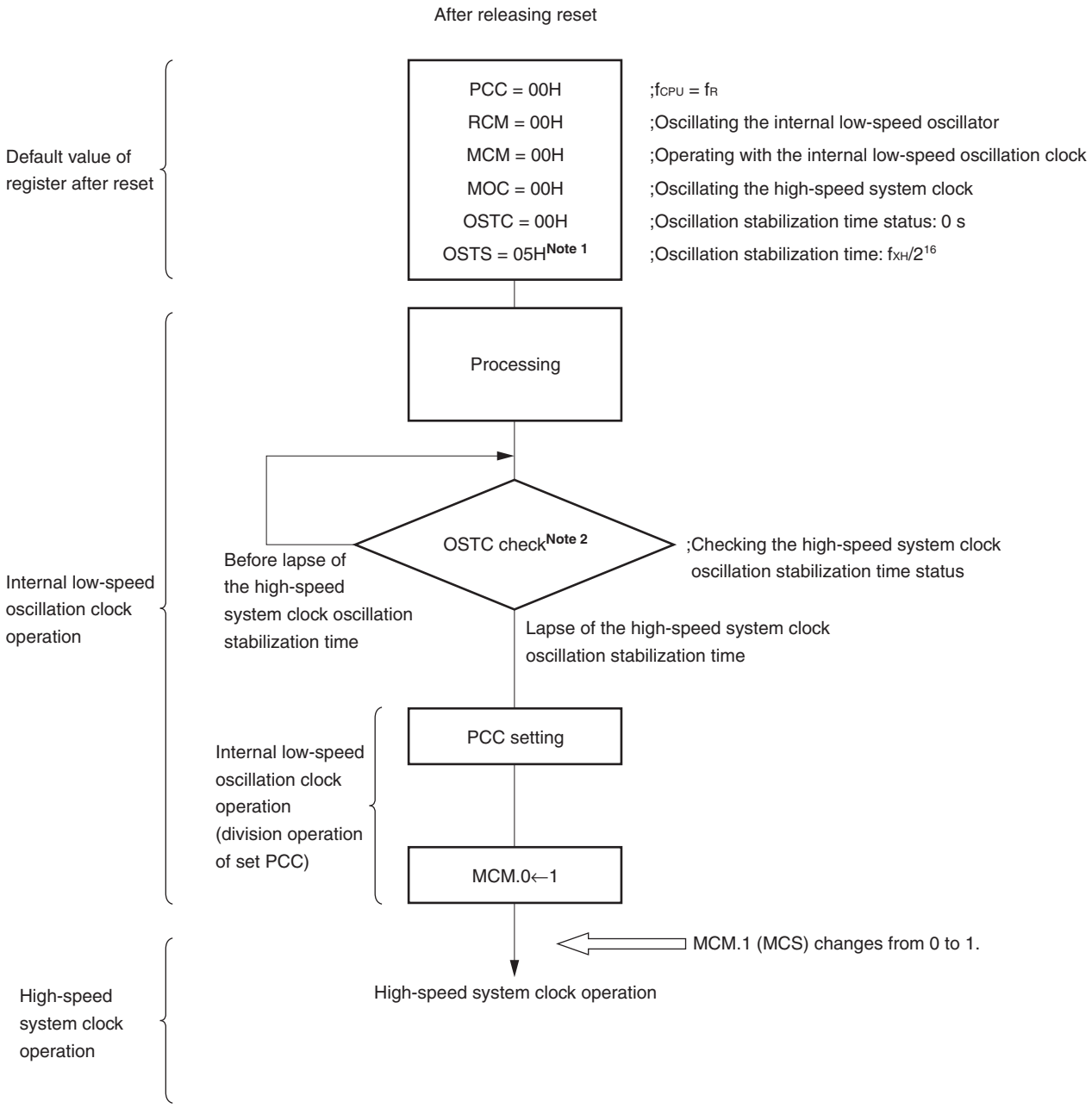
- **PCC2, PCC1, PCC0 = 0, 0, 1 (setting is permitted only for standard products and (A) grade products)**
- **PCC2, PCC1, PCC0 = 0, 1, 0**
- **PCC2, PCC1, PCC0 = 0, 1, 1**
- **PCC2, PCC1, PCC0 = 1, 0, 0**

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

5.8 Clock Selection Flowchart and Register Settings

5.8.1 Changing to high-speed system clock from internal low-speed oscillation clock

Figure 5-14. Changing to High-Speed System Clock from Internal Low-Speed Oscillation Clock (Flowchart)

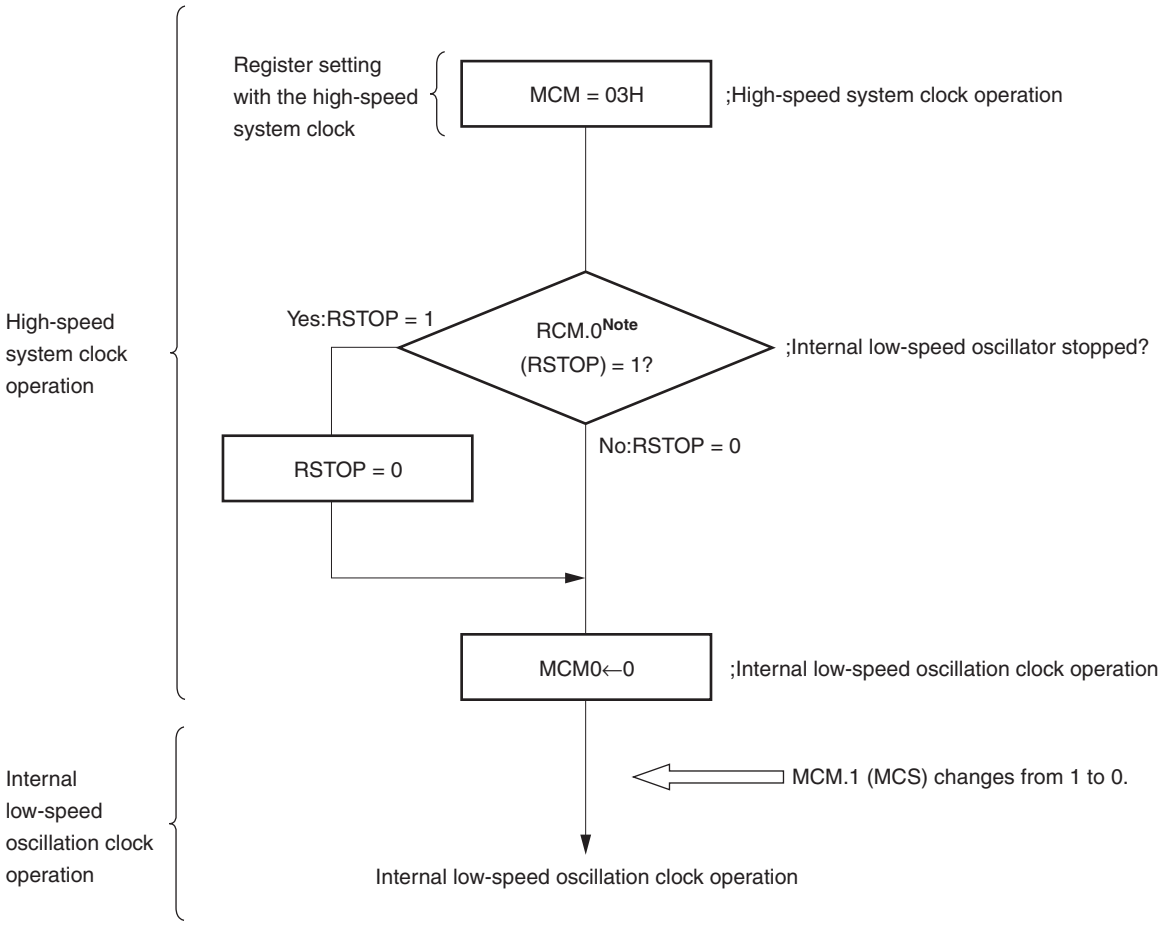


Notes 1. Setting the OSTS register is valid only when the STOP mode has been released with the system operating on the high-speed system clock.

2. Check the oscillation stabilization time of the high-speed system clock oscillator using the OSTC register after the reset signal has been released and select the high-speed system clock operation after the lapse of specified oscillation stabilization time. Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

5.8.2 Changing from high-speed system clock to internal low-speed oscillation clock

Figure 5-15. Changing from High-Speed System Clock to Internal Low-Speed Oscillation Clock (Flowchart)



Note This is necessary only when “clock can be stopped by software” is selected for the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

5.8.3 Register settings

Table 5-7. Clock and Register Settings

f_{CPU}	Mode	Setting Flag			Status Flag
		MCM Register	MOC Register	RCM Register	MCM Register
		MCM0	MSTOP	RSTOP[Note 1]	MCS
High-speed system clock[Note 2]	Internal low-speed Oscillation clock oscillating	1	0	0	1
	Internal low-speed Oscillation clock stopped	1	0	1	1
Internal low-speed oscillation clock	High-speed system clock oscillating	0	0	0	0
	High-speed system clock stopped	0	1	0	0

Notes 1. This is valid only when “clock can be stopped by software” is selected for the internal low-speed oscillator by mask option (option byte when using a flash memory version).

2. Do not set MSTOP to 1 during high-speed system clock operation (oscillation of high-speed system clock is not stopped even when MSTOP = 1).

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

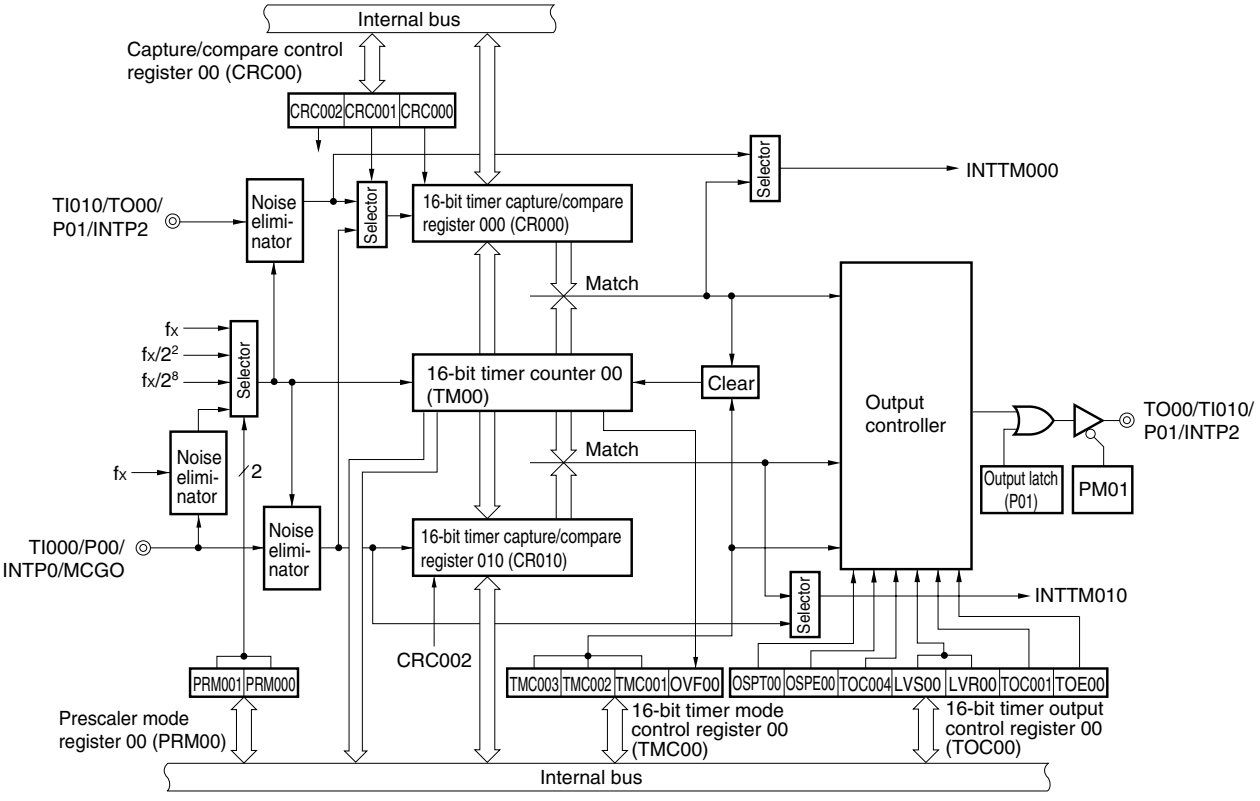
16-bit timer/event counter 00 includes the following hardware.

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) 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

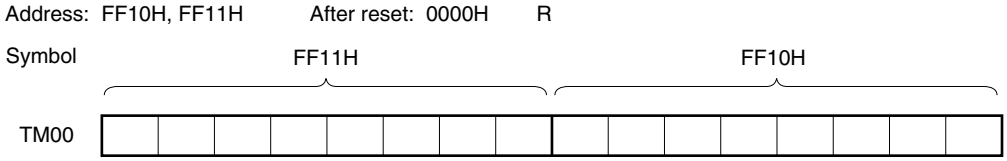


(1) 16-bit timer counter 00 (TM00)

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)



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 TI000 is input in the mode in which clear & start occurs when inputting the valid edge of TI000

<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

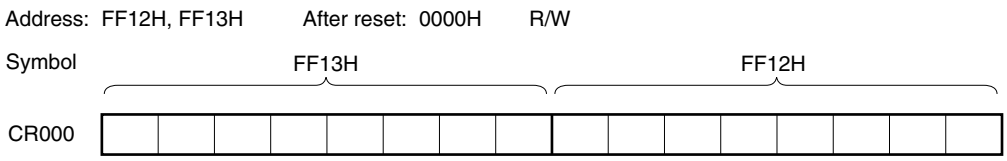
(2) 16-bit timer capture/compare register 000 (CR000)

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)



- **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 pin valid edge is set using prescaler mode register 00 (PRM00) (see **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. In the free-running mode and in the clear mode using the valid edge of TI000, if CR000 is cleared to 0000H, an interrupt request (INTTM000) is generated when the value of CR000 changes from 0000H to 0001H following TM00 overflow (FFFFH). INTTM000 is generated after TM00 and CR000 match, after the valid edge of the TI010 pin is detected, or after the timer is cleared by a one-shot trigger.

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

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 a timer count stop and capture trigger input conflict, the captured data is undefined.

5. Do not rewrite CR000 during TM00 operation.

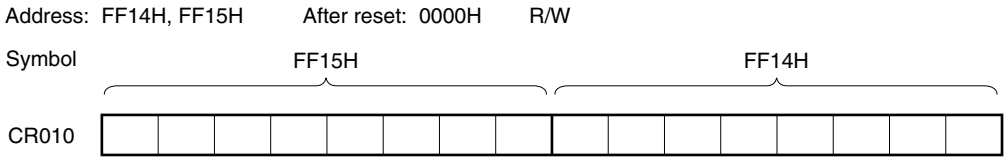
(3) 16-bit timer capture/compare register 010 (CR010)

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 valid edge is set by prescaler mode register 00 (PRM00) (see **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).
INTTM010 is generated after TM00 and CR010 match, after the valid edge of the TI000 pin is detected, or after 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 a timer count stop and capture trigger input conflict, the captured data is undefined.

3. CR010 can be rewritten during TM00 operation. For details, see 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. Set 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

<R>

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	<When used as compare register> Generated on match between TM00 and CR000, or match between TM00 and CR010 <When used as capture register> Generated on TI000 valid edge or TI010 valid edge
0	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 valid edge	
1	0	0	Clear & start occurs on TI000 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 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 pin using prescaler mode register 00 (PRM00).

3. If any of the following modes: the mode in which clear & start occurs on match between TM00 and CR000, the mode in which clear & start occurs at the TI000 valid edge, or free-running mode is selected, 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
1	Captures on valid edge of TI000 by reverse phase [Note]

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

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

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. 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 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. LVS00 and LVR00 are 0 when they are 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, and OSPE00: Timer output operation setting

<2> Set LVS00 and 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 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 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

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

PRM001	PRM000	Count clock selection
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]

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

Cautions 1. When the internal low-speed oscillation clock is selected as the clock to be supplied to the CPU, the clock of the internal low-speed oscillator is divided and supplied as the count clock. If the count clock is the internal low-speed oscillation clock, the operation of 16-bit timer/event counter 00 is not guaranteed. When an external clock is used and when the internal low-speed oscillation clock is selected and supplied to the CPU, the operation of 16-bit timer/event counter 00 is not guaranteed, either, because the internal low-speed oscillation 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 TI000 is to be set for the count clock, do not set the clear & start mode using the valid edge of TI000 and the capture trigger.

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. if the TI000 or TI010 pin is high level when re-enabling operation after the operation has been stopped, the rising edge is not detected.

<R>

Caution 5. When the valid edge of the TI010 pin is used, P01 cannot be used as the timer output pin (TO00). When P01 is used as the TO00 pin, the valid edge of the TI010 pin cannot be used.

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

2. TI000, TI010: 16-bit timer/event counter 00 input pin

3. Figures in parentheses are for operation with fx = 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/INTP2 pin for timer output, set PM01 and the output latch of P01 to 0.

When using the P01/TO00/TI010/INTP2 pin for timer input, set PM01 to 1. The output latch of P01 at this time 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	1	1	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0, 1)
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 (see **Figure 6-10** for the set value).
<2> Set any value to the CR000 register.
<3> Set the count clock by using the PRM000 register.
<4> Set the TMC00 register to start the operation (see **Figure 6-10** for the set value).

Caution Do not rewrite CR000 during TM00 operation.

Remark For how to enable the INTTM000 interrupt, see **CHAPTER 14 INTERRUPT FUNCTIONS**.

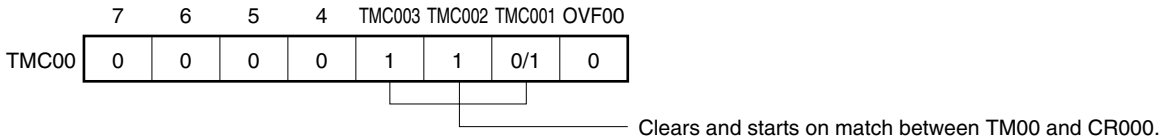
Interrupt requests are generated repeatedly using the count value preset in 16-bit timer capture/compare register 000 (CR000) as the interval.

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.

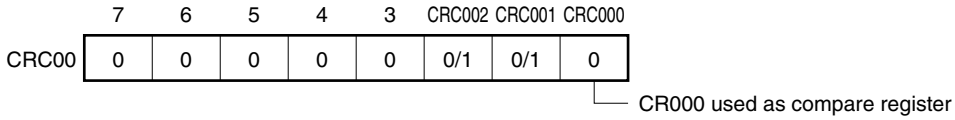
The count clock of 16-bit timer/event counter 00 can be selected with bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00).

Figure 6-10. Control Register Settings for Interval Timer Operation

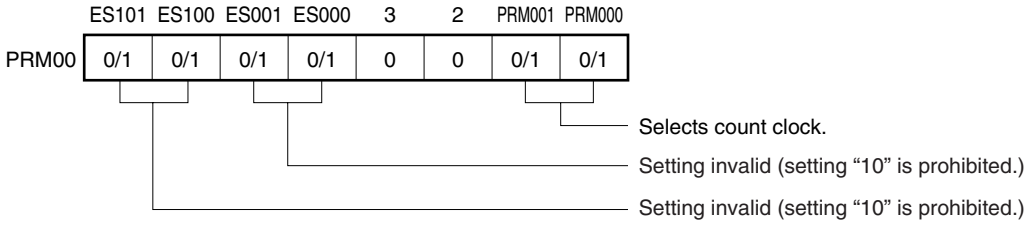
(a) 16-bit timer mode control register 00 (TMC00)



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

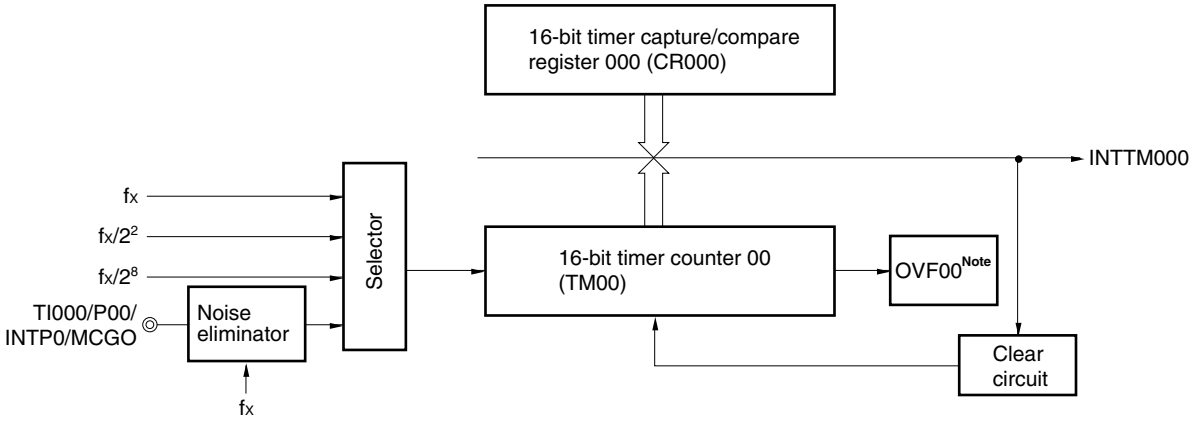


(c) Prescaler mode register 00 (PRM00)



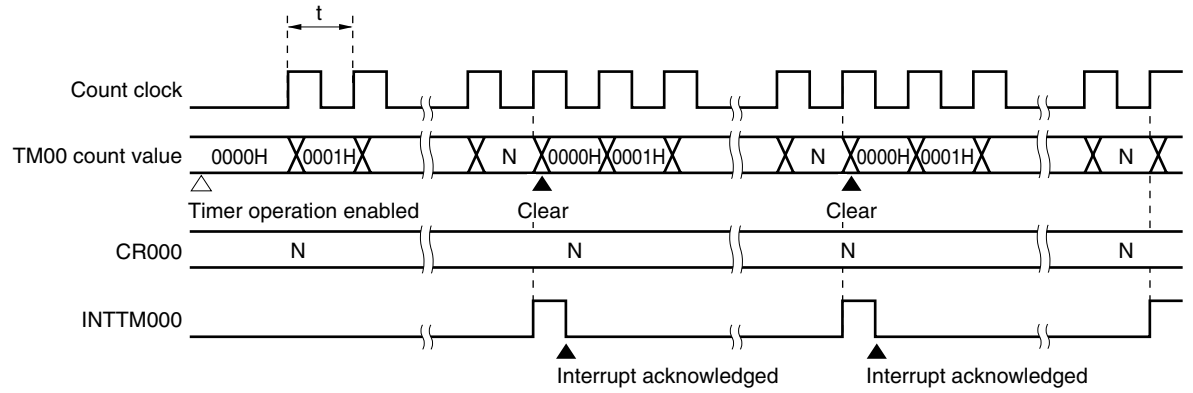
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See the description of the respective control registers for details.

Figure 6-11. Interval Timer Configuration Diagram



Note OVF00 is set to 1 only when CR000 is set to FFFFH.

Figure 6-12. Timing of Interval Timer Operation



Remark Interval time = (N + 1) × t

N = 0001H to FFFFH (settable range)

6.4.2 PPG output operation

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 (see **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 (see **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 (see **Figure 6-13** for the set value).

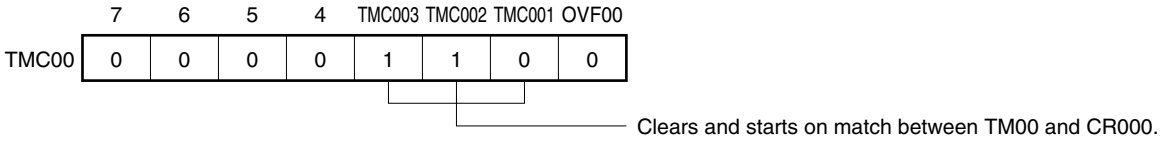
Caution To change the value of the duty factor (the value of the CR010 register) during operation, see Caution 2 in Figure 6-15 PPG Output Operation Timing.

Remarks 1. For the setting of the TO00 pin, see **6.3 (5) Port mode register 0 (PM0)**.
2. For how to enable the INTTM000 interrupt, see **CHAPTER 14 INTERRUPT FUNCTIONS**.

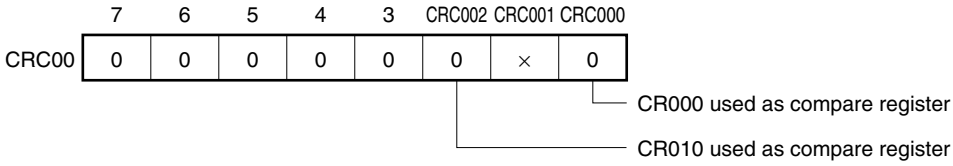
In the PPG output operation, rectangular waves are output from the TO00 pin with the pulse width and the cycle that correspond to the count values preset in 16-bit timer capture/compare register 010 (CR010) and in 16-bit timer capture/compare register 000 (CR000), respectively.

Figure 6-13. Control Register Settings for PPG Output Operation

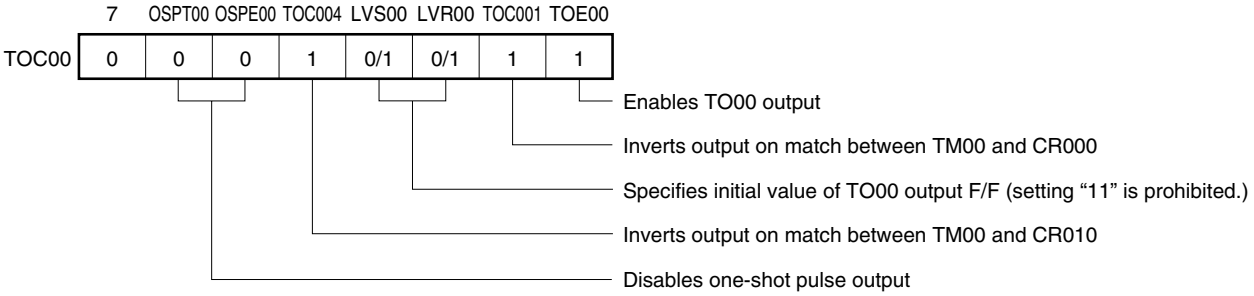
(a) 16-bit timer mode control register 00 (TMC00)



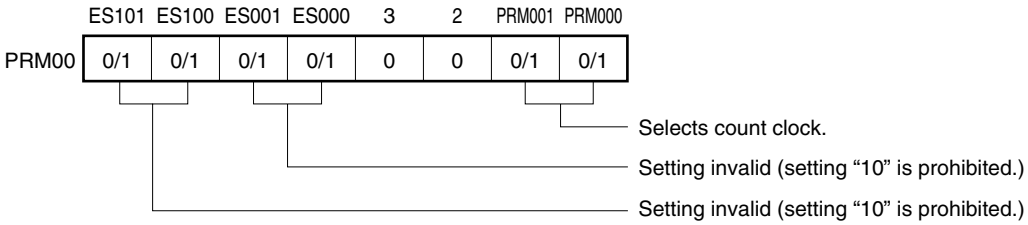
(b) Capture/compare control register 00 (CRC00)



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



(d) Prescaler mode register 00 (PRM00)



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

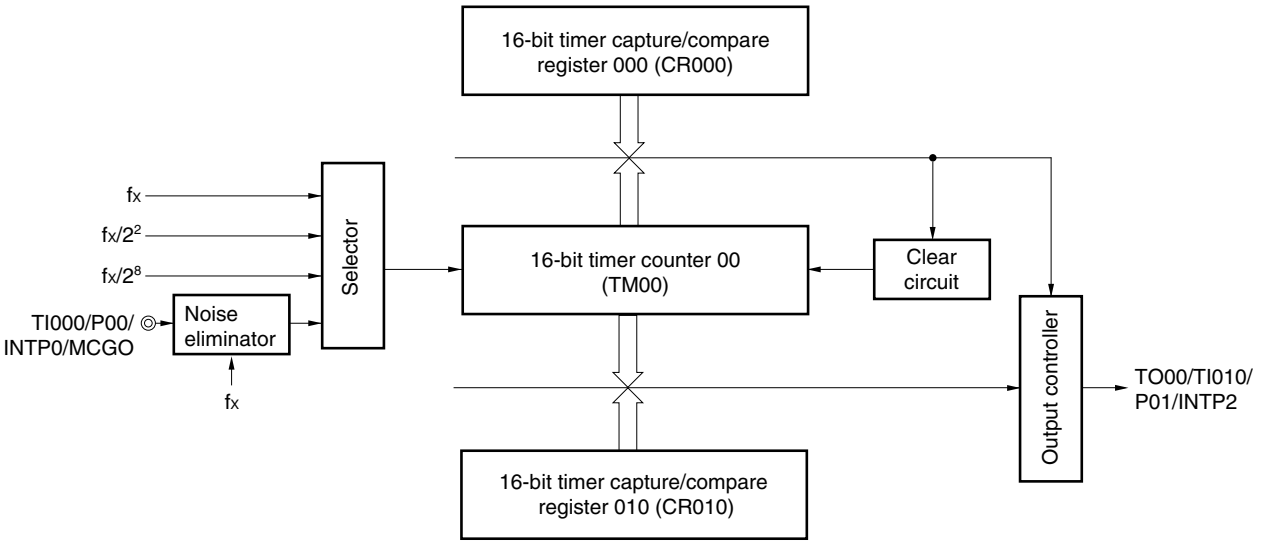
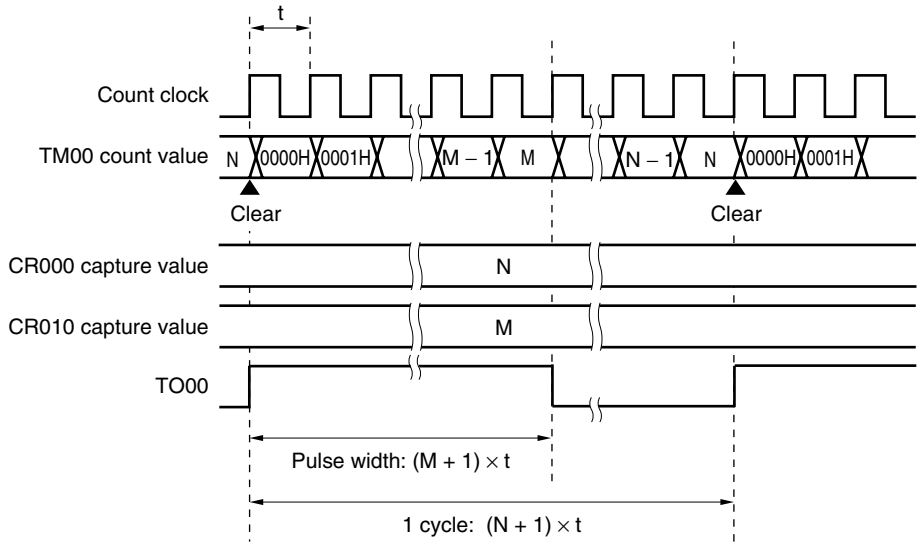


Figure 6-15. PPG Output Operation Timing



Cautions 1. Do not rewrite CR000 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 operation

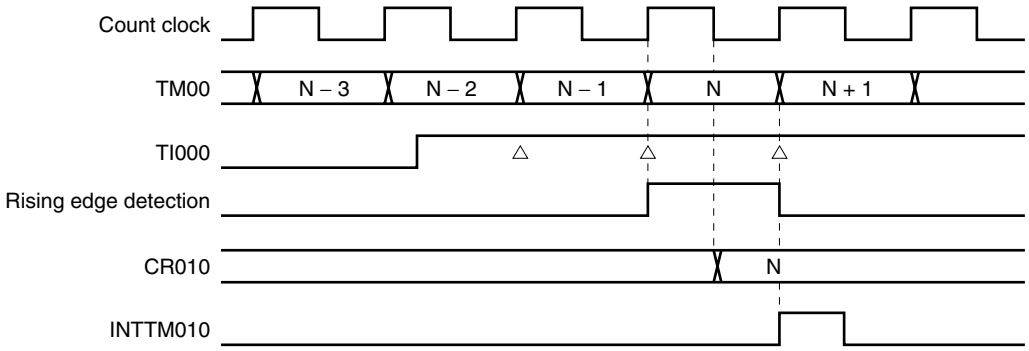
It is possible to measure the pulse width of the signals input to the TI000 pin and TI010 pin using 16-bit timer counter 00 (TM00).

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 (see **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 (see **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, see **6.3 (5) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 (or INTTM010) interrupt, see **CHAPTER 14 INTERRUPT FUNCTIONS**.

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

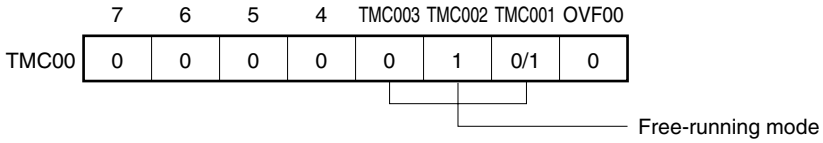
When 16-bit timer counter 00 (TM00) is operated in free-running mode, and the edge specified by prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an external interrupt request signal (INTTM010) is set.

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

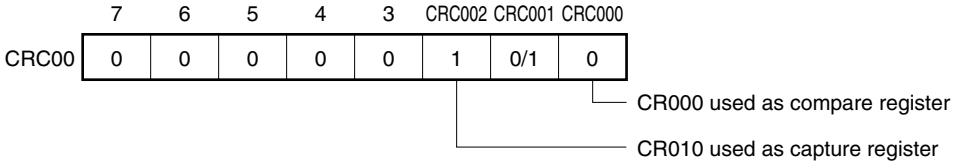
Sampling is performed using the count clock selected by PRM00, and a capture operation is only performed when the valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

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

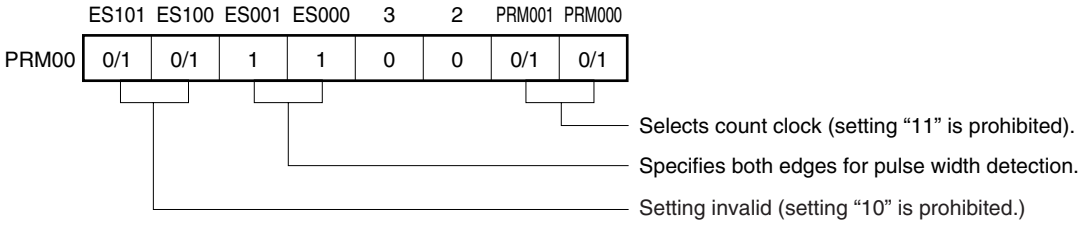
(a) 16-bit timer mode control register 00 (TMC00)



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



(c) Prescaler mode register 00 (PRM00)



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

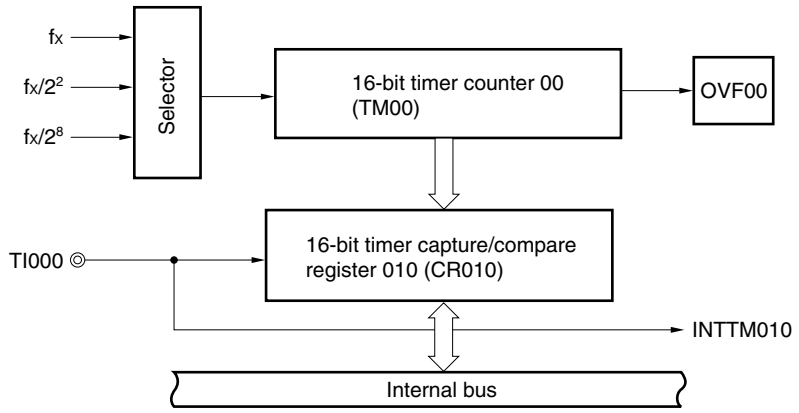
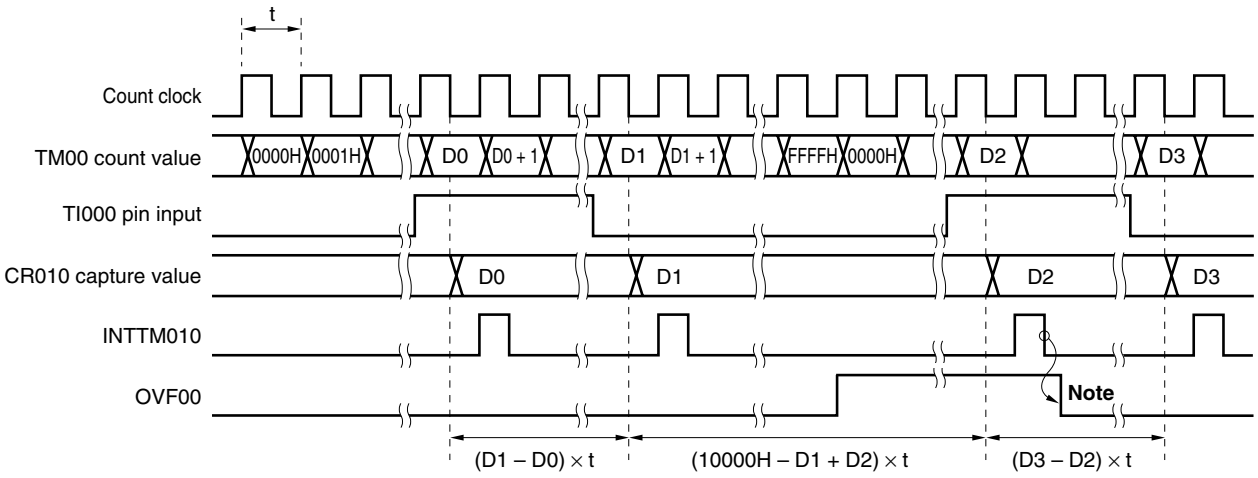


Figure 6-19. Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)



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.

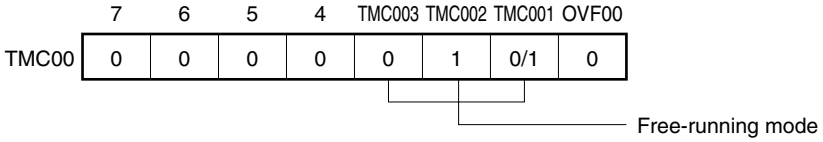
Also, when the edge specified by bits 6 and 7 (ES100 and ES101) of PRM00 is input to the TI010 pin, the value of TM00 is taken into 16-bit timer capture/compare register 000 (CR000) and an interrupt request signal (INTTM000) is set.

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.

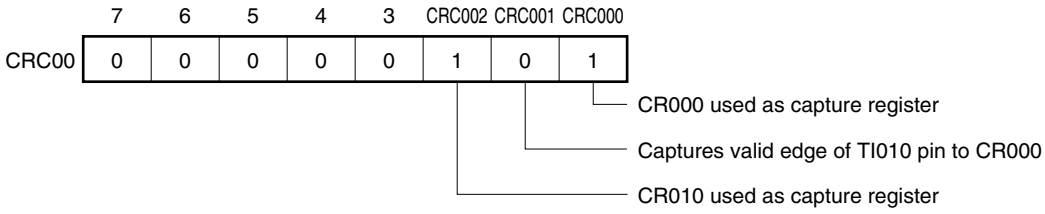
Sampling is performed at the interval selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when the valid level of the TI000 pin or TI010 pin is detected twice, thus eliminating noise with a short pulse width.

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

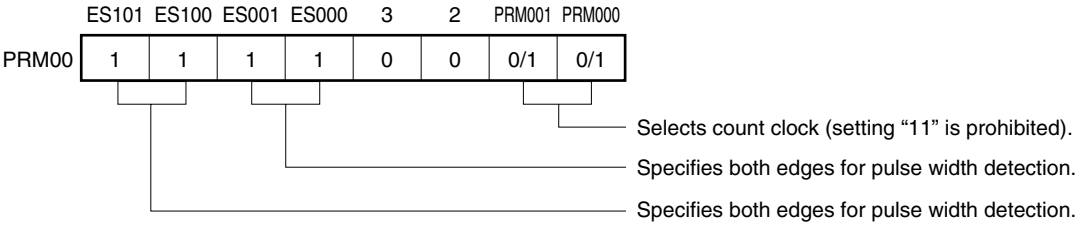
(a) 16-bit timer mode control register 00 (TMC00)



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

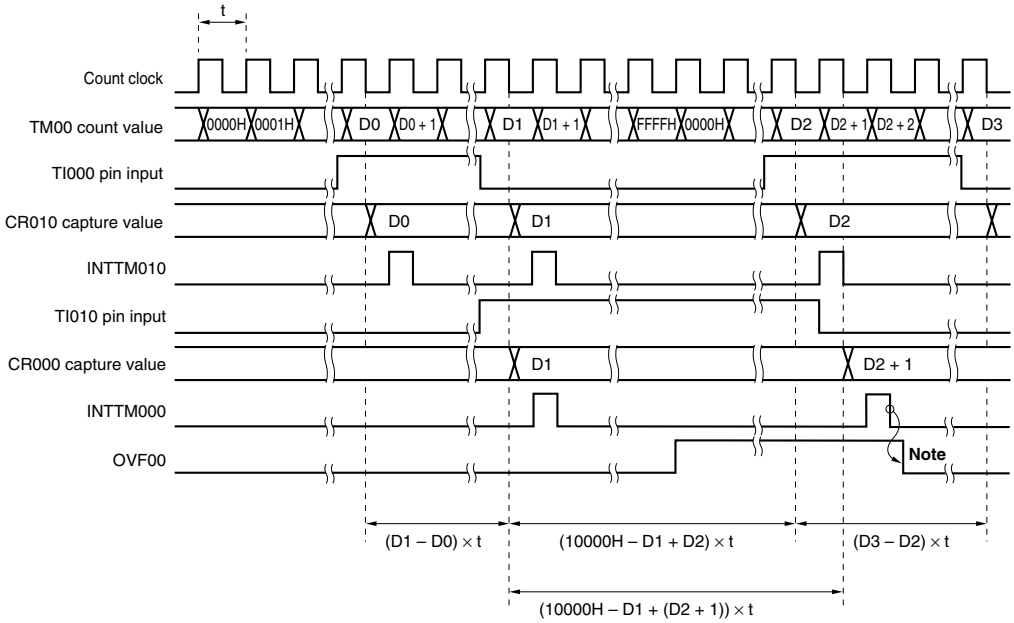


(c) Prescaler mode register 00 (PRM00)



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 rising or falling 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 at the interval selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when the 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

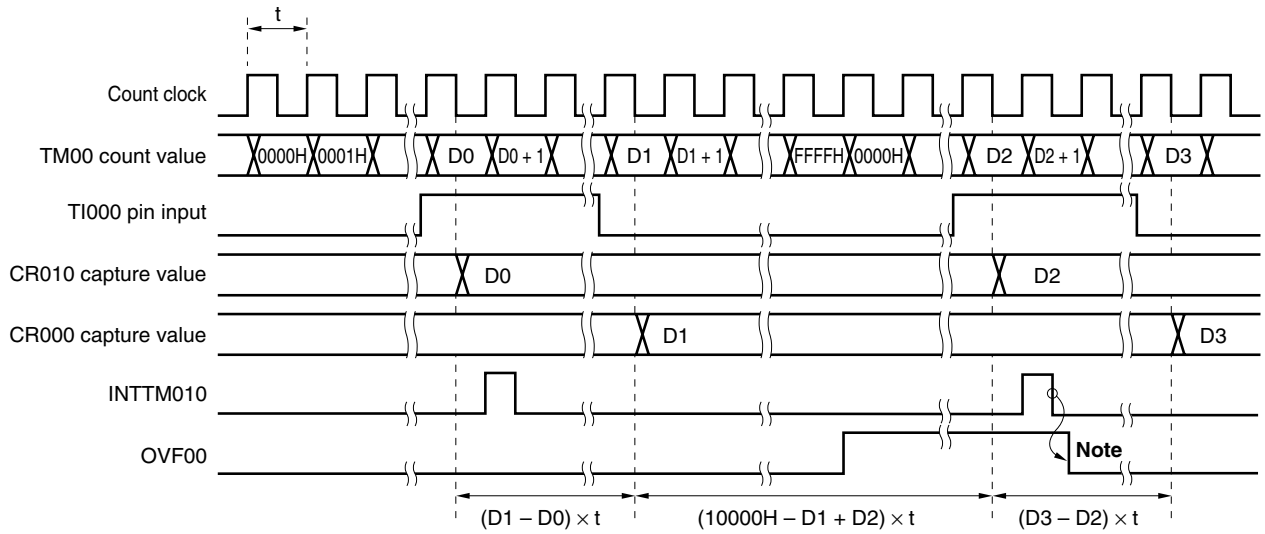
Selects count clock (setting “11” is prohibited).

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

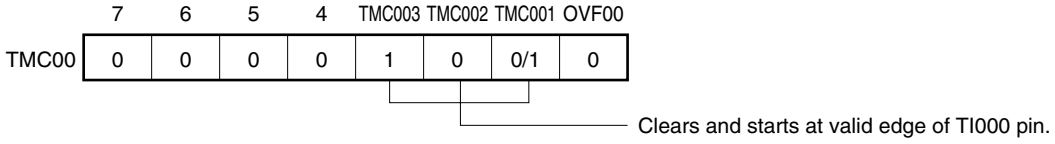
When input of a valid edge to the TI000 pin is detected, the count value of 16-bit timer counter 00 (TM00) is taken into 16-bit timer capture/compare register 010 (CR010), and then the pulse width of the signal input to the TI000 pin is measured by clearing TM00 and restarting the count operation.

Either of two edges—rising or falling—can be selected using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

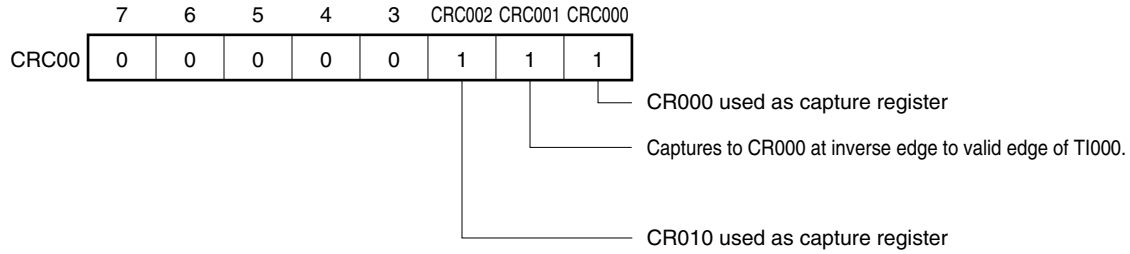
Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00) and a capture operation is only performed when the valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

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)



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



(c) Prescaler mode register 00 (PRM00)

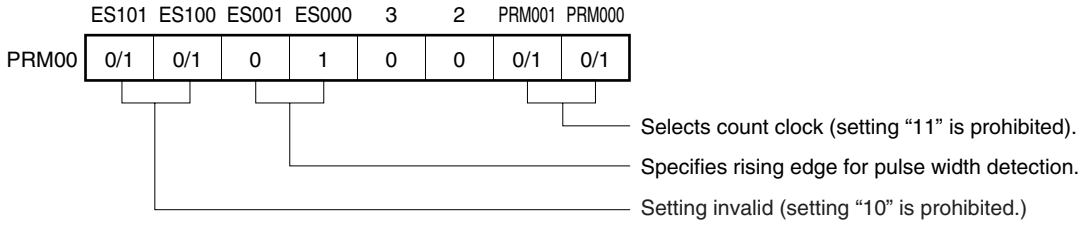
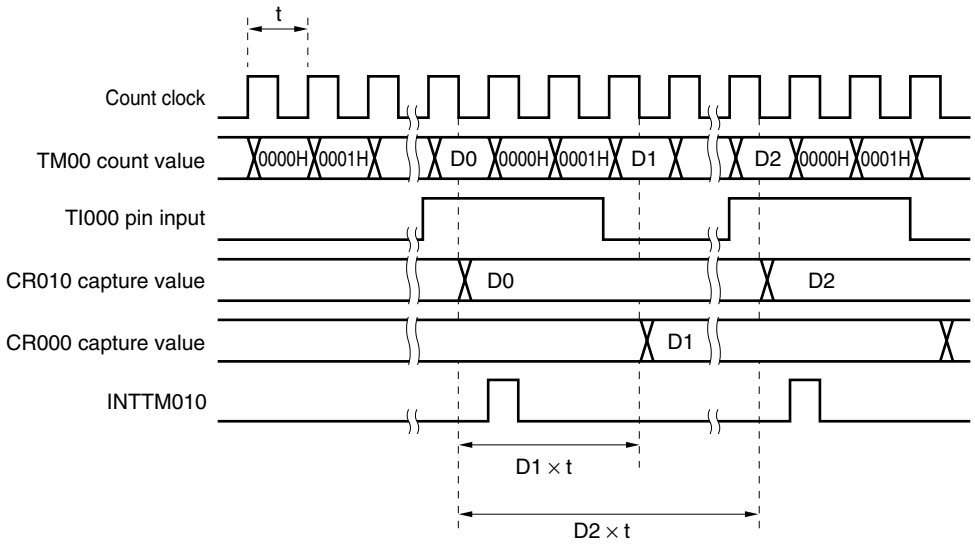


Figure 6-25. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.4.4 External event counter operation

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **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 (see **Figure 6-26** for the set value).

Remarks 1. For the setting of the TI000 pin, see **6.3 (5) Port mode register 0 (PM0)**.
2. For how to enable the INTTM000 interrupt, see **CHAPTER 14 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.

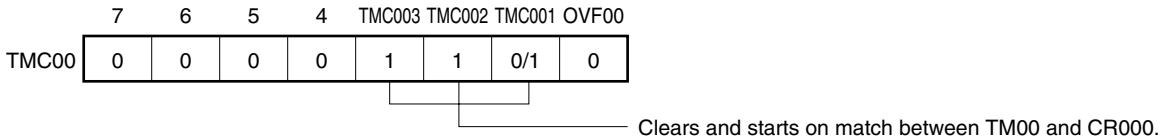
Input a value other than 0000H to CR000 (a count operation with 1-bit pulse cannot be carried out).

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).

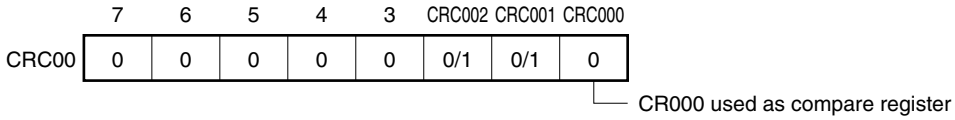
Sampling is performed using the internal clock (fx) and an operation is only performed when the valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-26. Control Register Settings in External Event Counter Mode (with Rising Edge Specified)

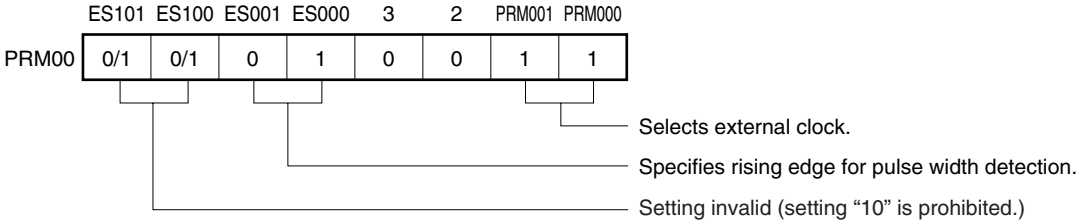
(a) 16-bit timer mode control register 00 (TMC00)



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

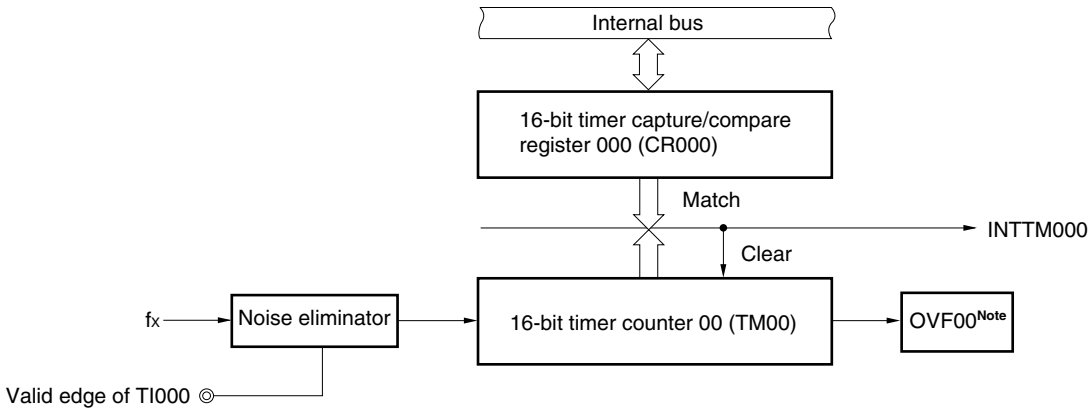


(c) Prescaler mode register 00 (PRM00)



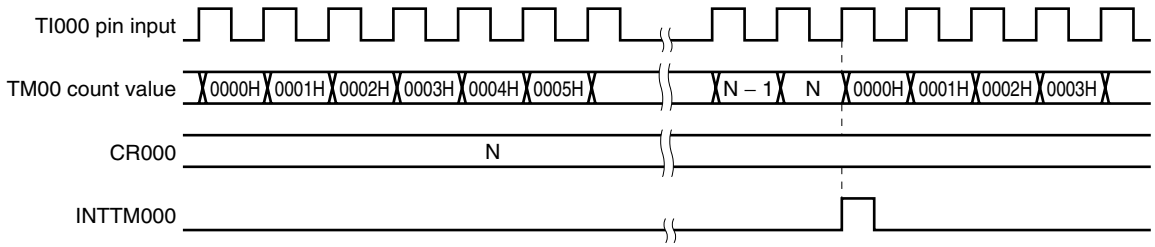
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See the description of the respective control registers for details.

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 (see **Figure 6-29** for the set value).
<3> Set the TOC00 register (see **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 (see **Figure 6-29** for the set value).

Caution Do not rewrite CR000 during TM00 operation.

Remarks 1. For the setting of the TO00 pin, see **6.3 (5) Port mode register 0 (PM0)**.
2. For how to enable the INTTM000 interrupt, see **CHAPTER 14 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 inverted 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.

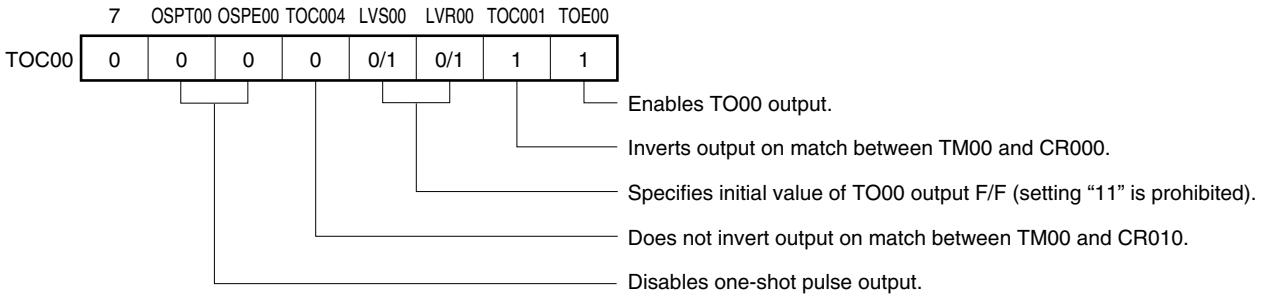
(b) Capture/compare control register 00 (CRC00)

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

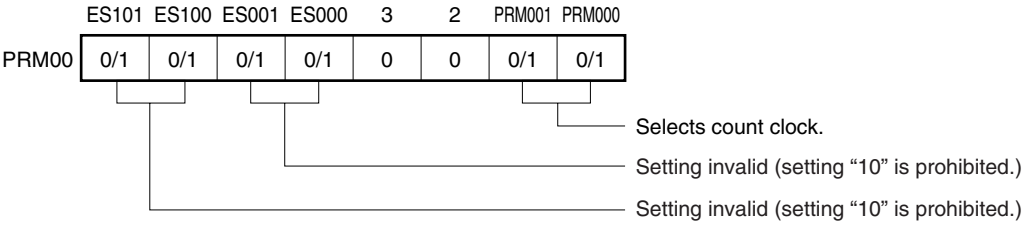
CR000 used as compare register

Figure 6-29. Control Register Settings in Square-Wave Output Mode (2/2)

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

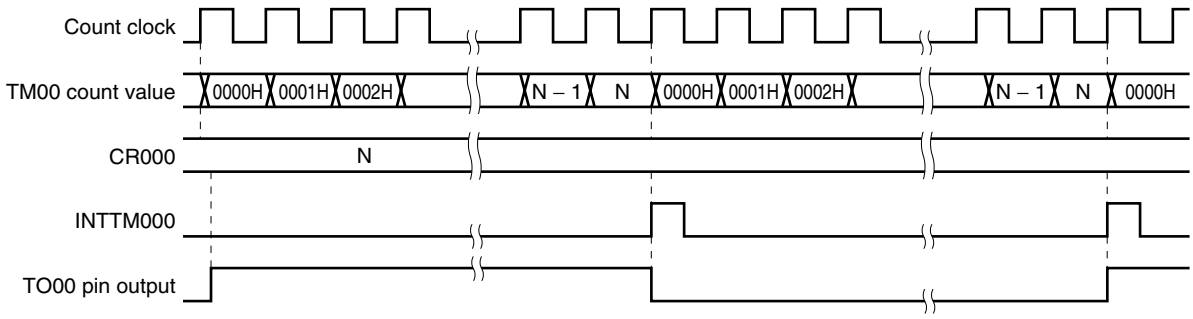


(d) Prescaler mode register 00 (PRM00)



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 (see **Figures 6-31** and **6-33** for the set value).
<3> Set the TOC00 register (see **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 (see **Figures 6-31** and **6-33** for the set value).

Remarks 1. For the setting of the TO00 pin, see **6.3 (5) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 (if necessary, INTTM010) interrupt, see **CHAPTER 14 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 set 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.

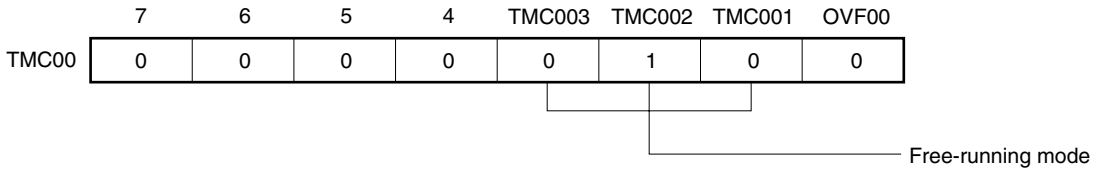
<R> **Cautions 1. Do not set the OSPT00 bit to 1 again while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is completed.**

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.

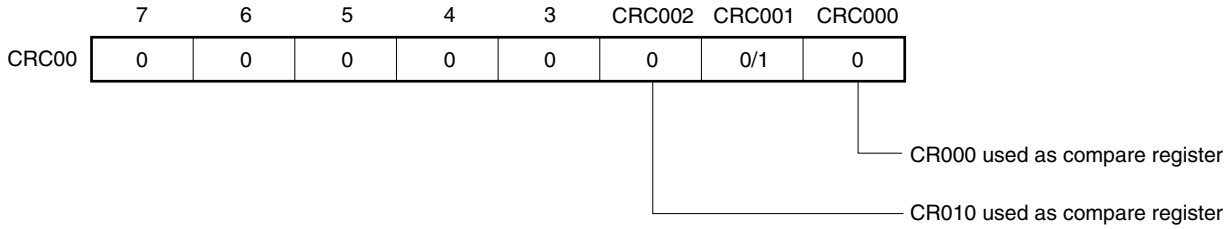
Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI000 pin or its alternate-function port pin, resulting in the output of a pulse at an undesired timing.

Figure 6-31. Control Register Settings for One-Shot Pulse Output with Software Trigger

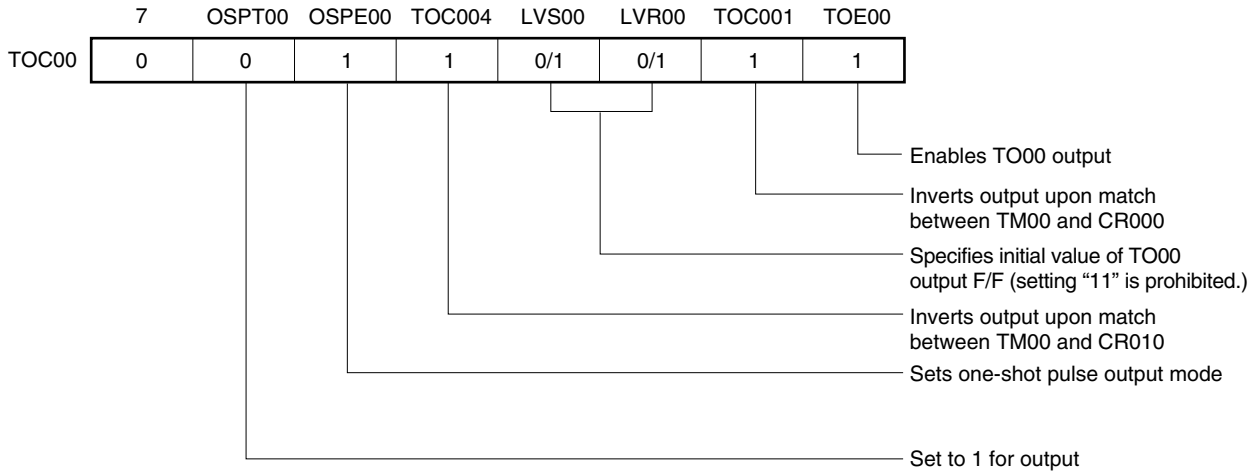
(a) 16-bit timer mode control register 00 (TMC00)



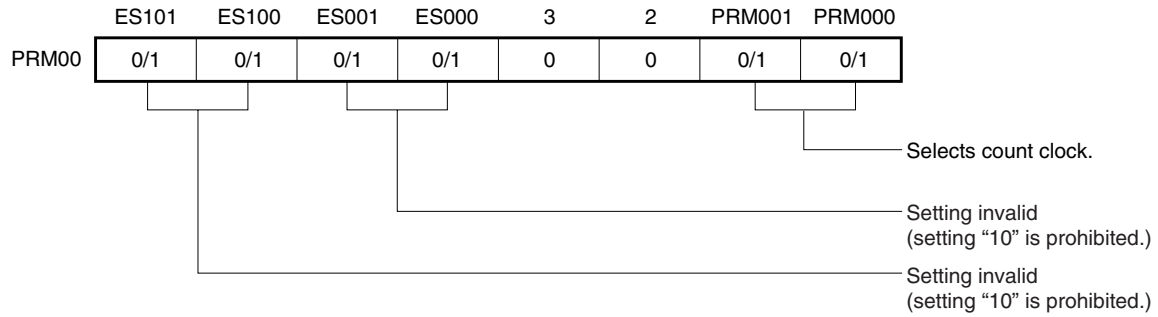
(b) Capture/compare control register 00 (CRC00)



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

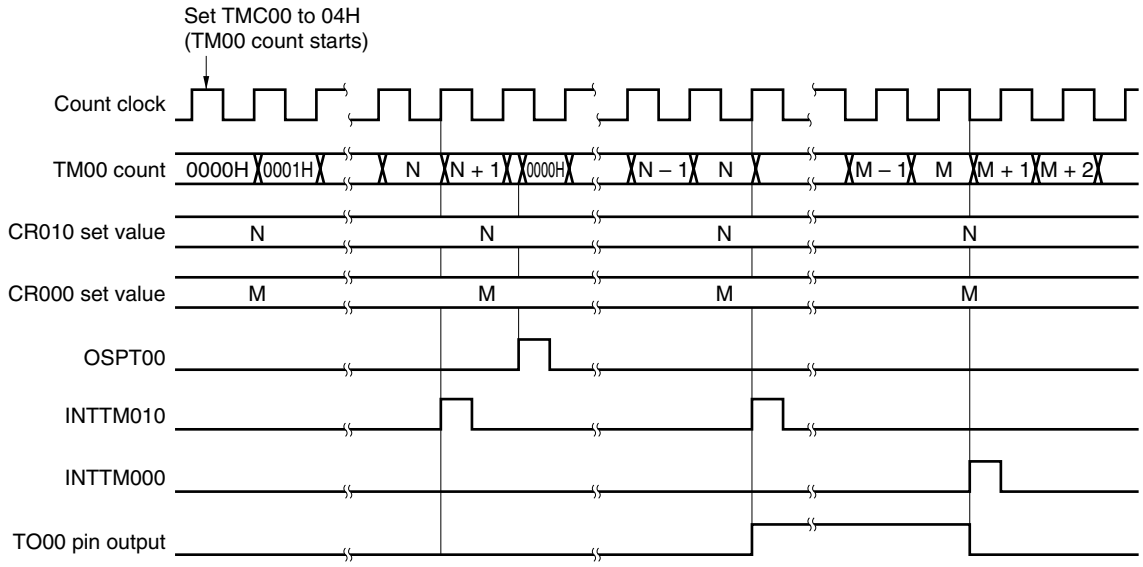


(d) Prescaler mode register 00 (PRM00)



Caution Do not set 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 the TMC003 and TMC002 bits are set to a value other than 00 (operation stop mode).

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, ES001) of prescaler mode register 00 (PRM00). The rising, falling, or both the rising and falling edges can be specified.

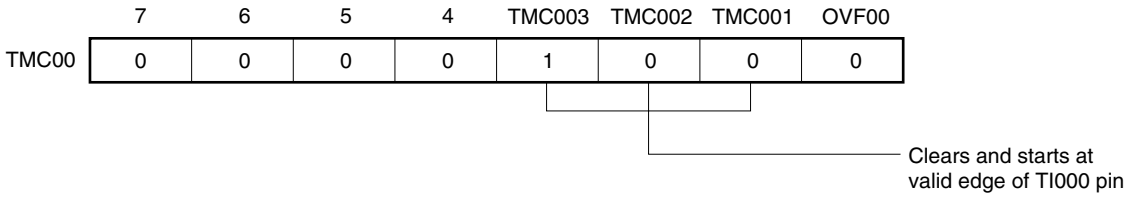
When the valid edge of the TI000 pin is detected, the 16-bit timer/event counter is cleared and started, and the output becomes active at the count value set in advance to 16-bit timer capture/compare register 010 (CR010). After that, the output becomes inactive at the count value set in advance to 16-bit 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.

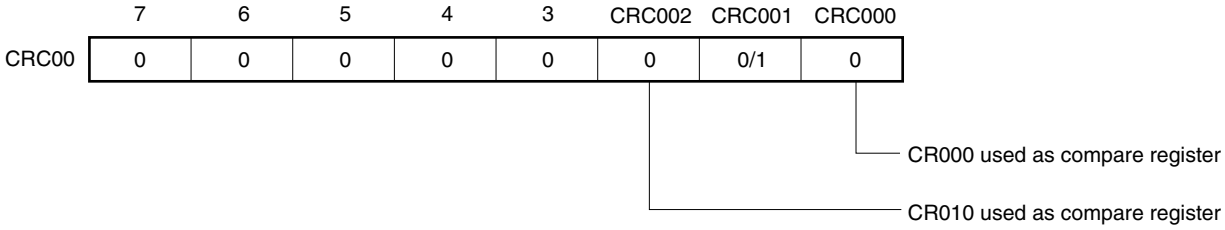
<R> **Caution Do not input the external trigger again while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is completed**

Figure 6-33. Control Register Settings for One-Shot Pulse Output with External Trigger (with Rising Edge Specified)

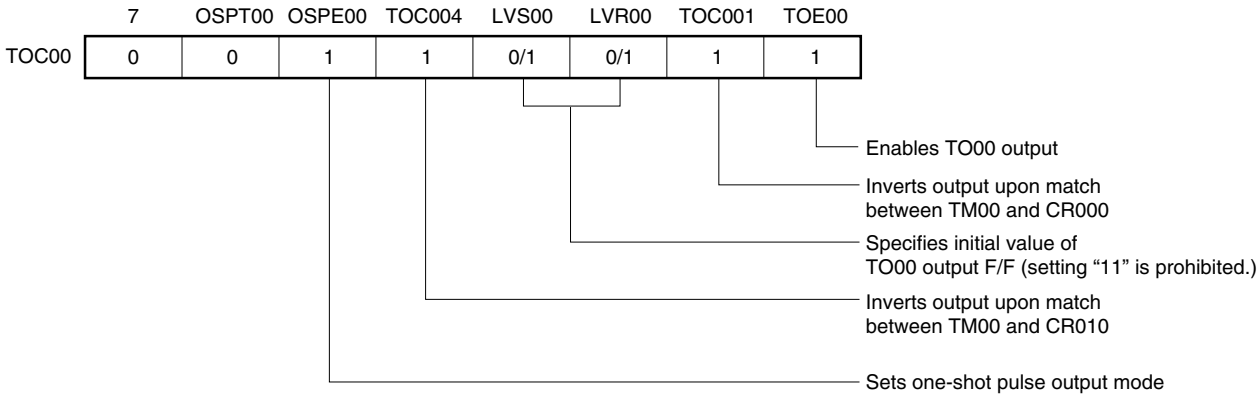
(a) 16-bit timer mode control register 00 (TMC00)



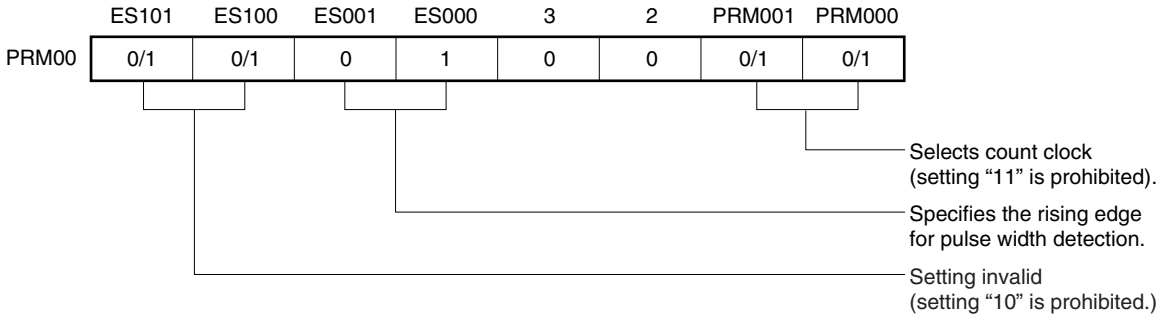
(b) Capture/compare control register 00 (CRC00)



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

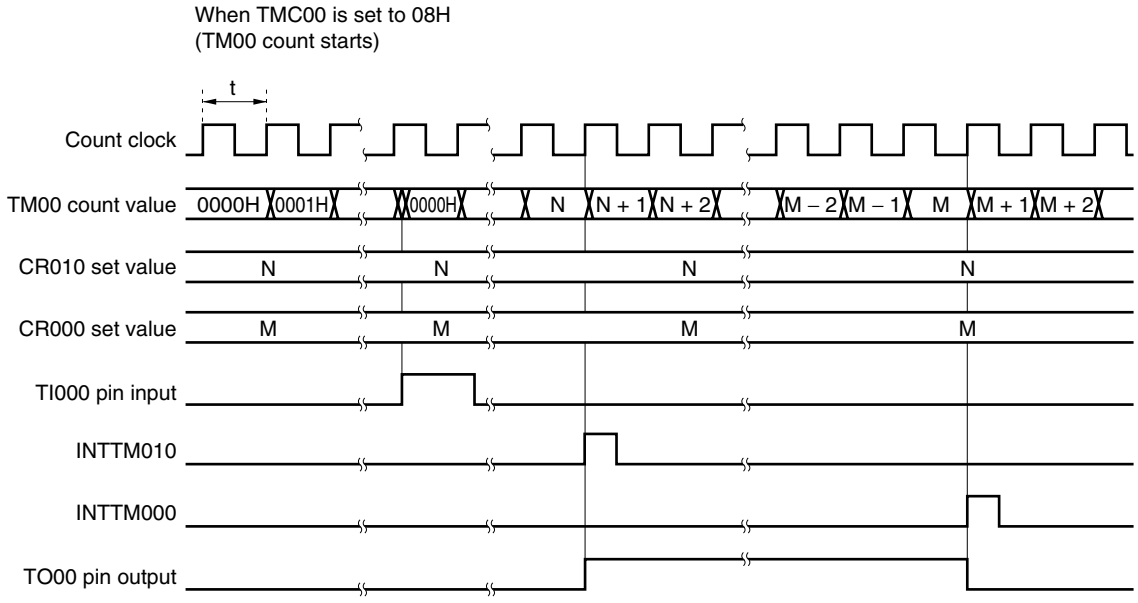


(d) Prescaler mode register 00 (PRM00)



Caution Do not set 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 the TMC002 and TMC003 bits are set to a value other than 00 (operation stop mode).

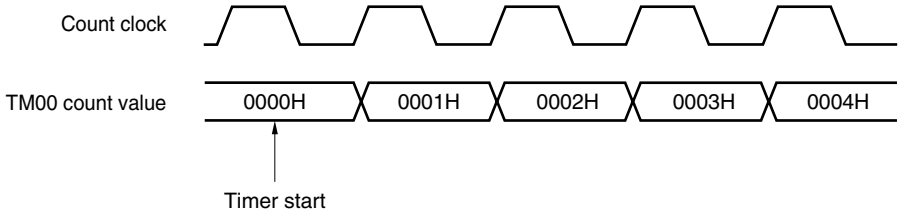
Remark N < M

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) Setting of 16-bit timer capture/compare register 000

In the mode in which clear & start occurs on match between TM00 and CR000, set 16-bit timer capture/compare register 000 (CR000) to a value 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 timing

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 setting 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

<R> Do not set the OSPT00 bit to 1 again while a one-shot pulse is being output.
To output the one-shot pulse again, wait until the current one-shot pulse output is completed.

(b) One-shot pulse output with external trigger

<R> Do not input the external trigger again while a one-shot pulse is being output.
To output the one-shot pulse again, wait until the current one-shot pulse output is completed.

(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: the mode in which clear & start occurs on a match between TM00 and CR000, the mode in which clear & start occurs on a TI000 valid edge, or the free-running mode, is selected

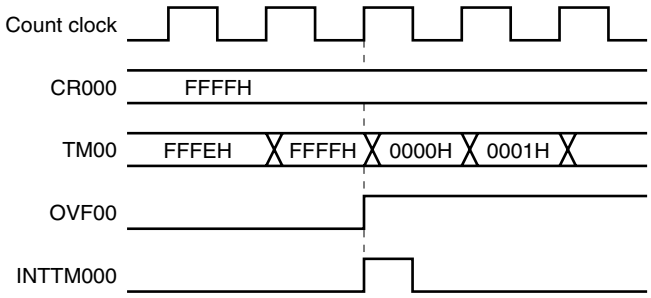
↓

CR000 is set to FFFFH

↓

TM00 is counted up from FFFFH to 0000H.

Figure 6-36. Operation Timing of OVF00 Flag

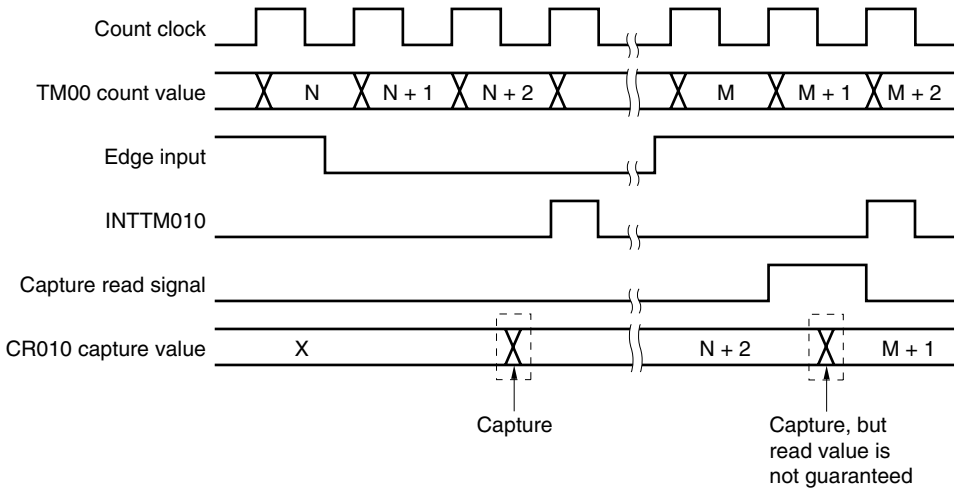


<2> Even if the OVF00 flag is cleared before the next count clock is counted (before TM00 becomes 0001H) after the occurrence of TM00 overflow, the OVF00 flag is re-set newly and clear is disabled.

(7) Conflicting operations

When a read period of the 16-bit timer capture/compare register (CR000/CR010) and a capture trigger input(CR000/CR010 used as capture register) conflict, the priority is given to the capture trigger input. 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 TM00 and CR000, one-shot pulse output is not possible because an overflow does not occur.

(9) Capture operation

<1> If TI000 valid edge is specified as the count clock, a capture operation by the capture register specified as the trigger for TI000 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 <R> therefore when pulling up the TI000 or TI010 pin. However, if the TI000 or TI010 pin is high level when reenabling operation after the operation has been stopped, the rising edge is not detected.

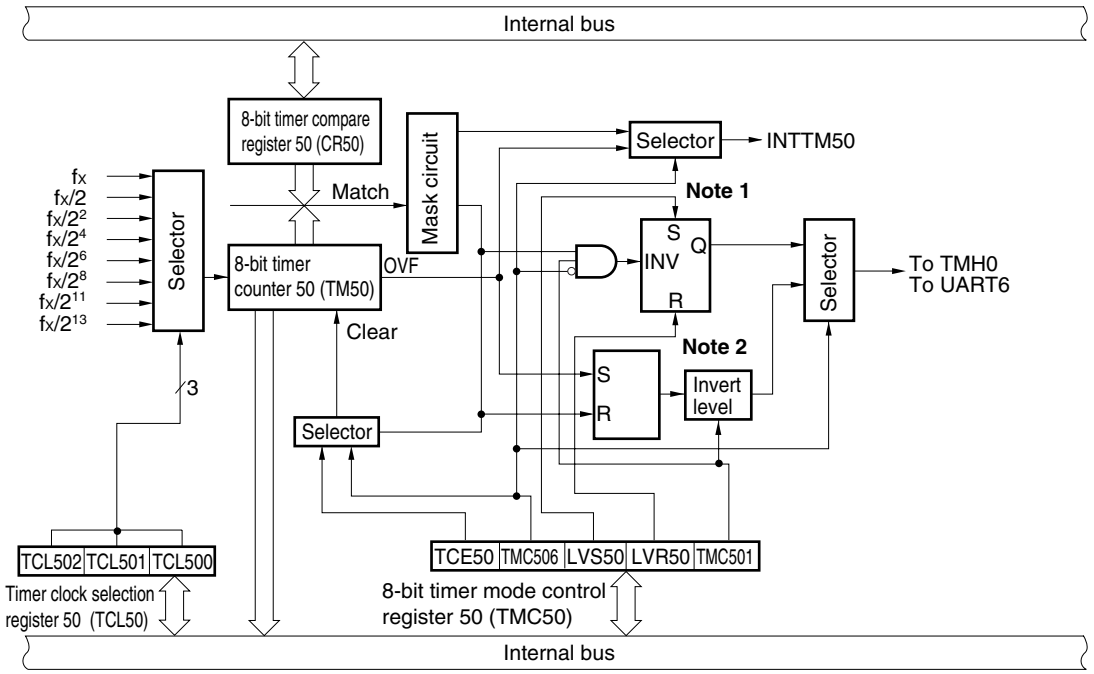
<2> The sampling clock used to eliminate noise differs when the TI000 valid edge is used as the count clock and when it is used as a capture trigger. In the former case, the count clock is f_X, and in the latter case the count clock is selected by prescaler mode register 00 (PRM00). The capture operation is started only after 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 50

8-bit timer 50 can be used as an interval timer or operating clock of TMH0 and UART6.

Figure 7-1 shows the block diagram of 8-bit timer 50.

Figure 7-1. Block Diagram of 8-Bit Timer 50



Notes 1. Timer output F/F

2. PWM output F/F

7.2 Configuration of 8-Bit Timer 50

8-bit timer 50 includes the following hardware.

Table 7-1. Configuration of 8-Bit Timer 50

Item	Configuration
Timer register	8-bit timer counter 50 (TM50)
Register	8-bit timer compare register 50 (CR50)
Control registers	Timer clock selection register 50 (TCL50) Timer clock switch control register (CSEL) 8-bit timer mode control register 50 (TMC50)

(1) 8-bit timer counter 50 (TM50)

TM50 is an 8-bit register that counts the count pulses and is read-only.

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

Figure 7-2. Format of 8-Bit Timer Counter 50 (TM50)

Address: FF16H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM50								

In the following situations, the count value is cleared to 00H.

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

<2> When TCE50 is cleared

<3> When TM50 and CR50 match in clear & start mode if this mode was entered upon a match of TM50 and CR50 values.

(2) 8-bit timer compare register 50 (CR50)

CR50 can be read and written by an 8-bit memory manipulation instruction.

The value set in CR50 is constantly compared with the 8-bit timer counter 50 (TM50) count value, and an interrupt request (INTTM50) is generated if they match.

The value of CR50 can be set within 00H to FFH.

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

Figure 7-3. Format of 8-Bit Timer Compare Register 50 (CR50)

Address: FF17H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR50								

Cautions 1. In the clear & start mode entered on a match of TM50 and CR50 (TMC506 = 0), do not write other values to CR50 during operation.

2. In PWM mode, make the CR50 rewrite period 3 count clocks of the count clock (clock selected by TCL50) or more.

7.3 Registers Controlling 8-Bit Timer 50

The following three registers are used to control 8-bit timer 50.

- Timer clock selection register 50 (TCL50)
- Timer clock switch control register (CSEL)
- 8-bit timer mode control register 50 (TMC50)

(1) Timer clock selection register 50 (TCL50)

This register sets the count clock of 8-bit timer 50.

TCL50 can be set by an 8-bit memory manipulation instruction.

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

Figure 7-4. 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	
0	0	0	Count stopped	
0	0	1		
0	1	0	f_X (10 MHz)	
0	1	1	$f_X/2$ (5 MHz)	
1	0	0	$f_X/2^2$ (2.5 MHz)	When CSEL2[Note 1] = 0
			$f_X/2^4$ (625 kHz)	When CSEL2[Note 1] = 1
1	0	1	$f_X/2^6$ (156.25 kHz)	
1	1	0	$f_X/2^8$ (39.06 kHz)	When CSEL3[Note 2] = 0
			$f_X/2^{11}$ (4.88 kHz)	When CSEL3[Note 2] = 1
1	1	1	$f_X/2^{13}$ (1.22 kHz)	

Notes 1. Check the setting of bit 2 (CSEL2) of the timer clock switch control register (CSEL) before setting TCL502, TCL501, and TCL500 to 1, 0, and 0, respectively (refer to **Figure 7-5 Format of Timer Clock Switch Control Register (CSEL)**). Do not rewrite CSEL2 during timer operation while TCL502, TCL501, and TCL500 are set to 1, 0, and 0, respectively.

2. Check the setting of bit 3 (CSEL3) of the timer clock switch control register (CSEL) before setting TCL502, TCL501, and TCL500 to 1, 1, and 0, respectively (refer to **Figure 7-5 Format of Timer Clock Switch Control Register (CSEL)**). Do not rewrite CSEL3 during timer operation while TCL502, TCL501, and TCL500 are set to 1, 1, and 0, respectively.

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

2. Be sure to set bits 3 to 7 to 0.

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

2. Figures in parentheses apply to operation at f_X = 10 MHz.

(2) Timer clock switch control register (CSEL)

This register is used to switch the selection clock.

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

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

Figure 7-5. Format of Timer Clock Switch Control Register (CSEL)

Address: FF71H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSEL	0	0	0	0	CSEL3	CSEL2	CSEL1	CSEL0

CSEL3	Count clock when TCL502, TCL501, TCL500 = 1, 1, 0
0	$f_X/2^8$ (39.06 kHz)
1	$f_X/2^{11}$ (4.88 kHz)

CSEL2	Count clock when TCL502, TCL501, TCL500 = 1, 0, 0
0	$f_X/2^2$ (2.5 MHz)
1	$f_X/2^4$ (625 kHz)

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

<R> **2.** Bits 1 (CSEL1) and 0 (CSEL0) of CSEL are used to switch the selection clock of the 8-bit timer H1 and H0, respectively (see **8.3 (2) Timer clock switch control register**).

(3) 8-bit timer mode control register 50 (TMC50)

TMC50 is a register that performs the following four types of settings.

<1> 8-bit timer counter 50 (TM50) count operation control
<2> 8-bit timer counter 50 (TM50) 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

TMC50 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-6. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF6BH After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	0
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	0

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	Clear & start mode by 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

Cautions 1. The settings of LVS50 and LVR50 are valid in other than PWM mode.

2. Perform <1> to <3> below in the following order, not at the same time.

<1> Set TMC501 and TMC506: Operating mode setting

<2> Set LVS50 and LVR50 (Caution 1): Timer output F/F setting

<3> Set TCE50

3. Stop operation before rewriting TMC506.

Remarks 1. In PWM mode, PWM output is made inactive by setting TCE50 to 0.

2. If LVS50 and LVR50 are read, 0 is read.

7.4 Operations of 8-Bit Timer 50

7.4.1 Operation as interval timer

8-bit timer 50 operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 50 (CR50).

When the count value of 8-bit timer counter 50 (TM50) matches the value set to CR50, counting continues with the TM50 value cleared to 0 and an interrupt request signal (INTTM50) is generated.

The count clock of TM50 can be selected with bits 0 to 2 (TCL500 to TCL502) of timer clock selection register 50 (TCL50) and bits 2 and 3 (CSEL2, CSEL3) of timer clock switch control register (CSEL).

Setting

<1> Set each register.
- TCL50, CSEL: Select the count clock.
- CR50: Compare value
- TMC50: Select count operation stop. (TMC50 = 000000×0B × = Don't care)

<2> After TCE50 = 1 is set, the count operation starts.

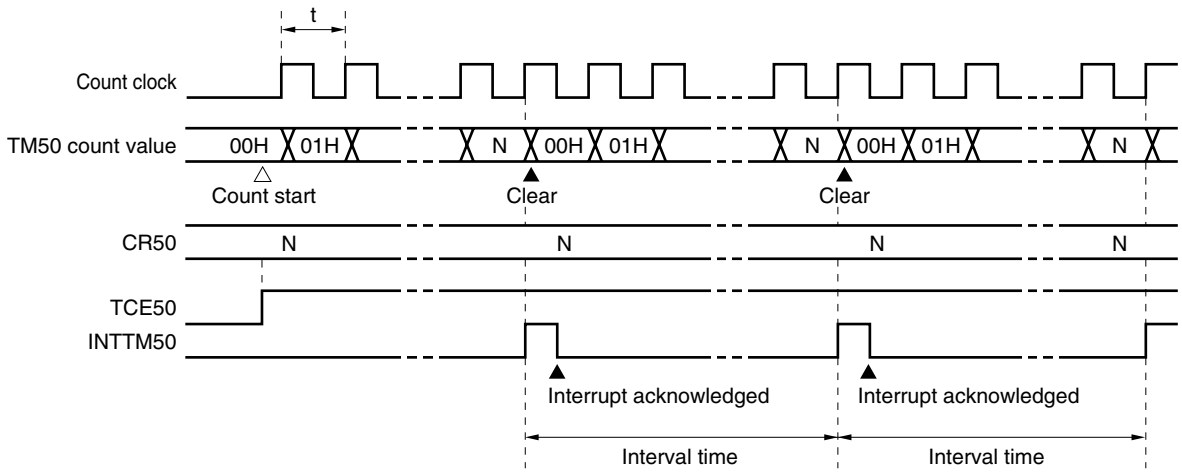
<3> If the values of TM50 and CR50 match, INTTM50 is generated (TM50 is cleared to 00H).

<4> INTTM50 is generated repeatedly at the same interval. Set TCE50 to 0 to stop the count operation.

Caution Do not write other values to CR50 during operation.

Figure 7-7. Interval Timer Operation Timing (1/2)

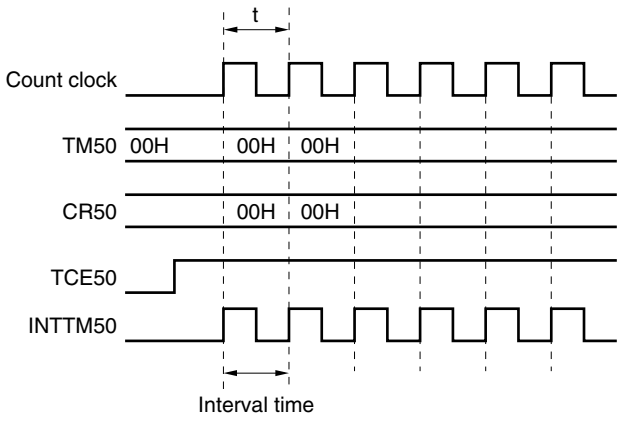
(a) Basic operation



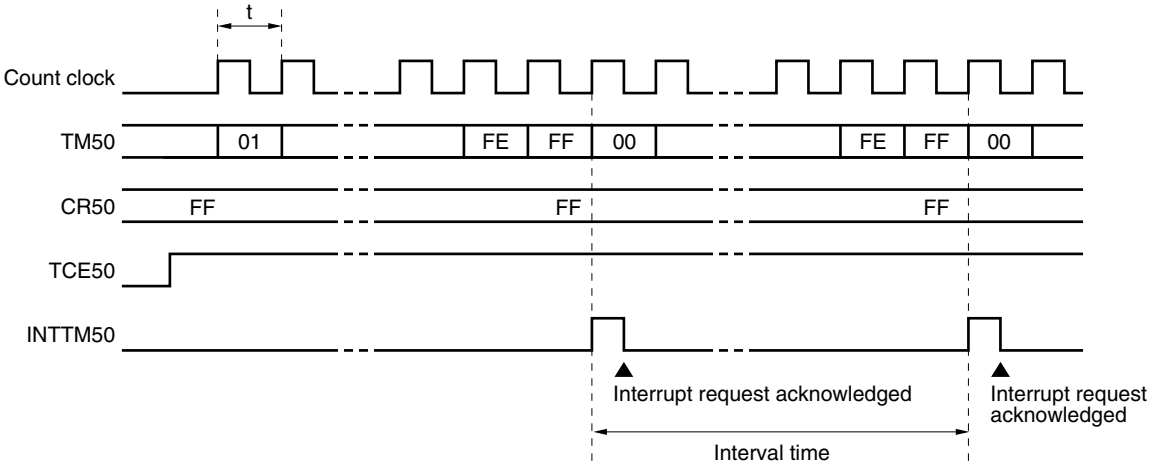
Remark Interval time = (N + 1) × t
N = 01H to FFH

Figure 7-7. Interval Timer Operation Timing (2/2)

(b) When CR50 = 00H



(c) When CR50 = FFH



7.4.2 Operation as operating clock of TMH0 and UART6

8-bit timer 50 can be used as the operating clock of TMH0 and UART6.

(1) In clear & start mode entered on match of TM50 and CR50 (TMC506 = 0)

The timer output F/F is inverted at intervals determined by the count value preset to CR50. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- TCL50: Select the count clock.
- CR50: Compare value
- TMC50: Stop the count operation, select clear & start mode entered on a match of TM50 and CR50.

LVS50	LVR50	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F inversion enabled
(TMC50 = 00001010B or 00000110B)

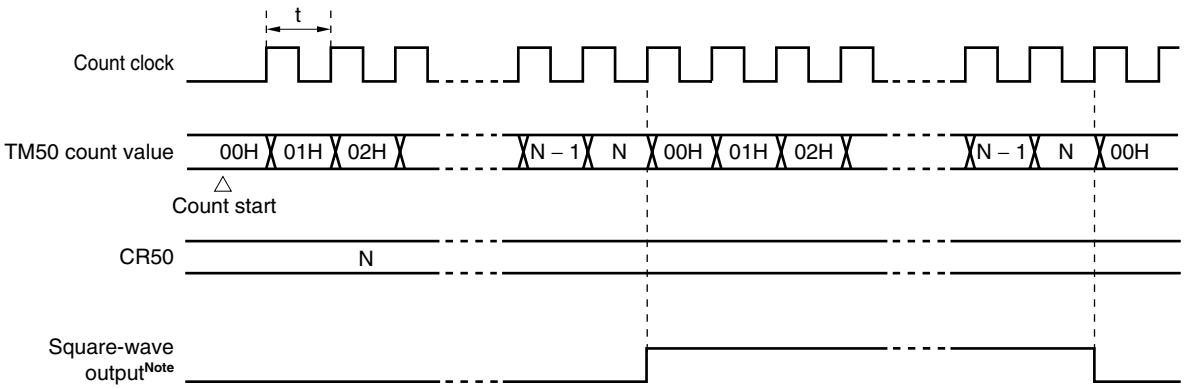
<2> After TCE50 = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM50 and CR50.
After INTTM50 is generated, TM50 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.
The frequency is as follows.
Frequency = 1/2t (N + 1)
(N: 00H to FFH)

Caution Do not write other values to CR50 during operation.

Figure 7-8. Square-Wave Output Operation Timing



Note The initial value of square-wave output can be set by bits 2 and 3 (LVR50, LVS50) of 8-bit timer mode control register 50 (TMC50).

(2) In PWM mode (TMC506 = 1)

The duty pulse is determined by the value set to 8-bit timer compare register 50 (CR50).

Set the active level width of the PWM pulse with CR50 (CR50 = 80H) so that the duty will be 50%; the active level can be selected with bit 1 of TMC50 (TMC501).

The count clock can be selected with bits 0 to 2 (TCL500 to TCL502) of timer clock selection register 50 (TCL50).

Caution In PWM mode, make the CR50 rewrite period 3 count clocks of the count clock (clock selected by TCL50) or more.

Setting

<1> Set each register.

- TCL50: Select the count clock.
- CR50: Compare value (80H)
- TMC50: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC501	Active Level Selection
0	Active-high
1	Active-low

(TMC50 = 01000000B or 01000010B)

<2> The count operation starts when TCE50 = 1.
Set TCE50 to 0 to stop the count operation.

PWM output operation

<1> PWM output outputs an inactive level until an overflow occurs.

<2> When an overflow occurs, the active level is output.
The active level is output until CR50 matches the count value of 8-bit timer counter 50 (TM50).

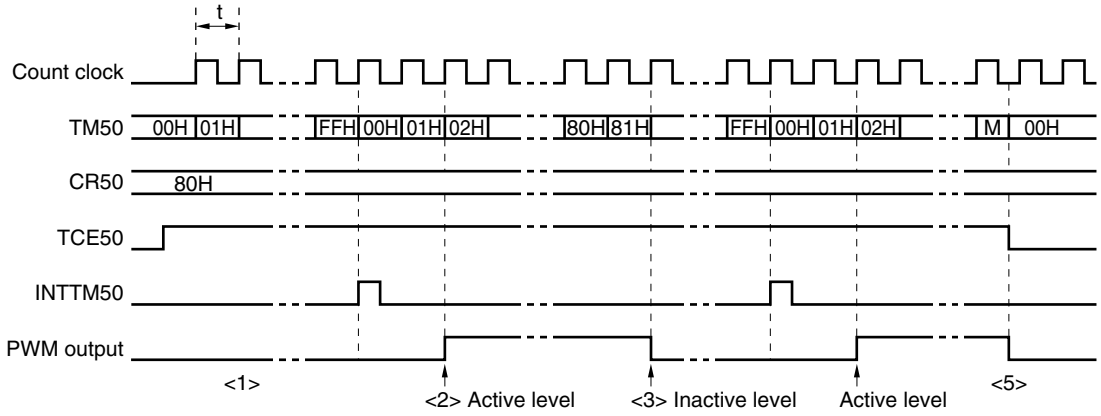
<3> After the CR50 matches the count value, the inactive level is output until an overflow occurs again.

<4> Operations <2> and <3> are repeated until the count operation stops.

<5> When the count operation is stopped with TCE50 = 0, PWM output becomes inactive.
For details of timing, see **Figure 7-9**.
The cycle, active-level width, and duty are as follows.

- Cycle = 2^8t
- Active-level width = Nt
- Duty = $N/2^8$
  (N = 80H)

Figure 7-9. PWM Output Operation Timing (CR50 = 80H, Active Level = H)



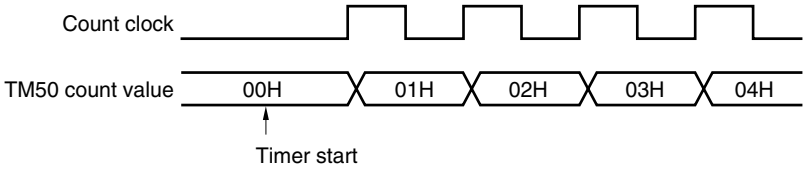
Remark <1> to <3> and <5> in Figure 7-9 correspond to <1> to <3> and <5> in PWM output operation in **7.4.2 (2) In PWM mode (TMC506 = 1)**.

7.5 Cautions on 8-Bit Timer 50

(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 8-bit timer counter 50 (TM50) is started asynchronously to the count clock.

Figure 7-10. 8-Bit Timer Counter 50 Start Timing



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

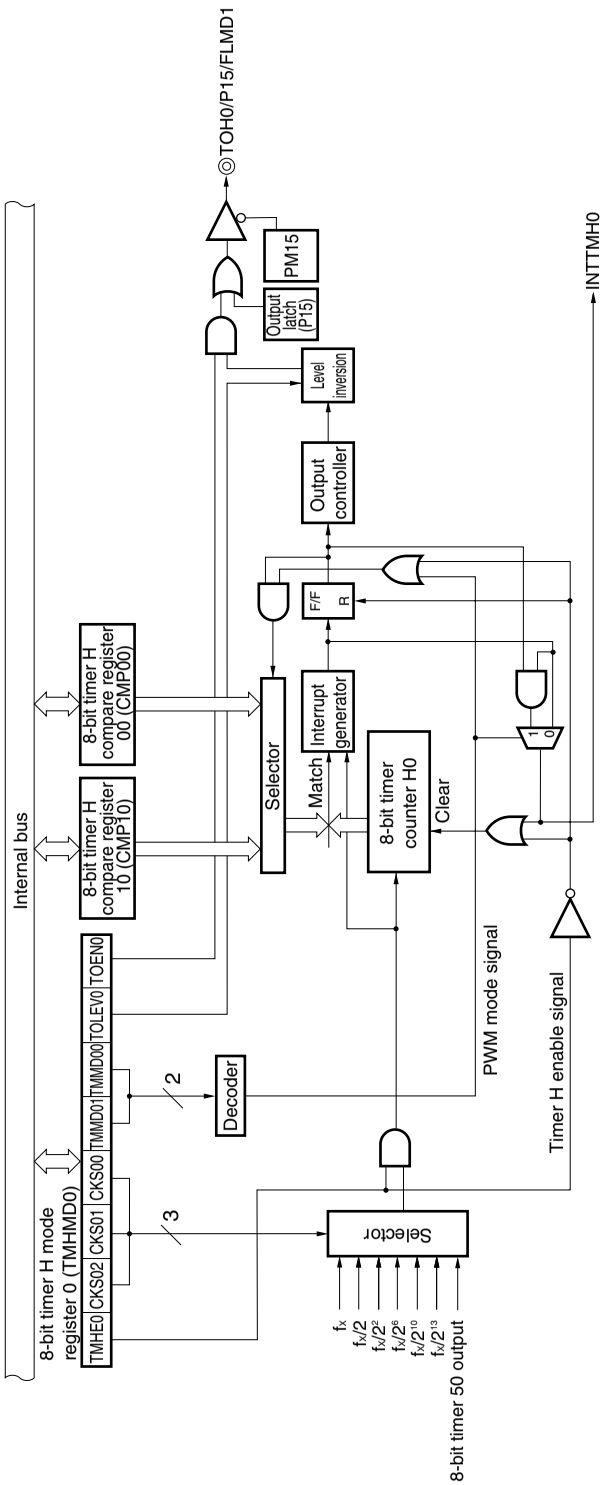
Item	Configuration
Timer register	8-bit timer counter Hn
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	TOHn
Control registers	8-bit timer H mode register n (TMHMDn) Timer clock switch control register (CSEL) 8-bit timer H carrier control register 1 (TMCYC1)[Note] Alternate-function pin switch register (PSEL)[Note] Port mode register 1 (PM1) Port register 1 (P1)

Note 8-bit timer H1 only

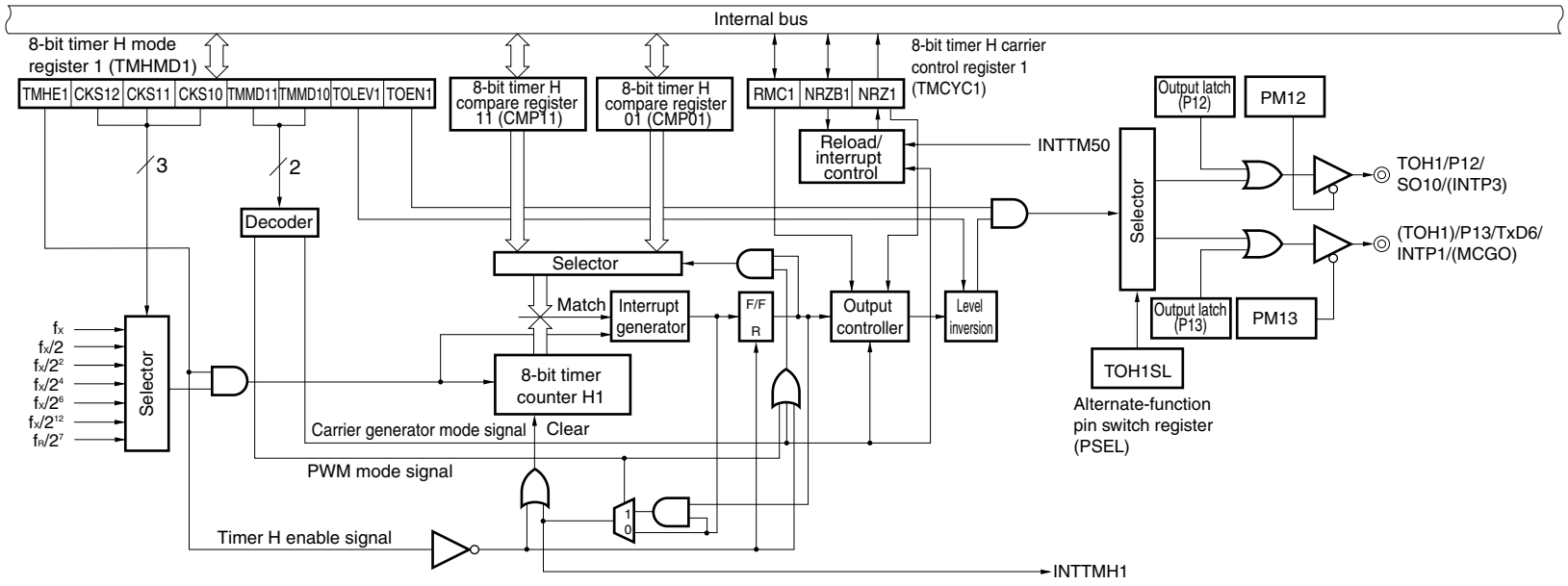
Remark n = 0, 1

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/written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register 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/written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register 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)								

The CMP1n register can be rewritten during timer count operation.

In the carrier generator mode, an interrupt request signal (INTTMHn) is generated if the timer count value and CMP1n value match after setting CMP1n. The timer count value is cleared at the same time. If the CMP1n value is rewritten during timer operation, transfer 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

8-bit timers H0 and H1 are controlled by the following six types of registers.

- 8-bit timer H mode register n (TMHMDn)
- Timer clock switch control register (CSEL)
- 8-bit timer H carrier control register 1 (TMCYC1)[Note]
- Alternate-function pin switch register (PSEL)[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 this register 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	
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 1]	When CSEL0[Note 2] = 0
			$f_X/2^{13}$ (1.22 kHz)	When CSEL0[Note 2] = 1
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. When the TM50 output is selected as the count clock, observe the following.

- PWM mode (TMC506 = 1)

  Set the clock so that the duty will be 50% and start the operation of 8-bit timer/event counter 50 in advance.
- Clear & start mode entered on match of TM50 and CR50 (TMC506 = 0)

  Enable the timer F/F inversion operation (TMC501 = 1) and start the operation of 8-bit timer/event counter 50 in advance.

2. Check the setting of bit 0 (CSEL0) of the timer clock switch control register (CSEL) before setting CKS02, CKS01, and CKS00 to 1, 0, and 1, respectively (refer to **Figure 8-7 Format of Timer Clock Switch Control Register (CSEL)**). Do not rewrite CSEL0 during timer operation while CKS02, CKS01, and CKS00 are set to 1, 0, and 1, respectively.

Cautions 1. When the internal low-speed oscillation clock is selected as the clock to be supplied to the CPU, the clock of the internal low-speed oscillator is divided and supplied as the count clock. If the count clock is the internal low-speed oscillation 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. fx: High-speed system clock oscillation frequency

2. Figures in parentheses apply to operation at fx = 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	
0	0	0	f_X (10 MHz)	
0	0	1	$f_X/2^2$ (2.5 MHz)	When CSEL1[Note] = 0
			$f_X/2$ (5 MHz)	When CSEL1[Note] = 1
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 Check the setting of bit 1 (CSEL1) of the timer clock switch control register (CSEL) before setting CKS12, CKS11, and CKS10 to 0, 0, and 1, respectively (refer to **Figure 8-7 Format of Timer Clock Switch Control Register (CSEL)**). Do not rewrite CSEL1 during timer operation while CKS12, CKS11, and CKS10 are set to 0, 0, and 1, respectively.

Cautions 1. When the internal low-speed oscillation clock is selected as the clock to be supplied to the CPU, the clock of the internal low-speed oscillator is divided and supplied as the count clock. If the count clock is the internal low-speed oscillation 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.

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 the CMP11 register).

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 TM50.

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

2. f_R: Internal low-speed oscillation clock oscillation frequency

3. Figures in parentheses apply to operation at f_X = 10 MHz, f_R = 240 kHz (TYP.).

(2) Timer clock switch control register (CSEL)

This register is used to switch the selection clock.

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

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

Figure 8-7. Format of Timer Clock Switch Control Register (CSEL)

Address: FF71H After reset: 00H R/W

	7	6	5	4	3	2	1	0
CSEL	0	0	0	0	CSEL3	CSEL2	CSEL1	CSEL0

CSEL1	Count clock when CKS12, CKS11, CKS10 = 0, 0, 1
0	$f_X/2^2$ (2.5 MHz)
1	$f_X/2$ (5 MHz)

CSEL0	Count clock when CKS02, CKS01, CKS00 = 1, 0, 1
0	TM50 output
1	$f_X/2^{13}$ (1.22 kHz)

Remarks 1. CKS12, CKS11, and CKS10: Bits 6 to 4 of 8-bit timer H mode register 1 (TMHMD1)
CKS02, CKS01, and CKS00: Bits 6 to 4 of 8-bit timer H mode register 0 (TMHMD0)

2. f_X: High-speed system clock oscillation frequency

<R> **3.** Bits 3 (CSEL3) and 2 (CSEL2) of CSEL are used to switch the selection clock of the 8-bit timer 50 (see **7.3 (2) Timer clock switch control register**).

4. Figures in parentheses apply to operation at f_X = 10 MHz.

(3) 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 this register to 00H.

Figure 8-8. 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 at rising edge of INTTM50 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM50 signal input

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.

(4) Alternate-function pin switch register (PSEL)

This register is used to select the TOH1 pin.

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

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

Figure 8-9. Format of Alternate-Function Pin Switch Register (PSEL)

Address: FF70H After reset: 00H R/W

	7	6	<5>	<4>	3	2	<1>	<0>
PSEL	0	0	TOH1SL	MCGSL	0	0	INTP1SL	INTP3SL

TOH1SL	TOH1 pin selection
0	P12/SO10/TOH1/(INTP3)
1	P13/TxD6/INTP1/(TOH1)/(MCGO)

Caution Set bit 7 (TMHE1) of 8-bit timer H mode register 1 (TMHMD1) to 0 before rewriting the TOH1SL bit.

(5) Port mode register 1 (PM1)

This register is used to set input/output for port 1 in 1-bit units.

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

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

Figure 8-10. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

	7	6	5	4	3	2	1	0
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

When using the P12/TOH1/SO10/(INTP3), P13/(TOH1)/TxD6/INTP1/(MCGO), or P15/TOH0/FLMD1 pin as a timer output, set the port mode register and port output latch as follows.

- **P12/TOH1/SO10/(INTP3) is used as timer output (bit 5 (TOH1SL) of PSEL register = 0)**
  Bit 2 (PM12) of port mode register 1: Cleared to 0
  Bit 2 (P12) of port 1: Cleared to 0

- **P13/(TOH1)/TxD6/INTP1/(MCGO) is used as timer output (bit 5 (TOH1SL) of PSEL register = 1)**
  Bit 3 (PM13) of port mode register 1: Cleared to 0
  Bit 3 (P13) of port 1: Cleared to 0

- **P15/TOH0/FLMD1 is used as timer output (setting of PSEL register is not necessary)**
  Bit 5 (PM15) of port mode register 1: Cleared to 0
  Bit 5 (P15) of port 1: Cleared to 0

8.4 Operation of 8-Bit Timers H0 and H1

8.4.1 Operation as interval timer

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.

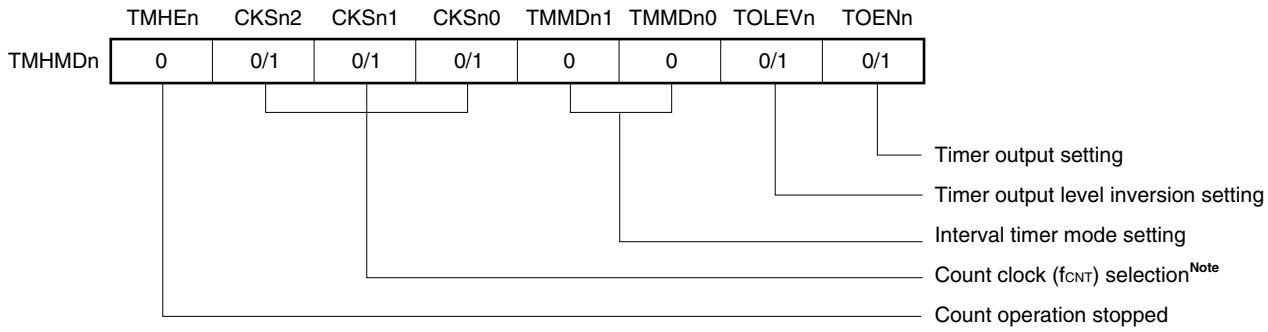
(1) Usage

Generates the INTTMHn signal repeatedly at the same interval.

<1> Set each register.

Figure 8-11. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register n (TMHMDn)



Note Check the setting of bit 0 (CSEL0) of the timer clock switch control register (CSEL) before setting CKS02, CKS01, and CKS00 to 1, 0, and 1, respectively, and check the setting of bit 1 (CSEL1) of the CSEL register before setting CKS12, CKS11, and CKS10 to 0, 0, and 1, respectively (refer to **Figure 8-7 Format of Timer Clock Switch Control Register (CSEL)**).

(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, set TMHEn to 0.

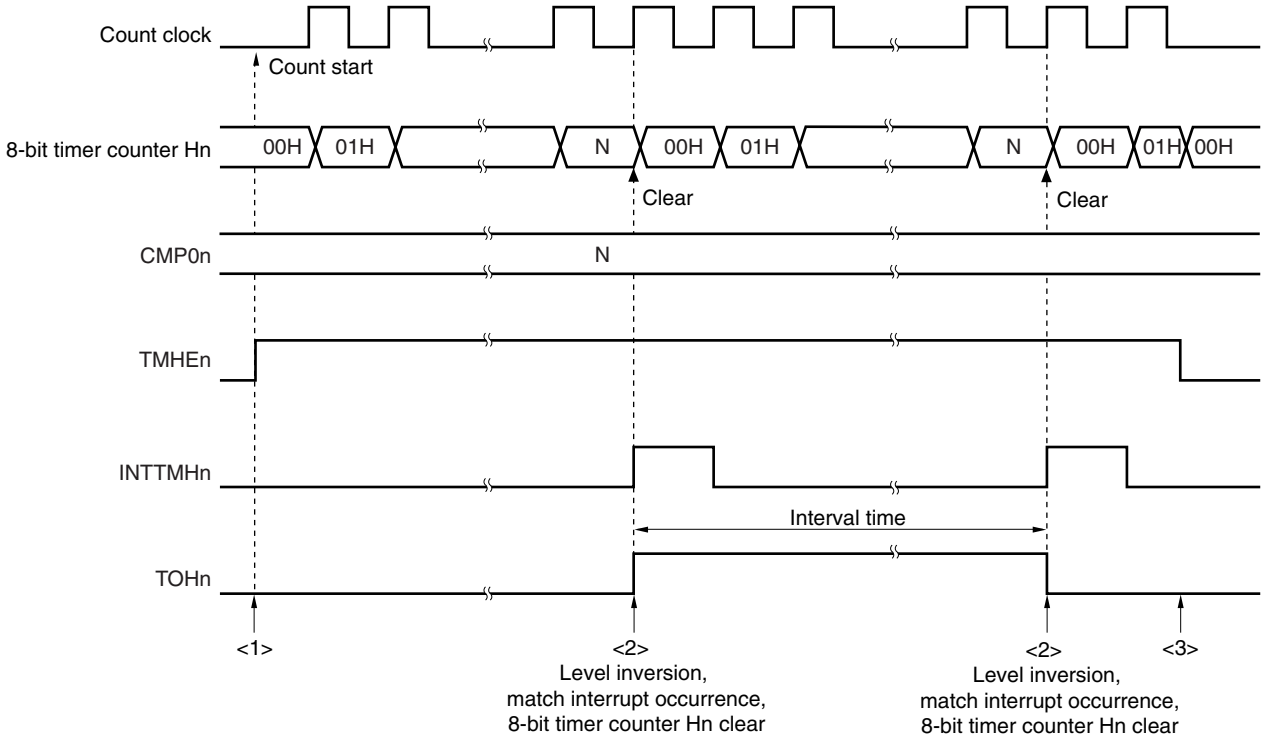
Remark n = 0, 1

(2) Timing chart

The timing of the interval timer/square-wave output operation is shown below.

Figure 8-12. 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.

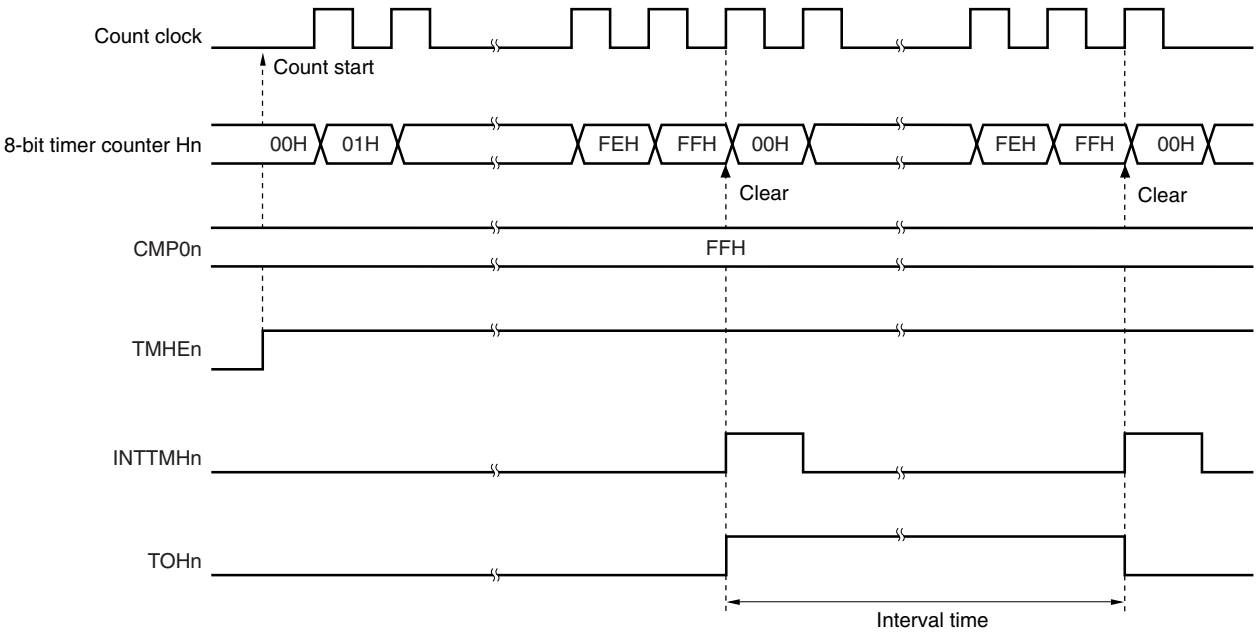
<2> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, the TOHn output level is inverted, and the INTTMHn signal is output.

<3> The INTTMHn signal and TOHn output become inactive by setting 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

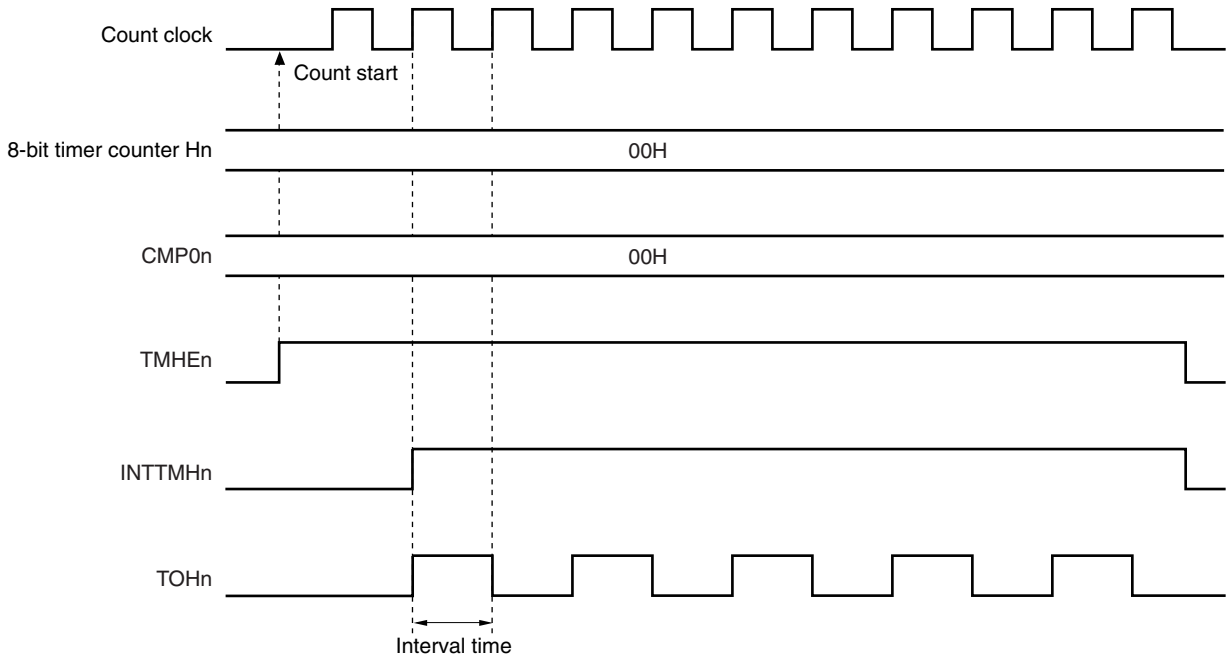
Figure 8-12. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP0n = FFH



<R>

(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.

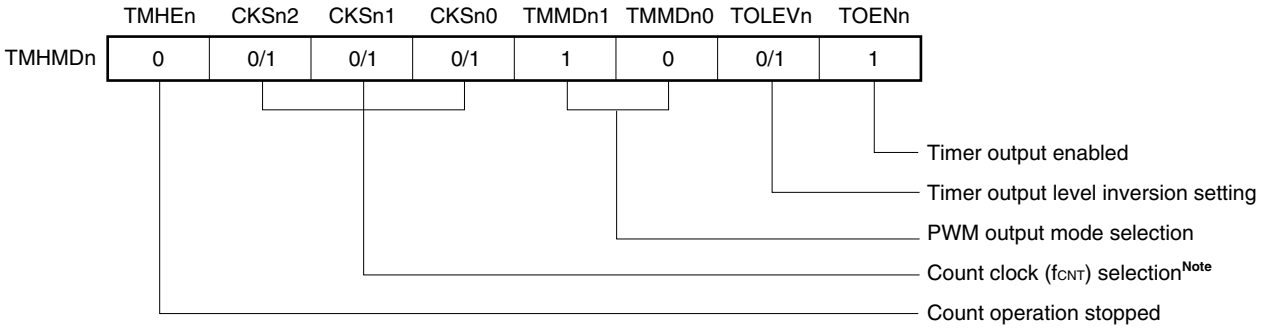
(1) Usage

In PWM output mode, a pulse for which an arbitrary duty and arbitrary cycle can be set is output.

<1> Set each register.

Figure 8-13. Register Setting in PWM Output Mode

(i) Setting timer H mode register n (TMHMDn)



Note Check the setting of bit 0 (CSEL0) of the timer clock switch control register (CSEL) before setting CKS02, CKS01, and CKS00 to 1, 0, and 1, respectively, and check the setting of bit 1 (CSEL1) of the CSEL register before setting CKS12, CKS11, and CKS10 to 0, 0, and 1, respectively (refer to **Figure 8-7 Format of Timer Clock Switch Control Register (CSEL)**).

(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).

(2) Timing chart

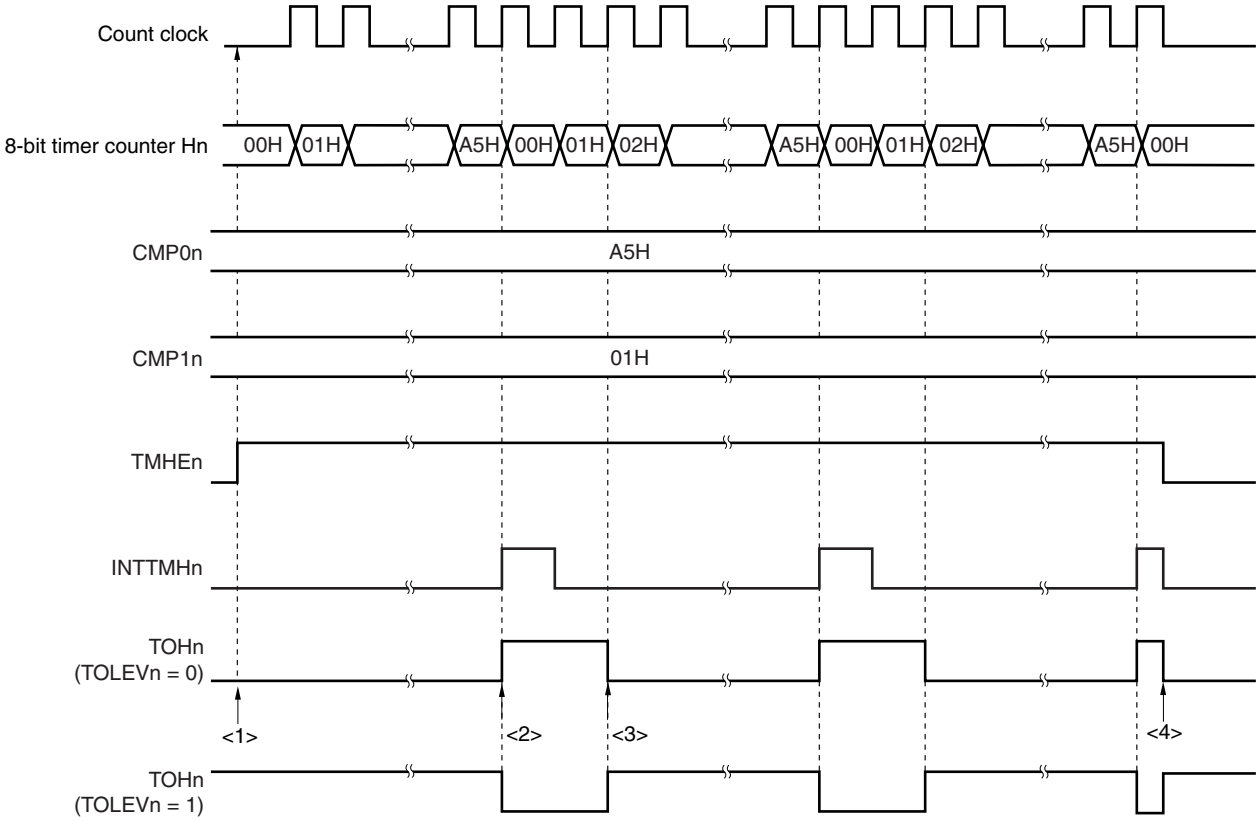
The operation timing in PWM output mode is shown below.

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

Remark n = 0, 1

Figure 8-14. 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.

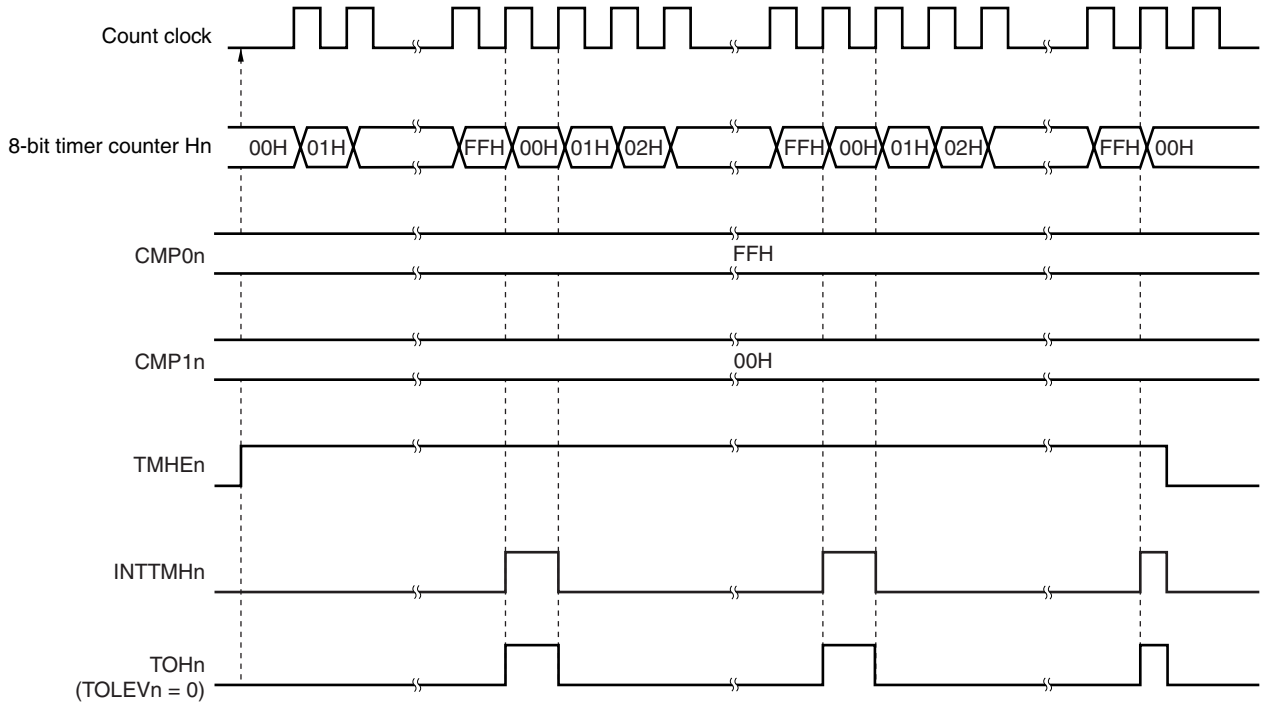
<3> When the values of 8-bit timer counter Hn and the CMP1n register match, the level of the TOHn output is returned. At this time, the value of 8-bit timer counter Hn is not cleared and the INTTMHn signal is not output.

<4> Setting the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

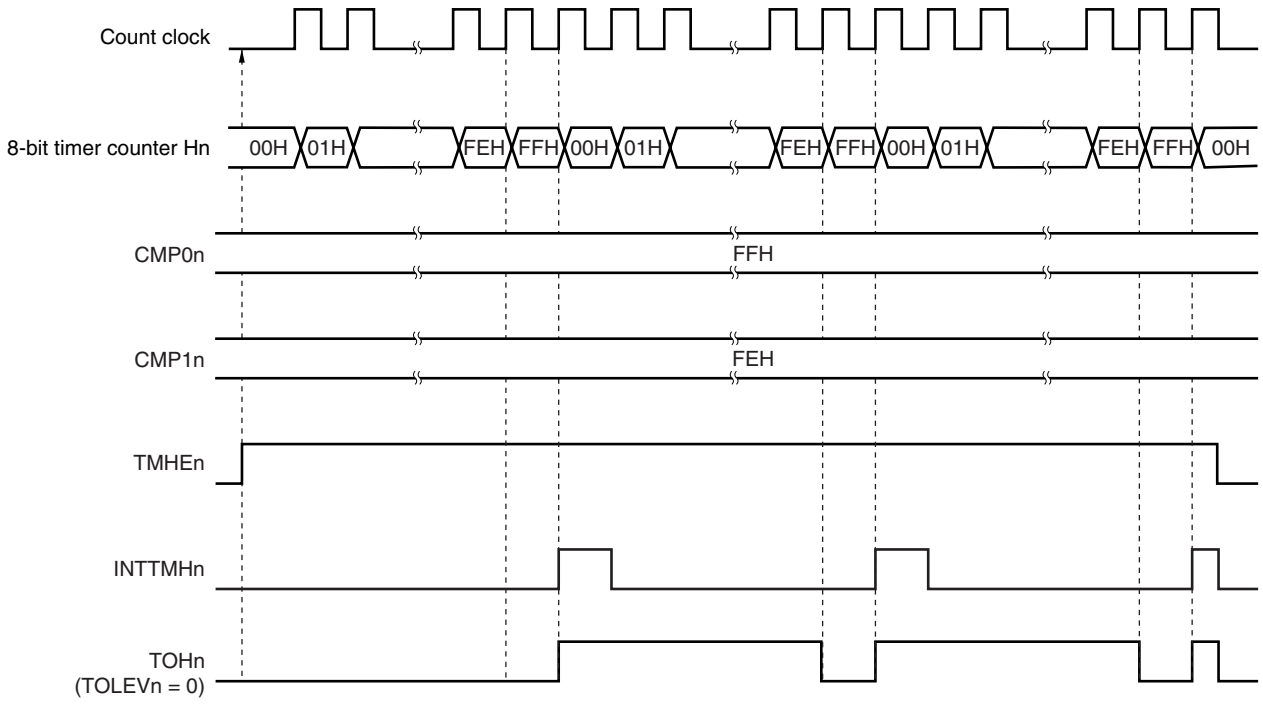
Remark n = 0, 1

Figure 8-14. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP0n = FFH, CMP1n = 00H



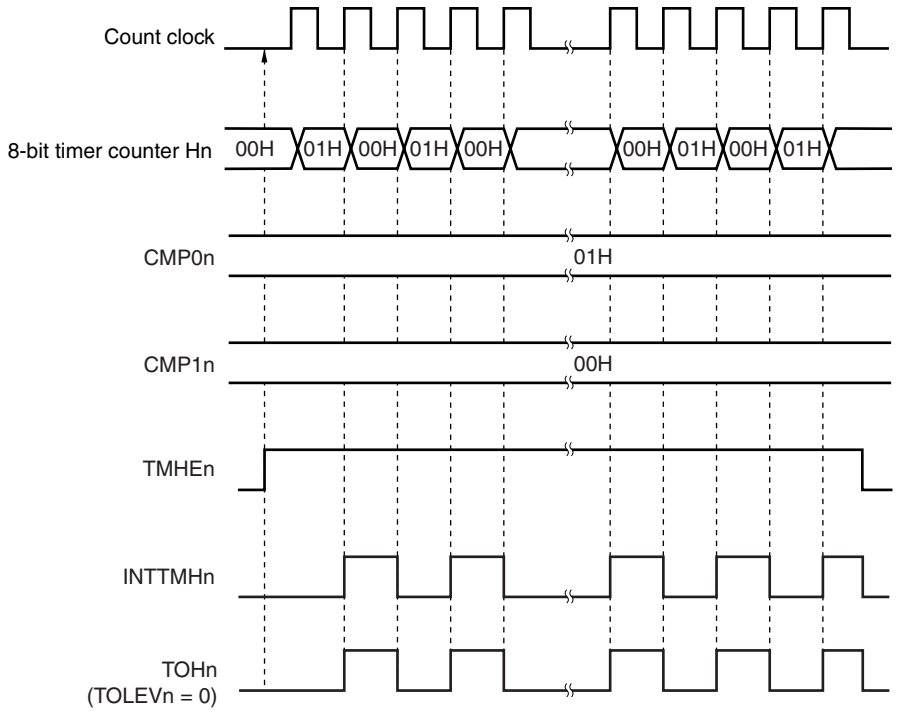
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0, 1

Figure 8-14. Operation Timing in PWM Output Mode (3/4)

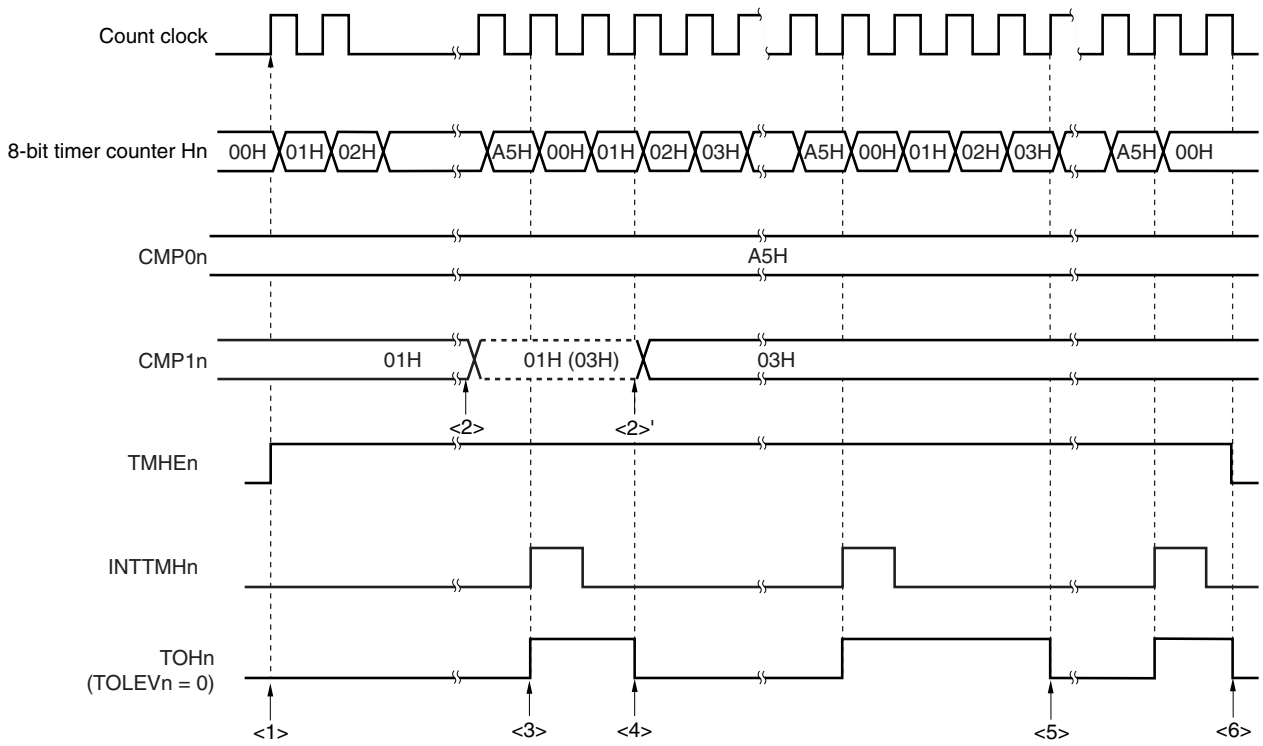
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0, 1

Figure 8-14. 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 TMHEn = 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> Setting the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

Remark n = 0, 1

8.4.3 Operation as carrier generator mode (8-bit timer H1 only)

The carrier clock generated by 8-bit timer H1 is output in the cycle set by 8-bit timer 50.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by 8-bit timer 50, 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.

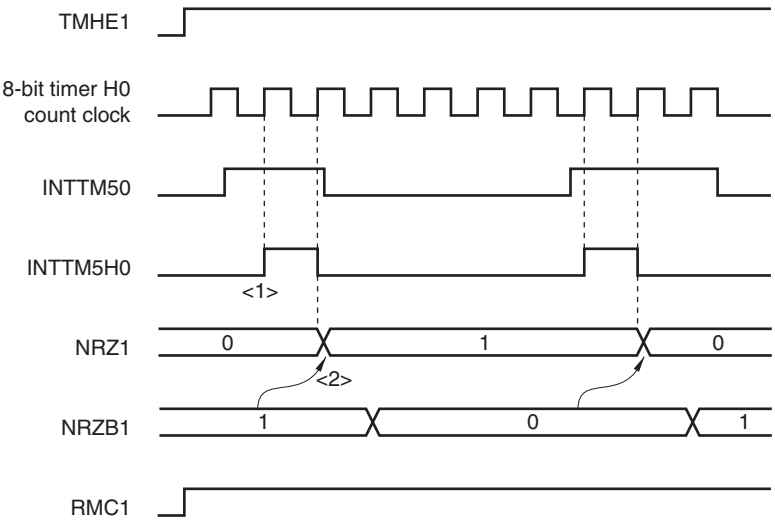
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM50) of 8-bit timer 50 and the NRZB1 and RMC1 bits of 8-bit timer H carrier control register 1 (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZB1 Bit	Output
0	0	Low-level output
<R> 0	1	High-level output at rising edge of INTTM50 signal input
1	0	Low-level output
<R> 1	1	Carrier pulse output at rising edge of INTTM50 signal input

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 INTTM50 signal is synchronized with the 8-bit timer H1 count clock and output as the INTTM5H0 signal. The INTTM5H0 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 8-15. Transfer Timing



<1> The INTTM50 signal is synchronized with the count clock of 8-bit timer H1 and is output as the INTTM5H0 signal.

<2> The value of the NRZB1 bit is transferred to the NRZ1 bit at the second clock from the rising edge of the INTTM5H0 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 50 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When 8-bit timer 50 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

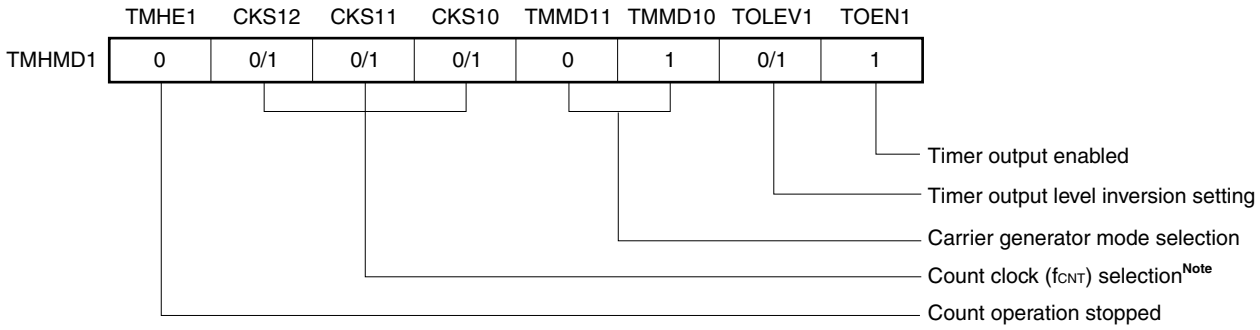
(3) Usage

Outputs an arbitrary carrier clock from the TOH1 pin.

<1> Set each register.

Figure 8-16. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)



Note Check the setting of bit 1 (CSEL1) of the timer clock switch control register (CSEL) before setting CKS12, CKS11, and CKS10 to 0, 0, and 1, respectively (refer to **Figure 8-7 Format of Timer Clock Switch Control Register (CSEL)**).

(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) TCL50 and TMC50 register setting

- Refer to **7.3 Registers Controlling 8-Bit Timer 50**.

<2> When TMHE1 = 1, 8-bit timer H1 starts counting.

<3> When TCE50 of 8-bit timer mode control register 50 (TMC50) is set to 1, 8-bit timer 50 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 INTTM50 signal is synchronized with the count clock of 8-bit timer H1 and output as the INTTM5H0 signal. The INTTM5H0 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, set 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 TM50.

(4) Timing chart

The carrier output control timing is shown below.

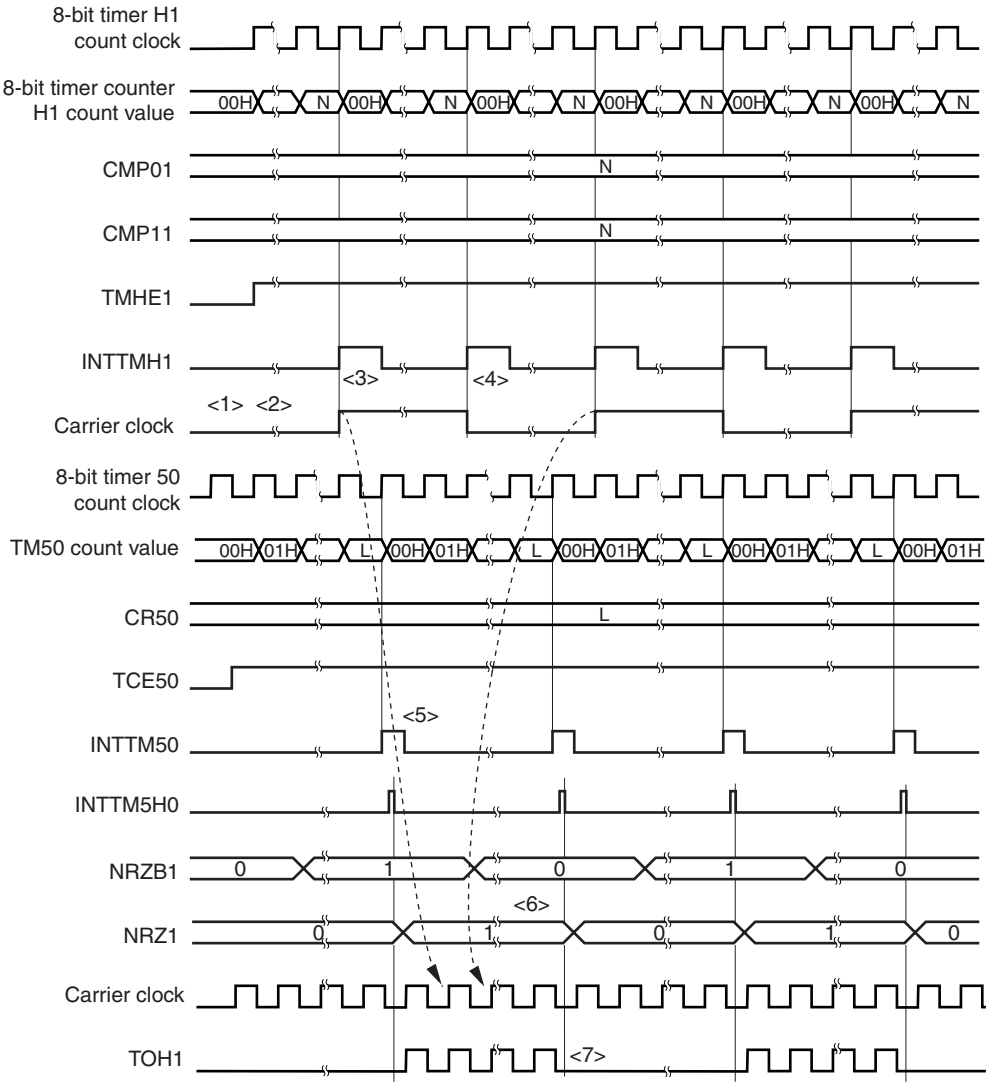
Cautions 1. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

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 8-17. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = N, CMP11 = N



<1> When TMHE1 = 0 and TCE50 = 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.

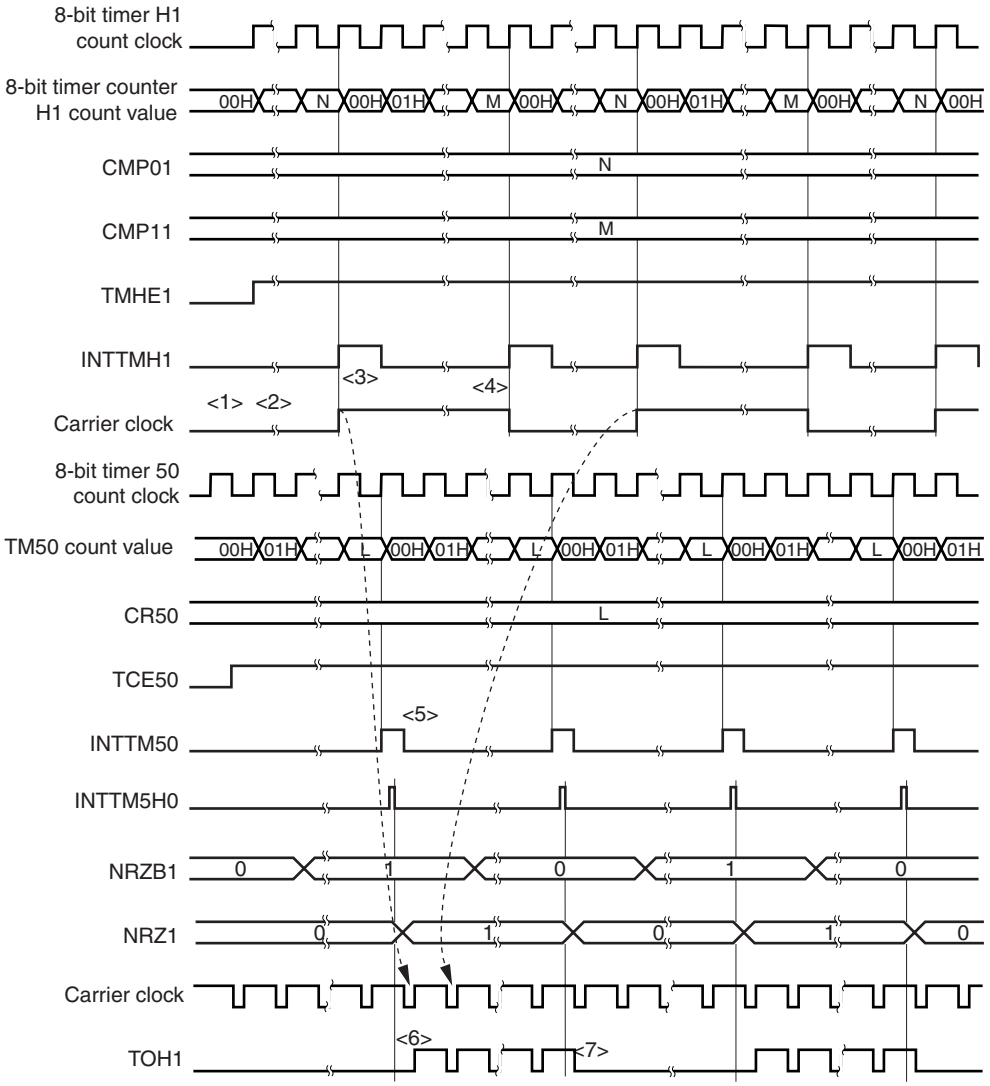
<5> When the INTTM50 signal is generated, it is synchronized with 8-bit timer H1 count clock and output as the INTTM5H0 signal.

<6> The INTTM5H0 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 8-17. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = N, CMP11 = M



<1> When TMHE1 = 0 and TCE50 = 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.

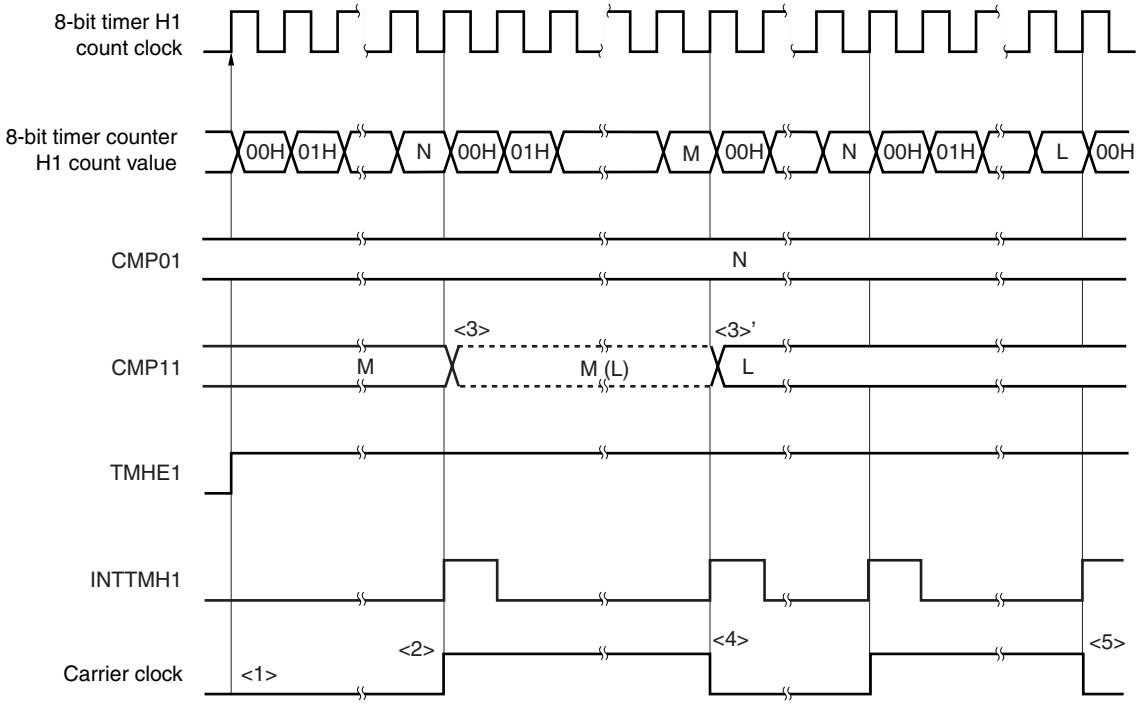
<5> When the INTTM50 signal is generated, it is synchronized with 8-bit timer H1 count clock and output as the INTTM5H0 signal.

<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 8-17. 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).

CHAPTER 9 WATCHDOG TIMER

9.1 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 16 RESET FUNCTION**.

<R>

Table 9-1. Loop Detection Time of Watchdog Timer

Loop Detection Time	
During Internal Low-Speed Oscillation Clock Operation	During High-Speed System Clock Operation
$2^{11}/f_R$ (4.27 ms)	$2^{13}/f_{XH}$ (819.2 μs)
$2^{12}/f_R$ (8.53 ms)	$2^{14}/f_{XH}$ (1.64 ms)
$2^{13}/f_R$ (17.07 ms)	$2^{15}/f_{XH}$ (3.28 ms)
$2^{14}/f_R$ (34.13 ms)	$2^{16}/f_{XH}$ (6.55 ms)
$2^{15}/f_R$ (68.27 ms)	$2^{17}/f_{XH}$ (13.11 ms)
$2^{16}/f_R$ (136.53 ms)	$2^{18}/f_{XH}$ (26.21 ms)
$2^{17}/f_R$ (273.07 ms)	$2^{19}/f_{XH}$ (52.43 ms)
$2^{18}/f_R$ (546.13 ms)	$2^{20}/f_{XH}$ (104.86 ms)

Remarks 1. f_R: Internal low-speed oscillation clock frequency

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

3. Figures in parentheses apply to operation at f_R = 480 kHz (MAX.), f_{XH} = 10 MHz.

The operation mode of the watchdog timer (WDT) is switched according to the mask option (option byte if using a flash memory version) setting of the internal low-speed oscillation clock as shown in Table 9-2.

Table 9-2. Mask Option Setting and Watchdog Timer Operation Mode

	Mask Option	
	Internal Low-Speed Oscillator Cannot Be Stopped	Internal Low-Speed Oscillator Can Be Stopped by Software
Watchdog timer clock source	Fixed to f_R Note 1.	• Selectable by software (f_{XH}, f_R or stopped) • When reset is released: f_R
Operation after reset	Operation starts with the maximum interval ($2^{18}/f_R$).	Operation starts with the maximum interval ($2^{18}/f_R$).
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, the internal low-speed oscillator absolutely cannot be stopped (except during reset).

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_{XH}, clock supply to the watchdog timer is stopped under the following conditions.

- When f_{XH} 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_{XH} and if f_R is stopped by software before execution of the STOP instruction
- In HALT/STOP mode

Remarks 1. f_R: Internal low-speed oscillation clock frequency

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

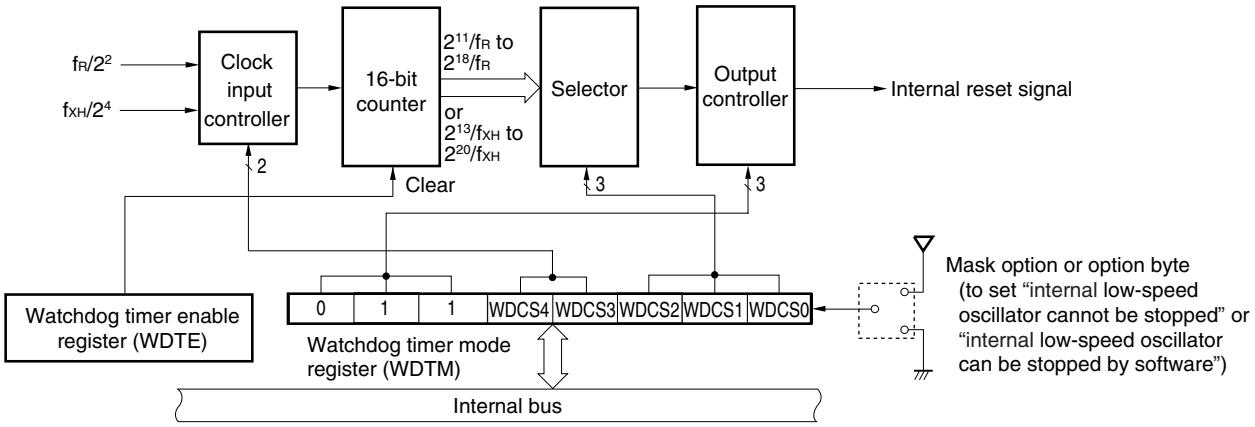
9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer mode register (WDTM) Watchdog timer enable register (WDTE)

Figure 9-1. Block Diagram of Watchdog Timer



9.3 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 this register to 67H.

Figure 9-2. 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	Internal low-speed oscillation clock (f_R)
0	1	High-speed system clock (f_{XH})
1	×	Watchdog timer operation stopped

<R>

WDCS2[Note 2]	WDCS1[Note 2]	WDCS0[Note 2]	Overflow time setting	
			During internal low-speed oscillation clock operation	During high-speed system clock operation
0	0	0	$2^{11}/f_R$ (4.27 ms)	$2^{13}/f_{XH}$ (819.2 μs)
0	0	1	$2^{12}/f_R$ (8.53 ms)	$2^{14}/f_{XH}$ (1.64 ms)
0	1	0	$2^{13}/f_R$ (17.07 ms)	$2^{15}/f_{XH}$ (3.28 ms)
0	1	1	$2^{14}/f_R$ (34.13 ms)	$2^{16}/f_{XH}$ (6.55 ms)
1	0	0	$2^{15}/f_R$ (68.27 ms)	$2^{17}/f_{XH}$ (13.11 ms)
1	0	1	$2^{16}/f_R$ (136.53 ms)	$2^{18}/f_{XH}$ (26.21 ms)
1	1	0	$2^{17}/f_R$ (273.07 ms)	$2^{19}/f_{XH}$ (52.43 ms)
1	1	1	$2^{18}/f_R$ (546.13 ms)	$2^{20}/f_{XH}$ (104.86 ms)

Notes 1. If “internal low-speed oscillator cannot be stopped” is specified by a mask option, this cannot be set. The internal low-speed oscillation clock will be selected no matter what value is written.

2. Reset is released at the maximum cycle (WDCS2, WDCS1, WDCS0 = 1, 1, 1).

Cautions 1. If data is written to WDTM, a wait cycle is generated. For details, refer to CHAPTER 28 CAUTIONS FOR WAIT.

2. Set bits 7, 6, and 5 to 0, 1, and 1, respectively (when “internal low-speed oscillator clock cannot be stopped” is selected by a mask option, other values are ignored).

3. After reset is released, WDTM can be written only once by an 8-bit memory manipulation instruction. If writing attempted a second time, an internal reset signal is generated. If the source clock of the watchdog timer is stopped, however, an internal reset signal is generated when the source clock of the watchdog timer starts operating again.

4. WDTM cannot be set by a 1-bit memory manipulation instruction.

5. When “internal low-speed oscillator can be stopped by software” is selected by a mask option and the watchdog timer is stopped by setting WDCS4 to 1, the watchdog timer does not operate even if WDCS4 is cleared to 0 again. An internal reset signal is not generated.

Remarks 1. f_R: Internal low-speed oscillation clock frequency

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

3. ×: Don’t care

4. Figures in parentheses apply to operation at f_R = 480 kHz (MAX.), f_{XH} = 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 this register to 9AH.

Figure 9-3. 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 of the watchdog timer is stopped, however, an internal reset signal is generated when the source clock of the watchdog timer starts operating again.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated. If the source clock of the watchdog timer is stopped, however, an internal reset signal is generated when the source clock of the watchdog timer starts operating again.

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 9-4. Relationship Between Watchdog Timer Operation and Internal Reset Signal Generated by Watchdog Timer

Watchdog Timer Operation / Internal Reset Signal Generation Source	"Internal low-Speed Oscillator Cannot Be Stopped" Is Set by Mask Option (Watchdog Timer Always Operating)	"Internal low-Speed Oscillator Can Be Stopped by Software" Is Set by Mask Option		
		During Watchdog Timer Operation	Watchdog Timer Stopped	
			Set WDCS4 to 1	Source Clock of Watchdog Timer Stopped
Watchdog timer overflow	An internal reset signal is generated.	An internal reset signal is generated.	–	–
Writing to WDTM for second time	An internal reset signal is generated.	An internal reset signal is generated.	An internal reset signal is not generated. Watchdog timer does not resume operation.	An internal reset signal is generated when the source clock of the watchdog timer starts operating again.
Writing value other than ACH to WDTE	An internal reset signal is generated.	An internal reset signal is generated.	An internal reset signal is not generated.	An internal reset signal is generated when the source clock of the watchdog timer starts operating again.
Accessing WDTE using 1-bit memory manipulation instruction				

9.4 Operation of Watchdog Timer

9.4.1 Watchdog timer operation when "Internal low-speed Oscillator cannot be stopped" is selected by mask option

The operation clock of watchdog timer is fixed to the internal low-speed oscillation 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 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: Internal low-speed oscillation clock
   - Cycle: $2^{18}/f_R$ (543.13 ms: At operation with f_R = 480 kHz (MAX.))
   - 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 (internal low-speed oscillation 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 internal low-speed oscillation clock can be selected as the count source, so after STOP instruction execution, clear the watchdog timer using the interrupt request of TMH1 before the watchdog timer overflows. If this processing is not performed, an internal reset signal is generated when the watchdog timer overflows after STOP instruction execution.

9.4.2 Watchdog timer operation when "Internal low-speed oscillator can be stopped by software" is selected by mask option

The operation clock of the watchdog timer can be selected as either the internal low-speed oscillation clock or the high-speed system clock.

After reset is released, operation is started at the maximum cycle of the internal low-speed oscillation clock (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: Internal low-speed oscillation clock frequency (f_R)
   - Cycle: $2^{18}/f_R$ (546.13 ms: At operation with f_R = 480 kHz (MAX.))
   - 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).
     Internal low-speed oscillation clock (f_R)
     High-speed system clock (f_{XH})
     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 **9.4.3 Watchdog timer operation in STOP mode** and **9.4.4 Watchdog timer operation in HALT mode**.

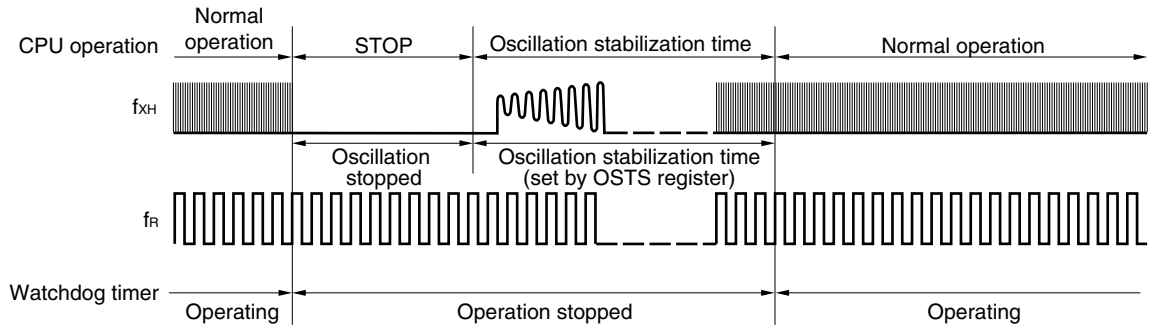
9.4.3 Watchdog timer operation in STOP mode (when "Internal low-speed oscillator can be stopped by software" is selected by mask option)

The watchdog timer stops counting during STOP instruction execution regardless of whether the high-speed system clock or the internal low-speed oscillation clock is being used.

(1) When the CPU clock and the watchdog timer operation clock are the high-speed system clock (f_{XH}) 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.

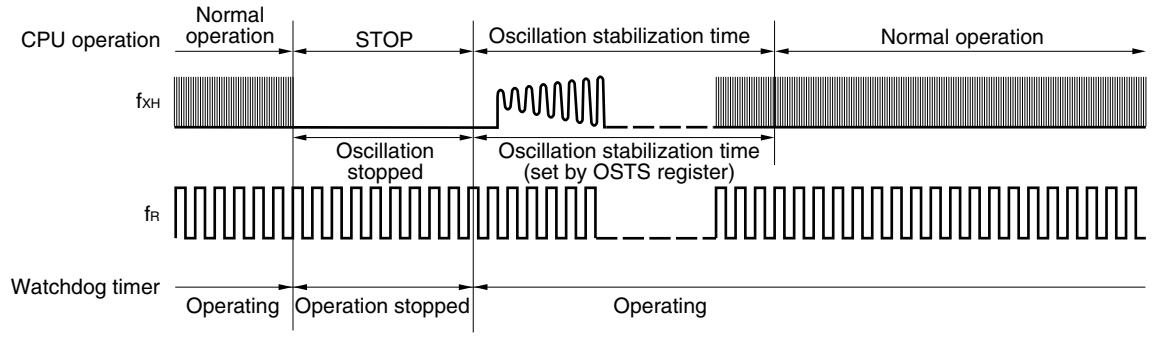
Figure 9-4. Operation in STOP Mode (CPU Clock and WDT Operation Clock: High-Speed System Clock)



(2) When the CPU clock is the high-speed system clock (f_{XH}) and the watchdog timer operation clock is the internal low-speed oscillation 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 9-5. Operation in STOP Mode
(CPU Clock: High-Speed System Clock, WDT Operation Clock: Internal Low-Speed Oscillation Clock)



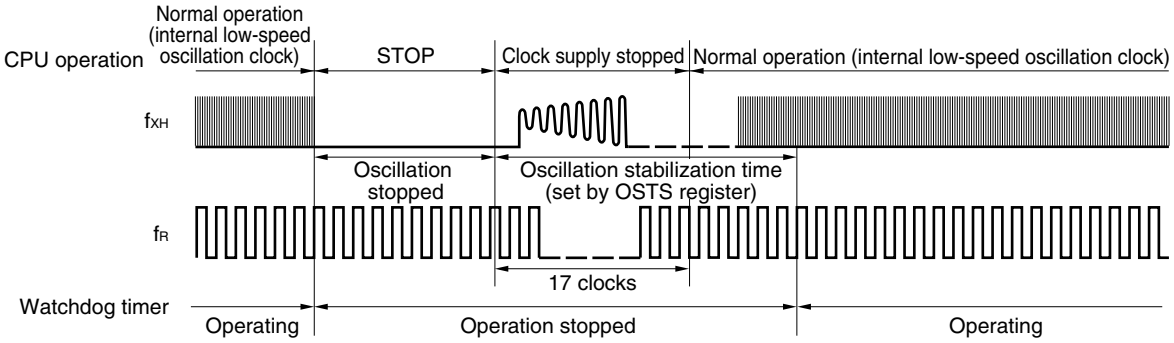
(3) When the CPU clock is the internal low-speed oscillation clock (f_R) and the watchdog timer operation clock is the high-speed system clock (f_{XH}) when the STOP instruction is executed

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.

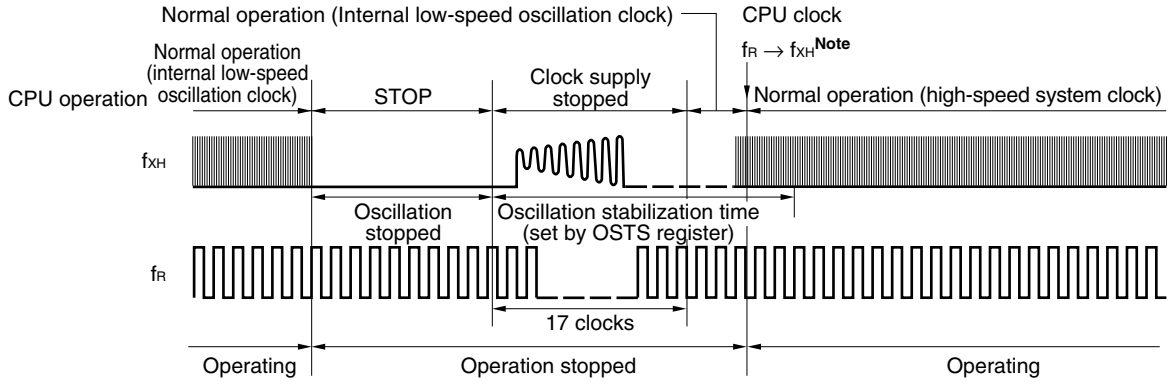
<1> The oscillation stabilization time set by the oscillation stabilization time select register (OSTS) elapses.
<2> The CPU clock is switched to the high-speed system clock (f_{XH}).

Figure 9-6. Operation in STOP Mode
(CPU Clock: Internal Low-Speed Oscillation 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



<2> Timing when counting is started after the CPU clock is switched to the high-speed system clock (f_{XH})

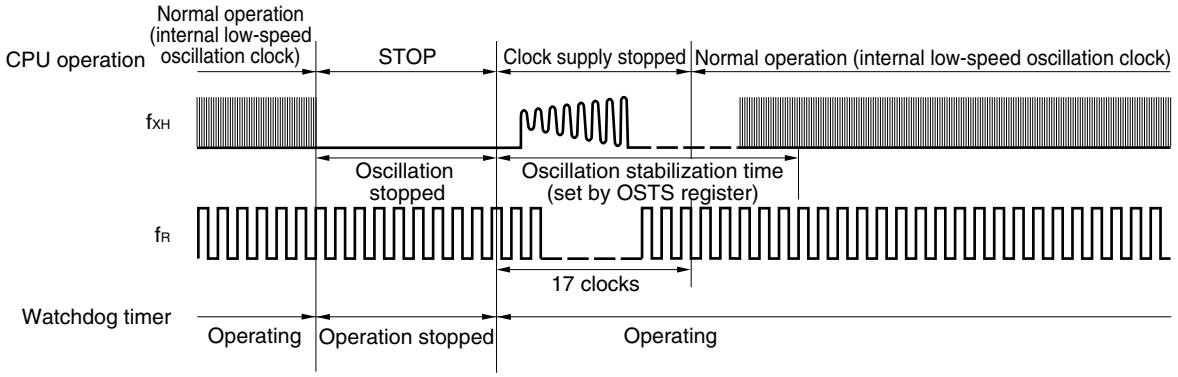


Note Confirm the oscillation stabilization time of f_{XH} using the oscillation stabilization time counter status register (OSTC).

(4) When the CPU clock and the watchdog timer operation clock are the internal low-speed oscillator 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.

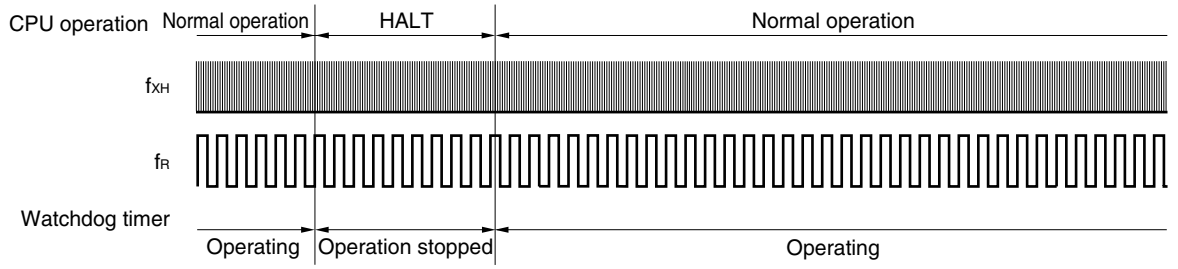
Figure 9-7. Operation in STOP Mode
(CPU Clock and WDT Operation Clock: Internal Low-Speed Oscillation Clock)



9.4.4 Watchdog timer operation in HALT mode (when "Internal low-speed oscillator can be stopped by software" is selected by mask option)

The watchdog timer stops counting during HALT instruction execution regardless of whether the CPU clock is the high-speed system clock (f_{XH}) or the internal low-speed oscillation clock (f_R), or whether the operation clock of the watchdog timer is the high-speed system clock (f_{XH}) or the internal low-speed oscillation 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 9-8. Operation in HALT Mode



CHAPTER 10 A/D CONVERTER

10.1 Function of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of up to four channels (ANI0 to ANI3) with a resolution of 10 bits.

The A/D converter has the following two functions.

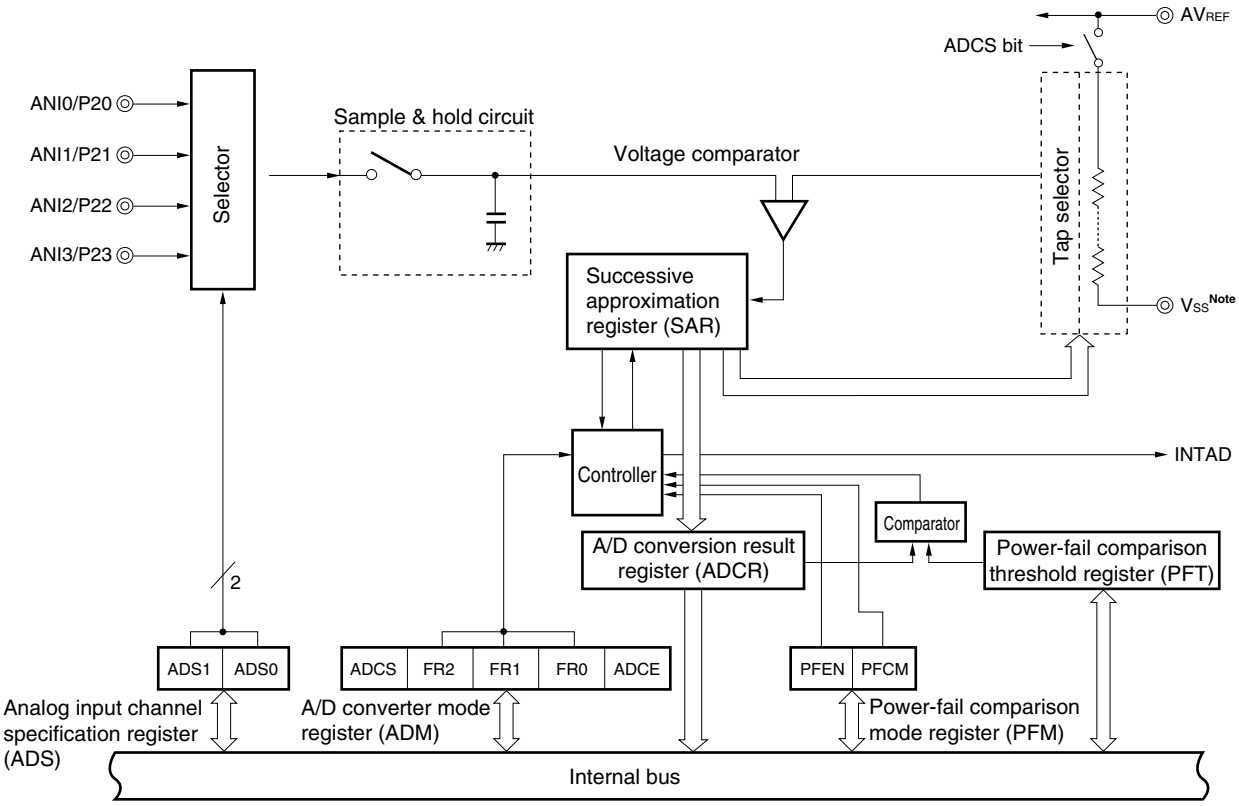
(1) 10-bit resolution A/D conversion

10-bit resolution A/D conversion is carried out repeatedly for one channel selected from analog inputs ANI0 to ANI3. 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 values of the A/D conversion result (ADCR register value) and power-fail comparison threshold register (PFT) are compared. INTAD is generated only when a comparative condition has been matched.

Figure 10-1. Block Diagram of A/D Converter



Note V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

Table 10-1. 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 ANI3 pins

These are the analog input pins of the 4-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.

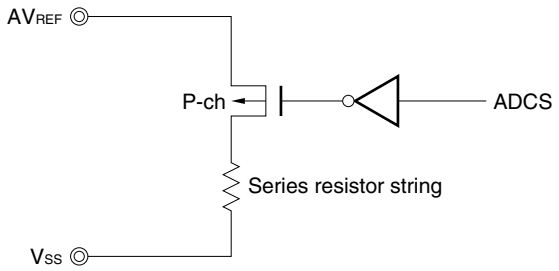
(2) Sample & hold circuit

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 V_{SS}, and generates a voltage to be compared with the analog input signal.

Figure 10-2. 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 ANI3 is converted into a digital signal, based on the voltage applied across AV_{REF} and V_{SS}.

(9) V_{SS} pin

The V_{SS} pin is the ground potential pin.

Caution V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

(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).

10.3 Registers Used in A/D Converter

The A/D converter uses the following five registers.

- 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 this register to 00H.

Figure 10-3. 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]			
				f_X = 2 MHz	f_X = 8.38 MHz	f_X = 10 MHz
0	0	0	288/f_X	144 μs	34.3 μs	28.8 μs
0	0	1	240/f_X	120 μs	28.6 μs	24.0 μs
0	1	0	192/f_X	96 μs	22.9 μs	19.2 μs
1	0	0	144/f_X	72 μs	17.2 μs	14.4 μs
1	0	1	120/f_X	60 μs	14.3 μs	12.0 μs
1	1	0	96/f_X	48 μs	11.5 μs	9.6 μs
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 as follows.

- Standard products, (A) grade products: 14 μs or longer but less than 100 μs
- (A1) grade products: 14 μs or longer but less than 60 μs
- (A2) grade products: 16 μs or longer but less than 48 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.

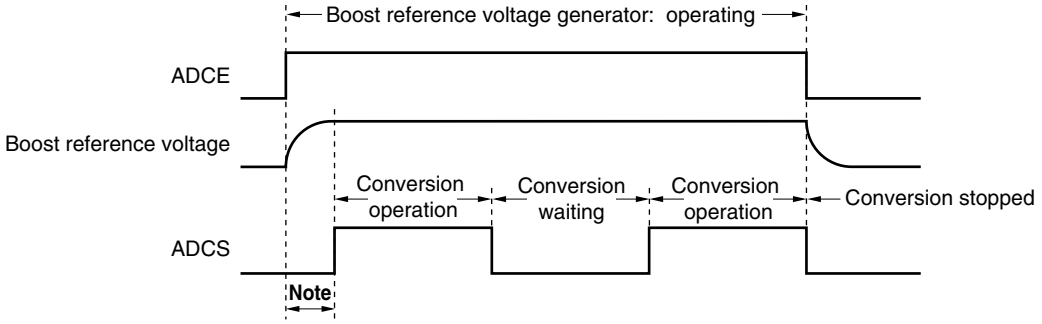
Remark f_X: High-speed system clock oscillation frequency

Table 10-2. 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 10-4. 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 A/D converter sampling time and A/D conversion start delay time, refer to 10.6 Cautions for A/D Converter (11).

3. If data is written to ADM, a wait cycle is generated. For details, refer to CHAPTER 28 CAUTIONS FOR WAIT.

Remark fx: High-speed system clock oscillation frequency

(2) Analog input channel specification register (ADS)

This register specifies the analog voltage input port to be A/D converted.

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

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

Figure 10-5. 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	0	ADS1	ADS0

ADS1	ADS0	Analog input channel specification
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

Cautions 1. Be sure to set bits 2 to 7 of ADS to 0.

2. If data is written to ADS, a wait cycle is generated. For details, refer to CHAPTER 28 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 10-6. 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. For details, see CHAPTER 28 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 this register to 00H.

Figure 10-7. 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. For details, refer to CHAPTER 28 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 this register to 00H.

Figure 10-8. 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. For details, refer to CHAPTER 28 CAUTIONS FOR WAIT.

10.4 A/D Converter Operations

10.4.1 Basic operations of A/D converter

<R> <1> Set ADCE to 1.

<R> <2> Select the channel and the conversion time to be used in the analog input mode by using ADS1, ADS0, and FR2 to FR0.

<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.

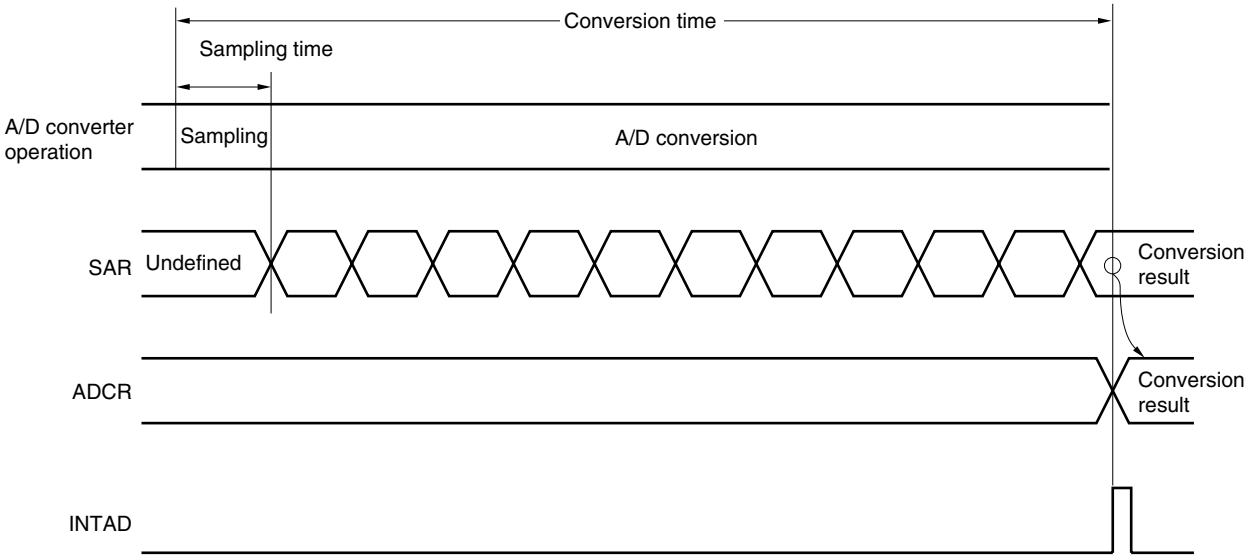
<11> Repeat steps <4> to <10>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <3>. To restart A/D conversion from the status of ADCE = 0, however, start from <2>.

<R> **Cautions 1. Make sure the period of <1> to <3> is 14 μs or more.**

<R> **2. It is no problem if the order of <1> and <2> is reversed.**

<R> **3. <1> can be omitted. However, do not use the first conversion in this case.**

Figure 10-9. 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.

10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3) and the theoretical A/D conversion result (stored in the A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

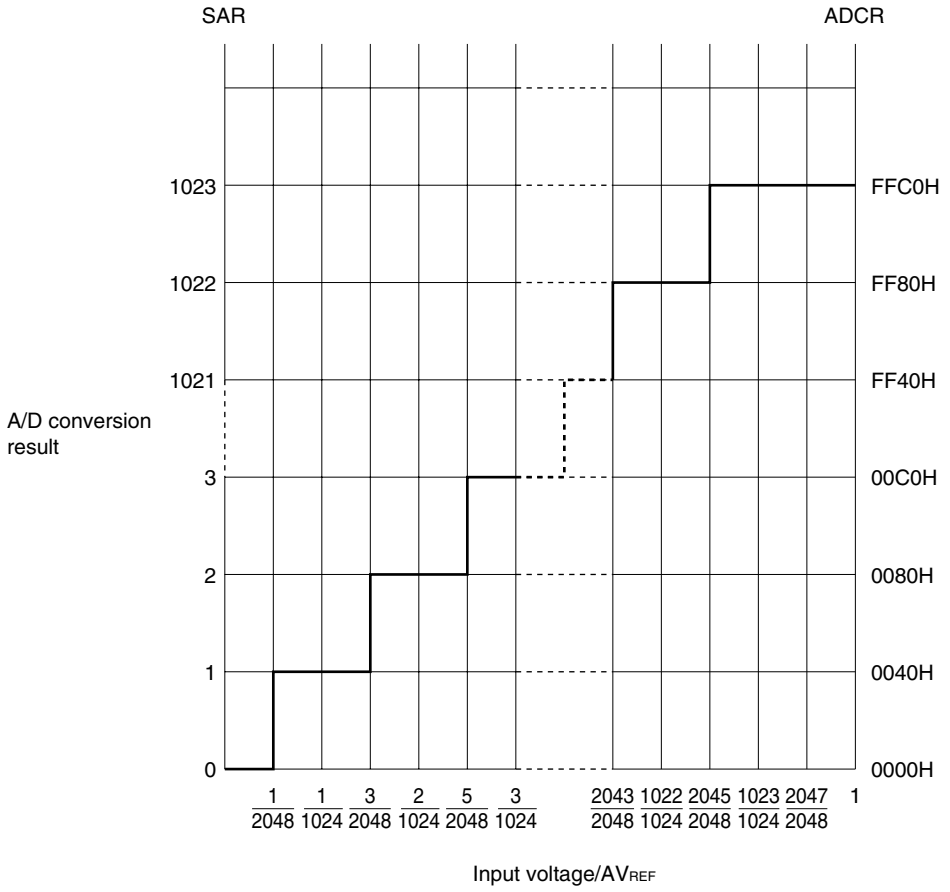
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 10-10 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-10. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 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 ANI3 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 of 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)

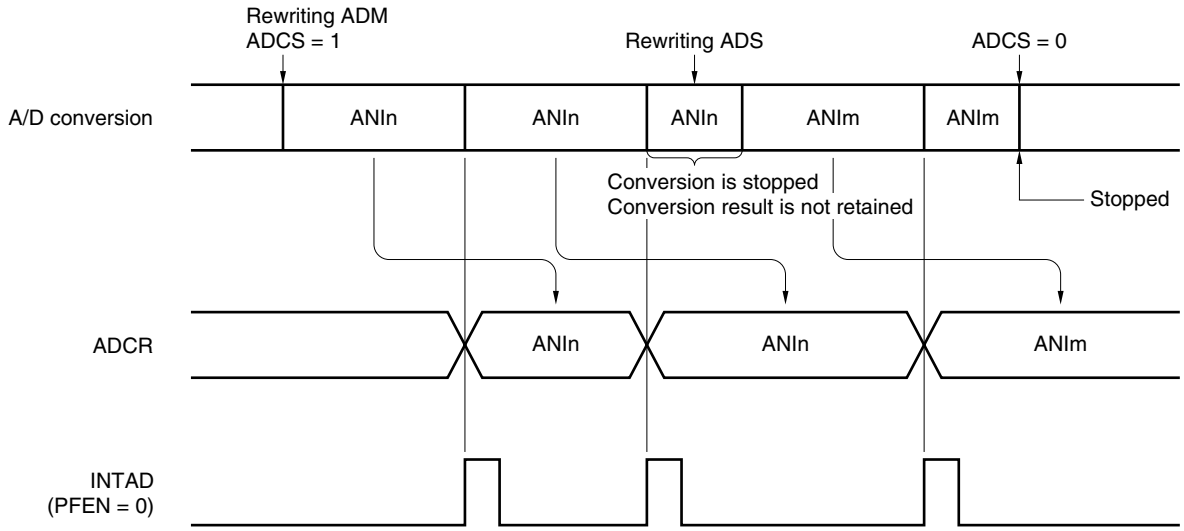
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 0, A/D conversion of the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) is started.

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 next A/D conversion has started and when one A/D conversion has been completed, the A/D conversion operation after that 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 10-11. A/D Conversion Operation



Remarks 1. n = 0 to 3

2. m = 0 to 3

(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, which 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.

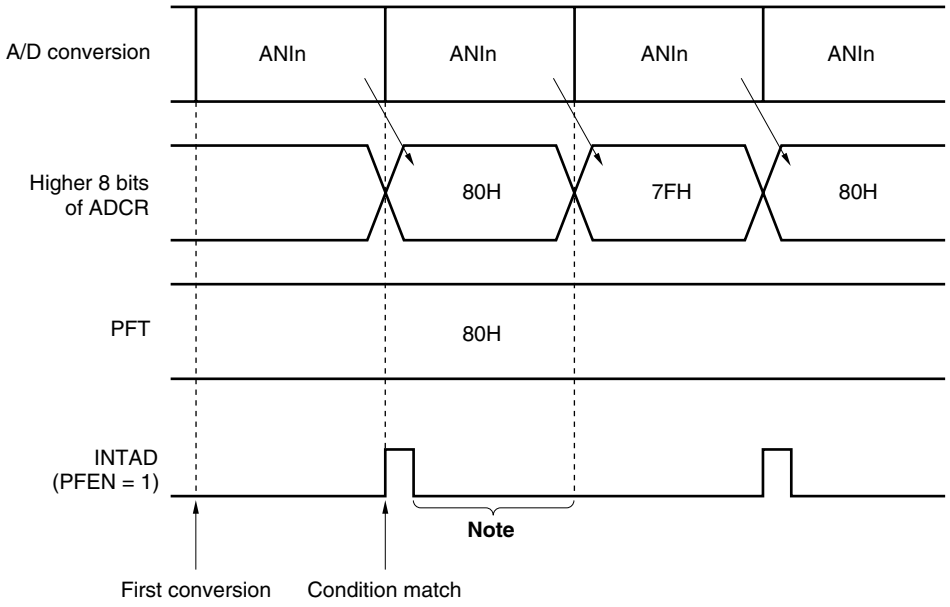
<1> When PFEN = 1 and PFCM = 0

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 10-12. 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 3

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 1 and 0 (ADS1, 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 and start the A/D conversion operation.
  - <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 1 and 0 (ADS1, ADS0) of ADS and start the A/D conversion operation.
  - <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).
  - <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 1 and 0 (ADS1, 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 1 and 0 (ADS1, 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.

10.5 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.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±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 10-13. Overall Error

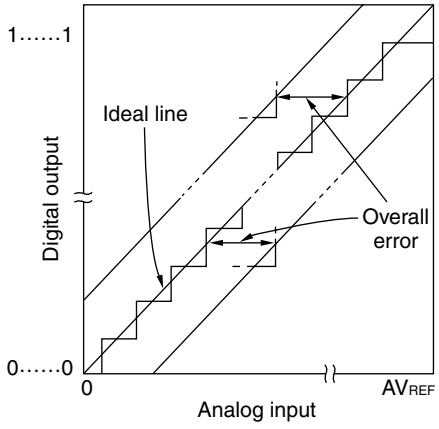
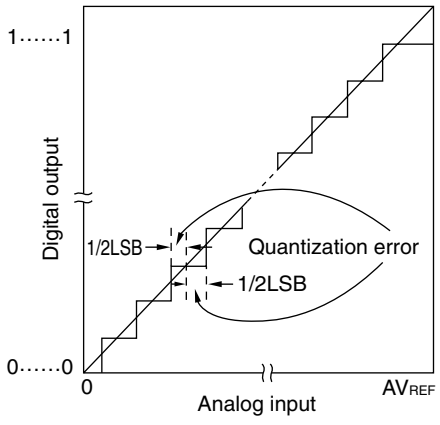


Figure 10-14. Quantization Error



(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 10-15. Zero-Scale Error

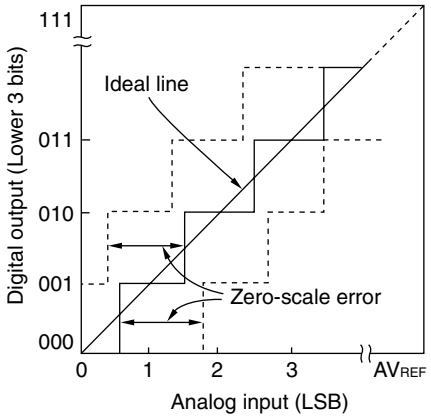


Figure 10-16. Full-Scale Error

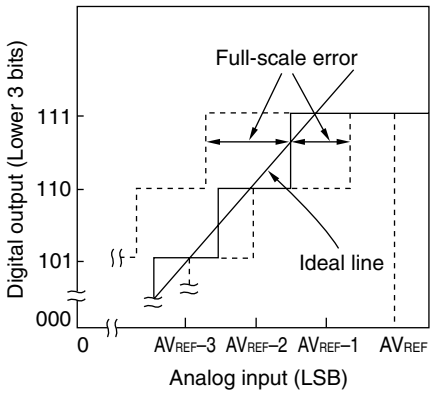


Figure 10-17. Integral Linearity Error

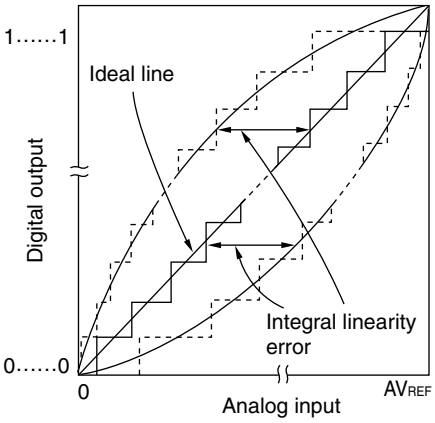
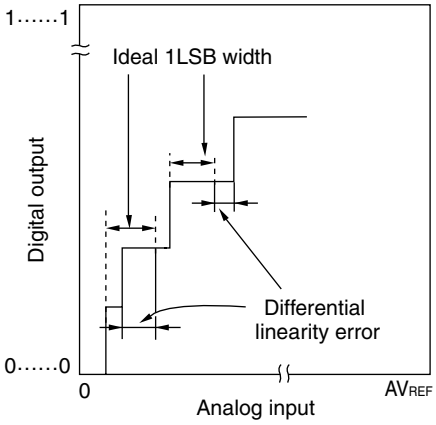


Figure 10-18. Differential Linearity Error



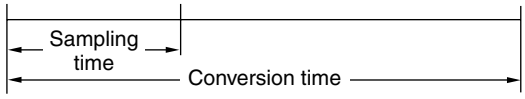
(8) Conversion time

This expresses the time from when the analog input voltage was applied to the time when the digital output was 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.



10.6 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 setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 (see **Figure 10-2**).

(2) Input range of ANI0 to ANI3

Observe the rated range of the ANI0 to ANI3 input voltage. If a voltage of AV_{REF} or higher and V_{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.

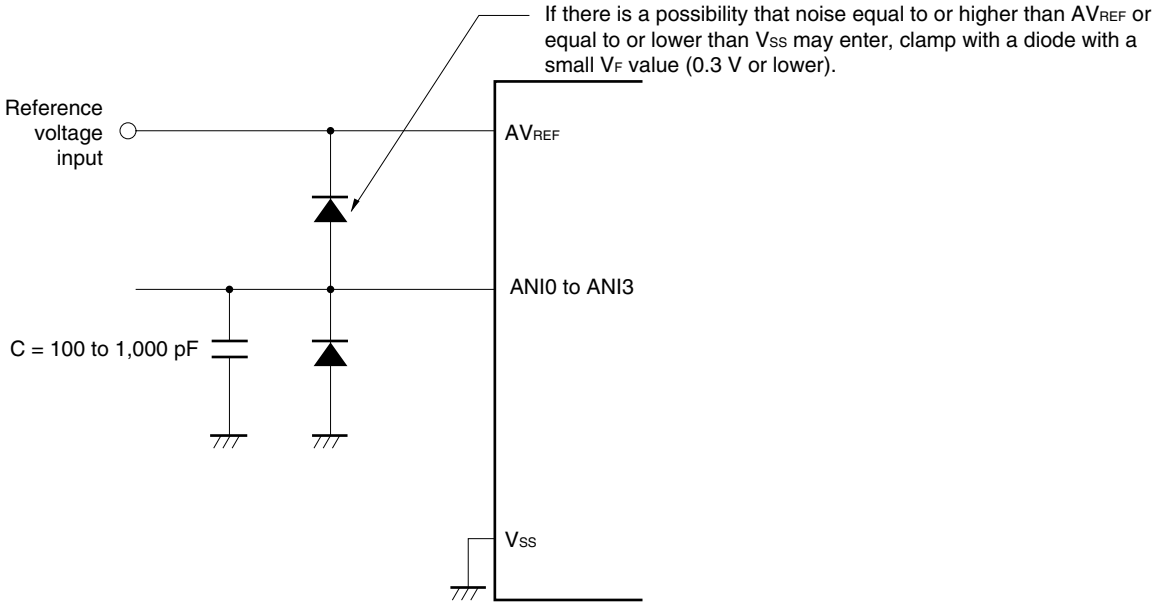
<2> Conflict between ADCR write and A/D converter mode register (ADM) write or analog input channel specification register (ADS) write upon the end of conversion

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 AV_{REF} pin and pins ANI0 to ANI3. 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 10-19, to reduce noise.

Figure 10-19. Analog Input Pin Connection



(5) ANI0/P20 to ANI3/P23

<1> The analog input pins (ANI0 to ANI3) are also used as input port pins (P20 to P23).
When A/D conversion is performed with any of ANI0 to ANI3 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 ANI3 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 ANI3 pins (see **Figure 10-19**).

(7) AV_{REF} pin input impedance

A series resistor string of several tens of 10 kΩ is connected between the AV_{REF} and V_{SS} 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 AV_{REF} and V_{SS} pins, resulting in a large reference voltage error.

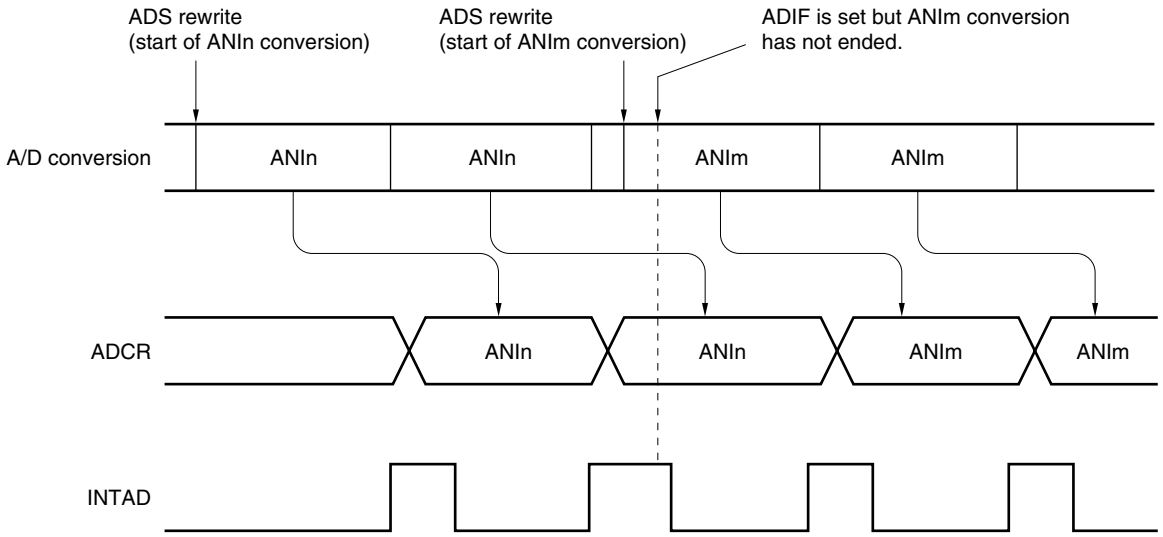
(8) Interrupt request flag (ADIF)

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 10-20. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 3

2. m = 0 to 3

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 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 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 10-21 and Table 10-3.

Figure 10-21. Timing of A/D Converter Sampling and A/D Conversion Start Delay

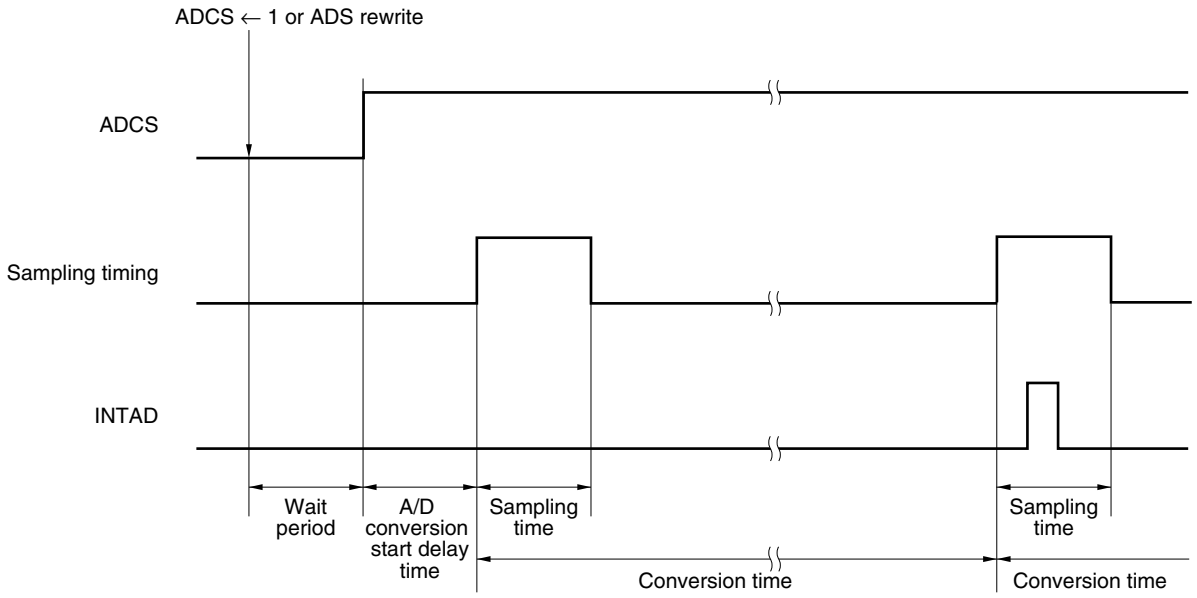


Table 10-3. A/D Converter Sampling Time and A/D Conversion Start Delay Time (ADM Set Value)

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 28 CAUTIONS FOR WAIT**.

Remark fx: High-speed system clock oscillation frequency

(12) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-22. Internal Equivalent Circuit of ANIn Pin

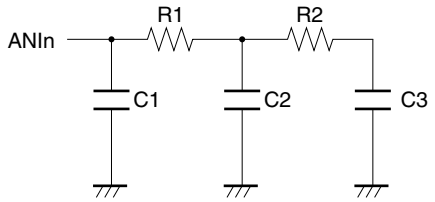


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	R2	C1	C2	C3	
					Mask ROM Version	Flash Memory Version
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF	0.6 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF	0.6 pF

Remarks 1. The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

2. n = 0 to 3

CHAPTER 11 SERIAL INTERFACE UART6

11.1 Functions of Serial Interface UART6

Serial interface UART6 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfer is not executed and can enable a reduction in the power consumption.
For details, refer to **11.4.1 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, see **11.4.2 Asynchronous serial interface (UART) mode** and **11.4.3 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 is 13-bit length output.
- 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 UART6 is used in the LIN communication.

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automotive network.

LIN uses single-master communication, and up to 15 slaves can be connected to one master.

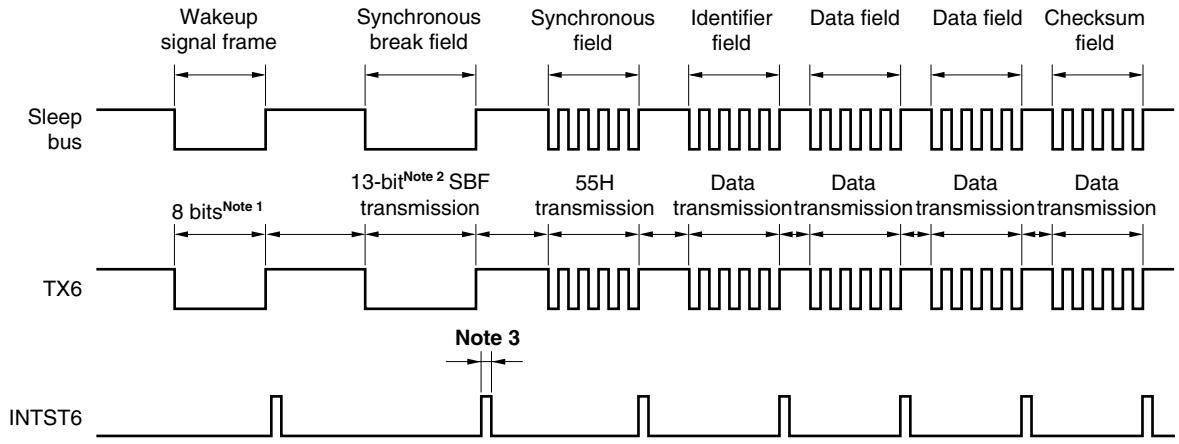
A LIN slave is used to control switches, actuators, and sensors, which are connected to the LIN master via the LIN.

The LIN master is usually connected to a network such as CAN (Controller Area Network). The LIN bus is a single-wire type and each node is connected to the bus via a transceiver conforming to ISO9141.

The LIN protocol defines that the master transmits frames that include baud rate information, and a slave receives this information and corrects the baud rate error to that of the master. Therefore, communication is enabled if the baud rate error of the slave is within ±15%.

Figures 11-1 and 11-2 outline the transmission and reception operations of LIN.

Figure 11-1. LIN Transmission Operation



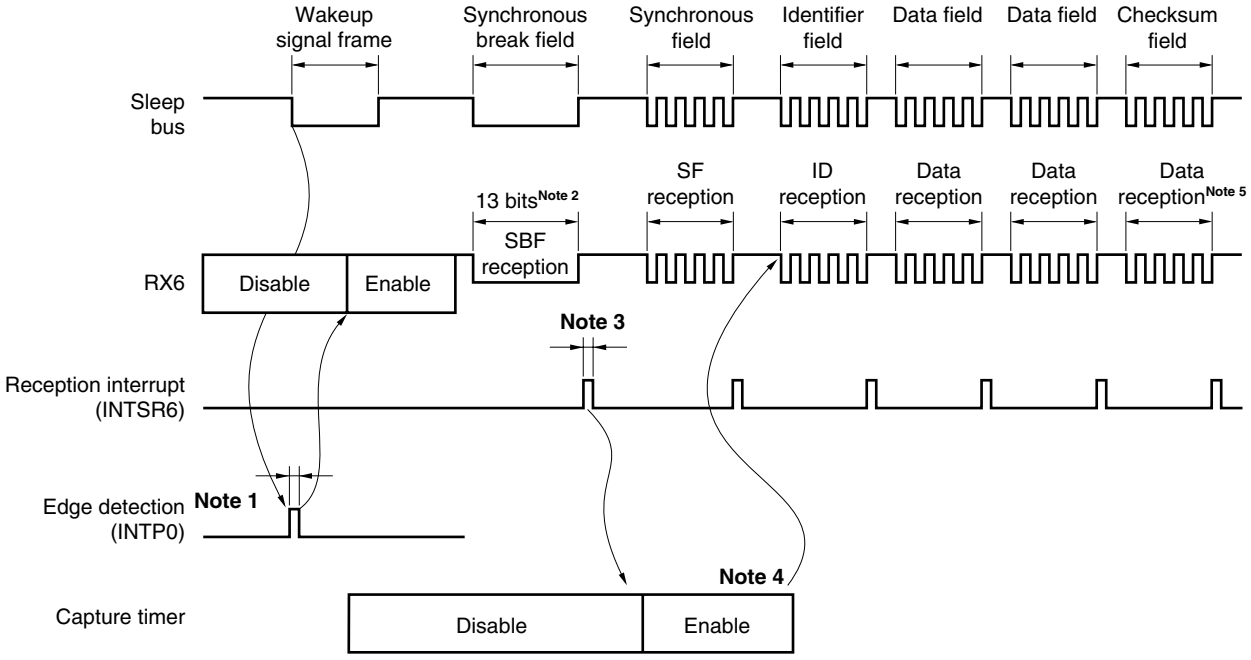
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The synchronous break field is output by hardware. The output width is adjusted by baud rate generator control register 6 (BRGC6) (see **11.4.2 (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 11-2. 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).

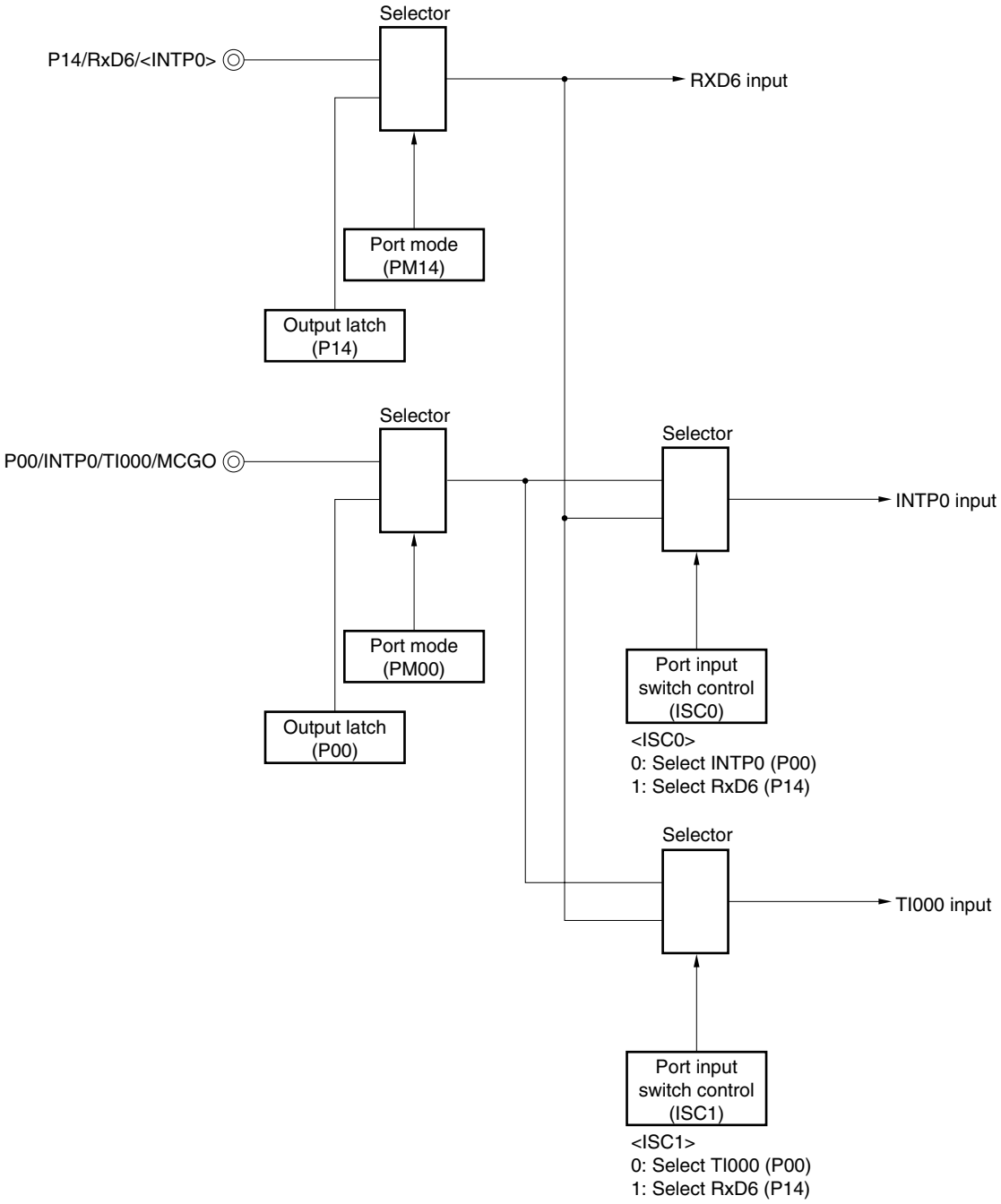
5. Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

To perform a LIN receive operation, use a configuration like the one shown in Figure 11-3.

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 signal 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 11-3. Port Configuration for LIN Reception Operation



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (see **Figure 11-11**)

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 synchronous field (SF) length and divides it by the number of bits.
- Serial interface UART6

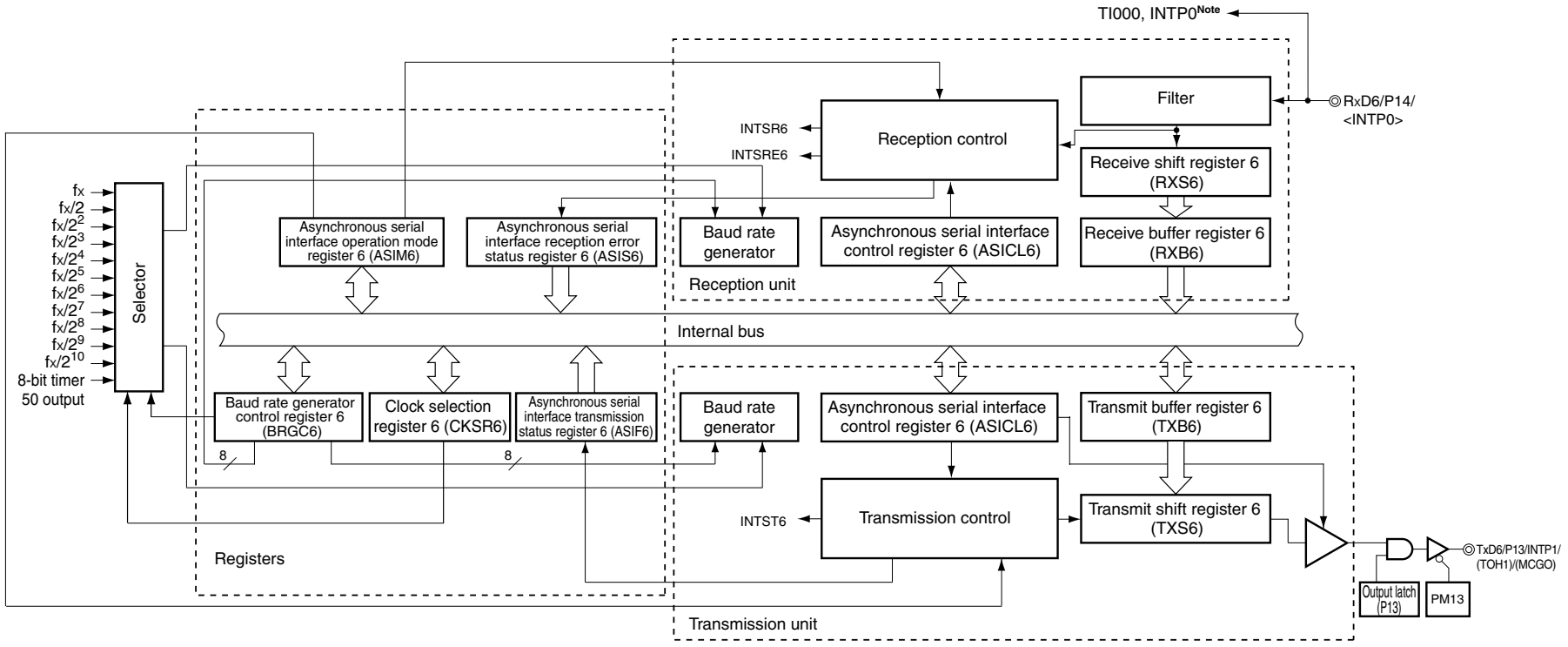
11.2 Configuration of Serial Interface UART6

Serial interface UART6 includes the following hardware.

Table 11-1. 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 11-4. 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 this register 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 this register 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.

11.3 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 this register 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 11-5. 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 the 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 11-5. 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 UART6 is used in the LIN communication operation.

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, bit 7 (POWER6) of ASIM6 = 0, or bit 5 (RXE6) of ASIM6 = 0 clears this register to 00H. 00H is read when this register is read.

Figure 11-6. 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 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. For details, refer to CHAPTER 28 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, bit 7 (POWER6) of ASIM6 = 0, or bit 5 (RXE6) of ASIM6 = 0 clears this register to 00H

Figure 11-7. 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 this register 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 11-8. 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
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]
Other than above				Setting prohibited

Note When the TM50 output is selected as the base clock, observe the following.

- PWM mode (TMC506 = 1)
  Set the clock so that the duty will be 50% and start the operation of 8-bit timer/event counter 50 in advance.
- Clear & start mode entered on match of TM50 and CR50 (TMC506 = 0)
  Enable the timer F/F inversion operation (TMC501 = 1) and start the operation of 8-bit timer/event counter 50 in advance.

Cautions 1. When the internal oscillation clock is selected as the clock to be supplied to the CPU, the clock of the internal oscillator is divided and supplied as the count clock. If the base clock is the internal oscillation 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 are for 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 this register 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 11-9. 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 this register 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). However, do not set both SBRT6 and SBTT6 to 1 by a refresh operation during SBF reception (SBRT6 = 1) or SBF transmission (until INTST6 occurs since SBTT6 has been set (1)), because it may re-trigger SBF reception or SBF transmission.

Figure 11-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6)

Address: FF58H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	0	1	0	1	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

DIR6	First bit specification
0	MSB
1	LSB

TXDLV6	Enables/disables inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

Note Bits 2 to 5 and 7 are read-only.

<R> **Cautions 1. In the case of an SBF reception error, return the mode to the SBF reception mode. The status of SBRF6 flag is held (1).**

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 rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.

(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 this register to 00H.

Figure 11-11. 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 (P00)
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/INTP1/(TOH1)/(MCGO) pin for serial interface data output, clear PM13 to 0 and set the output latch of P13 to 1.

When using the P14/RxD6/<INTP0> 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 this register to FFH.

Figure 11-12. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

11.4 Operation of Serial Interface UART6

Serial interface UART6 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

11.4.1 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 this register 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/<INTP0> and TxD6/P13/INTP1/(TOH1)/(MCGO) pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

11.4.2 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 (see **Figure 11-8**).
<2> Set the BRGC6 register (see **Figure 11-9**).
<3> Set bits 0 to 4 (ISRM6, SL6, CL6, PS60, PS61) of the ASIM6 register (see **Figure 11-5**).
<4> Set bits 0 and 1 (TXDLV6, DIR6) of the ASICL6 register (see **Figure 11-10**).
<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 11-2. Relationship Between Register Settings and Pins

POWER6	TXE6	RXE6	PM13	P13	PM14	P14	UART6 Operation	Pin Function	
								TxD6/P13/INTP1/ (TOH1)/(MCGO)	RxD6/P14/<INTP0>
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

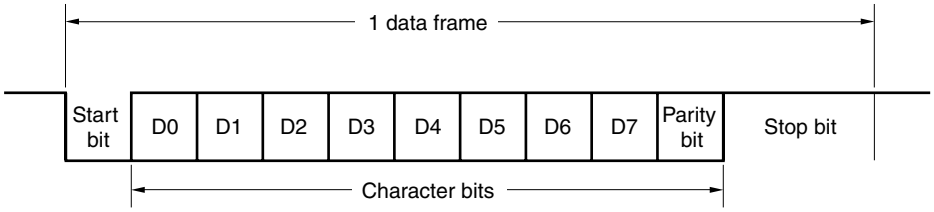
(2) Communication operation

(a) Normal transmit/receive data format

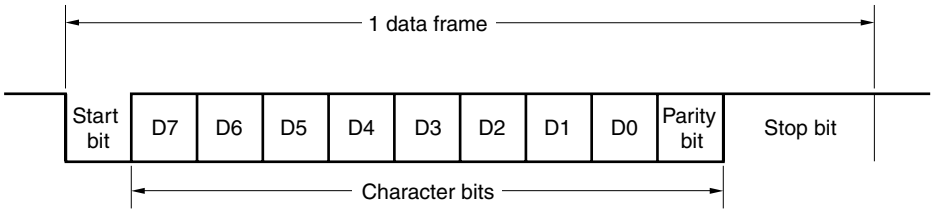
Figure 11-13 shows the format and waveform example of the normal transmit/receive data.

Figure 11-13. 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

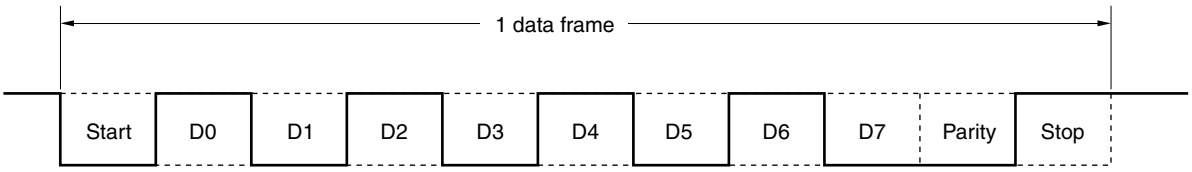
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6 (ASIM6).

Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).

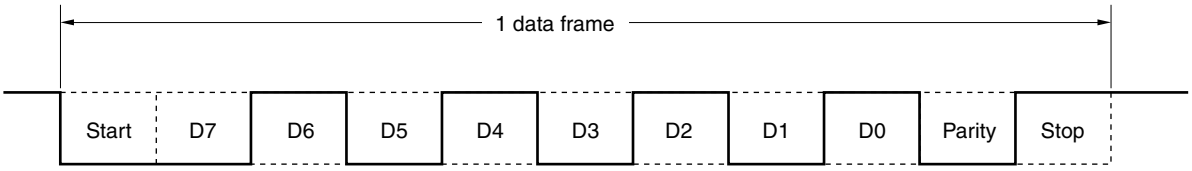
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 11-14. Example of Normal UART Transmit/Receive Data Waveform

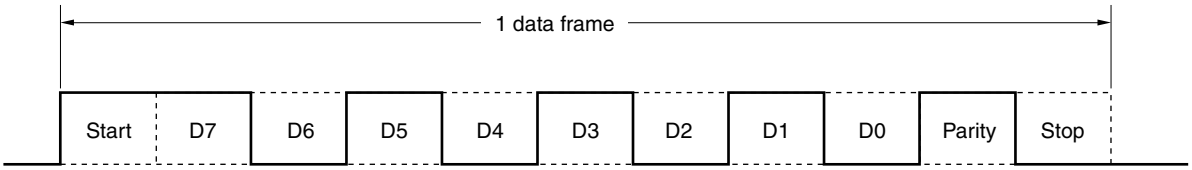
1. Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



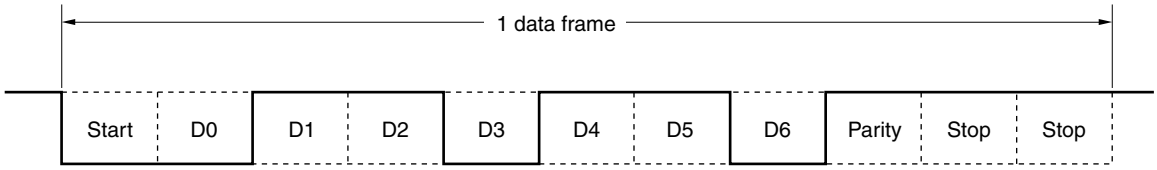
2. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



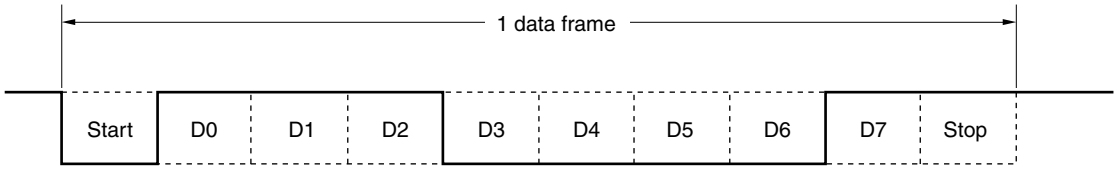
3. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6 pin inverted output



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 used in LIN communication operation.

(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.

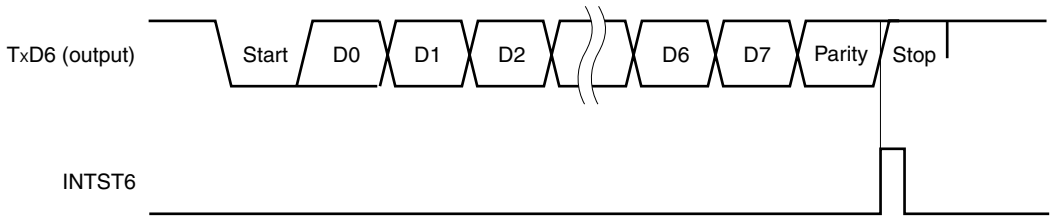
When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the data is sequentially output from TXS6 to the TxD6 pin. When transmission is completed, the parity bit and stop bit set by ASIM6 are appended and a transmission completion interrupt request (INTST6) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6.

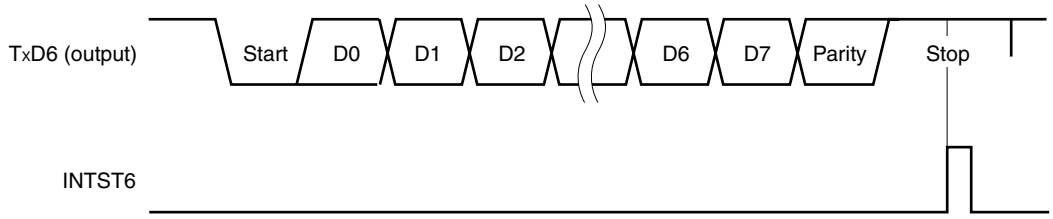
Figure 11-15 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

Figure 11-15. 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 used in LIN communication, 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.

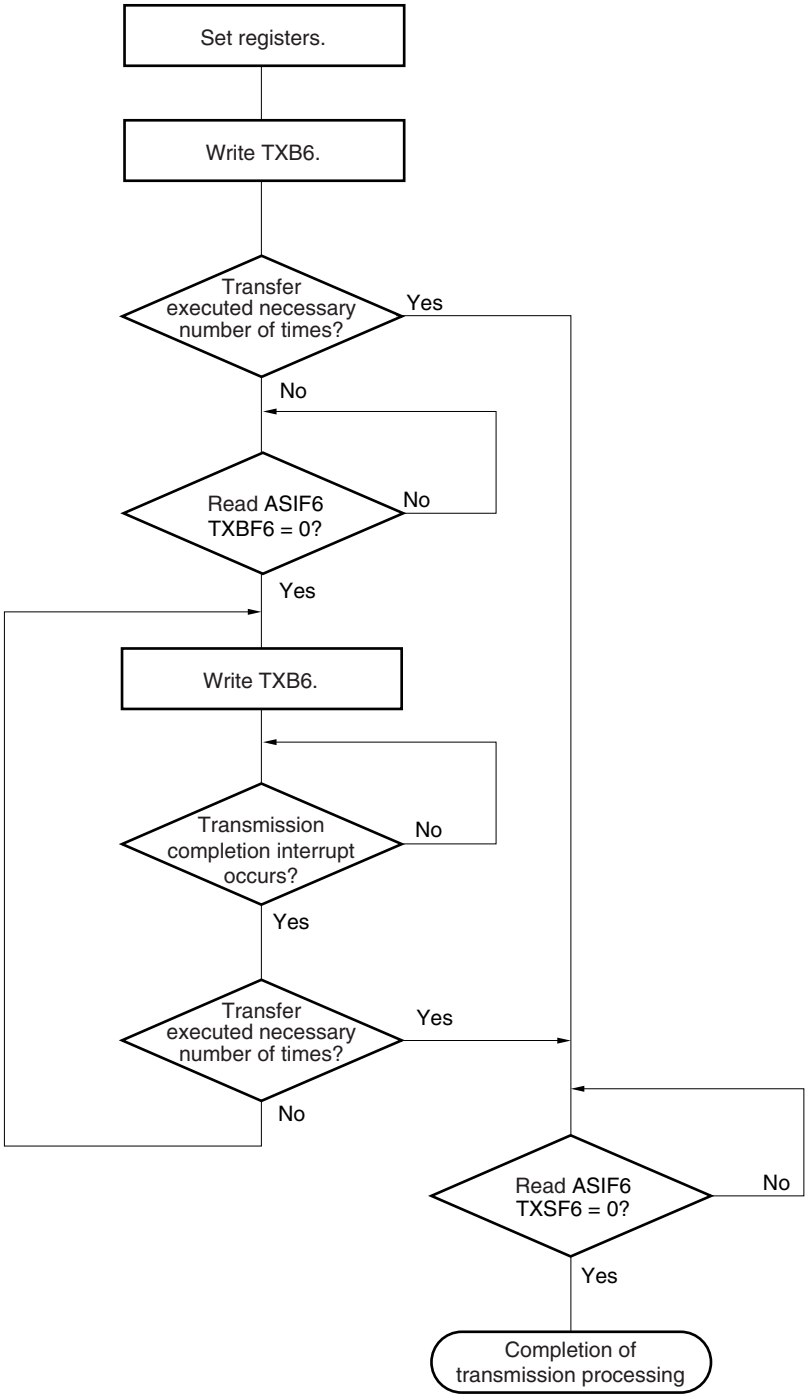
TXSF6	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

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 11-16 shows an example of the continuous transmission processing flow.

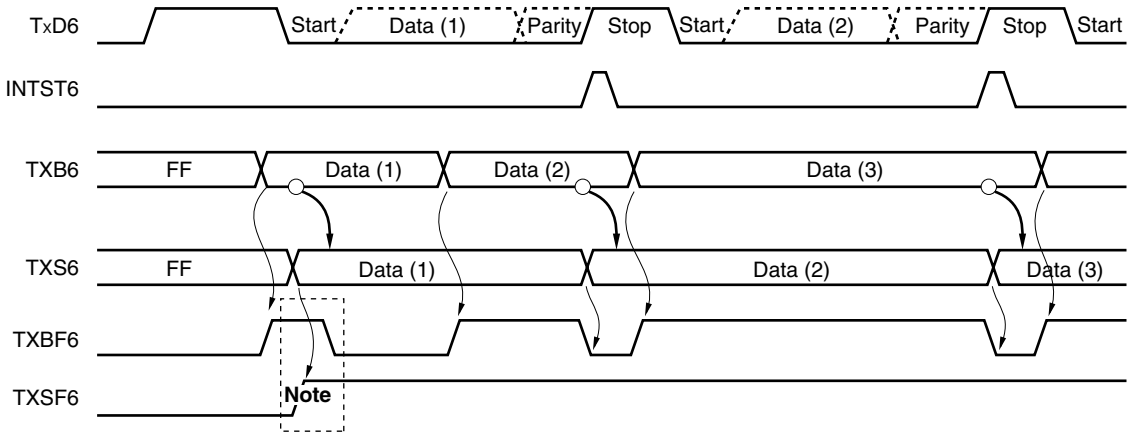
Figure 11-16. Example of Continuous Transmission Processing Flow



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 11-17 shows the timing of starting continuous transmission, and Figure 11-18 shows the timing of ending continuous transmission.

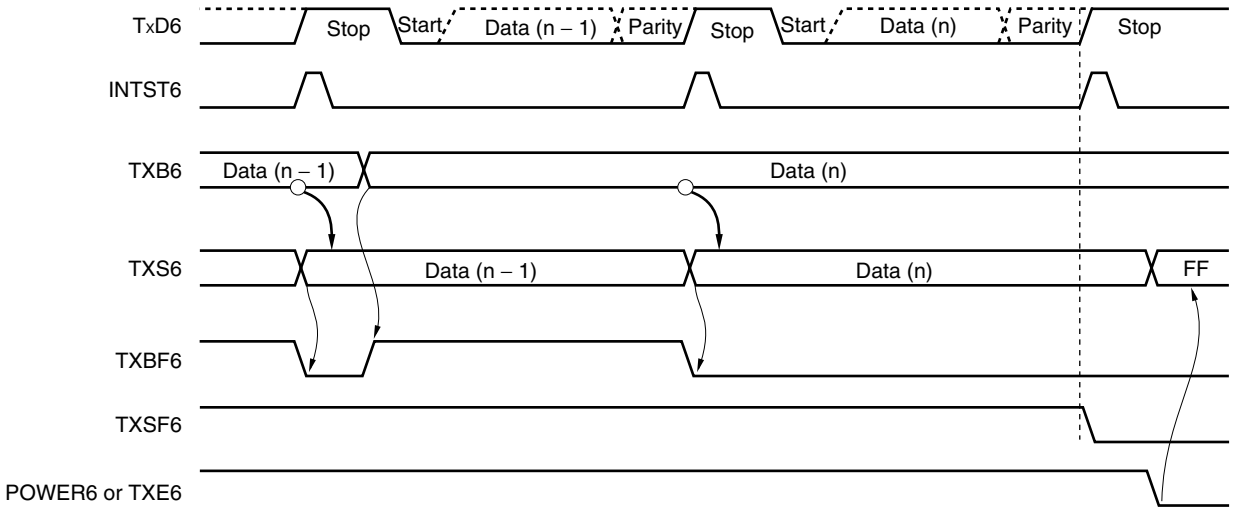
Figure 11-17. Timing of Starting Continuous Transmission



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 11-18. 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

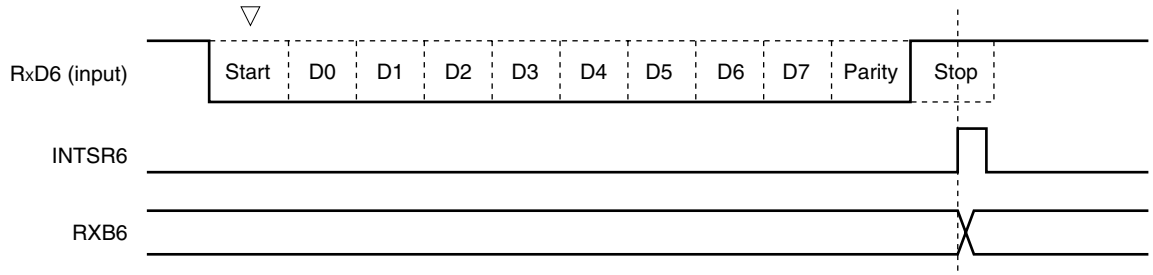
Reception is enabled and the RxD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

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 11-19). 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 11-19. 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) (see **Figure 11-6**).

The contents of ASIS6 are reset to 0 when ASIS6 is read.

Table 11-3. Cause of Reception Error

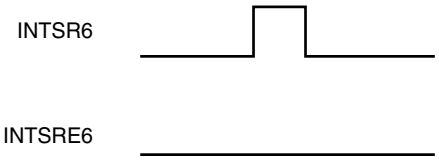
Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).

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.

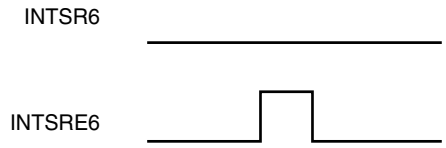
Figure 11-20. Reception Error Interrupt

1. If ISRM6 is cleared to 0 (reception completion interrupt (INTSR6) and error interrupt (INTSRE6) are separated)

(a) No error during reception

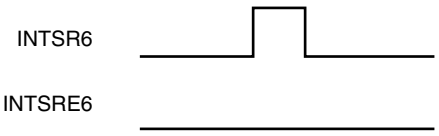


(b) Error during reception

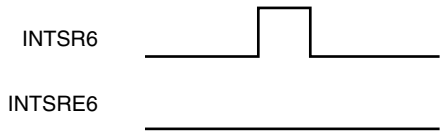


2. If ISRM6 is set to 1 (error interrupt is included in INTSR6)

(a) No error during reception



(b) Error during reception



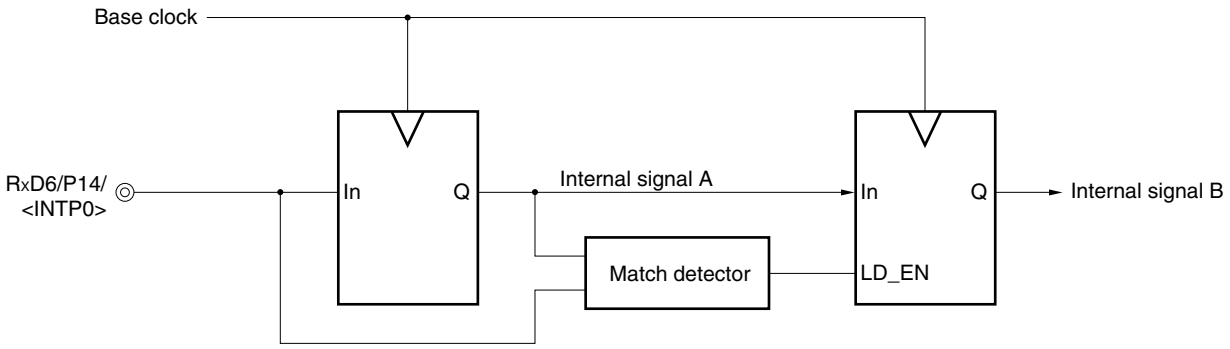
(g) Noise filter of receive data

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 11-21, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 11-21. Noise Filter Circuit



(h) SBF transmission

When the device is used in LIN communication operation, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, see **Figure 11-1 LIN Transmission Operation**.

SBF transmission is used to transmit an SBF length that is a low-level width of 13 bits or more by adjusting the baud rate value of the ordinary 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 adjust this 10-bit low level to the targeted SBF length.

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 the baud rate generator control register 6 (BRGC6) is set to 130.

- 13-bit SBF length = 0.2 μs $\times$ 130 $\times$ 2 $\times$ 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 $\times$ 169 $\times$ 2 $\times$ 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 11-22. Example of Setting Procedure of SBF Transmission (Flowchart)

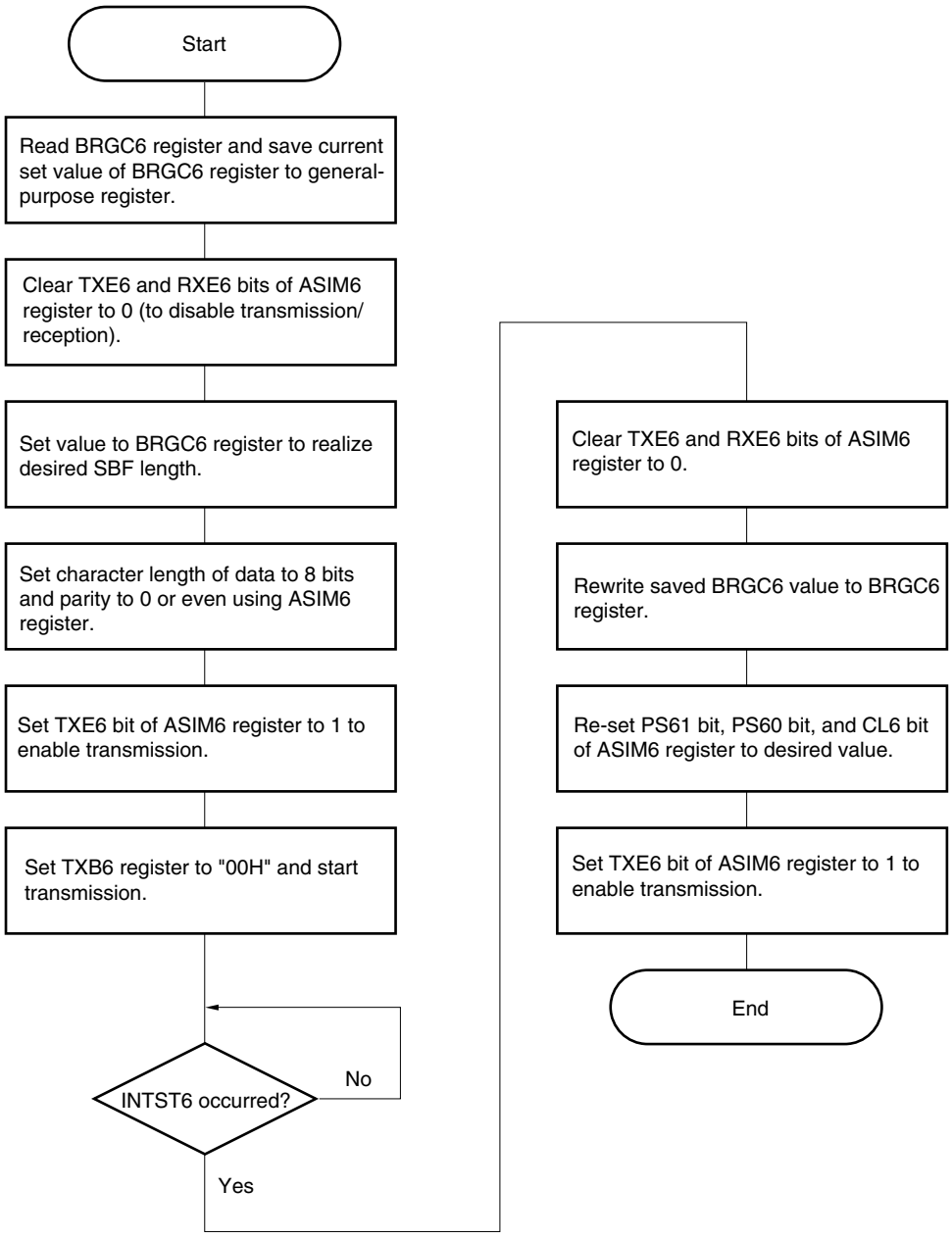
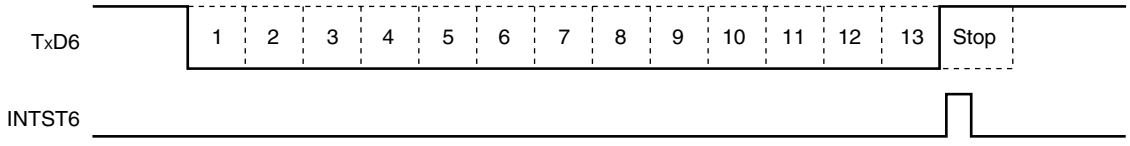


Figure 11-23. SBF Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Transmission completion interrupt request

(i) SBF reception

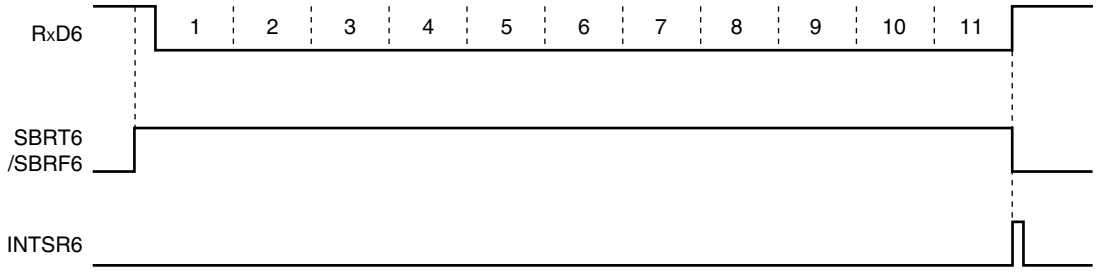
When the device is used in LIN communication operation, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, refer to **Figure 11-2 LIN Reception Operation**.

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.

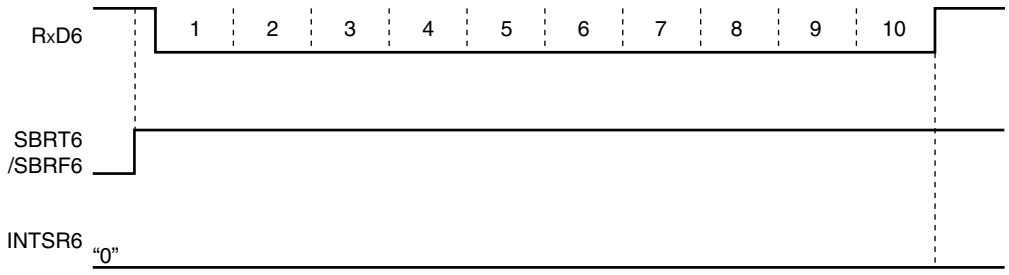
When the start bit has been detected, reception is started, and serial data is sequentially stored in receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal 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 11-24. 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

11.4.3 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 the 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 the low level when POWER6 = 0.

- Transmission counter

  This counter stops, 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.

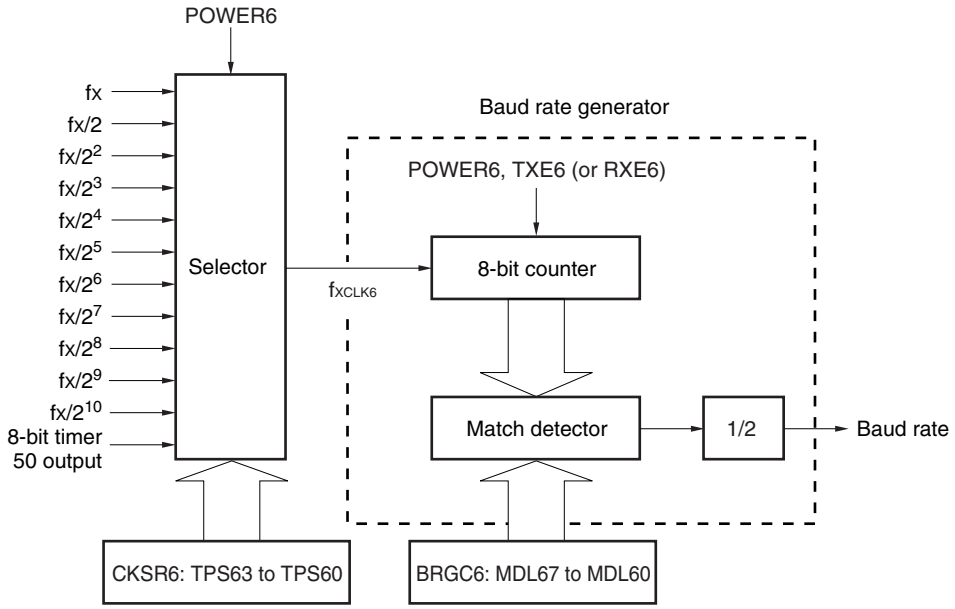
- Reception counter

  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 11-25. 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]

f_{XCLK6}: Frequency of the 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) = 151515 [bps]

Error = (151515/153600 − 1) × 100
= −1.357 [%]

(3) Example of setting baud rate

Table 11-4. 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

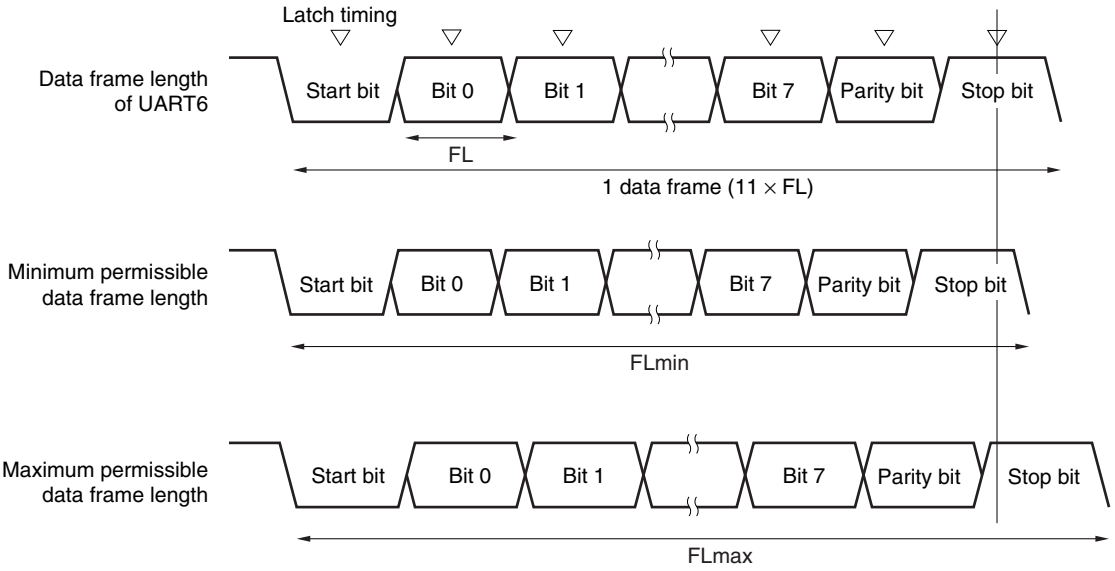
Remark TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock (f_{XCLK6}))
k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) (k = 8, 9, 10, ..., 255)
fx: High-speed system clock oscillation frequency
ERR: Baud rate error

(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 11-26. Permissible Baud Rate Range During Reception



As shown in Figure 11-26, 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: $FLmin = 11 \times FL \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2}Brate$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

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 11-5. 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%

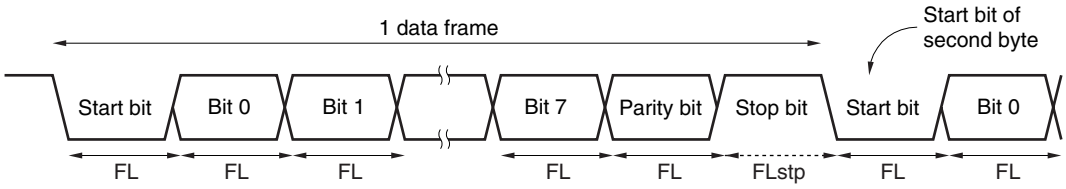
Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

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 11-27. 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 12 SERIAL INTERFACE CSI10

12.1 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, see **12.4.1 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 for connecting peripheral ICs and display controllers with a clocked serial interface.

For details, see **12.4.2 3-wire serial I/O mode**.

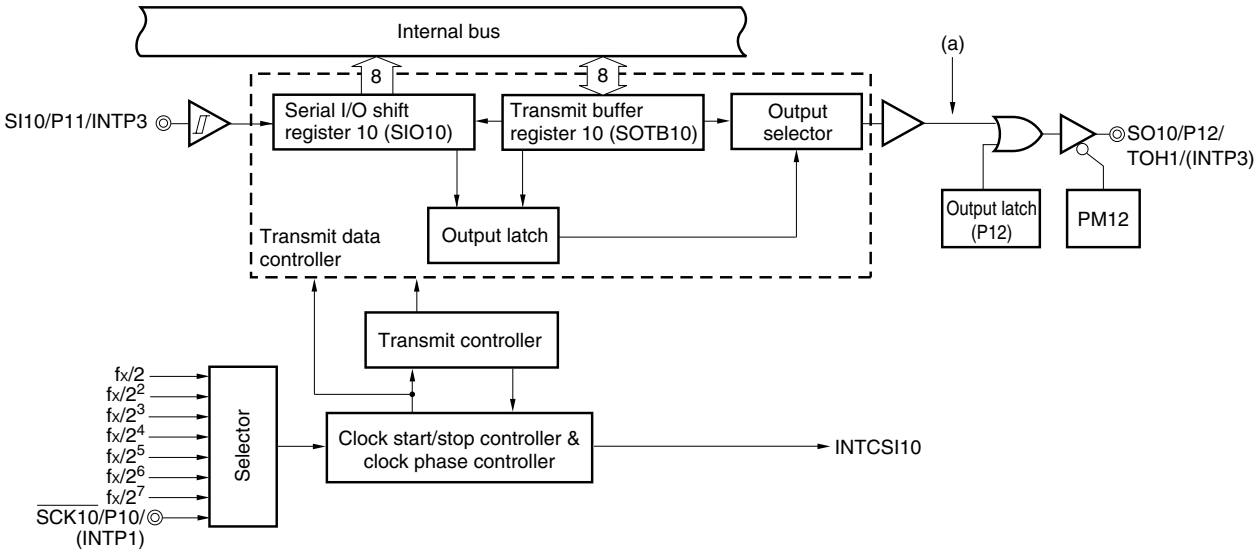
12.2 Configuration of Serial Interface CSI10

Serial interface CSI10 includes the following hardware.

Table 12-1. 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 1 (PM1) Port register 1 (P1)

Figure 12-1. 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 this register 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 or vice versa.

This register can be read by an 8-bit memory manipulation instruction.

Reception is started by reading data from SIO10 when 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 this register to 00H.

Caution Do not access SIO10 when CSOT10 = 1 (during serial communication).

12.3 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)

This register is used to select the operation mode and enable or disable operation.

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

$\overline{\text{RESET}}$ input sets this register to 00H.

Figure 12-2. 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	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

<R> **2.** When using P10/$\overline{\text{SCK10}}$/(INTP1), and P12/SO10/TOH1/(INTP3) as a general-purpose port, set CSIM10 in the default status (00H).

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)

This register is used to select the phase of the data clock and set the count clock.

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

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

Figure 12-3. 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	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

Cautions 1. When the internal oscillation clock is selected as the clock supplied to the CPU, the clock of the internal 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).

<R> **3. When using P10/$\overline{\text{SCK10}}$/(INTP1) and P12/SO10/TOH1/(INTP3) as general-purpose port, set CSIC10 in the default status (00H).**

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/$\overline{\text{SCK10}}$/(INTP1) as the clock output pins of the serial interface, clear PM10 to 0 and set the output latch of P10 to 1.

When using P12/SO10/TOH1/(INTP3) as the data output pins, clear PM12 and the output latch of P12 to 0.

When using P10/$\overline{\text{SCK10}}$/(INTP1) as the clock input pins of the serial interface, and P11/SI10/INTP3 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 this register to FFH.

Figure 12-4. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 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

12.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P10/$\overline{\text{SCK10}}$/(INTP1), P11/SI10/INTP3, and P12/SO10/TOH1/(INTP3) 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].

<R>

Notes 1. When using P10/$\overline{\text{SCK10}}$/(INTP1), and P12/SO10/TOH1/(INTP3) as a general-purpose port, set CSIM10 in the default status (00H).

2. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

12.4.2 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 (see **Figure 12-3**).

<2> Set bits 0, 4, and 6 (CSOT10, DIR10, and TRMD10) of the CSIM10 register (see **Figure 12-2**).

<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 12-2. Relationship Between Register Settings and Pins

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 Operation	Pin Function		
									P11/SI10 /INTP3	P12/SO10 /TOH1 /(INTP3)	P10/$\overline{\text{SCK10}}$ /(INTP1)
0	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	Stop	P11 /INTP3	P12 /TOH1 /(INTP3)	P10 /(INTP1)Note2
1	0	1	×	×Note 1	×Note 1	1	×	Slave receptionNote 3	SI10	P12 /TOH1 /(INTP3)	$\overline{\text{SCK10}}$ (input)Note 3
1	1	×Note 1	×Note 1	0	0	1	×	Slave transmissionNote 3	P11 /INTP3	SO10	$\overline{\text{SCK10}}$ (input)Note 3
1	1	1	×	0	0	1	×	Slave transmission/ receptionNote 3	SI10	SO10	$\overline{\text{SCK10}}$ (input)Note 3
1	0	1	×	×Note 1	×Note 1	0	1	Master reception	SI10	P12 /TOH1 /(INTP3)	$\overline{\text{SCK10}}$ (output)
1	1	×Note 1	×Note 1	0	0	0	1	Master transmission	P11 /INTP3	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/$\overline{\text{SCK10}}$/(INTP1) 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 12-5. Timing in 3-Wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)

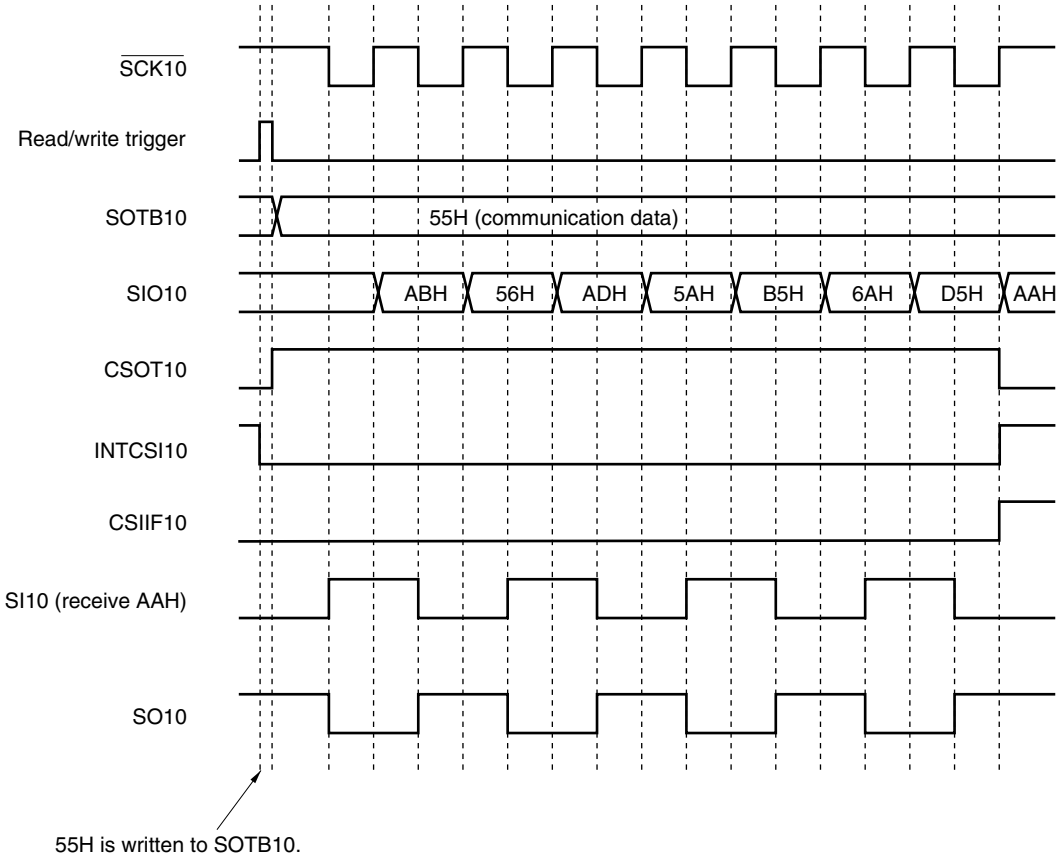


Figure 12-5. Timing in 3-Wire Serial I/O Mode (2/2)

(2) Transmission/reception timing (Type 2; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

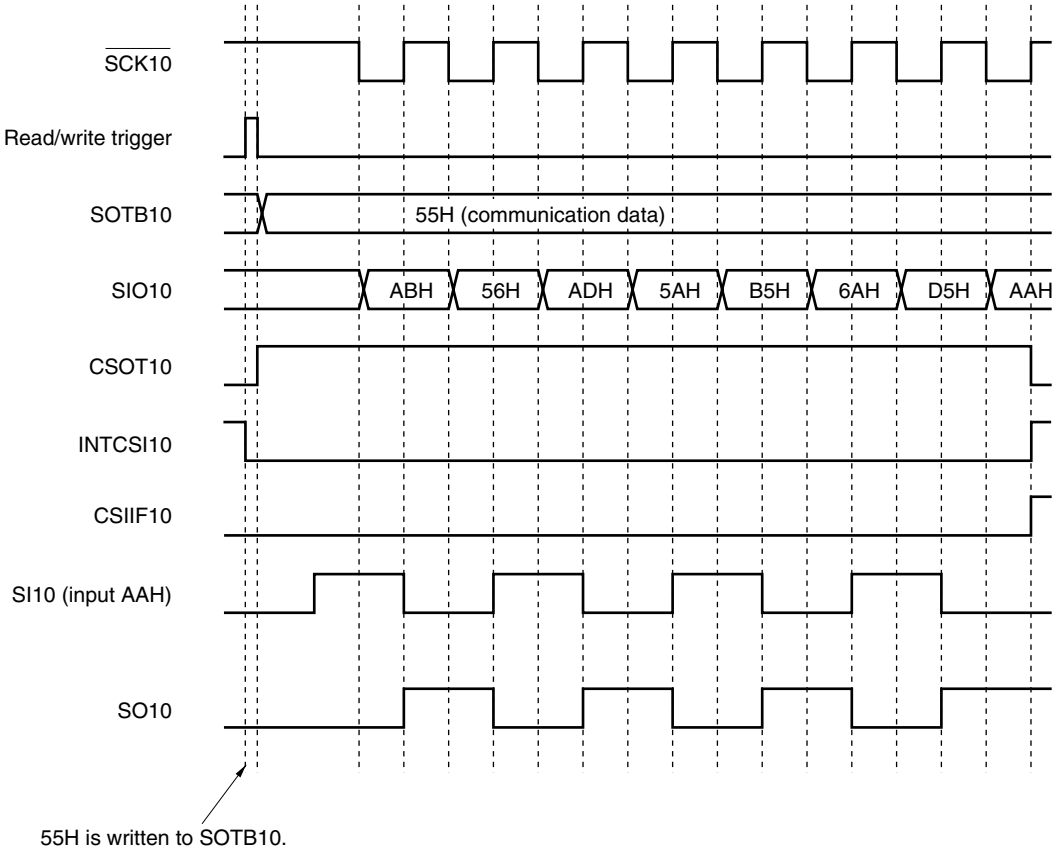
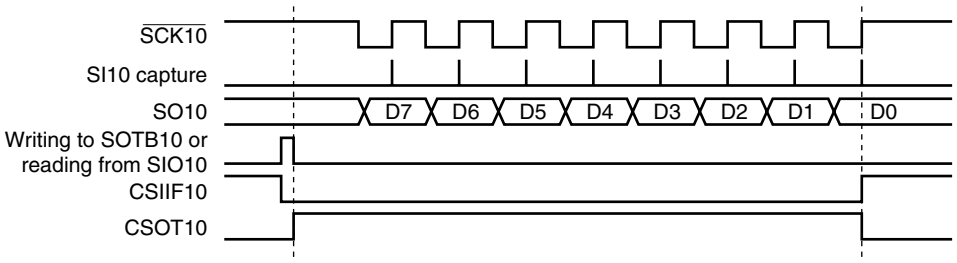


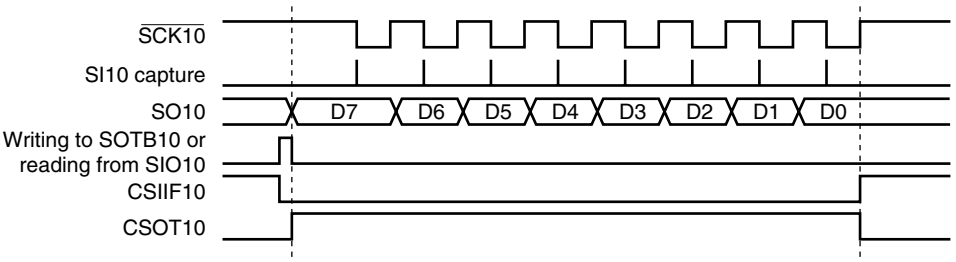
Figure 12-6. Timing of Clock/Data Phase

(a) Type 1; CKP10 = 0, DAP10 = 0

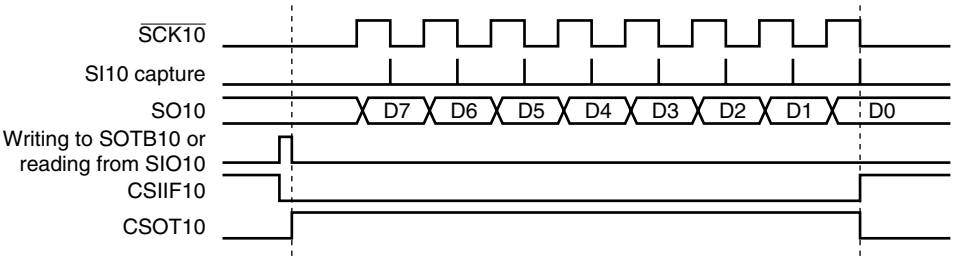


(b) Type 2; CKP10 = 0, DAP10 = 1

<R>

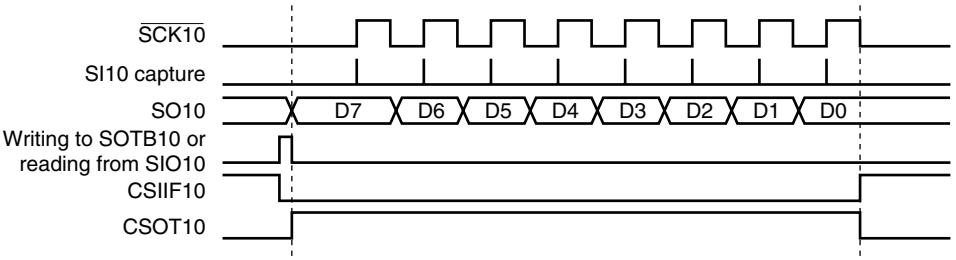


(c) Type 3; CKP10 = 1, DAP10 = 0



(d) Type 4; CKP10 = 1, DAP10 = 1

<R>

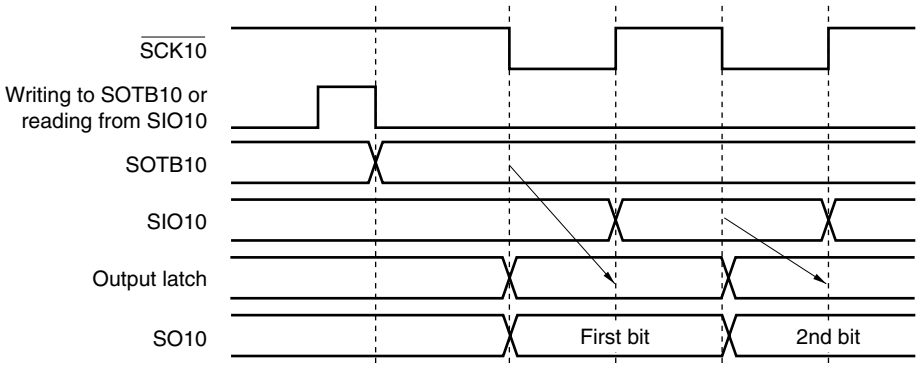


(3) Timing of output to SO10 pin (first bit)

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 12-7. Output Operation of First Bit

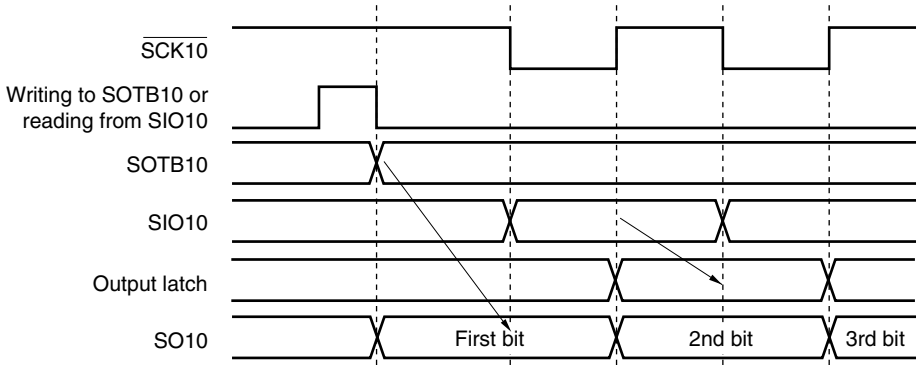
(1) When CKP10 = 0, DAP10 = 0 (or CKP10 = 1, DAP10 = 0)



The first bit is directly latched by the SOTB10 register to the output latch at the falling (or rising) edge of the 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 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 SCK10, and the data is output from the SO10 pin.

(2) When CKP10 = 0, DAP10 = 1 (or CKP10 = 1, DAP10 = 1)



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 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.

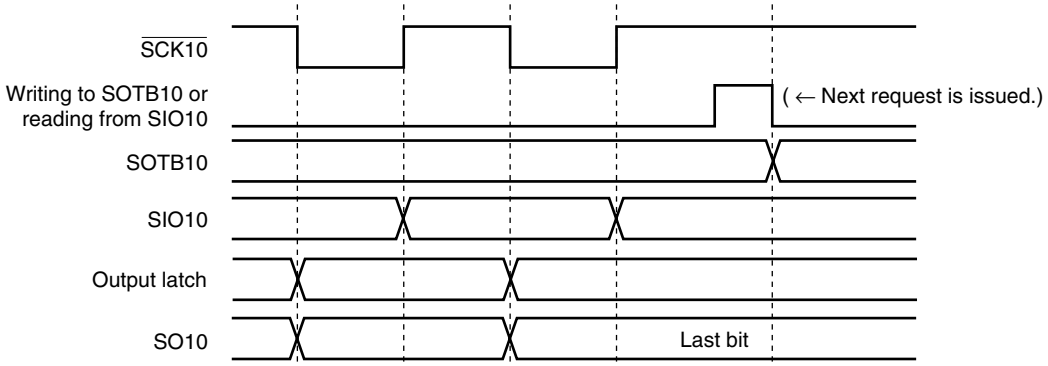
The second and subsequent bits are latched by the SIO10 register to the output latch at the next rising (or falling) edge of SCK10, and the data is output from the SO10 pin.

(4) Output value of SO10 pin (last bit)

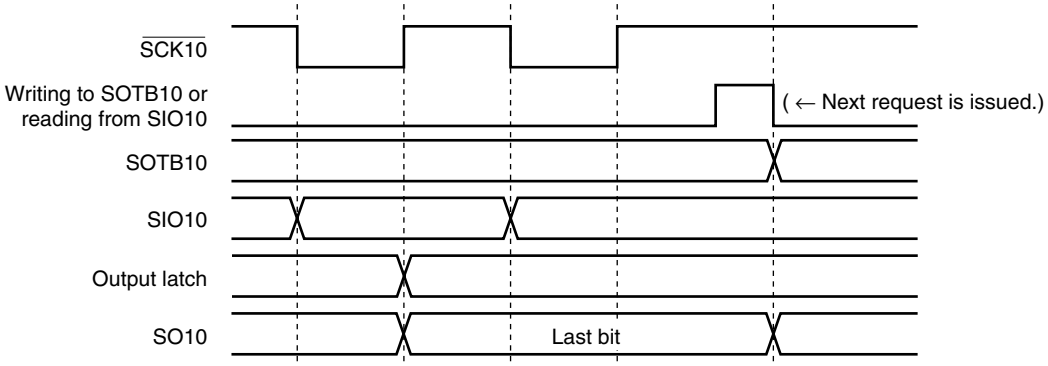
After communication has been completed, the SO10 pin holds the output value of the last bit.

Figure 12-8. Output Value of SO10 Pin (Last Bit)

(1) Type 1; when CKP10 = 0 and DAP10 = 0 (or CKP10 = 1, DAP10 = 0)



(2) Type 2; when CKP10 = 0 and DAP10 = 1 (or CKP10 = 1, DAP10 = 1)



(5) SO10 output (see (a) in Figure 12-1)

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 12-3. 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/TOH1/(INTP3) pin is determined by PM12 and P12 as well as SO10 output.

2. Status after reset

Caution If a value is written to TRMD10, DAP10, and DIR10, the output value of SO10 changes.

CHAPTER 13 MANCHESTER CODE GENERATOR

13.1 Functions of Manchester Code Generator

The following three types of modes are available for the Manchester code generator.

(1) Operation stop mode

This mode is used when output by the Manchester code generator/bit sequential buffer is not performed. This mode reduces the power consumption.

For details, refer to **13.4.1 Operation stop mode**.

(2) Manchester code generator mode

This mode is used to transmit Manchester code from the MCGO pin.

The transfer bit length can be set and transfers of various bit lengths are enabled. Also, the output level of the data transfer and LSB- or MSB-first can be set for 8-bit transfer data.

(3) Bit sequential buffer mode

This mode is used to transmit bit sequential data from the MCGO pin.

The transfer bit length can be set and transfers of various bit lengths are enabled. Also, the output level of the data transfer and LSB- or MSB-first can be set for 8-bit transfer data.

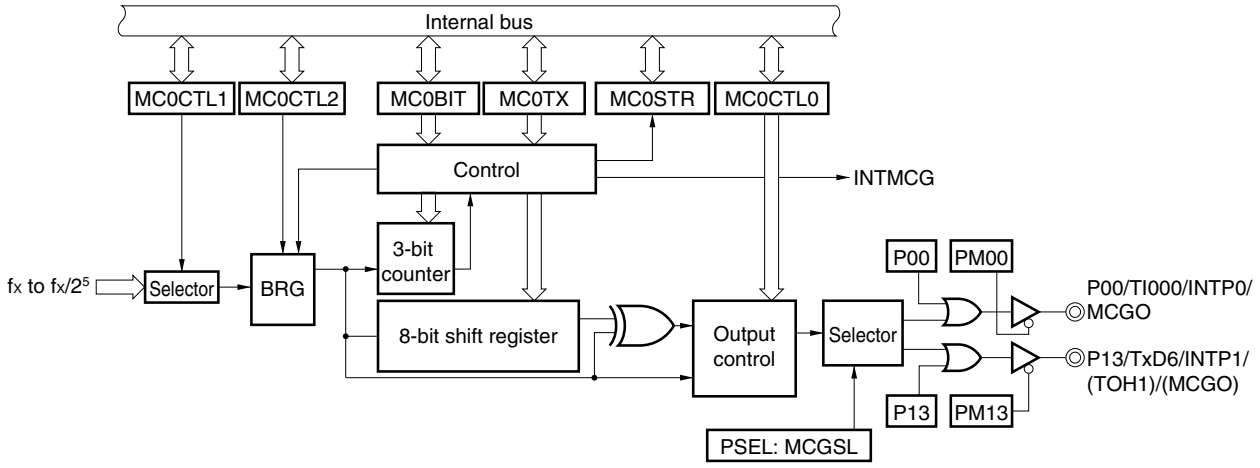
13.2 Configuration of Manchester Code Generator

The Manchester code generator includes the following hardware.

Table 13-1. Configuration of Manchester Code Generator

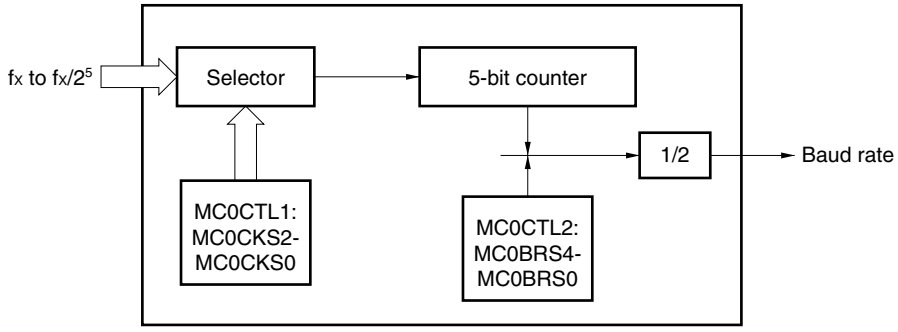
Item	Configuration
Registers	MCG transmit buffer register (MC0TX) MCG transmit bit count specification register (MC0BIT)
Control registers	MCG control register 0 (MC0CTL0) MCG control register 1 (MC0CTL1) MCG control register 2 (MC0CTL2) MCG status register (MC0STR) Port mode registers 0, 1 (PM0, PM1) Port registers 0, 1 (P0, P1)

Figure 13-1. Block Diagram of Manchester Code Generator



Remark BRG: Baud rate generator
fx: High-speed system clock oscillation frequency
MC0BIT: MCG transmit bit count specification register
MC0CTL2 to MC0CTL0: MCG control registers 2 to 0
MC0STR: MCG status register
MC0TX: MCG transmit buffer register
MCGSL: Bit 0 of PSEL register
PSEL: Alternate-function pin switch register

Figure 13-2. Block Diagram of Baud Rate Generator



Remark fx: High-speed system clock oscillation frequency
MC0CTL2, MC0CTL 1: MCG control registers 2, 1
MC0CKS2 to MC0CKS0: Bits 2 to 0 of MC0CTL1 register
MC0BRS4 to MC0BRS0: Bits 4 to 0 of MC0CTL2 register

(1) MCG transmit buffer register (MC0TX)

This register is used to set the transmit data. A transmit operation starts when data is written to MC0TX while bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) is 1.

The data written to MC0TX is converted into serial data by the 8-bit shift register, and output to the MCGO pin.

Manchester code or bit sequential data can be set as the output code using bit 1 (MC0OSL) of MCG control register 0 (MC0CTL0).

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

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

(2) MCG transmit bit count specification register (MC0BIT)

This register is used to set the number of transmit bits.

Set the transmit bit count to this register before setting the transmit data to MC0TX.

In continuous transmission, the number of transmit bits to be transmitted next needs to be written after the occurrence of a transmission start interrupt (INTMCG). However, if the next transmit count is the same number as the previous transmit count, this register does not need to be written.

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

$\overline{\text{RESET}}$ input sets this register to 07H.

Figure 13-3. Format of MCG Transmit Bit Count Specification Register (MC0BIT)

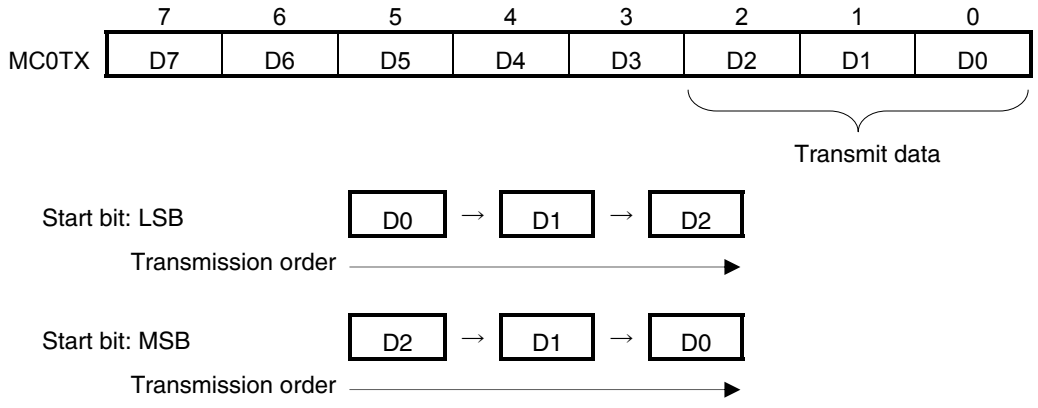
Address: FF65H After reset: 07H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MC0BIT	0	0	0	0	0	MC0BIT2	MC0BIT1	MC0BIT0

MC0BIT2	MC0BIT1	MC0BIT0	Transmit bit count setting
0	0	0	1 bit
0	0	1	2 bits
0	1	0	3 bits
0	1	1	4 bits
1	0	0	5 bits
1	0	1	6 bits
1	1	0	7 bits
1	1	1	8 bits

<R> **Remark** When the number of transmit bits is set as 7 bits or smaller, the lower bits are always transmitted regardless of MSB/LSB settings as the transmission start bit.

ex. When the number of transmit bits is set as 3 bits, and D7 to D0 are written to MCG transmit buffer register (MC0TX)



13.3 Registers Controlling Manchester Code Generator

The following six types of registers are used to control the Manchester code generator.

- MCG control register 0 (MC0CTL0)
- MCG control register 1 (MC0CTL1)
- MCG control register 2 (MC0CTL2)
- MCG status register (MC0STR)
- Port mode registers 0, 1 (PM0, PM1)
- Port registers 0, 1 (P0, P1)

(1) MCG control register 0 (MC0CTL0)

This register is used to set the operation mode and to enable/disable the operation.

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

$\overline{\text{RESET}}$ input sets this register to 10H.

Figure 13-4. Format of MCG Control Register 0 (MC0CTL0)

Address: FF60H After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

MC0DIR	First bit specification
0	MSB
1	LSB

MC0OSL	Data format
0	Manchester code
1	Bit sequential data

MC0OLV	Output level when transmission suspended
0	Low level
1	High level

Caution Clear (0) the MC0PWR bit before rewriting the MC0DIR, MC0OSL, and MC0OLV bits (it is possible to rewrite these bits by an 8-bit memory manipulation instruction at the same time when the MC0PWR bit is set (1)).

(2) MCG control register 1 (MC0CTL1)

This register is used to set the base clock of the Manchester code generator.

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

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

Figure 13-5. Format of MCG Control Register 1 (MC0CTL1)

Address: FF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	Base clock (f_{XCLK}) selection
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	$f_X/2^3$ (1.25 MHz)
1	0	0	$f_X/2^4$ (625 kHz)
1	0	1	$f_X/2^5$ (312.5 kHz)
1	1	0	
1	1	1	

Caution Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0CKS2 to MC0CKS0 bits.

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

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

(3) MCG control register 2 (MC0CTL2)

This register is used to set the transmit baud rate.

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

$\overline{\text{RESET}}$ input sets this register to 1FH.

Figure 13-6. Format of MCG Control Register 2 (MC0CTL2)

Address: FF62H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	Output clock selection of 5-bit counter
0	0	0	×	×	4	$f_{XCLK}/4$
0	0	1	0	0	4	$f_{XCLK}/4$
0	0	1	0	1	5	$f_{XCLK}/5$
0	0	1	1	0	6	$f_{XCLK}/6$
0	0	1	1	1	7	$f_{XCLK}/7$
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	0	1	29	$f_{XCLK}/29$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0BRS4 to MC0BRS0 bits.

2. The value from further dividing the output clock of the 5-bit counter by 2 is the baud rate value.

Remarks 1. f_{XCLK}: Frequency of the base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

2. k: Value set by the MC0BRS4 to MC0BRS0 bits (k = 4, 5, 6, 7, …., 31)

3. ×: Don't care

(4) MCG status register (MC0STR)

This register is used to indicate the operation status of the Manchester code generator.

This register can be read by a 1-bit or 8-bit memory manipulation instruction. Writing to this register is not possible.

$\overline{\text{RESET}}$ input or setting MC0PWR = 0 clears this register to 00H.

Figure 13-7. Format of MCG Status Register (MC0STR)

Address: FF63H After reset: 00H R

Symbol	<7>	6	5	4	3	2	1	0
MC0STR	MC0TSF	0	0	0	0	0	0	0

MC0TSF	Data transmission status
0	• $\overline{\text{RESET}}$ input • MC0PWR = 0 • If the next transfer data is not written to MC0TX when a transmission is completed
1	Transmission operation in progress

Caution This flag always indicates 1 during continuous transmission. Do not initialize a transmission operation without confirming that this flag has been cleared.

13.4 Operation of Manchester Code Generator

The Manchester code generator has the three modes described below.

- Operation stop mode
- Manchester code generator mode
- Bit sequential buffer mode

13.4.1 Operation stop mode

Transmissions are not performed in the operation stop mode. Therefore, the power consumption can be reduced. In addition, the P00/TI00/INTP0/MCGO and P13/TxD6/INTP1/(TOH1)/(MCGO) pins are used as an ordinary I/O port in this mode.

(1) Register description

MCG control register 0 (MC0CTL0) is used to set the operation stop mode.

To set the operation stop mode, clear bit 7 (MC0PWR) of MC0CTL0 to 0.

(a) MCG control register 0 (MC0CTL0)

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

$\overline{\text{RESET}}$ input sets this register to 10H.

Address: FF60H After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

13.4.2 Manchester code generator mode

This mode is used to transmit data in Manchester code format using the MCGO pin.

(1) Register description

MCG control register 0 (MC0CTL0), MCG control register 1 (MC0CTL1), and MCG control register 2 (MC0CTL2) are used to set the Manchester code generator mode.

(a) MCG control register 0 (MC0CTL0)

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

$\overline{\text{RESET}}$ input sets this register to 10H.

Address: FF60H After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

MC0DIR	First bit specification
0	MSB
1	LSB

MC0OSL	Data format
0	Manchester code
1	Bit sequential data

MC0OLV	Output level when transmission suspended
0	Low level
1	High level

Caution Clear (0) the MC0PWR bit before rewriting the MC0DIR, MC0OSL, and MC0OLV bits (it is possible to rewrite these bits by an 8-bit memory manipulation instruction at the same time when the MC0PWR bit is set (1)).

(b) MCG control register 1 (MC0CTL1)

This register is used to set the base clock of the Manchester code generator.

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

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

Address: FF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	Base clock (f_{XCLK}) selection
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	$f_X/2^3$ (1.25 MHz)
1	0	0	$f_X/2^4$ (625 kHz)
1	0	1	$f_X/2^5$ (312.5 kHz)
1	1	0	
1	1	1	

Caution Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0CKS2 to MC0CKS0 bits.

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

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

(c) MCG control register 2 (MC0CTL2)

This register is used to set the transmit baud rate.

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

$\overline{\text{RESET}}$ input sets this register to 1FH.

Address: FF62H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	Output clock selection of 5-bit counter
0	0	0	×	×	4	$f_{XCLK}/4$
0	0	1	0	0	4	$f_{XCLK}/4$
0	0	1	0	1	5	$f_{XCLK}/5$
0	0	1	1	0	6	$f_{XCLK}/6$
0	0	1	1	1	7	$f_{XCLK}/7$
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	0	1	29	$f_{XCLK}/29$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0BRS4 to MC0BRS0 bits.

2. The value from further dividing the output clock of the 5-bit counter by 2 is the baud rate value.

Remarks 1. f_{XCLK}: Frequency of the base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

2. k: Value set by the MC0BRS4 to MC0BRS0 bits (k = 4, 5, 6, 7, …., 31)

3. ×: Don't care

<1> Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\dfrac{f_{XCLK}}{2 \times k}$ [bps]

f_{XCLK}: Frequency of base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

k: Value set by the MC0BRS4 to MC0BRS0 bits of the MC0CTL2 register (k = 4, 5, 6, ..., 31)

<2> Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$

Caution Keep the baud rate error during transmission to within the permissible error range at the reception destination.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz
Set value of MC0BRS4 to MC0BRS0 bits of MC0CTL2 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)
= 2,500,000/(2 × 16) = 78125 [bps]

Error = (78,125/76,800 − 1) × 100
= 1.725 [%]

<3> Example of setting baud rate

Baud Rate [bps]	f_X = 10.0 MHz				f_X = 8.38 MHz				f_X = 8.0 MHz				f_X = 6.0 MHz			
	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]
4800	–	–	–	–	5, 6, or 7	27	4850	1.03	5, 6, or 7	26	4808	0.16	5, 6, or 7	20	4688	–2.34
9600	5, 6, or 7	16	9766	1.73	4	27	9699	1.03	5, 6, or 7	13	9615	0.16	4	20	9375	–2.34
19200	5	8	19531	1.73	3	27	19398	1.03	4	13	19231	0.16	4	10	18750	–2.34
31250	4	10	31250	0	2	17	30809	–1.41	4	8	31250	0	2	24	31250	0
38400	4	8	39063	1.73	2	27	38796	1.03	3	13	38462	0.16	2	20	37500	–2.34
56000	3	11	56818	1.46	2	19	55132	–1.55	3	9	55556	–0.79	1	27	55556	–0.79
62500	2	20	62500	0	2	17	61618	–1.41	3	8	62500	0	2	12	62500	0
76800	2	16	78125	1.73	1	27	77592	1.03	2	13	76923	0.16	2	10	75000	–2.34
115200	1	22	113636	–1.36	2	9	116389	1.03	1	17	117647	2.12	1	13	115385	0.16
125000	1	20	125000	0	1	17	123235	–1.41	1	16	125000	0	1	12	125000	0
153600	1	16	156250	1.73	2	7	149643	–2.58	1	13	153846	0.16	1	10	150000	–2.34
250000	1	10	250000	0	1	8	261875	4.75	1	8	250000	0	1	6	250000	0
					0	17	246471	–1.41								

Remark MC0CKS2 to MC0CKS0: Bits 2 to 0 of MCG control register 1 (MC0CTL1) (setting of base clock (f_{XCLK}))
k: Value set by bits 4 to 0 (MC0BRS4 to MC0BRS0) of MCG control register 2 (MC0CTL2) (k = 4, 5, 6, …, 31)
fx: High-speed system clock oscillation frequency
ERR: Baud rate error

(d) Alternate-function pin switch register (PSEL)

This register is used to select the MCGO pin.

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

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

Address: FF70H After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	2	<1>	<0>
PSEL	0	0	TOH1SL	MCGSL	0	0	INTP1SL	INTP3SL

MCGSL	MCGO pin selection
0	P00/TI000/INTP0/MCGO
1	P13/TxD6/INTP1/(TOH1)/(MCGO)

Caution Clear bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) to 0 before rewriting the MCGSL bit.

(e) Port mode registers 0, 1 (PM0, PM1)

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

When using the P00/TI000/INTP0/MCGO and P13/TxD6/INTP1/(TOH1)/(MCGO) pins for Manchester code output, clear PM00 and PM13 to 0 and clear the output latches of P00 and P13 to 0.

PM0 and PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Address: FF20H After reset: FFH R/W

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

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

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Port settings

(a) When P00/TI000/INTP0/MCGO is set as Manchester code output

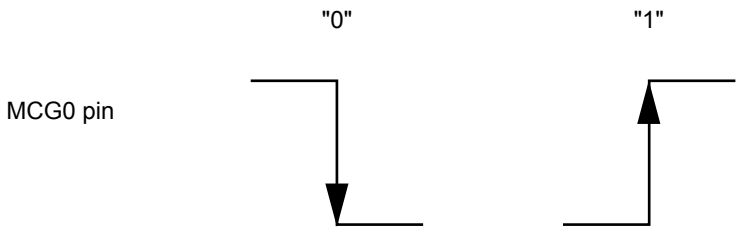
Bit 0 of port mode register 0 (PM00): Cleared to 0
Bit 0 of port 0 (P00): Cleared to 0

(b) When P13/TxD6/INTP1/(TOH1)/(MCGO) is set as Manchester code output

Bit 3 (PM13) of port mode register 1: Cleared to 0
Bit 3 (P13) of port 1: Cleared to 0

<R> **(3) Format of "0" and "1" of Manchester code output**

The format of "0" and "1" of Manchester code output in μPD780862 Subseries is as follows.



(3) Transmit operation

In Manchester code generator mode, data is transmitted in 1- to 8-bit units. Data bits are transmitted in Manchester code format. Transmission is enabled if bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) is set to 1.

The output value while a transmission is suspended can be set by using bit 0 (MC0OLV) of the MC0CTL0 register.

A transmission starts by writing a value to the MCG transmit buffer register (MC0TX) after setting the transmit data bit length to the MCG transmit bit count specification register (MC0BIT). At the transmission start timing, the MC0BIT value is transferred to the 3-bit counter and the data of MC0TX is transferred to the 8-bit shift register. An interrupt request signal (INTMCG) occurs at the timing that the MC0TX value is transferred to the 8-bit shift register. The 8-bit shift register is continuously shifted by the baud rate clock, and signal that is XORed with the baud rate clock is output from the MCGO pin.

When continuous transmission is executed, the next data is set to MC0BIT and MC0TX during data transmission after INTMCG occurs.

To transmit continuously, writing the next transfer data to MC0TX must be complete within the period (3) and (4) in Figure 13-8. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

Figure 13-8. Timing of Manchester Code Generator Mode (LSB First) (1/4)

(1) Transmit timing (MC0OLV = 1, total transmit bit length = 8 bits)

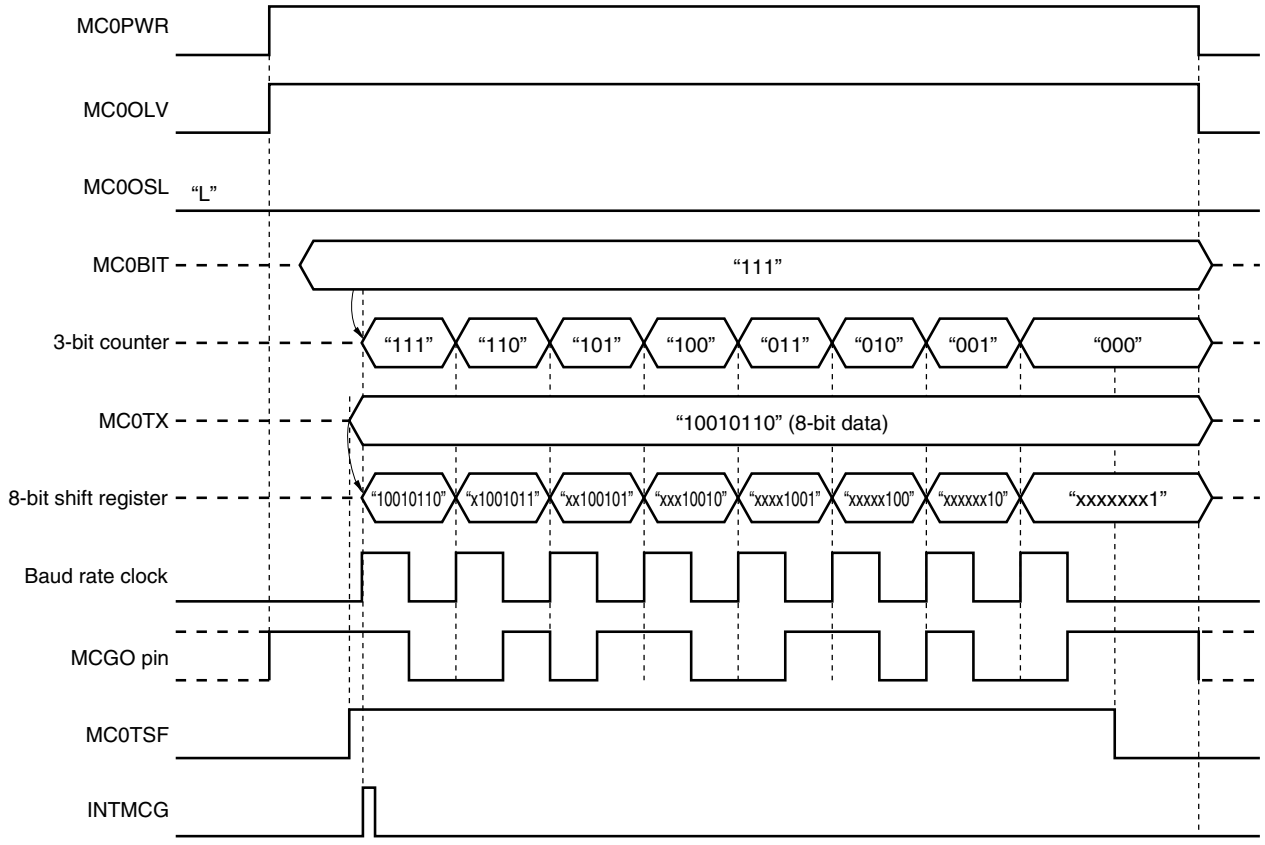


Figure 13-8. Timing of Manchester Code Generator Mode (LSB First) (2/4)

(2) Transmit timing (MC0OLV = 0, total transmit bit length = 8 bits)

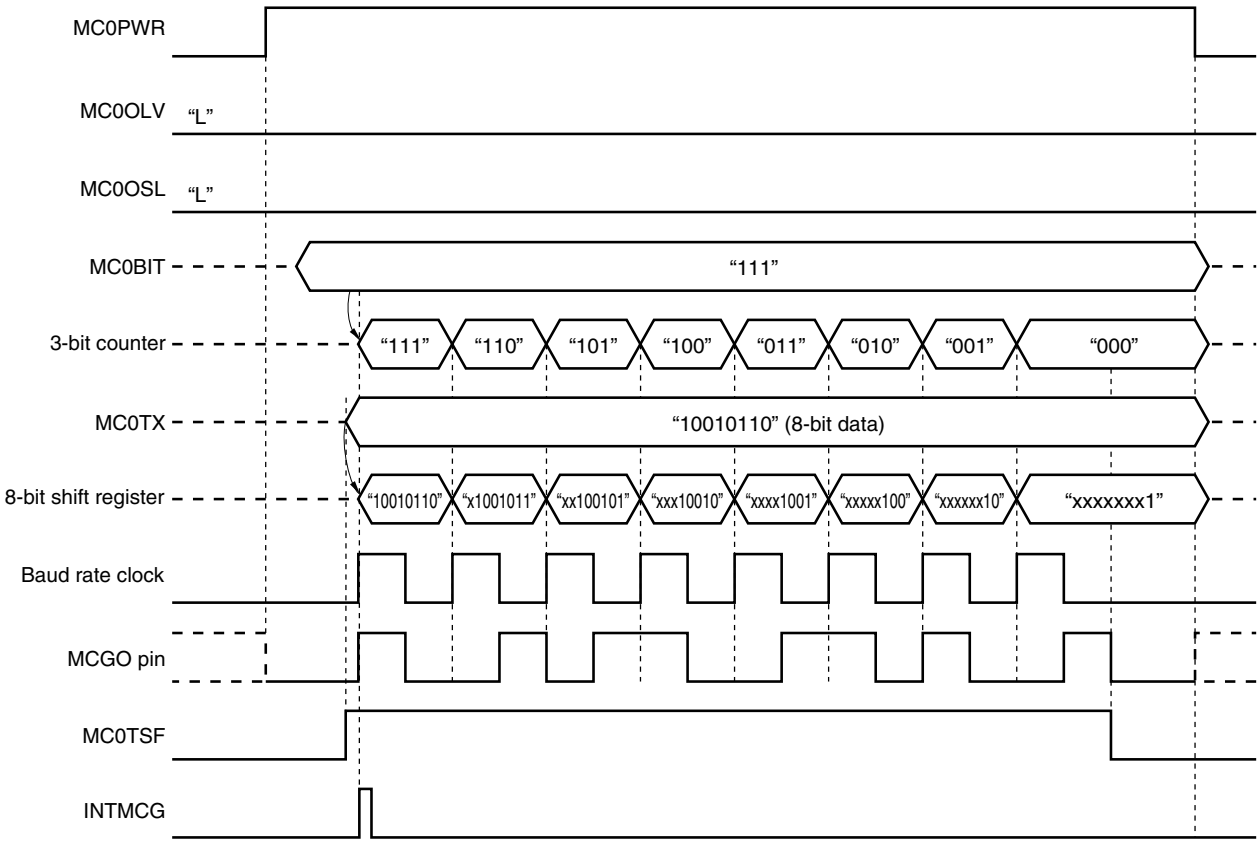
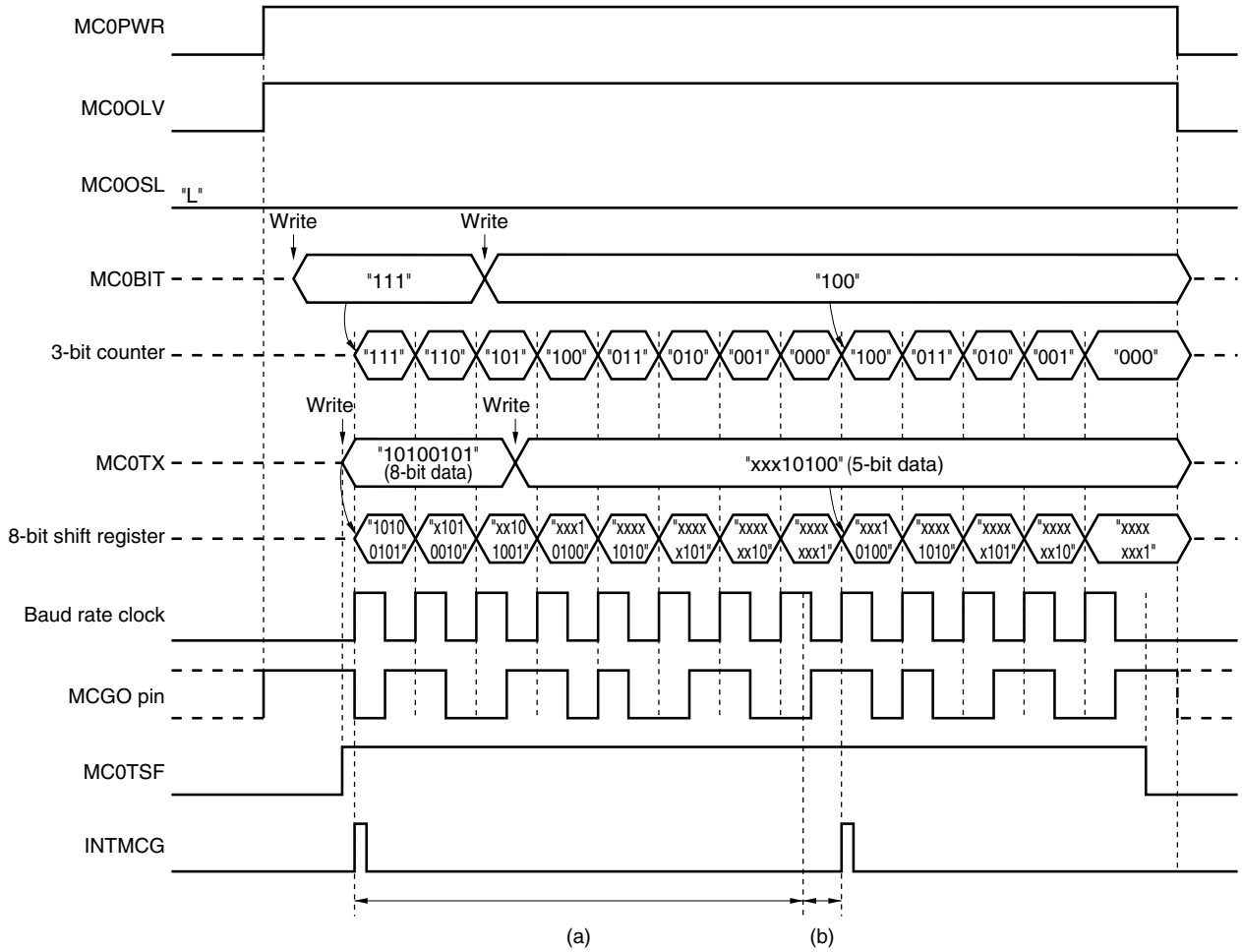


Figure 13-8. Timing of Manchester Code Generator Mode (LSB First) (3/4)

(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

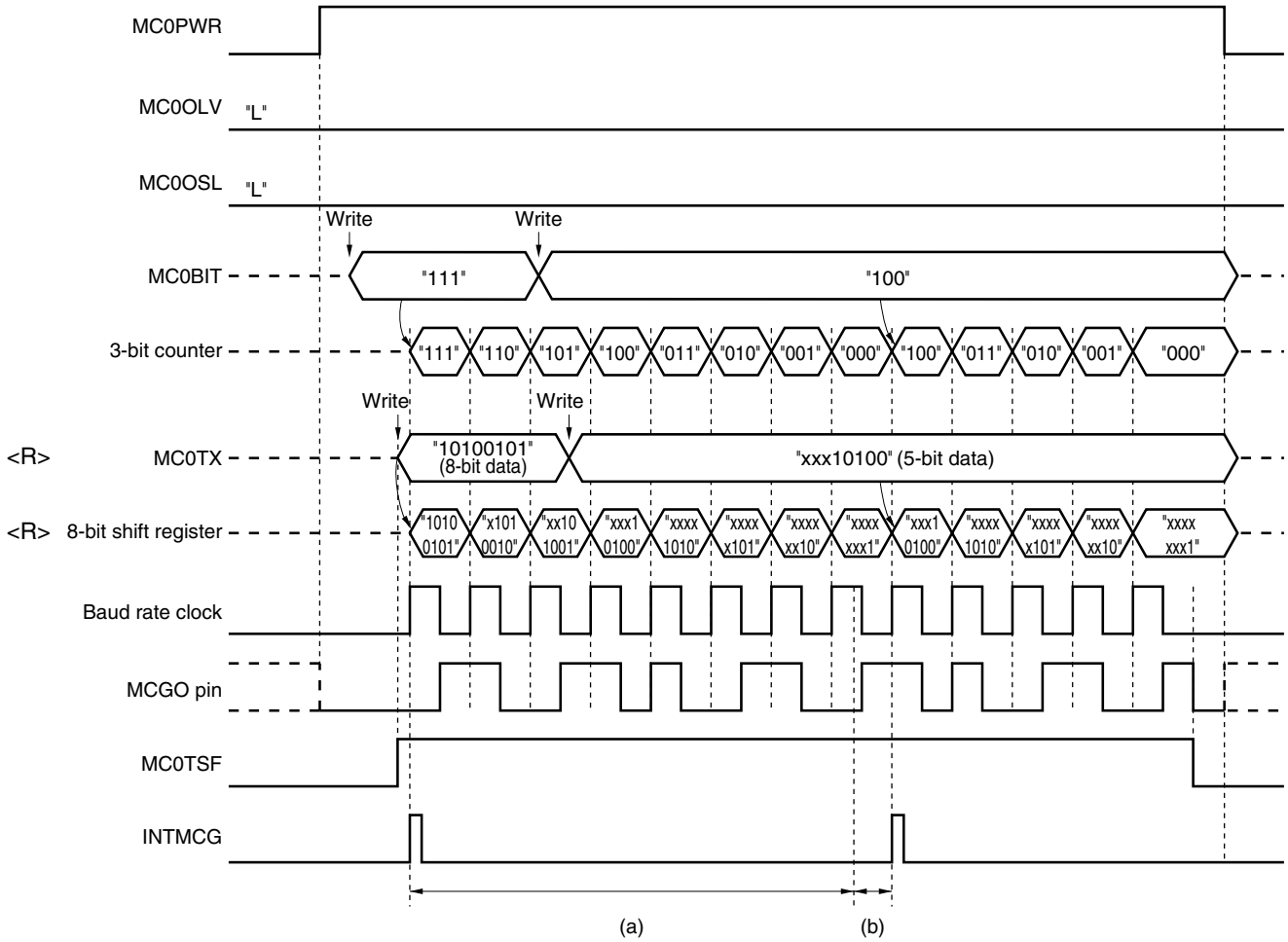
f_{XCLK}: Frequency of the operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

Figure 13-8. Timing of Manchester Code Generator Mode (LSB First) (4/4)

(4) Transmit timing (MC0OLV = 0, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

f_{XCLK}: Frequency of the operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

13.4.3 Bit sequential buffer mode

The bit sequential buffer mode is used to output sequential signals using the MCGO pin.

(1) Register description

The MCG control register 0 (MC0CTL0), MCG control register 1 (MC0CTL1), and MCG control register 2 (MC0CTL2) are used to set the bit sequential buffer mode.

(a) MCG control register 0 (MC0CTL0)

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

$\overline{\text{RESET}}$ input sets this register to 10H.

Address: FF60H After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

MC0DIR	First bit specification
0	MSB
1	LSB

MC0OSL	Data format
0	Manchester code
1	Bit sequential data

MC0OLV	Output level when transmission suspended
0	Low level
1	High level

Caution Clear (0) the MC0PWR bit before rewriting the MC0DIR, MC0OSL, and MC0OLV bits (it is possible to rewrite these bits by an 8-bit memory manipulation instruction at the same time when the MC0PWR bit is set (1)).

(b) MCG control register 1 (MC0CTL1)

This register is used to set the base clock of the Manchester code generator.

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

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

Address: FF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	Base clock (f_{XCLK}) selection
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	$f_X/2^3$ (1.25 MHz)
1	0	0	$f_X/2^4$ (625 kHz)
1	0	1	$f_X/2^5$ (312.5 kHz)
1	1	0	
1	1	1	

Caution Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0CKS2 to MC0CKS0 bits.

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

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

(c) MCG control register 2 (MC0CTL2)

This register is used to set the transmit baud rate.

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

$\overline{\text{RESET}}$ input sets this register to 1FH.

Address: FF62H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	Output clock selection of 5-bit counter
0	0	0	×	×	4	$f_{XCLK}/4$
0	0	1	0	0	4	$f_{XCLK}/4$
0	0	1	0	1	5	$f_{XCLK}/5$
0	0	1	1	0	6	$f_{XCLK}/6$
0	0	1	1	1	7	$f_{XCLK}/7$
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	0	1	29	$f_{XCLK}/29$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0BRS4 to MC0BRS0 bits.

2. The value from further dividing the output clock of the 5-bit counter by 2 is the baud rate value.

Remarks 1. f_{XCLK}: Frequency of the base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

2. k: Value set by the MC0BRS4 to MC0BRS0 bits (k = 4, 5, 6, 7, …., 31)

3. ×: Don't care

<1> Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\dfrac{f_{XCLK}}{2 \times k}$ [bps]

f_{XCLK}: Frequency of base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

k: Value set by the MC0BRS4 to MC0BRS0 bits of the MC0CTL2 register (k = 4, 5, 6, ..., 31)

<2> Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$

Caution Keep the baud rate error during transmission to within the permissible error range at the reception destination.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz
Set value of MC0BRS4 to MC0BRS0 bits of MC0CTL2 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)
= 2,500,000/(2 × 16) = 78125 [bps]

Error = (78,125/76,800 − 1) × 100
= 1.725 [%]

<3> Example of setting baud rate

Baud Rate [bps]	f_X = 10.0 MHz				f_X = 8.38 MHz				f_X = 8.0 MHz				f_X = 6.0 MHz			
	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]
4800	–	–	–	–	5, 6, or 7	27	4850	1.03	5, 6, or 7	26	4808	0.16	5, 6, or 7	20	4688	–2.34
9600	5, 6, or 7	16	9766	1.73	4	27	9699	1.03	5, 6, or 7	13	9615	0.16	4	20	9375	–2.34
19200	5	8	19531	1.73	3	27	19398	1.03	4	13	19231	0.16	4	10	18750	–2.34
31250	4	10	31250	0	2	17	30809	–1.41	4	8	31250	0	2	24	31250	0
38400	4	8	39063	1.73	2	27	38796	1.03	3	13	38462	0.16	2	20	37500	–2.34
56000	3	11	56818	1.46	2	19	55132	–1.55	3	9	55556	–0.79	1	27	55556	–0.79
62500	2	20	62500	0	2	17	61618	–1.41	3	8	62500	0	2	12	62500	0
76800	2	16	78125	1.73	1	27	77592	1.03	2	13	76923	0.16	2	10	75000	–2.34
115200	1	22	113636	–1.36	2	9	116389	1.03	1	17	117647	2.12	1	13	115385	0.16
125000	1	20	125000	0	1	17	123235	–1.41	1	16	125000	0	1	12	125000	0
153600	1	16	156250	1.73	2	7	149643	–2.58	1	13	153846	0.16	1	10	150000	–2.34
250000	1	10	250000	0	1	8	261875	4.75	1	8	250000	0	1	6	250000	0
					0	17	246471	–1.41								

Remark MC0CKS2 to MC0CKS0: Bits 2 to 0 of MCG control register 1 (MC0CTL1) (setting of base clock (f_{XCLK}))
k: Value set by bits 4 to 0 (MC0BRS4 to MC0BRS0) of MCG control register 2 (MC0CTL2) (k = 4, 5, 6, …, 31)
fx: High-speed system clock oscillation frequency
ERR: Baud rate error

(d) Alternate-function pin switch register (PSEL)

This register is used to select the MCGO pin.

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

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

Address: FF70H After reset: 00H R/W

<Symbol	7	6	<5>	<4>	3	2	<1>	<0>
PSEL	0	0	TOH1SL	MCGSL	0	0	INTP1SL	INTP3SL

MCGSL	MCGO pin selection
0	P00/TI000/INTP0/MCGO
1	P13/TxD6/INTP1/(TOH1)/(MCGO)

Caution Clear bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) to 0 before rewriting the MCGSL bit.

(e) Port mode registers 0, 1 (PM0, PM1)

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

When using the P00/TI000/INTP0/MCGO and P13/TxD6/INTP1/(TOH1)/(MCGO) pins for bit sequential data output, clear PM00 and PM13 to 0 and clear the output latches of P00 and P13 to 0.

PM0 and PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Address: FF20H After reset: FFH R/W

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

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

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Port settings

(a) When P00/TI000/INTP0/MCGO is set as bit sequential data output

Bit 0 of port mode register 0 (PM00): Cleared to 0

Bit 0 of port 0 (P00): Cleared to 0

(b) When P13/TxD6/INTP1/(TOH1)/(MCGO) is set as bit sequential data output

Bit 3 (PM13) of port mode register 1: Cleared to 0

Bit 3 (P13) of port 1: Cleared to 0

(3) Transmit operation

In bit sequential buffer mode, data is transmitted in 1- to 8-bit units. Transmission is enabled if bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) is set to 1.

The output value while transmission is suspended can be set by using bit 0 (MC0OLV) of the MC0CTL0 register.

A transmission starts by writing a value to the MCG transmit buffer register (MC0TX) after setting the transmit data bit length to the MCG transmit bit count specification register (MC0BIT). At the transmission start timing, the MC0BIT value is transferred to the 3-bit counter and data of MC0TX is transferred to the 8-bit shift register. An interrupt request signal (INTMCG) occurs at the timing that the MC0TX value is transferred to the 8-bit shift register. The 8-bit shift register is continuously shifted by the baud rate clock and is output from the MCGO pin.

When continuous transmission is executed, the next data is set to MC0BIT and MC0TX during data transmission after INTMCG occurs.

To transmit continuously, writing the next transfer data to MC0TX must be complete within the period (3) and (4) in Figure 13-9. Rewrite MC0BIT before writing to MC0TX during continuous transmission.

Figure 13-9. Timing of Bit Sequential Buffer Mode (LSB First) (1/4)

(1) Transmit timing (MC0OLV = 1, total transmit bit length = 8 bits)

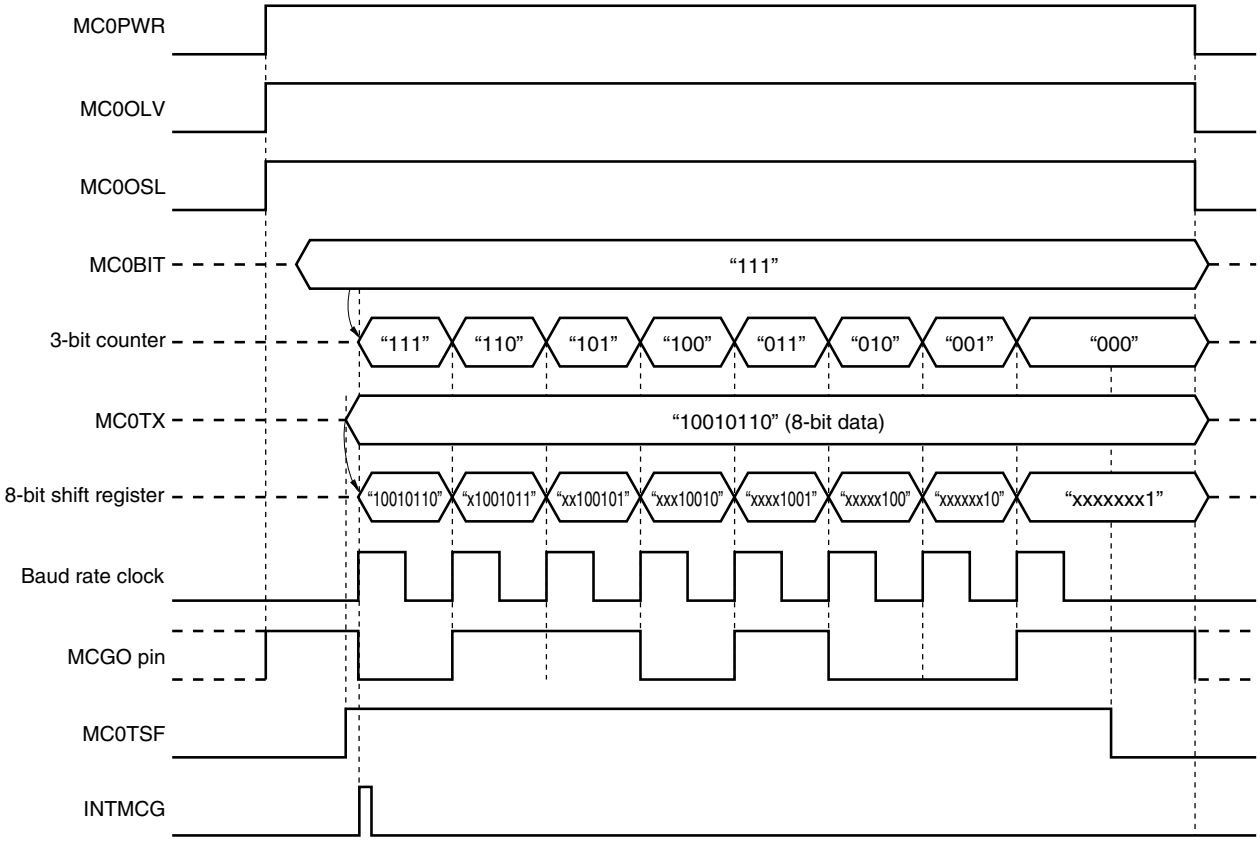


Figure 13-9. Timing of Bit Sequential Buffer Mode (LSB First) (2/4)

(2) Transmit timing (MC0OLV = 0, total transmit bit length = 8 bits)

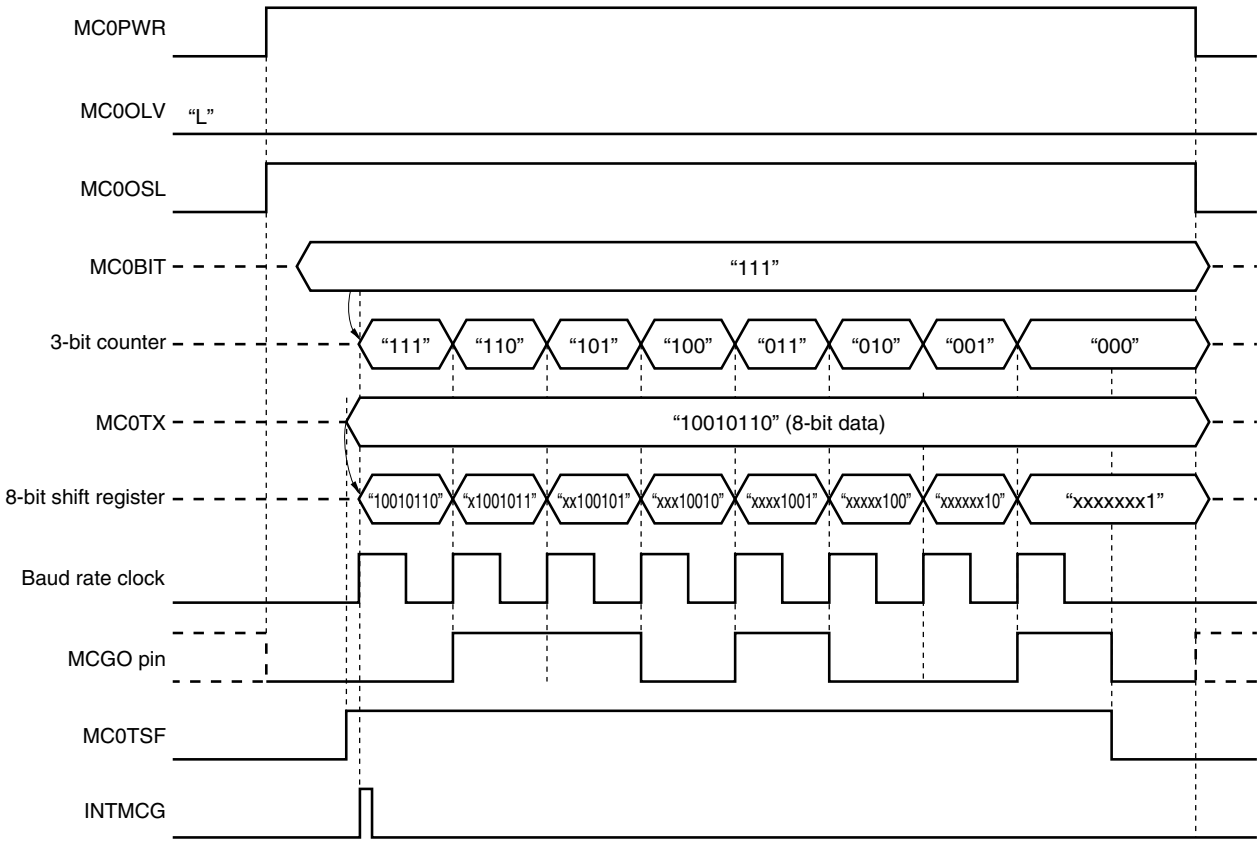
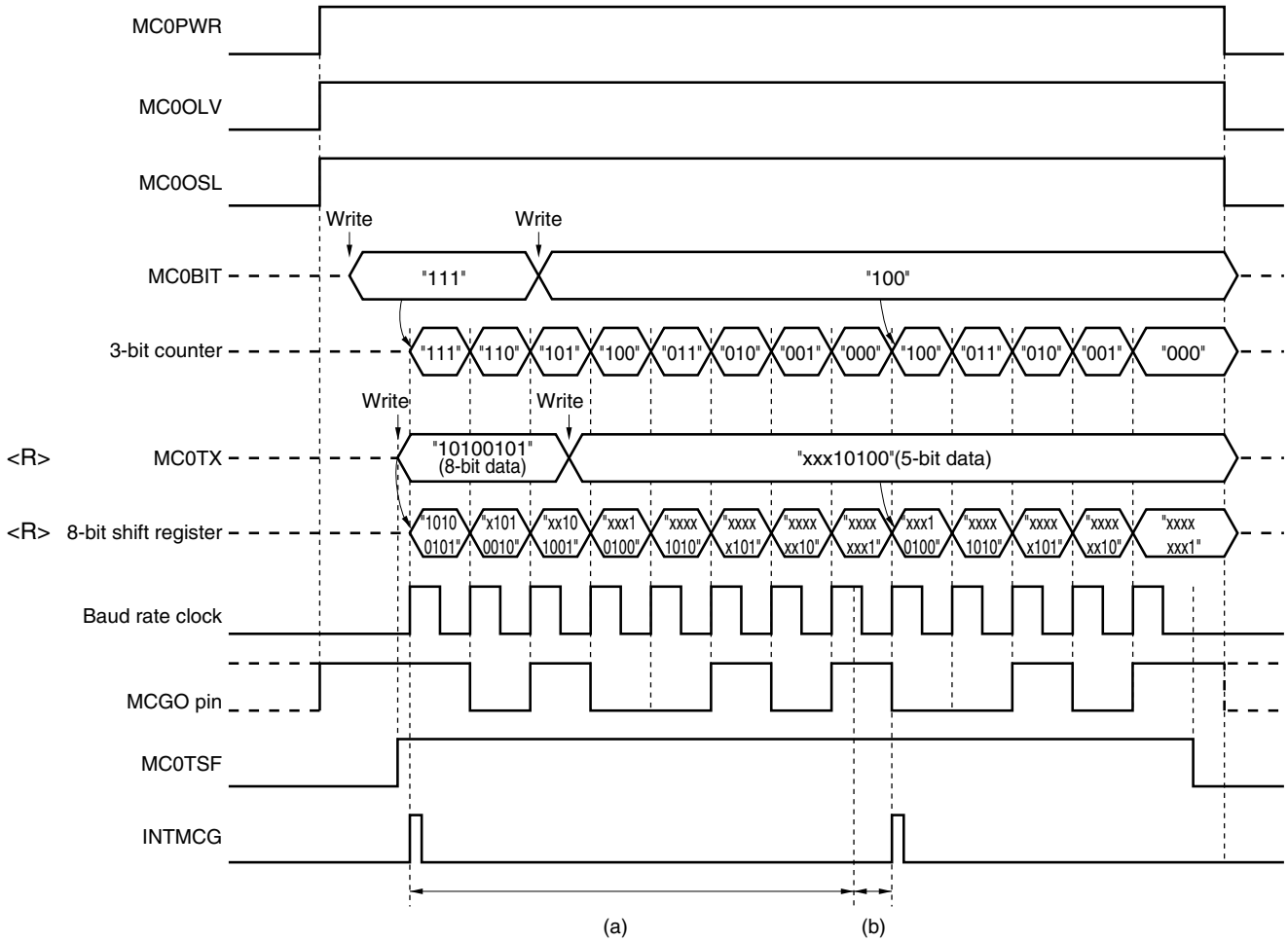


Figure 13-9. Timing of Bit Sequential Buffer Mode (LSB First) (3/4)

(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

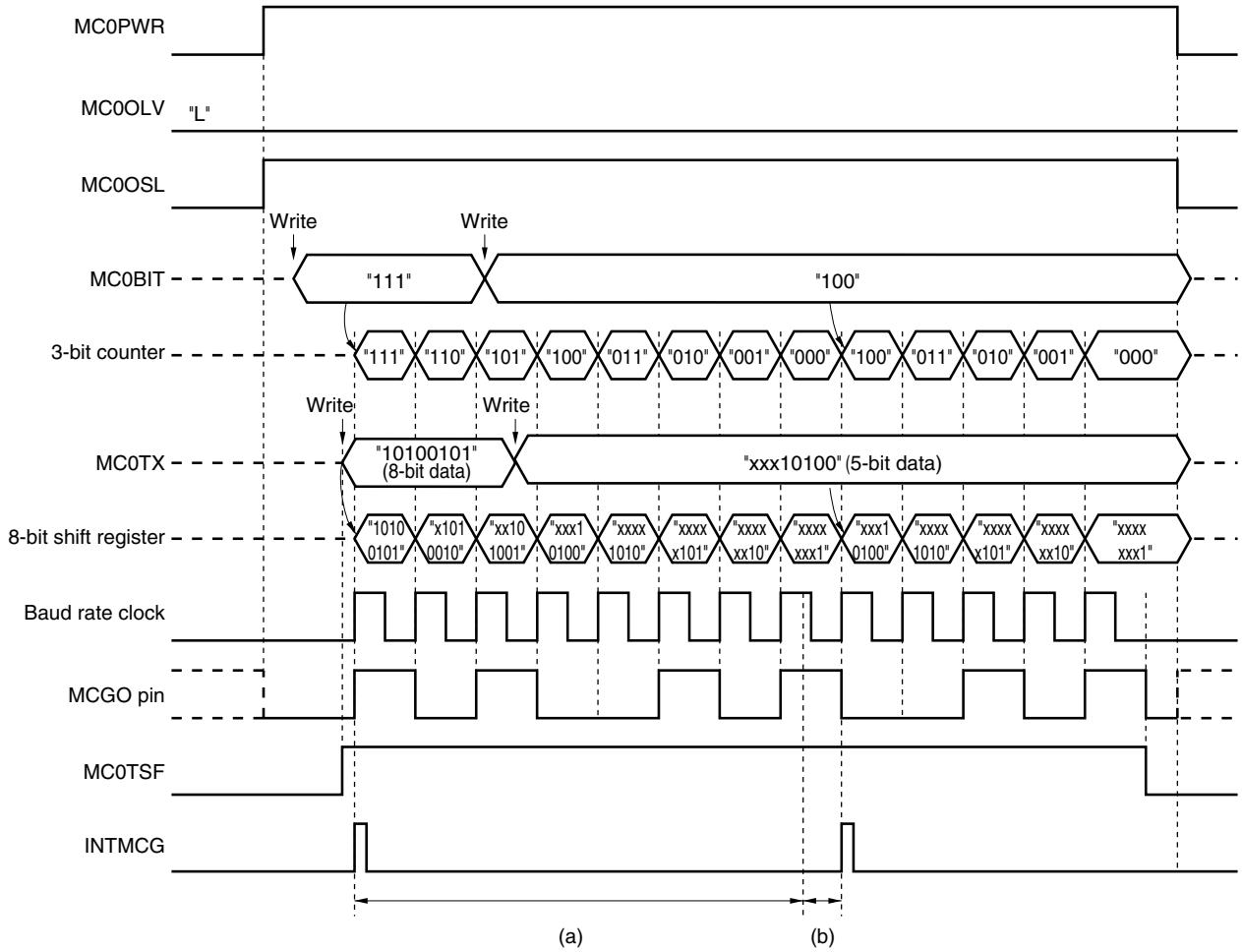
f_{XCLK}: Frequency of operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

Figure 13-9. Timing of Bit Sequential Buffer Mode (LSB First) (4/4)

(4) Transmit timing (MC0OLV = 0, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

f_{XCLK}: Frequency of operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

CHAPTER 14 INTERRUPT FUNCTIONS

14.1 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 of high-priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt is serviced according to its predetermined priority (see **Table 14-1**).

A standby release signal is generated and the STOP and HALT modes are released.

Four external interrupt requests and 12 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.

14.2 Interrupt Sources and Configuration

A total of 17 interrupt sources exist for maskable and software interrupts. In addition, maximum total of 5 reset sources are also provided (see **Table 14-1**).

Table 14-1. Interrupt Source List

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	INTMCG	End of Manchester code transmission	Internal	000EH	(A)
	6	INTSRE6	UART6 reception error generation		0012H	
	7	INTSR6	End of UART6 reception		0014H	
	8	INTST6	End of UART6 transmission		0016H	
	9	INTCSI10	End of CSI10 communication		0018H	
	10	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)		001AH	
	11	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		001CH	
	12	INTTM50	Match between TM50 and CR50 (when compare register is specified)		001EH	
<R>	13	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		0020H	
<R>	14	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	
	15	INTAD	End of A/D conversion		0024H	
Software	–	BRK	BRK instruction execution	–	003EH	(C)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on-clear			
		LVI	Low-voltage detection[Note 4]			
		Clock monitor	High-speed system clock 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 15 is the lowest.

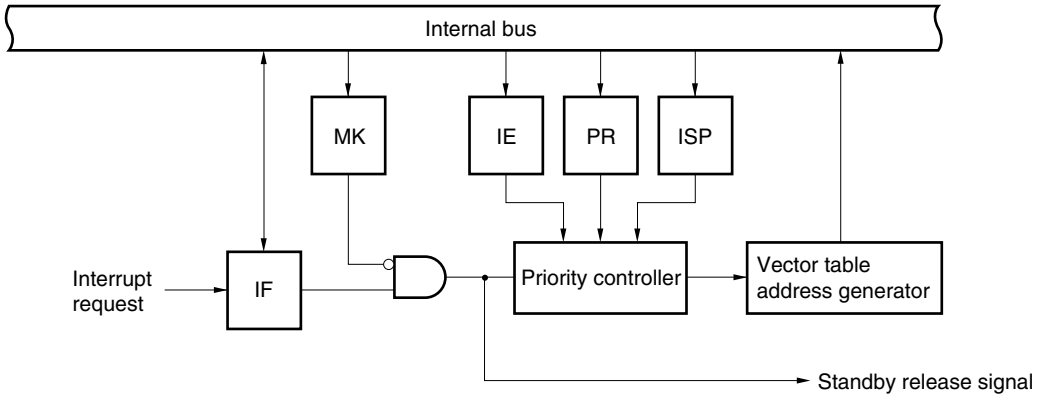
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 14-1.

3. When bit 1 (LVIMD) = 0 is selected for the low-voltage detection register (LVIM).

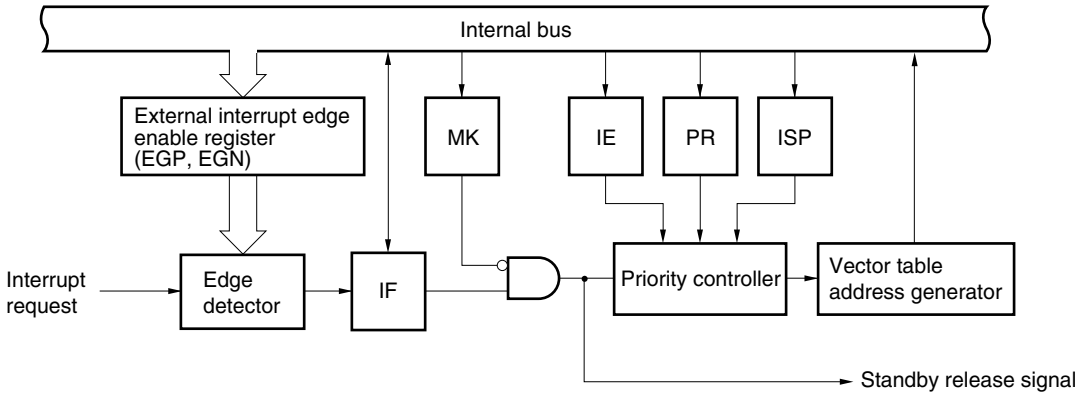
4. When LVIMD = 1 is selected.

Figure 14-1. Basic Configuration of Interrupt Function

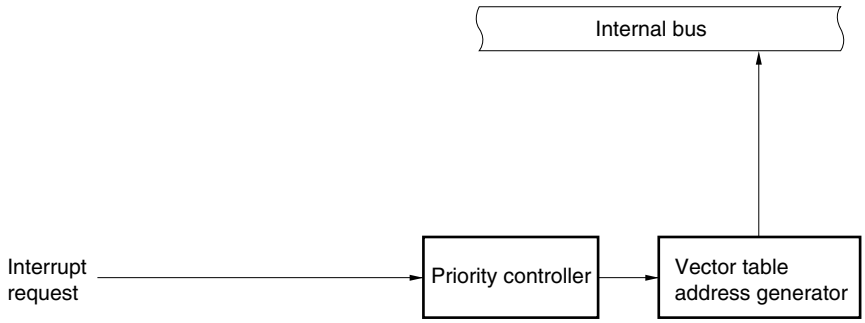
(A) Internal maskable interrupt



(B) External maskable interrupt (INTP0 to INTP3)



(C) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

14.3 Registers Controlling Interrupt Function

The following 8 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)
- Input switch control register (ISC)
- Alternate-function pin switch register (PSEL)

The following registers are used to select the INTP0, INTP1, and INTP3 pins, which are used for external interrupt requests.

- Input switch control register (ISC): INTP0
- Alternate-function pin switch register (PSEL): INTP1, INTP3

Table 14-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 14-2. 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	
INTMCG	MCGIF		MCGMK		MCGPR	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
INTST6	STIF6		STMK6		STPR6	
INTCSI10	CSIIF10		CSIMK10		CSIPR10	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD	ADIF	1F1L	ADMK	MK1L	ADPR	PR1L

(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 are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are set by a 16-bit memory manipulation instruction.

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

Figure 14-2. Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	0	MCGIF	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	CSIIF10	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF1L	0	0	0	0	0	0	0	ADIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Be sure to set bit 6 of IF0L and bits 1 to 7 of IF1L to 0.

2. 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.

3. 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 are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form 16-bit register MK0, they are set by a 16-bit memory manipulation instruction.

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

Figure 14-3. Format of Interrupt Mask Flag Register (MK0L, MK0H, MK1L)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	1	MCGMK	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	CSIMK10	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
MK1L	1	1	1	1	1	1	1	ADMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bit 6 of MK0L and bits 1 to 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 are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set by a 16-bit memory manipulation instruction.

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

Figure 14-4. Format of Priority Specification Flag Register (PR0L, PR0H, PR1L)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	1	MCGPR	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	CSIPR10	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR1L	1	1	1	1	1	1	1	ADPR

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bit 6 of PR0L and bits 1 to 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 INTP3.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 14-5. 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	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 3)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 14-3 shows the ports corresponding to EGPn and EGNn.

Table 14-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port		Interrupt Request Signal
EGP0	EGN0	P00 (ISC0 = 0)	P14 (ISC0 = 1)	INTP0
EGP1	EGN1	P13 (INTP1SL = 0)	P10 (INTP1SL = 1)	INTP1
EGP2	EGN2	P01		INTP2
EGP3	EGN3	P11 (INTP3SL = 0)	P12 (INTP3SL = 1)	INTP3

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 3

ISC0: Bit 0 of input switch control register (ISC)

INTP1SL: Bit 1 of alternate-function pin switch register (PSEL)

INTP3SL: Bit 0 of alternate-function pin switch register (PSEL)

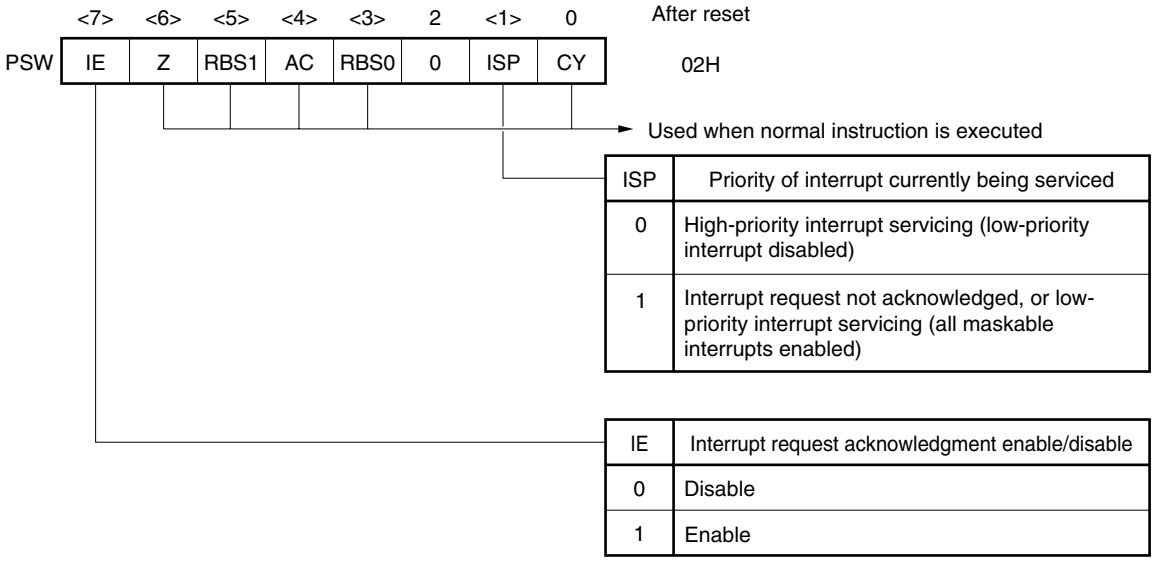
(5) Program status word (PSW)

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 14-6. Format of Program Status Word



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	Disable
1	Enable

(6) Input switch control register (ISC)

The input source is switched by setting ISC.

- When P00/INTP0/TI000/MCGO is used as an external interrupt input pin
  ISC0 = 0
- When P14/<INTP0>/RxD6 is used as an external interrupt input pin
  ISC0 = 1

This register can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 14-7. 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

ISC0	INTP0 input source selection
0	INTP0 (P00)
1	RxD6 (P14)

Remark When using the P00/INTP0/TI000/MCGO and P14/<INTP0>/RxD6 pins as external interrupt request inputs, set PM00 and PM14 to 1.

(7) Alternate-function pin switch register (PSEL)

This register is used to select the INTP1 and INTP3 pins.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 14-8. Format of Alternate-Function Pin Switch Register (PSEL)

Address: FF70H After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	2	<1>	<0>
PSEL	0	0	TOH1SL	MCGSL	0	0	INTP1SL	INTP3SL

INTP1SL	INTP1 pin selection
0	P13/TxD6/INTP1/(TOH1)/(MCGO)
1	P10/$\overline{\text{SCK10}}$/(INTP1)

INTP3SL	INTP3 pin selection
0	P11/SI10/INTP3
1	P12/SO10/TOH1/(INTP3)

Remark When using the P10/$\overline{\text{SCK10}}$/(INTP1), P11/SI10/INTP3, P12/TOH1/SO10/(INTP3), and P13/(TOH1)/TxD6/INTP1/(MCGO) pins as external interrupt request inputs, set PM10 to PM13 to 1.

14.4 Interrupt Servicing Operations

14.4.1 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 14-4 below.

For the interrupt request acknowledgment timing, see **Figures 14-10** and **14-11**.

Table 14-4. 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 interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

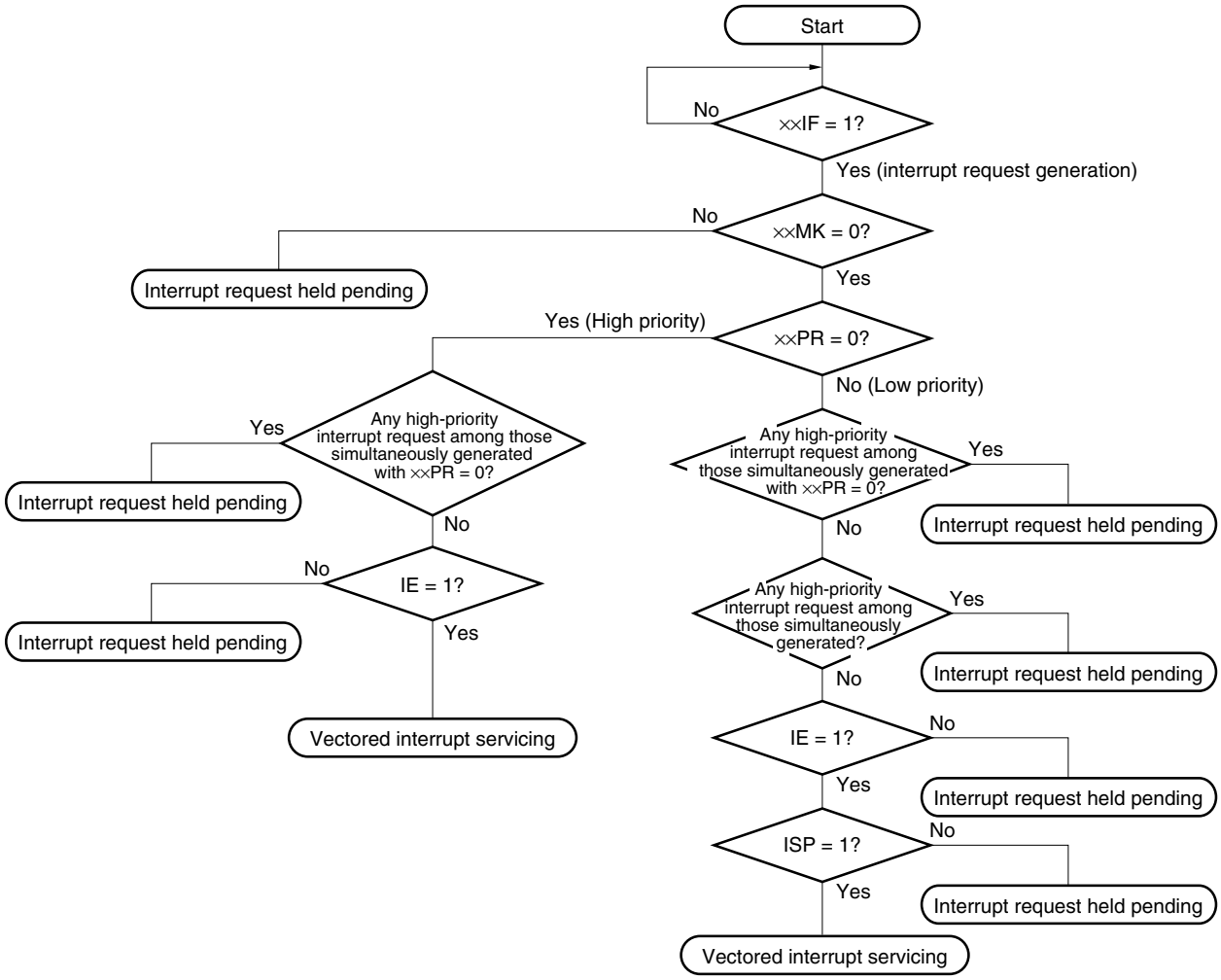
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 14-9 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 14-9. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag

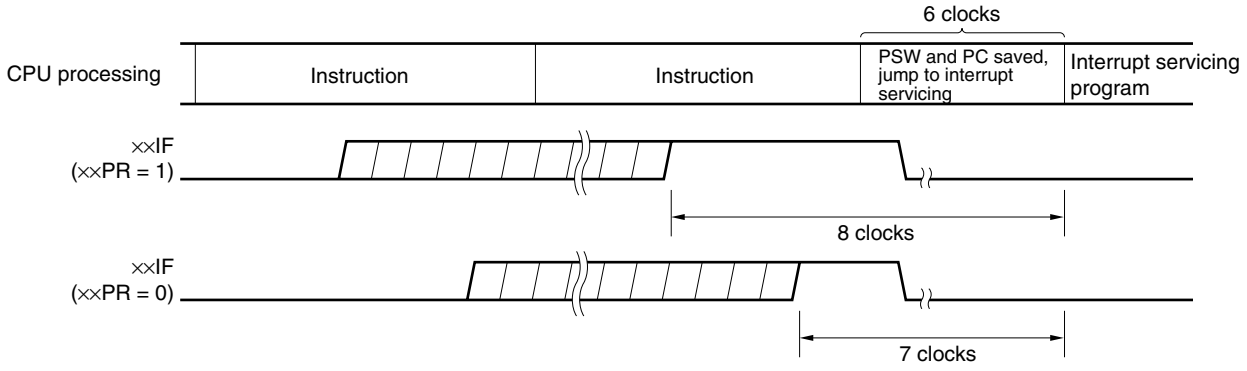
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)

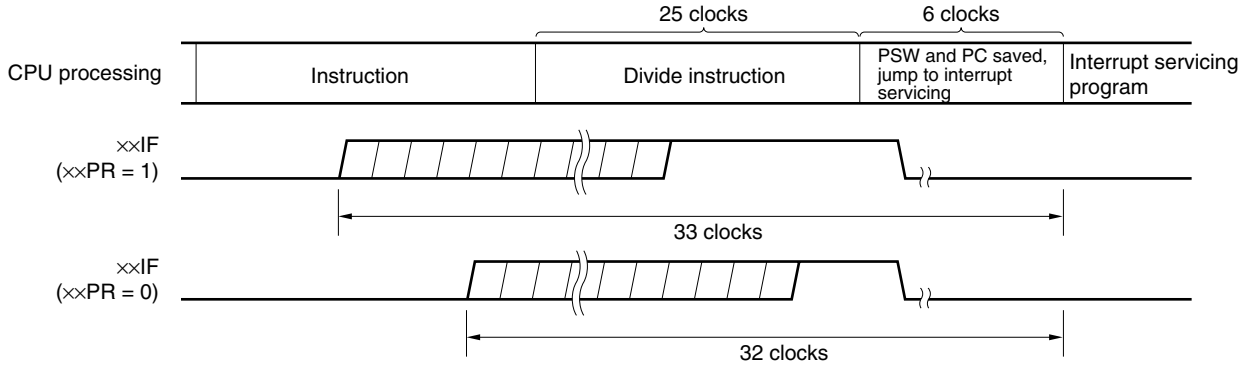
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = High-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 14-10. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 14-11. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

14.4.2 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.

14.4.3 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 14-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 14-12 shows multiple interrupt servicing examples.

Table 14-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request				Software Interrupt Request
		PR = 0		PR = 1		
		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. The 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.

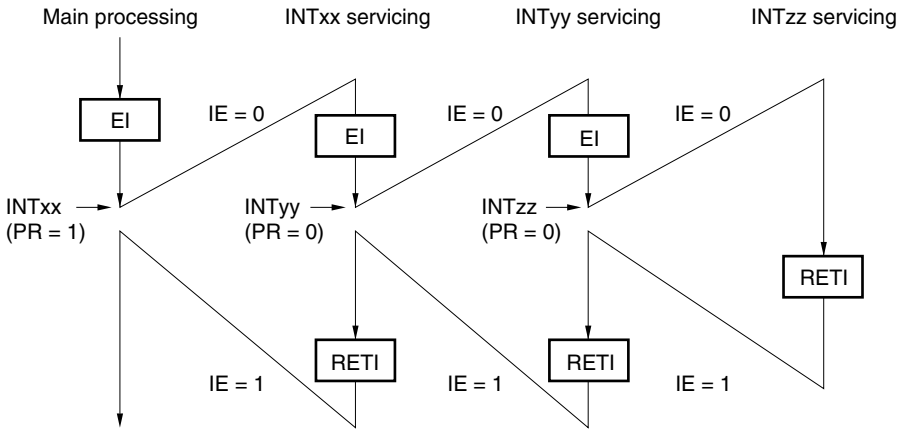
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level

PR = 1: Lower priority level

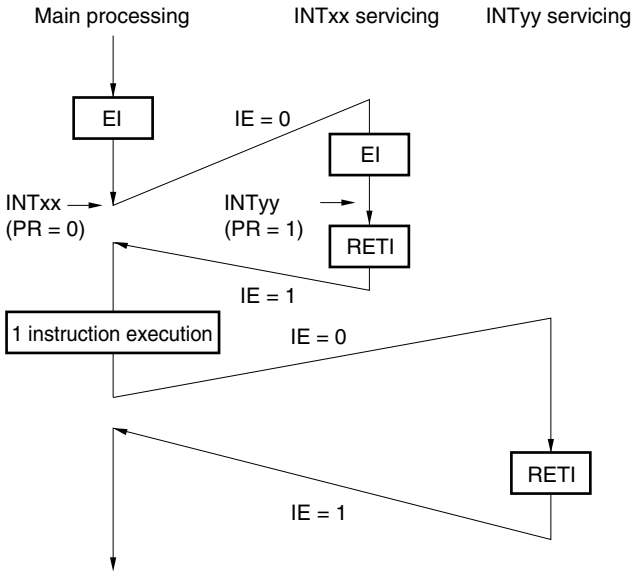
Figure 14-12. Examples of Multiple Interrupt Servicing (1/2)

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

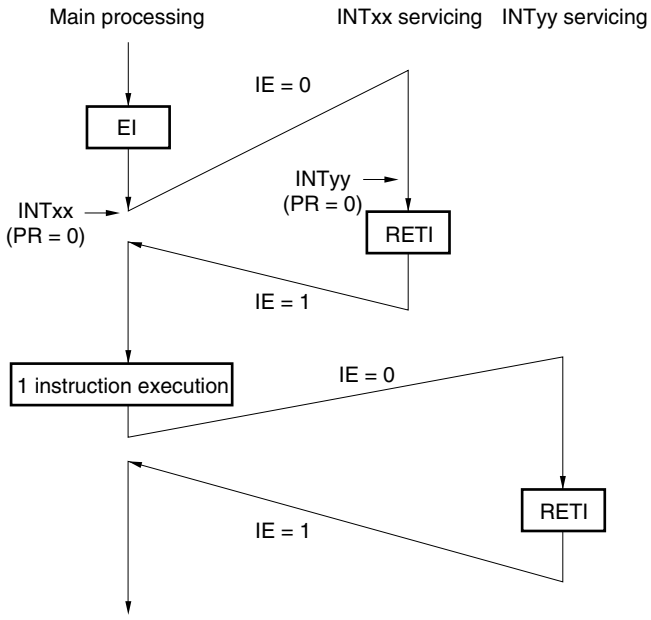


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 14-12. 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

14.4.4 Interrupt request hold

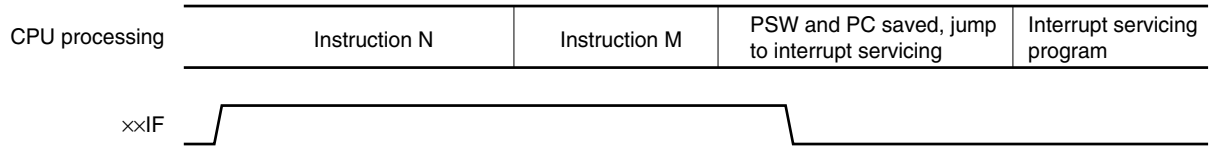
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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 14-13 shows the timing at which interrupt requests are held pending.

Figure 14-13. Interrupt Request Hold



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 15 STANDBY FUNCTION

15.1 Standby Function and Configuration

15.1.1 Standby function

Table 15-1. Relationship Between Operation Clocks in Each Operation Status

<R>

Status oscillation / Operation Mode	High-Speed System Clock Oscillator		Internal Low-Speed Oscillator			CPU Clock After Release	Prescaler Clock Supplied to Peripherals	
			Note 1	**Note 2**				
	MSTOP = 0	MSTOP = 1		RSTOP = 0	RSTOP = 1		MCM0 = 0	MCM0 = 1
Reset	Stopped	Stopped	Stopped			Internal Low-speed Oscillation clock	Stopped	
STOP			Oscillating	Oscillating	Stopped	**Note 3**	Stopped	
HALT	Oscillating					**Note 4**	Internal Low-speed Oscillation clock	High-speed system clock

Notes 1. When “Cannot be stopped” is selected for the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

2. When “Can be stopped by software” is selected for the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

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 the internal low-speed oscillator by a mask option (option byte when using a flash memory version).

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
RSTOP: Bit 0 of the internal low-speed oscillation 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. The HALT mode is intended to stop the CPU operation clock. If the high-speed system clock oscillator and internal low-speed oscillator are operating before the HALT mode is set, oscillation of the high-speed system clock and internal low-speed oscillation clock continues. In this mode, operating current is not decreased as much as in the STOP mode. However, 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 cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released, 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. When shifting to the STOP mode, be sure to stop the peripheral hardware operation before executing STOP instruction.

2. 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.

3. If the internal low-speed oscillator is operating before the STOP mode is set, oscillation of the internal low-speed oscillation clock cannot be stopped in the STOP mode. However, when the internal low-speed oscillation clock is used as the CPU clock, operation is stopped for $17/f_R$ (s) after STOP mode is released.

15.1.2 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, see **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 internal low-speed oscillation 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, or WDT), the STOP instruction or MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Caution Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

Figure 15-1. 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
1	0	0	0	0	$2^{11}/f_{XH}$ min. (204.8 μs min.)
1	1	0	0	0	$2^{13}/f_{XH}$ min. (819.2 μs min.)
1	1	1	0	0	$2^{14}/f_{XH}$ min. (1.64 ms min.)
1	1	1	1	0	$2^{15}/f_{XH}$ min. (3.28 ms min.)
1	1	1	1	1	$2^{16}/f_{XH}$ min. (6.55 ms min.)

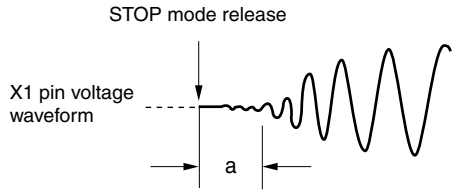
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the internal low-speed oscillation 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.



Remarks 1. Values in parentheses are reference values for operation with f_{XH} = 10 MHz.

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

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the oscillation stabilization wait time of the high-speed system clock when STOP mode is released. The wait time set by OSTS is valid only after the STOP mode is released while the high-speed system clock is selected as the CPU clock. Check the oscillation stabilization time by OSTC after the STOP mode is released when the internal low-speed oscillation clock is selected as the CPU clock.

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

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

Figure 15-2. 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
0	0	1	$2^{11}/f_{XH}$ (204.8 μs)
0	1	0	$2^{13}/f_{XH}$ (819.2 μs)
0	1	1	$2^{14}/f_{XH}$ (1.64 ms)
1	0	0	$2^{15}/f_{XH}$ (3.28 ms)
1	0	1	$2^{16}/f_{XH}$ (6.55 ms)
Other than above			Setting prohibited

<R> **Cautions 1. To set the STOP mode when the high-speed system clock is used as the CPU clock , set OSTS before executing a STOP instruction.**

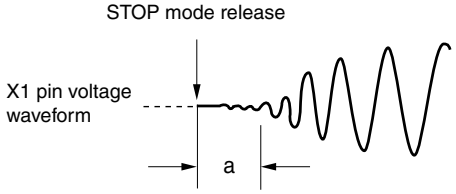
<R> **2. Before setting OSTS, confirm with OSTC that the desired oscillation stabilization time has elapsed.**

3. If the STOP mode is entered and then released while the internal low-speed oscillation 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 up to the oscillation stabilization time set by OSTS. Therefore, note with caution that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after the STOP mode is released.

4. 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.



Remarks 1. Values in parentheses are reference values for operation with f_{XH} = 10 MHz.

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

15.2 Standby Function Operation

15.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. The HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock or internal low-speed oscillation clock.

The operating statuses in the HALT mode are shown below.

Table 15-2. Operating Statuses in HALT Mode

HALT Mode Setting / Item		When HALT Instruction Is Executed While CPU Is Operating Using High-Speed System Clock		When HALT Instruction Is Executed While CPU Is Operating Using Internal Low-Speed Oscillation Clock	
		When Internal Low-Speed Oscillation Clock Continues	When Internal Low-Speed Oscillation Stopped[Note 1]	When High-Speed System Clock Oscillation Continues	When High-Speed System Clock Oscillation Stopped
System clock		Clock supply to CPU stops.			
CPU		Operation stopped			
Port (output latch)		Holds the status before HALT mode is set			
16-bit timer/event counter 00		Operable		Operation not guaranteed	
8-bit timer 50		Operable		Operation not guaranteed	
8-bit timer H0		Operable		Operation not guaranteed	
8-bit timer H1		Operable		Operation not guaranteed when count clock other than $f_R/2^7$ is selected	
Watchdog timer	Internal Low-speed oscillator cannot be stopped[Note 2]	Operable	–	Operable	
	Internal Low-speed oscillator can be stopped[Note 2]	Operation stopped			
A/D converter		Operable		Operation not guaranteed	
Serial interface	UART6	Operable		Operation not guaranteed	
	CSI10	Operable		Operation not guaranteed when serial clock other than external $\overline{SCK10}$ is selected	
Manchester code generator		Operable		Operation not guaranteed	
Clock monitor		Operable	Operation stopped	Operable	Operation stopped
Power-on-clear function		Operable			
Low-voltage detection function		Operable			
External interrupt		Operable			

Notes 1. When “Can be stopped by software” is selected for the internal low-speed oscillator by a mask option (option byte if a flash memory version is used) and the internal low-speed oscillator is stopped by software (for mask options and option bytes, see **CHAPTER 20 MASK OPTIONS/OPTION BYTE**).

2. For the internal low-speed oscillator, “Cannot be stopped” or “Can be stopped by software” can be selected by a mask option (option byte if a flash memory version is used).

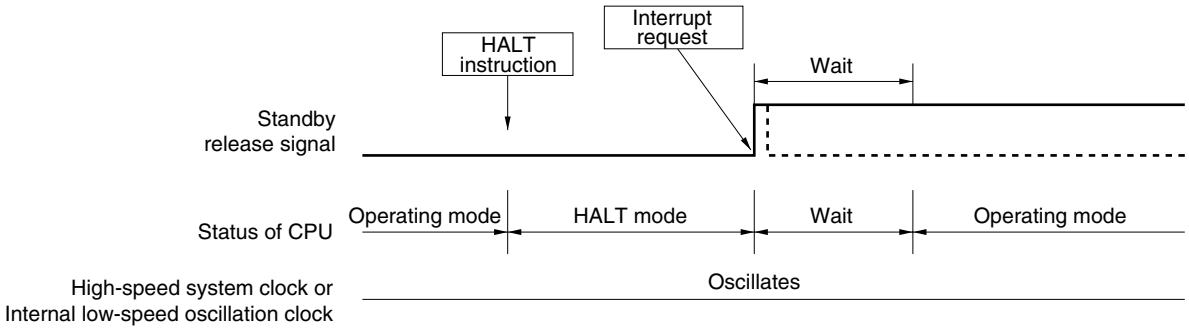
(2) HALT mode release

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 15-3. 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:

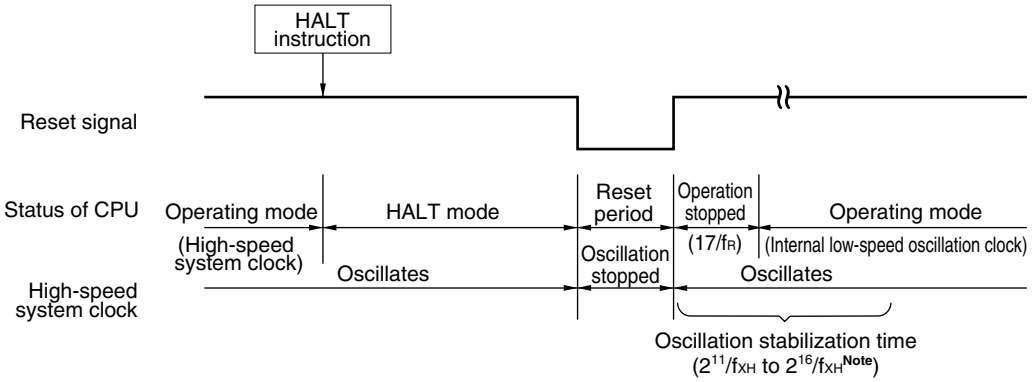
- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

<R> **(b) Release by reset signal (reset by $\overline{\text{RESET}}$ input, POC, LVI, clock monitor, or WDT)**

When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

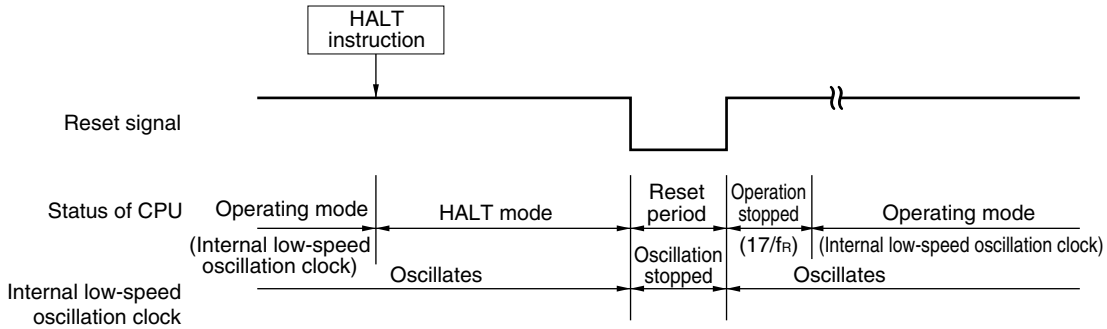
Figure 15-4. HALT Mode Release by Reset Signal

(1) When high-speed system clock is used as CPU clock



Note Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or Internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

(2) When Internal low-speed oscillation clock is used as CPU clock



Remarks 1. f_{XH}: High-speed system clock oscillation frequency

2. f_R: Internal low-speed oscillation clock frequency

Table 15-3. 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
Reset signal	–	–	×	×	Reset processing

×: Don't care

15.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction. It can be set regardless of whether the CPU clock before the setting was the high-speed system clock or internal low-speed oscillation 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 15-4. Operating Statuses in STOP Mode

HALT Mode Setting / Item		When STOP Instruction Is Executed While CPU Is Operating Using High-Speed System Clock		When STOP Instruction Is Executed While CPU Is Operating Using Internal Low-Speed oscillation Clock
		When Internal Low-Speed Oscillation clock Continues	When Internal Low-Speed Oscillation Clock Stopped[Note 1]	
System clock		Only high-speed system clock oscillator oscillation stops. Clock supply to CPU stops.		
CPU		Operation stopped		
Port (output latch)		Holds the status before STOP mode is set		
16-bit timer/event counter 00		Operation stopped		
8-bit timer 50		Operation stopped		
8-bit timer H0		Operation stopped		
8-bit timer H1		Operable[Note 2]	Operation stopped	Operable[Note 2]
Watch-dog timer	Internal low-speed oscillator cannot be stopped[Note 3]	Operable	–	Operable
	Internal low-speed oscillator can be stopped[Note 3]	Operation stopped		
A/D converter		Operation stopped		
Serial interface	UART6	Operation stopped		
	CSI10	Operable only when external $\overline{SCK10}$ is selected as serial clock		
Manchester code generator		Operation stopped		
Clock monitor		Operation stopped		
Power-on-clear function		Operable		
Low-voltage detection function		Operable		
External interrupt		Operable		

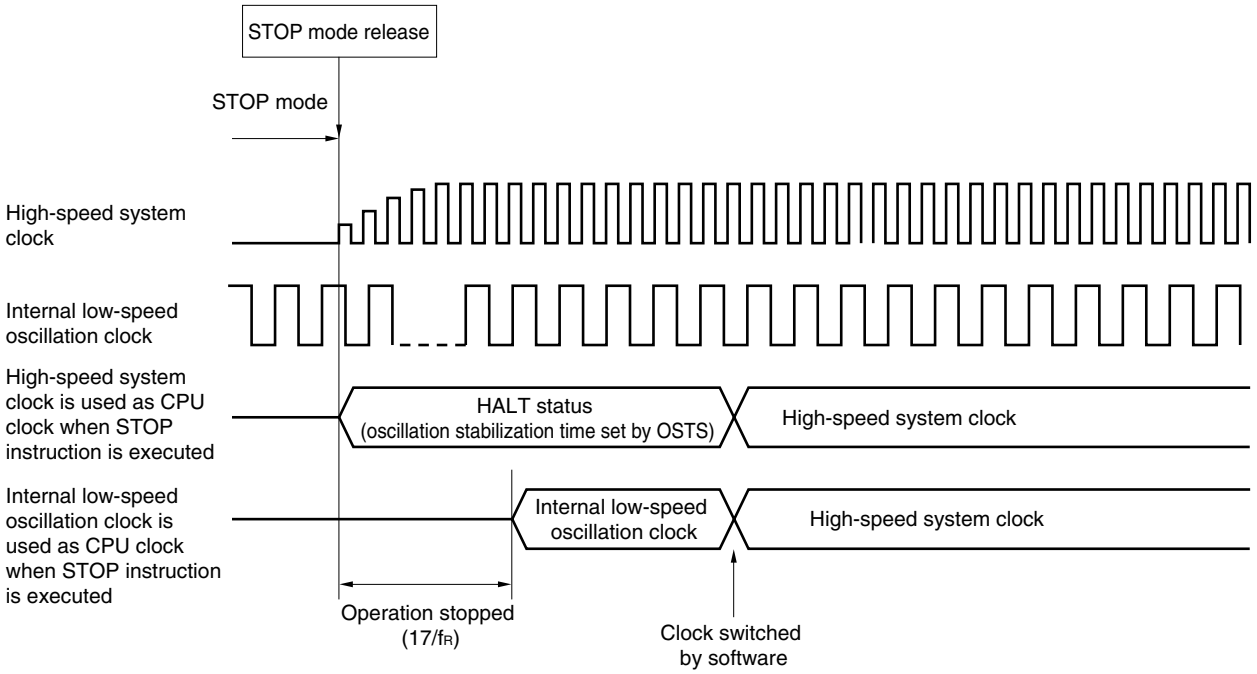
Notes 1. When “Can be stopped by software” is selected for the internal low-speed oscillator by a mask option (option byte if a flash memory version is used) and the internal low-speed oscillator is stopped by software (for mask options and option bytes, see **CHAPTER 20 MASK OPTIONS/OPTION BYTE**).

2. Operable only when $f_R/2^7$ is selected as count clock.

3. For the internal low-speed oscillator, “Cannot be stopped” or “Can be stopped by software” can be selected by a mask option (option byte if a flash memory version is used).

(2) STOP mode release

Figure 15-5. Operation Timing When STOP Mode Is Released



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 15-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock is used as CPU clock

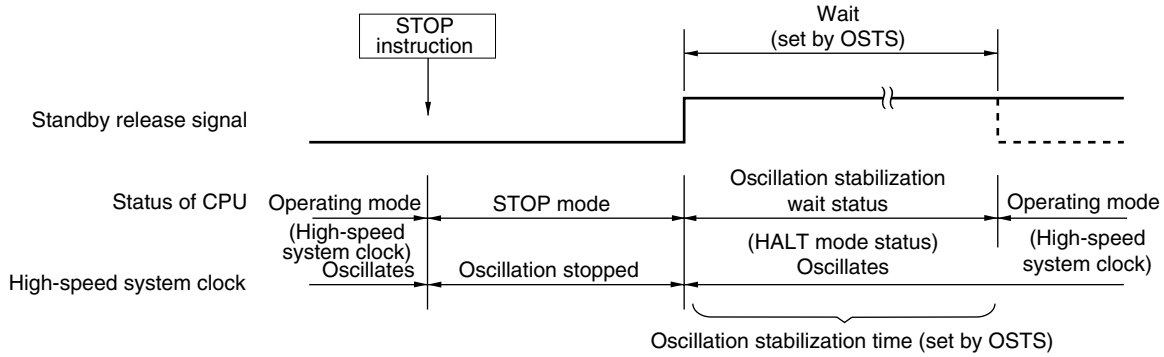
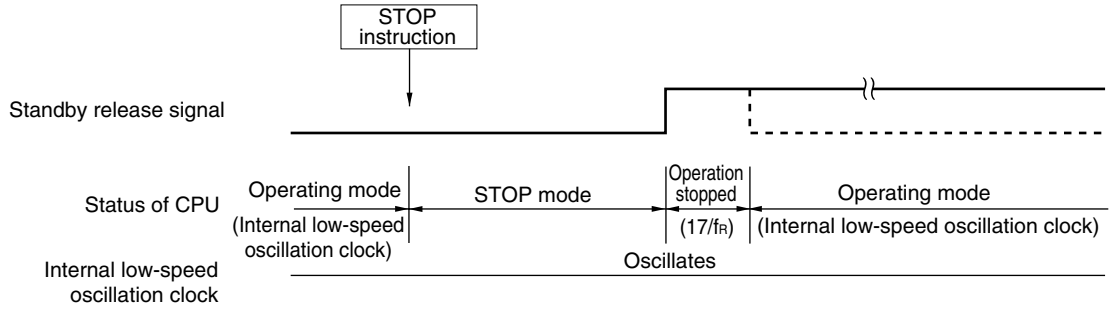


Figure 15-6. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When internal low-speed oscillation clock is used as CPU clock



Remarks 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

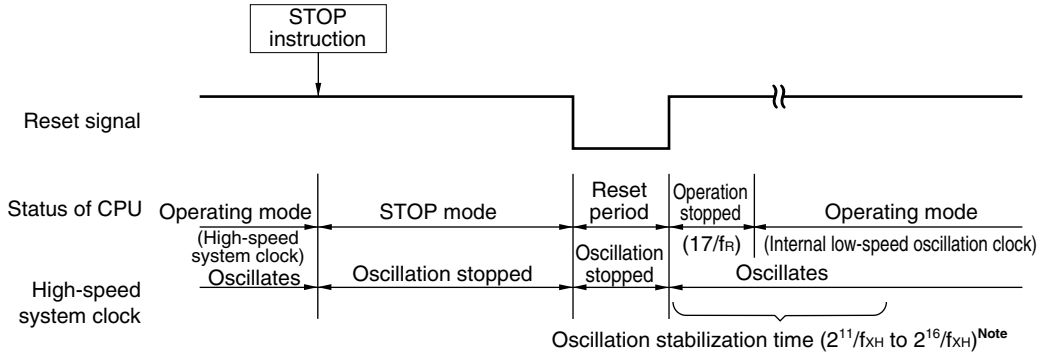
2. f_R: Internal low-speed oscillation clock frequency

<R> **(b) Release by reset signal (reset by $\overline{\text{RESET}}$ input, POC, LVI, clock monitor, or WDT)**

When the reset signal is generated, STOP mode is released and a reset operation is performed after the oscillation stabilization time has elapsed.

Figure 15-7. STOP Mode Release by Reset Signal (1/2)

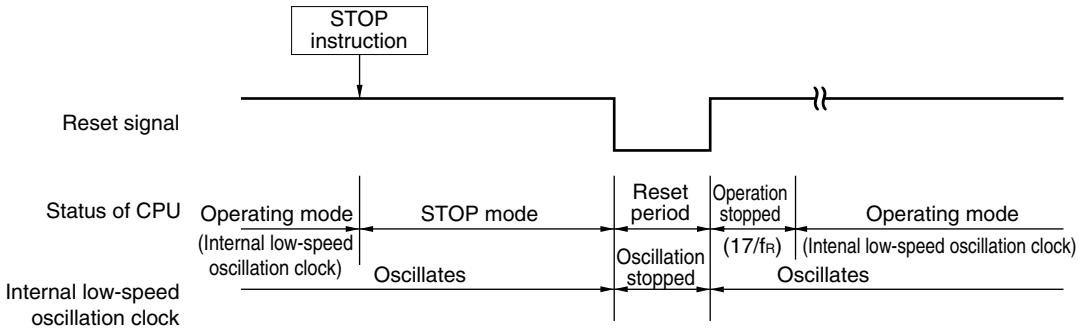
(1) When high-speed system clock is used as CPU clock



Note Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

Figure 15-7. STOP Mode Release by Reset Signal (2/2)

(2) When internal low-speed oscillation clock is used as CPU clock



Remarks 1. f_{XH}: High-speed system clock oscillation frequency

2. f_R: Internal low-speed oscillation clock frequency

Table 15-5. 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
Reset signal	–	–	×	×	Reset processing

×: Don't care

CHAPTER 16 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)

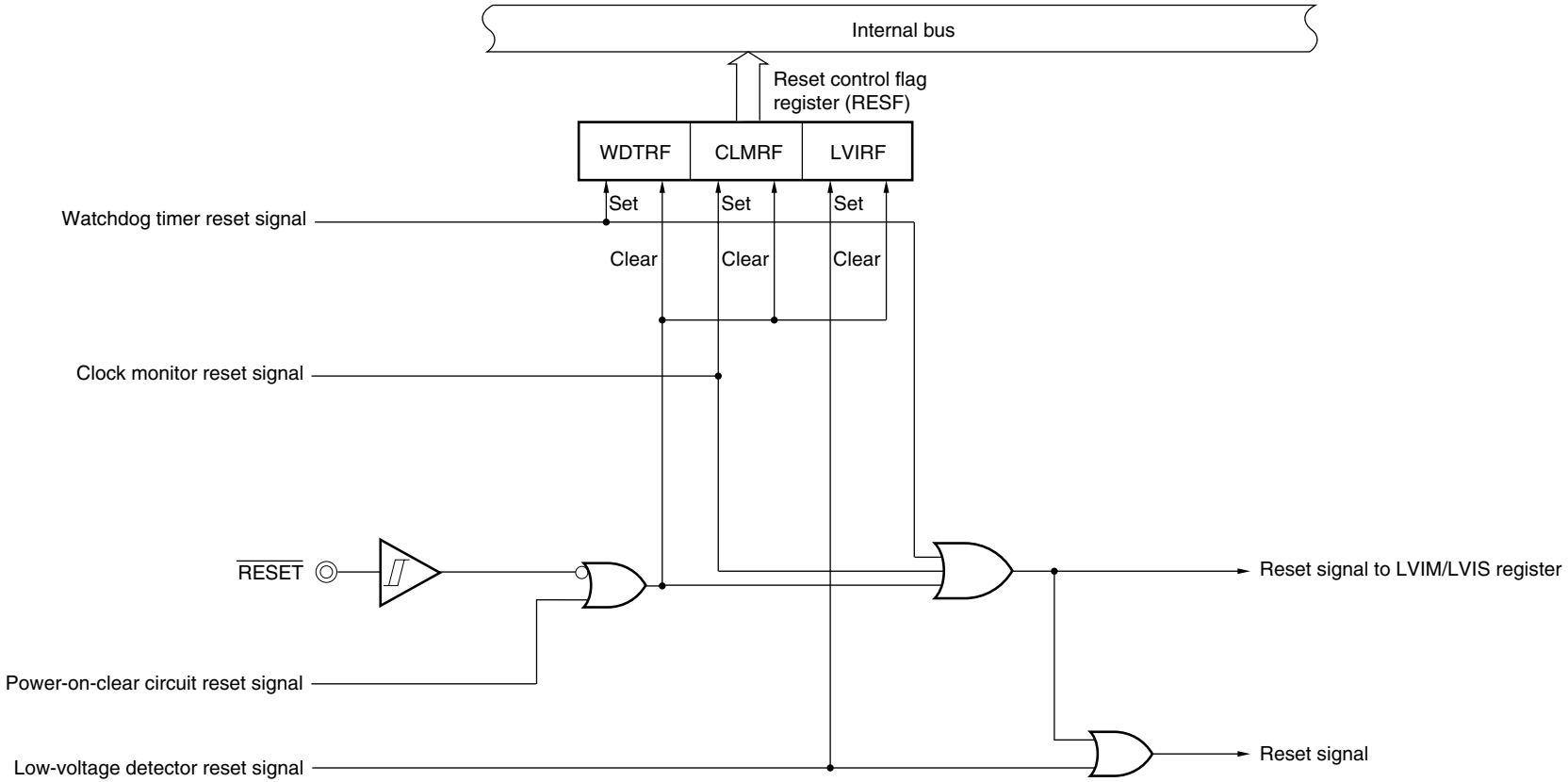
External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is input.

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 hardware is set to the status shown in Table 16-1. 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 internal low-speed oscillation clock after the CPU clock operation has stopped for 17/f_R (s). Reset by the watchdog timer or clock monitor source is automatically released after the reset, and program execution starts using the internal low-speed oscillation clock after the CPU clock operation has stopped for 17/f_R (s) (see **Figures 16-2** to **16-4**). 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 internal low-speed oscillation clock after the CPU clock operation has stopped for 17/f_R (s) (see **CHAPTER 18 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 19 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 internal low-speed oscillation 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 16-1. Block Diagram of Reset Function

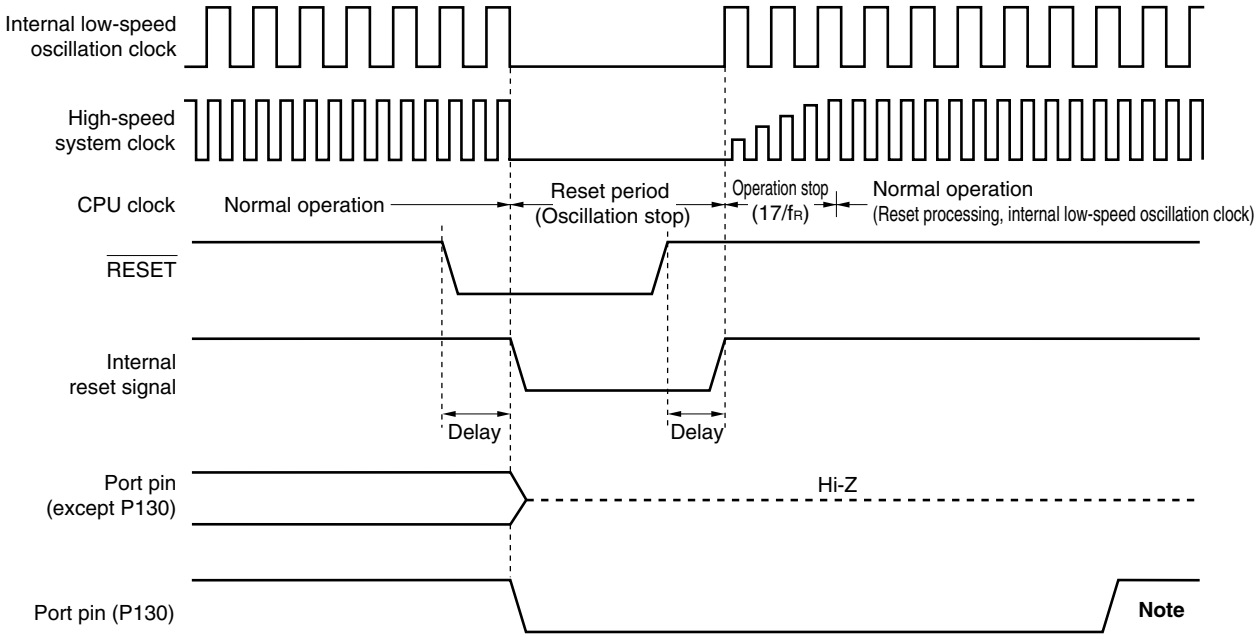


Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level selection register

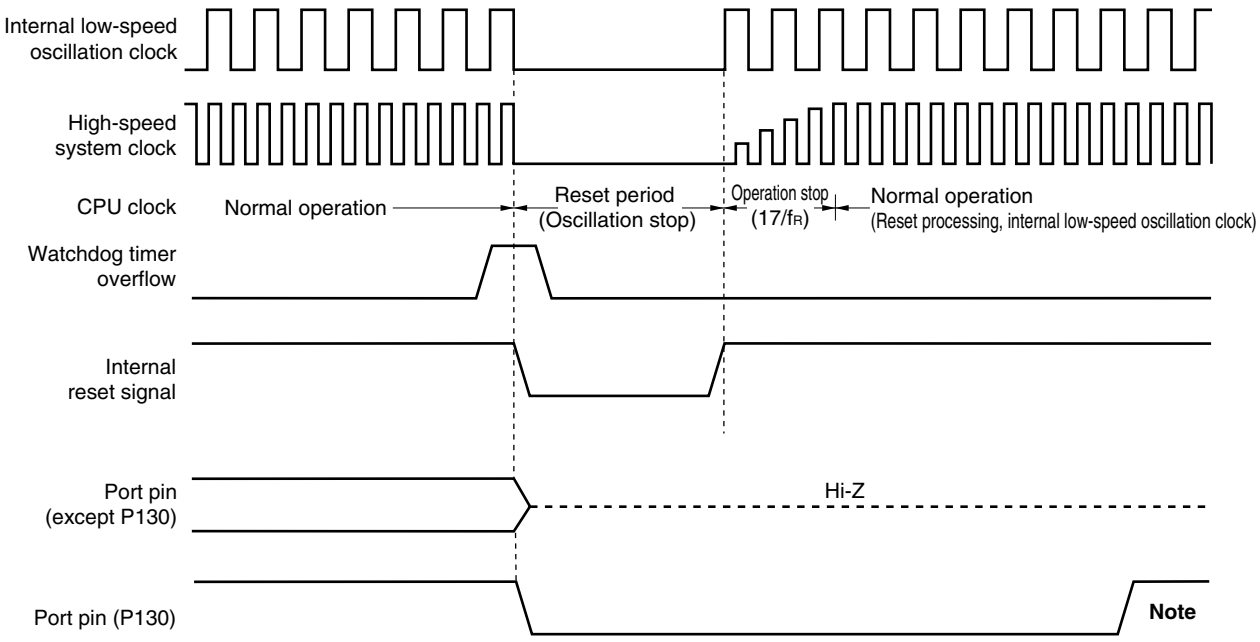
Figure 16-2. Timing of Reset by $\overline{\text{RESET}}$ Input



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 16-3. Timing of Reset Due to Watchdog Timer Overflow

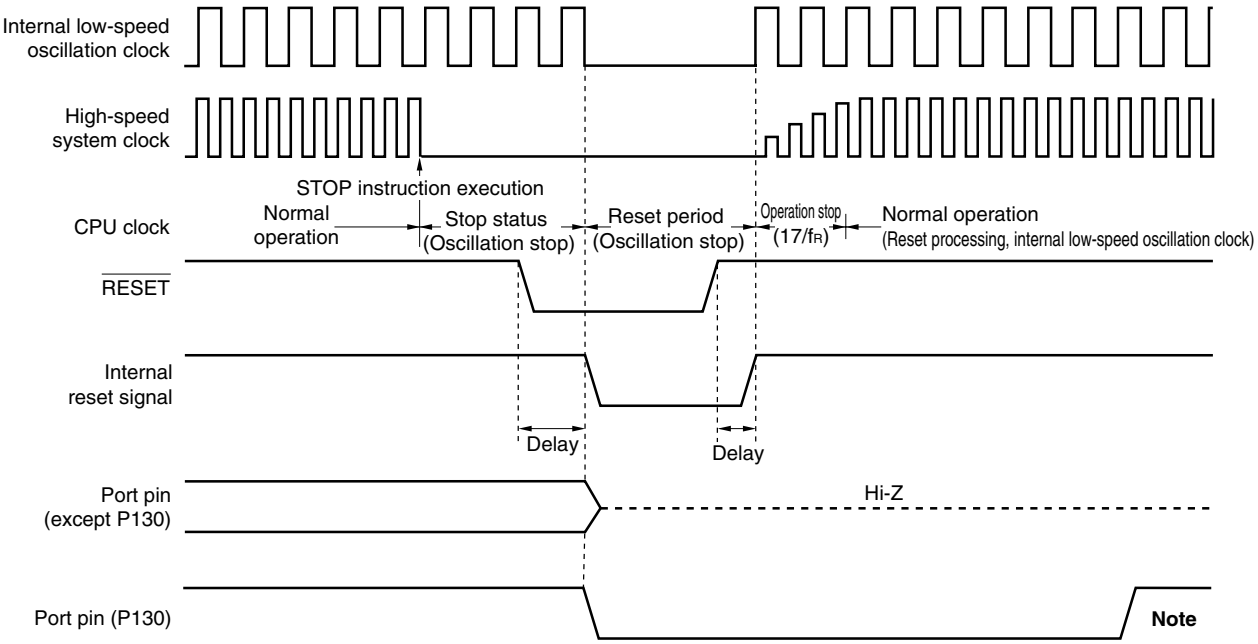


Note Set P130 to high-level output by software.

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 before reset is effected, the output signal of P130 can be dummy-output as the reset signal to the CPU.

Figure 16-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ 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 before 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, see **CHAPTER 18 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 19 LOW-VOLTAGE DETECTOR**.

Table 16-1. Hardware Statuses After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		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 P2, P13) (output latches)		00H (undefined only for P2)
Port mode registers (PM0, PM1)		FFH
Pull-up resistor option registers (PU0, PU1)		00H
Alternate-function pin switch register (PSEL)		00H
Input switch control register (ISC)		00H
Internal memory size switching register (IMS)		CFH
Processor clock control register (PCC)		00H
Internal low-speed oscillation 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
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 50	Timer counter 50 (TM50)	00H
	Compare register 50 (CR50)	00H
	Timer clock selection register 50 (TCL50)	00H
	Timer clock switch register (CSEL)	00H
	Mode control register 50 (TMC50)	00H
8-bit timer/event counters H0, H1	Compare registers 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	Mode registers (TMHMD0, TMHMD1)	00H
	Timer clock switch register (CSEL)	00H
	Carrier control register 1 (TMCYC1)[Note 3]	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. 8-bit timer H1 only.

Table 16-1. Hardware Statuses After Reset (2/2)

Hardware		Status After Reset
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 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
Manchester code generator	Transmit buffer register (MC0TX)	FFH
	Transmit bit count specification register (MC0BIT)	07H
	Control register 0 (MC0CTL0)	10H
	Control register 1 (MC0CTL1)	00H
	Control register 2 (MC0CTL2)	1FH
	Status register (MC0STR)	00H
Clock monitor	Mode register (CLM)	00H
Reset function	Reset control flag register (RESF)	00H[Note]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note]
	Low-voltage detection level selection register (LVIS)	00H[Note]
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

Note These values vary depending on the reset source.

<R>

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by CLM	Reset by LVI
RESF	WDTRF	Cleared (00H)	Cleared (00H)	Set (1)	Held	Held
	CLMRF			Held	Set (1)	Held
	LVIRF			Held	Held	Set (1)
LVIM				Cleared (00H)	Cleared (00H)	Held
LVIS						

16.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the μPD780862 Subseries. 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 16-5. 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 16-2.

Table 16-2. RESF Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ 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 17 CLOCK MONITOR

17.1 Functions of Clock Monitor

The clock monitor samples the high-speed system clock using the internal low-speed oscillation clock, 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 16 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) and during the oscillation stabilization time
- When the internal low-speed oscillation clock is stopped

Remark MSTOP: Bit 7 of the main OSC control register (MOC)

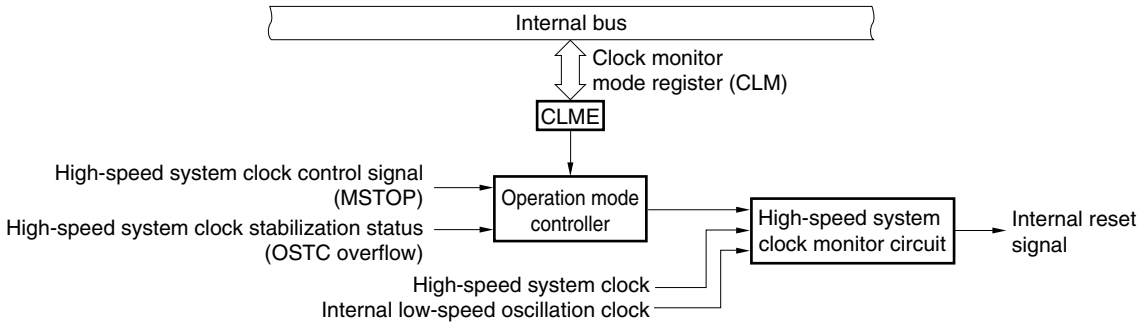
17.2 Configuration of Clock Monitor

The clock monitor includes the following hardware.

Table 17-1. Configuration of Clock Monitor

Item	Configuration
Control register	Clock monitor mode register (CLM)

Figure 17-1. Block Diagram of Clock Monitor



Remark MSTOP: Bit 7 of the main OSC control register (MOC)
OSTC: Oscillation stabilization time counter status register (OSTC)

17.3 Registers 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 this register to 00H.

Figure 17-2. 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.

3. The clock monitor stops operating during the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).

17.4 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>

Set bit 0 (CLME) of the clock monitor mode register (CLM) 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) and during the oscillation stabilization time
- When the internal low-speed oscillation clock is stopped

Remark MSTOP: Bit 7 of the main OSC control register (MOC)

Table 17-2. Operation Status of Clock Monitor (When CLME = 1)

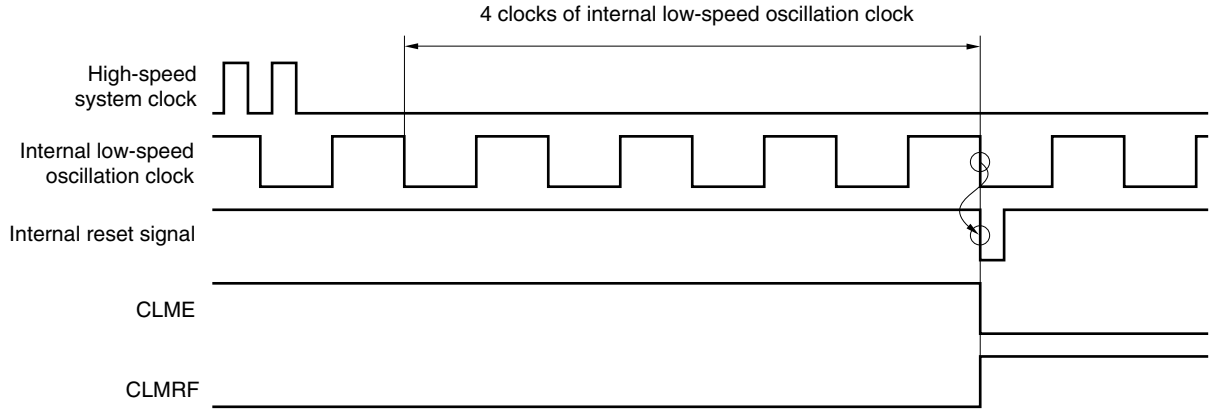
CPU Operation Clock	Operation Mode	High-Speed System Clock Status	Internal Low-Speed Oscillation Clock Status	Clock Monitor Status
High-speed system clock	STOP mode	Stopped	Oscillating	Stopped
			Stopped[Note]	
	$\overline{\text{RESET}}$ input		Oscillating	
			Stopped[Note]	
	Normal operation mode HALT mode	Oscillating	Oscillating	Operating
			Stopped[Note]	Stopped
Internal low-speed oscillation clock	STOP mode	Stopped	Oscillating	Stopped
	$\overline{\text{RESET}}$ input			
	Normal operation mode HALT mode	Oscillating		Operating
		Stopped		Stopped

Note The internal low-speed oscillation clock is stopped only when the “Internal low-speed oscillator can be stopped by software” is selected by a mask option (option byte if a flash memory version is used). If “Internal low-speed oscillator cannot be stopped” is selected, the internal low-speed oscillation clock cannot be stopped.

The clock monitor timing is as shown in Figure 17-3.

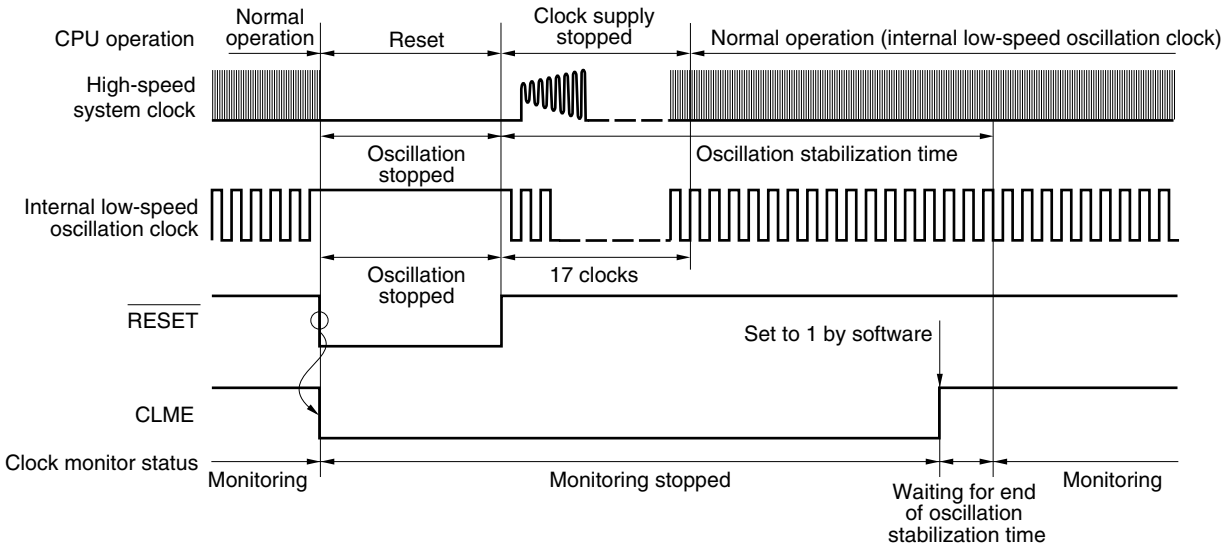
Figure 17-3. Timing of Clock Monitor (1/4)

(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)



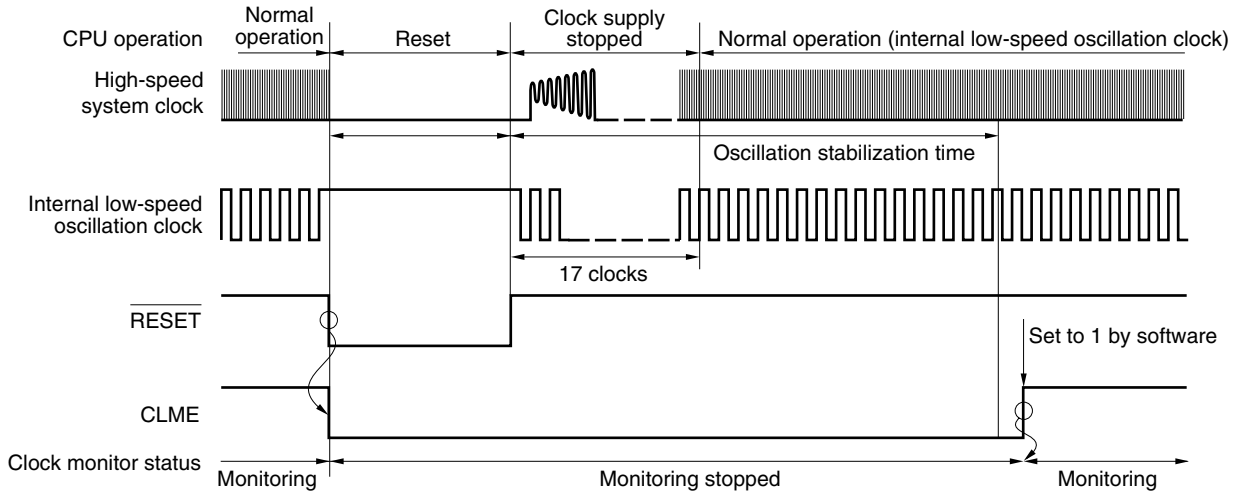
$\overline{\text{RESET}}$ input clears bit 0 (CLME) of the clock monitor mode register (CLM) to 0 and stops the clock monitor operation. Even if CLME is set to 1 by software during the oscillation stabilization time (OSTS register reset value = 05H ($2^{16}/f_{XH}$)) of the high-speed system clock, monitoring 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 or internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). 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 ($2^{16}/f_{XH}$)) has elapsed.

Figure 17-3. 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)

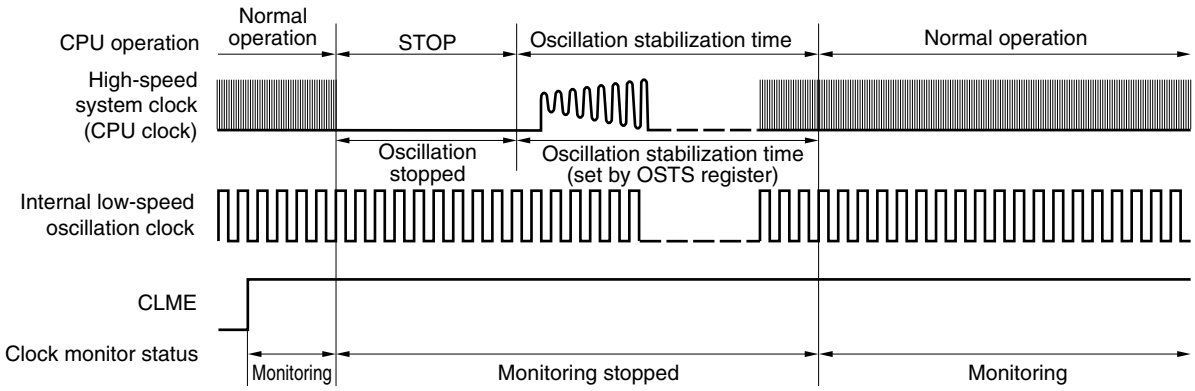


$\overline{\text{RESET}}$ input clears bit 0 (CLME) of the clock monitor mode register (CLM) to 0 and stops the clock monitor operation. When CLME is set to 1 by software at the end of the oscillation stabilization time (OSTS register reset value = 05H ($2^{16}/f_{XH}$)) of the high-speed system clock, monitoring is started.

Caution Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). 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 ($2^{16}/f_{XH}$)) has elapsed.

(4) Clock monitor status after STOP mode is released

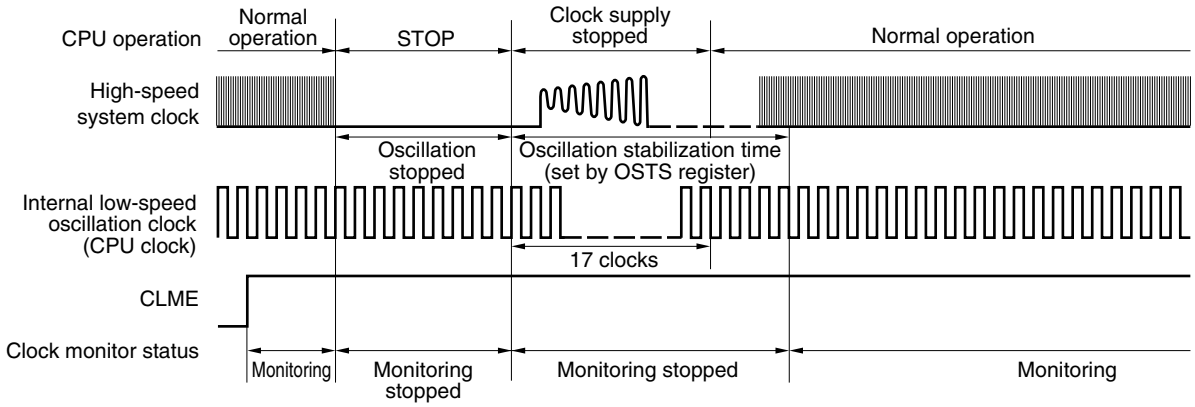
(CLME = 1 is set when CPU clock operates on high-speed system clock and before entering STOP mode)



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.

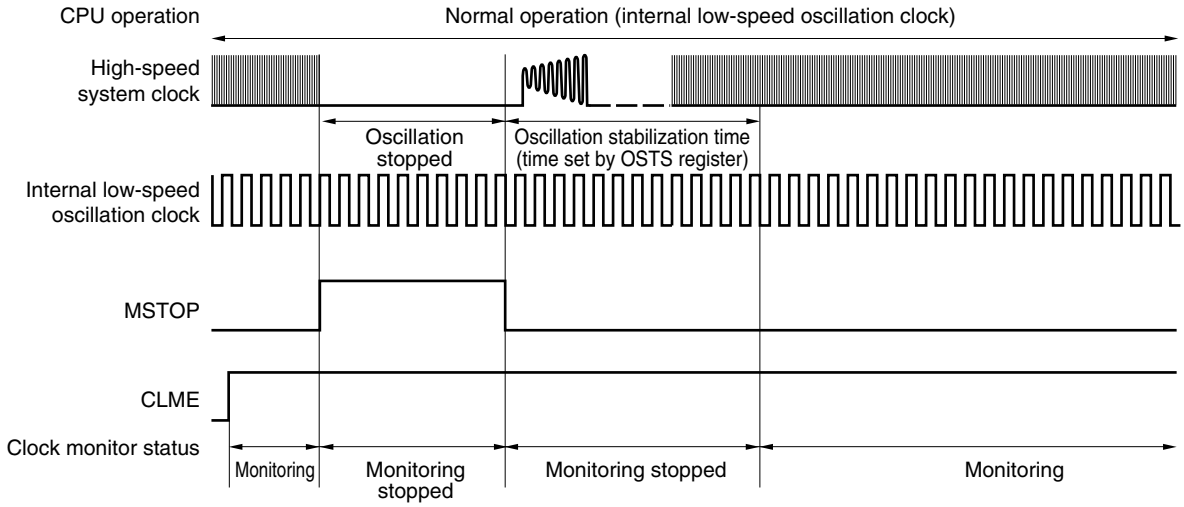
Figure 17-3. Timing of Clock Monitor (3/4)

(5) Clock monitor status after STOP mode is released
(CLME = 1 is set when CPU clock operates on internal low-speed oscillation 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.

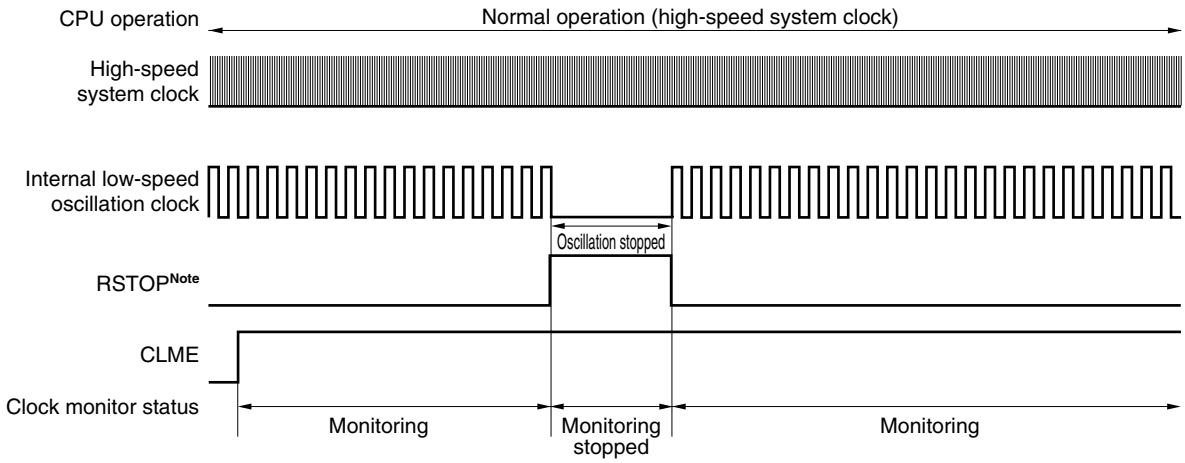
(6) Clock monitor status after high-speed system clock oscillation is stopped by software



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.

Figure 17-3. Timing of Clock Monitor (4/4)

(7) Clock monitor status after internal low-speed oscillation 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 internal low-speed oscillation clock is stopped, monitoring automatically starts after the internal low-speed oscillation clock is stopped. Monitoring is stopped when oscillation of the internal low-speed oscillation clock is stopped.

Note If it is specified by a mask option (option byte when using a flash memory version) that the internal low-speed oscillator cannot be stopped, the setting of bit 0 (RSTOP) of the internal low-speed oscillation 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 18 POWER-ON-CLEAR CIRCUIT

18.1 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} = 2.85 V ±0.15 V), and generates internal reset signal when $V_{DD} < V_{POC}$.

Cautions 1. If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

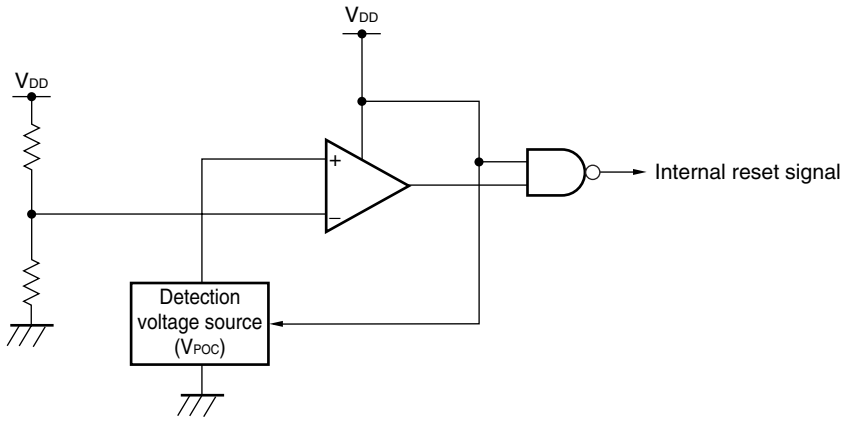
2. Although the supply voltage is V_{DD} = 2.7 to 5.5 V, use the product in a voltage range of 3.0 to 5.5 V because the detection voltage (V_{POC}) of the POC circuit is 2.85 V ±0.15 V.

Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), low-voltage-detector (LVI), 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 16 RESET FUNCTION**.

18.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 18-1.

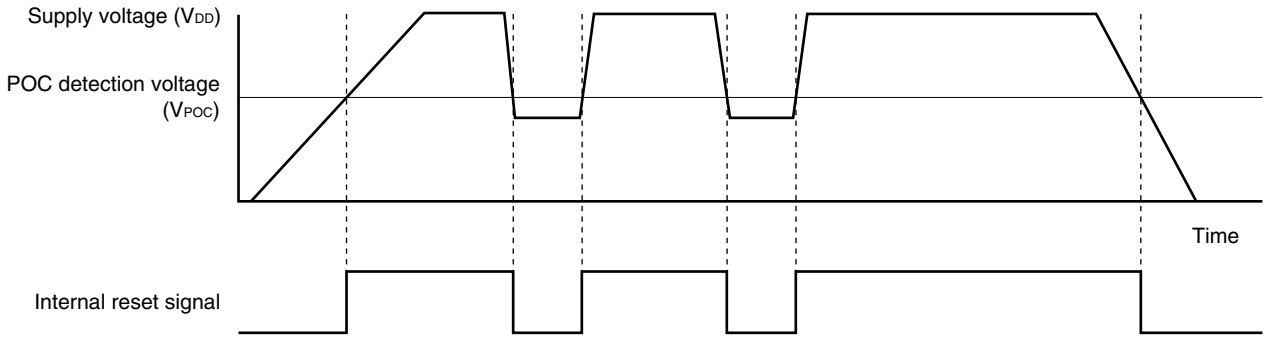
Figure 18-1. Block Diagram of Power-on-Clear Circuit



18.3 Operation of Power-on-Clear Circuit

In the power-on-clear circuit, the supply voltage (V_{DD}) and detection voltage (V_{POC} = 2.85 V ±0.15 V) are compared, and when $V_{DD} < V_{POC}$, an internal reset signal is generated.

Figure 18-2. Timing of Internal Reset Signal Generation in Power-on-Clear Circuit



18.4 Cautions for Power-on-Clear Circuit

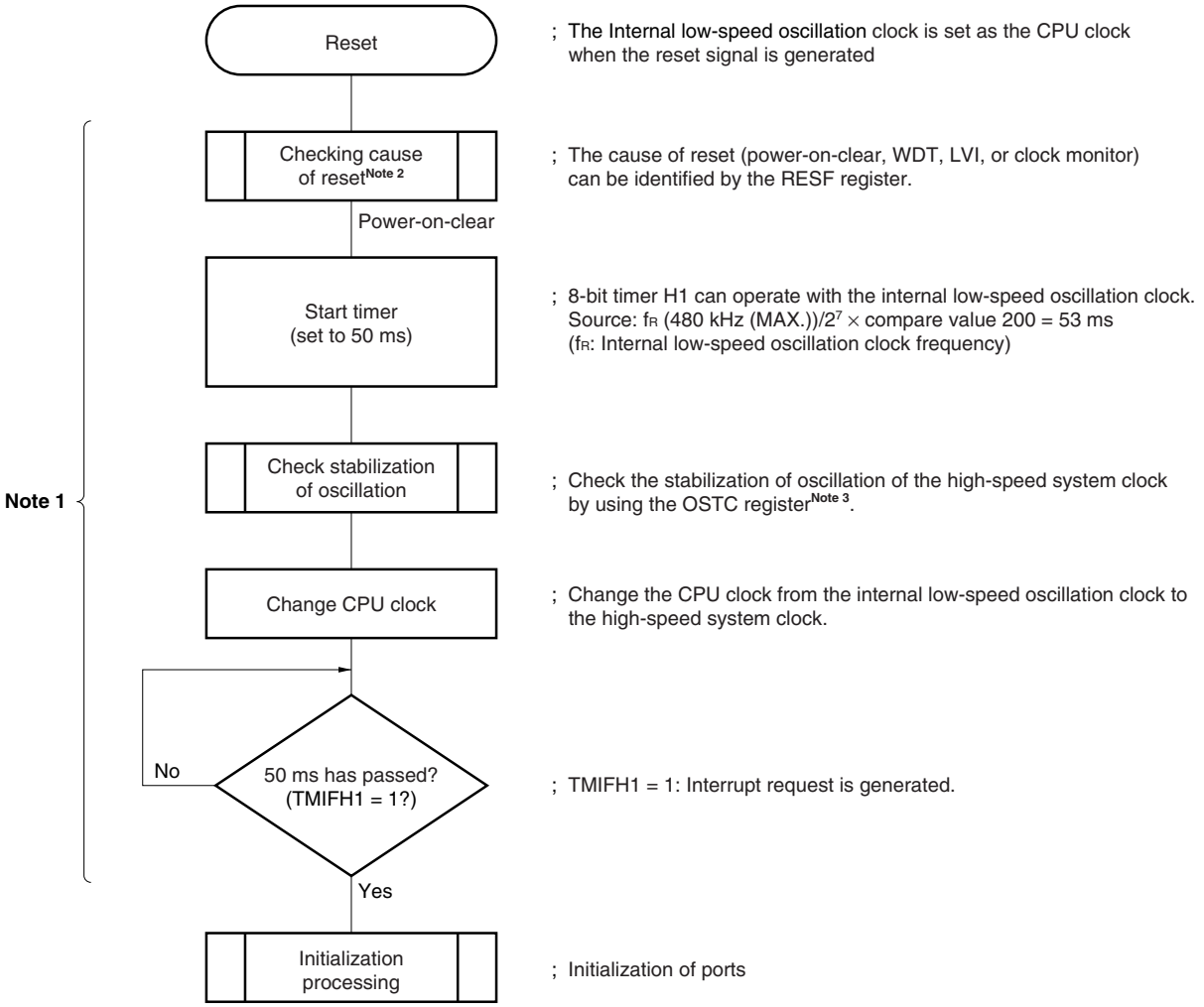
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC} = 2.85 V ±0.15 V), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the 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 18-3. 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



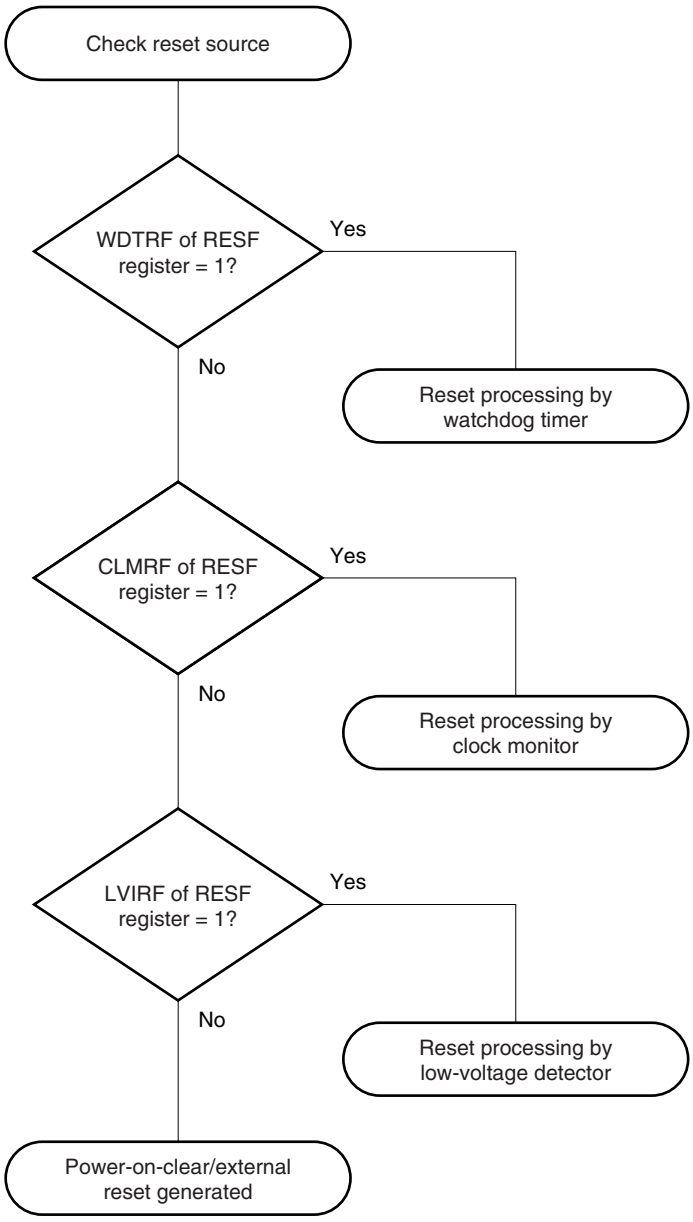
Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

3. Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

Figure 18-3. Example of Software Processing After Release of Reset (2/2)

• Checking reset source



CHAPTER 19 LOW-VOLTAGE DETECTOR

19.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has the following functions.

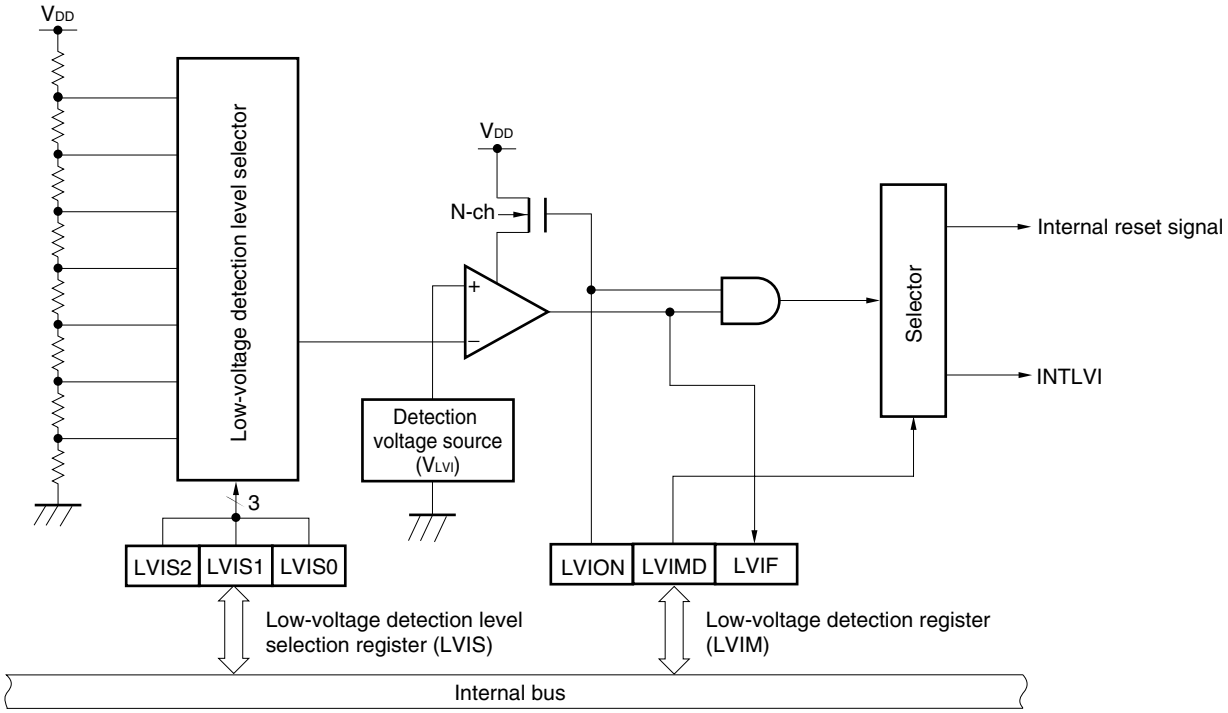
- Compares supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an internal interrupt signal or internal reset signal when $V_{DD} < V_{LVI}$.
- Detection levels (seven levels) of supply voltage can be changed by software.
- Interrupt or reset function can be selected by software.
- 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 16 RESET FUNCTION**.

19.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown below.

Figure 19-1. Block Diagram of Low-Voltage Detector



19.3 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 this register to 00H.

Figure 19-2. 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	0	0	LVIMD	LVIF

LVION[Notes 2, 3]	Enables low-voltage detection operation
0	Disables operation
1	Operation starts

LVIMD[Note 2]	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 4]	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 a read-only bit.

2. LVION and LVIMD are cleared to 0 at a reset other than an LVI reset. These are not cleared to 0 at an LVI reset.

3. 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.

4. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

Cautions 1. To stop LVI, follow either of the procedures below.

- **When using 8-bit memory manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

2. Be sure to clear bits 2 to 6 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 this register to 00H.

Figure 19-3. 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	0	LVIS2	LVIS1	LVIS0

LVIS2	LVIS1	LVIS0	Detection level
0	0	0	V_{LVI0} (4.3 V ±0.2 V)
0	0	1	V_{LVI1} (4.1 V ±0.2 V)
0	1	0	V_{LVI2} (3.9 V ±0.2 V)
0	1	1	V_{LVI3} (3.7 V ±0.2 V)
1	0	0	V_{LVI4} (3.5 V ±0.2 V)
1	0	1	V_{LVI5} (3.3 V ±0.15 V)
1	1	0	V_{LVI6} (3.1 V ±0.15 V)
1	1	1	Setting prohibited

Caution Be sure to clear bits 3 to 7 to 0.

19.4 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

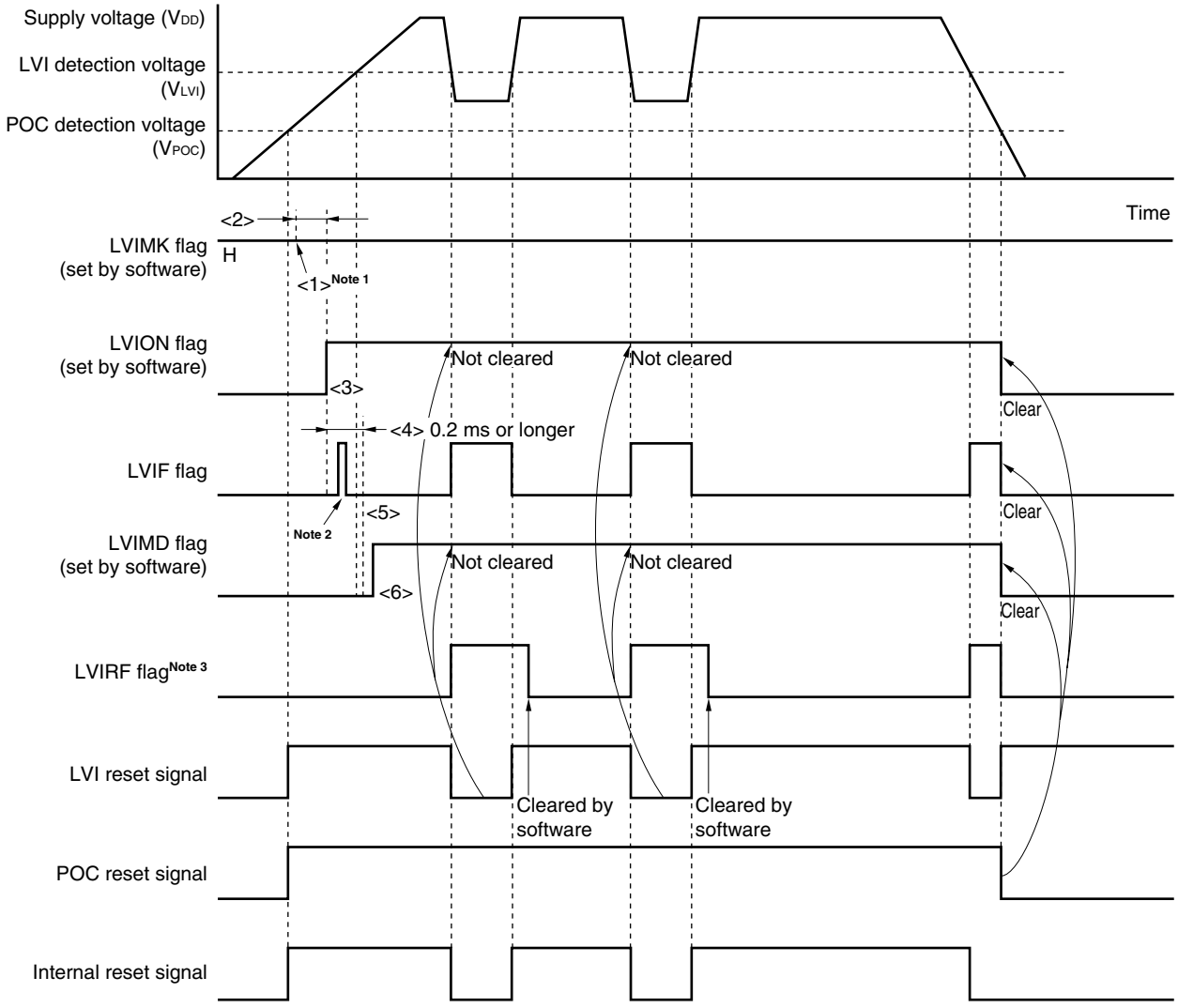
<1> Mask the LVI interrupt (LVIMK = 1).
<2> Set the detection voltage using bits 2 to 0 (LVIS2 to LVIS0) of the low-voltage detection level selection register (LVIS).
<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
<4> Use software to instigate a wait of at least 0.2 ms.
<5> Wait until it is checked that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" by 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 19-4 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 and LVION to 0 in that order.

Figure 19-4. 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, see **CHAPTER 16 RESET FUNCTION**.

Remark <1> to <6> in Figure 19-4 above correspond to <1> to <6> in the description of "when starting operation" in **19.4 (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 2 to 0 (LVIS2 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> Wait until it is checked that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" by 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 vector interrupts are used).

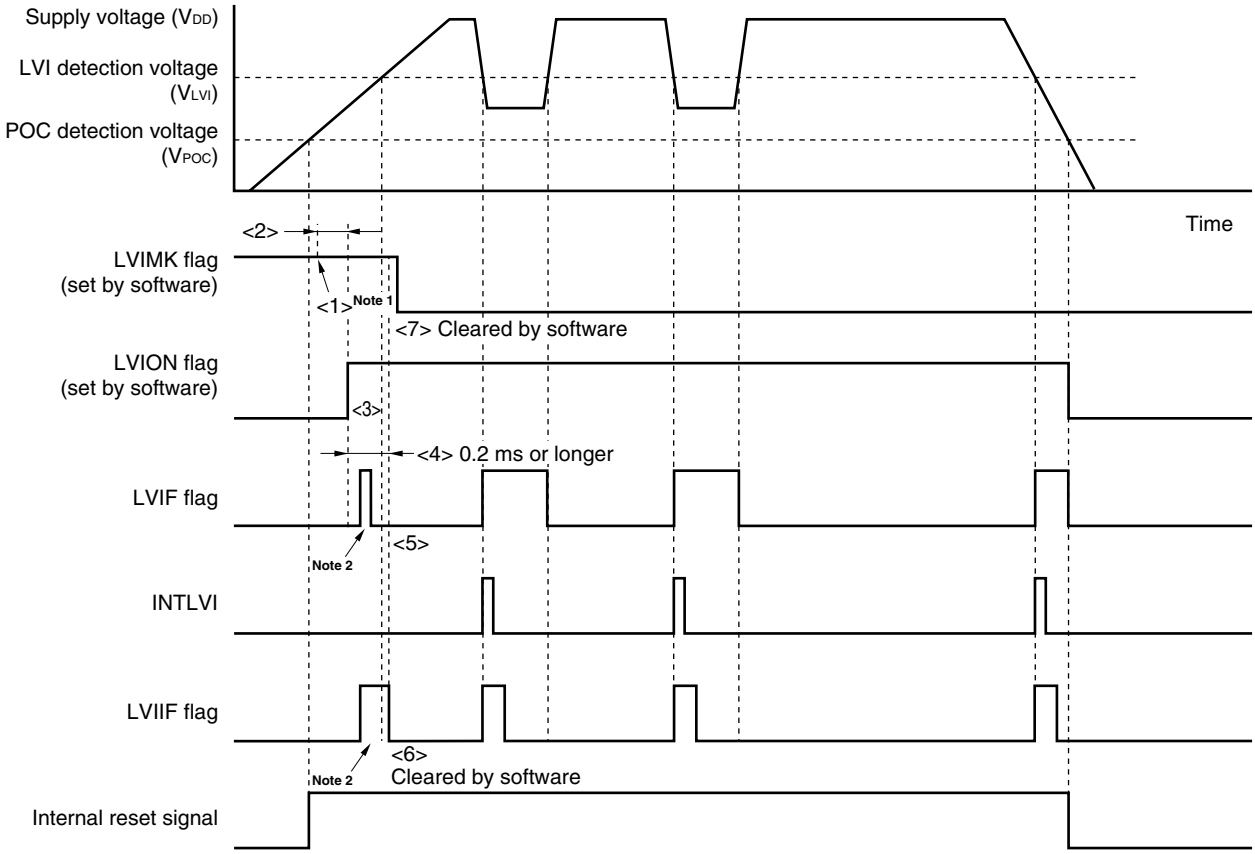
Figure 19-5 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

- 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 19-5. Timing of Low-Voltage Detector Interrupt Signal Generation



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 19-5 above correspond to <1> to <7> in the description of "when starting operation" in **19.4 (2) When used as interrupt**.

19.5 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 (a) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take action (b) below.

In this system, take the following actions.

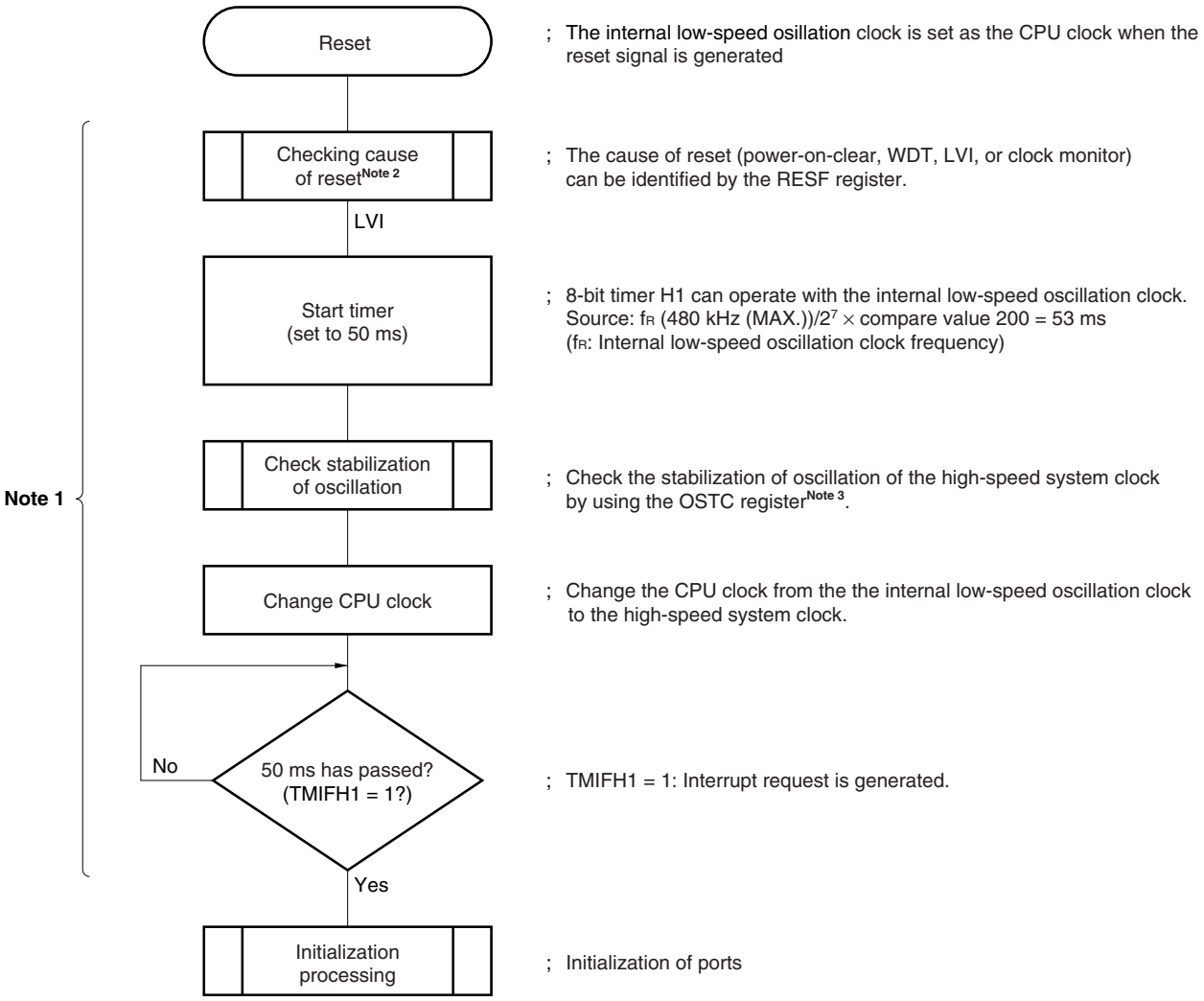
<Action>

(a) 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 19-6. 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



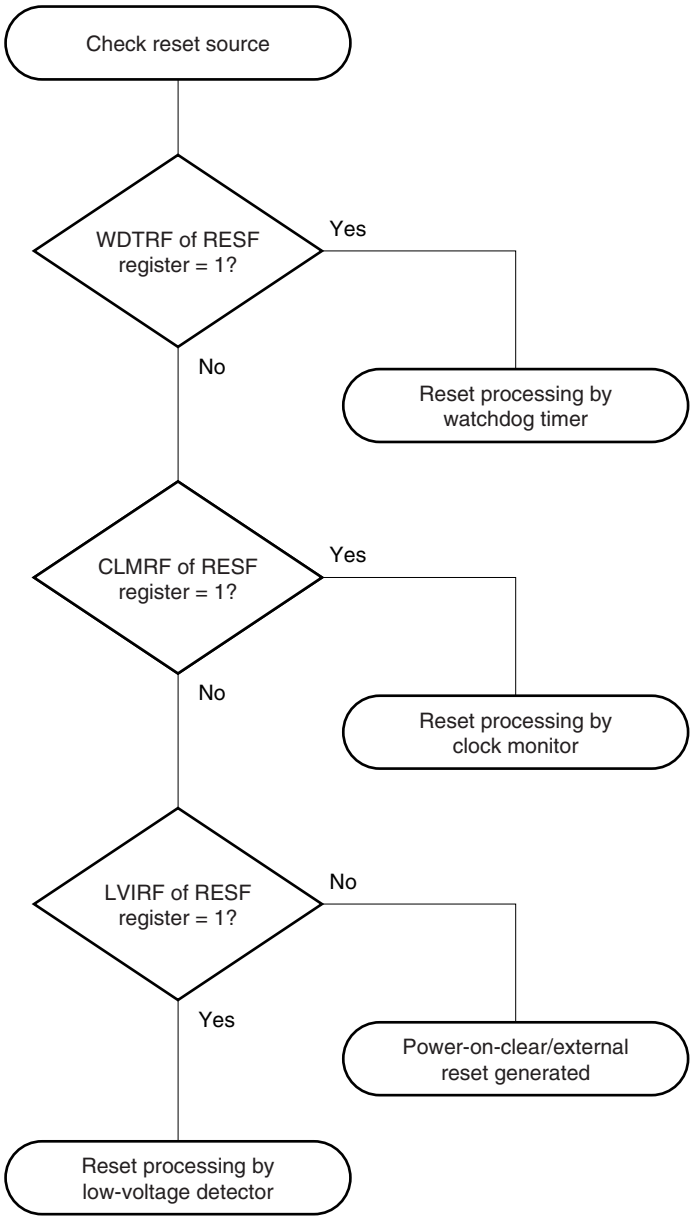
Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

3. Waiting for the oscillation stabilization time is not required when the external RC oscillation clock or internal high-speed oscillation clock is selected as the high-speed system clock by a mask option (option byte when using a flash memory version). Therefore, the CPU clock can be switched without reading the OSTC value.

Figure 19-6. Example of Software Processing After Release of Reset (2/2)

• Checking reset source



(b) 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})” with the LVIF flag, and then enable interrupts (EI).

CHAPTER 20 MASK OPTIONS/OPTION BYTE

20.1 Mask Options (Mask ROM Versions)

Mask ROM versions have the following mask options.

1. High-speed system clock oscillation selection
   - Crystal/ceramic oscillation
   - External RC oscillation
   - Internal high-speed oscillation
2. Internal low-speed oscillator oscillation
   - Cannot be stopped[Note]
   - Can be stopped by software

Note If “Internal low-speed oscillator cannot be stopped” is selected, the source clock of the watchdog timer is fixed to the internal low-speed oscillator clock, and it cannot be changed.

Caution Select crystal/ceramic oscillation or external RC oscillation when using an external clock.

20.2 Option Bytes (Flash Memory Versions)

In the flash memory versions, the functions equivalent to the mask options of the mask ROM versions can be realized by setting using option bytes.

Option bytes are prepared at address 0080H in the flash memory.

When using flash memory version products, be sure to set the mask option information to the option bytes.

Figure 20-1. Allocation of Option Bytes (Flash Memory Versions)

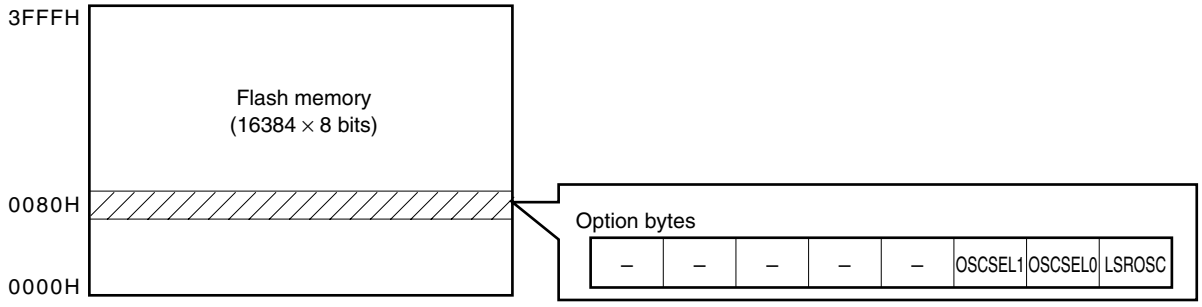


Figure 20-2. Format of Option Bytes (Flash Memory Versions)

Address: 0080H

7	6	5	4	3	2	1	0
0	0	0	0	0	OSCSEL1	OSCSEL0	LSROSC

OSCSEL1	OSCSEL0	High-speed system clock oscillation selection
0	0	Crystal/ceramic oscillation
0	1	External RC oscillation
1	×	Internal high-speed oscillation

LSROSC	Internal low-speed oscillator oscillation
0	Can be stopped by software
1	Cannot be stopped

Caution Select crystal/ceramic oscillation or external RC oscillation when using an external clock.

Remark An example of software coding for setting the option bytes is shown below.

```
OPT     CSEG    AT 0080H
OPTION: DB      03H     ; Set to option byte (external RC oscillation used/internal low-speed
                          oscillator cannot be stopped)
```

CHAPTER 21 FLASH MEMORY

The μPD78F0862 and 78F0862A are provided as the flash memory version of the μPD780862 Subseries.

The μPD78F0862 and 78F0862A replace the internal mask ROM of the μPD780862 with flash memory to which a program can be written, erased, and overwritten while mounted on the board. Table 21-1 lists the differences between the μPD78F0862, 78F0862A and the mask ROM versions.

Table 21-1. Differences Between μPD78F0862, 78F0862A and Mask ROM Versions

Item	μPD78F0862, 78F0862A	Mask ROM Versions
Internal ROM configuration	Flash memory	Mask ROM
Internal ROM capacity	16 KB[Note]	μPD780861: 8 KB μPD780862: 16 KB
Internal high-speed RAM capacity	768 bytes[Note]	μPD780861: 512 bytes μPD780862: 768 bytes
IC pin	None	Available
FLMD0, FLMD1 pins	Available	None
Electrical specifications	Refer to the description of electrical specifications.	

Note The same capacity as the mask ROM versions can be specified by means of the internal memory size switching register (IMS).

Cautions 1. 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.

<R> **2. μPD78F0862 and 78F0862A differ only in the flash memory characteristics. For details, refer to "Flash Memory Programming Characteristics" in the chapter of the electrical specifications.**

21.1 Internal Memory Size Switching Register

The μPD78F0862 and 78F0862A allow users to select the internal memory capacity using the internal memory size switching register (IMS) so that the same memory map as that of the mask ROM versions with a different internal memory capacity can be achieved.

IMS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to CFH.

Caution The initial value of IMS is "setting prohibited (CFH)". Be sure to set the value of the relevant mask ROM version at initialization.

Figure 21-1. 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	0	0	768 bytes
0	1	0	512 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
Other than above				Setting prohibited

The IMS settings required to obtain the same memory map as mask ROM versions are shown in Table 21-2.

Table 21-2. Internal Memory Size Switching Register Settings

Target Mask ROM Versions	IMS Setting
μPD780861	42H
μPD780862	04H

Caution When using a mask ROM version, be sure to set the value indicated in Table 21-2 to IMS.

21.2 Writing with Flash Programmer

Data can be written to the flash memory on-board or off-board, by using a dedicated flash programmer (FlashPro 4).

(1) On-board programming

The contents of the flash memory can be rewritten after the μPD78F0862 and 78F0862A have 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 μPD78F0862 and 78F0862A are mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 21-3. Wiring Between μPD78F0862, 78F0862A and Dedicated Flash Programmer

Pin Configuration of Dedicated Flash Programmer			With CSI10 + HS		With CSI10		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/TOH1/(INTP3)	11	SO10/P12/TOH1/(INTP3)	11	TxD6/P13/INTP1/(TOH1)/(MCGO)	12
SO/TxD	Output	Transmit signal	SI10/P11/INTP3	10	SI10/P11/INTP3	10	RxD6/P14/<INTP0>	13
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$/P10/(INTP1)	9	$\overline{\text{SCK10}}$/P10/(INTP1)	9	Not required	Not required
CLK	Output	Clock to μPD78F0862, 78F0862A	X1[CL1]	2	X1[CL1]	2	X1[CL1]	2
			X2[CL2]/P02Note	3	X2[CL2]/P02Note	3	X2[CL2]/P02Note	3
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	6	$\overline{\text{RESET}}$	6	$\overline{\text{RESET}}$	6
FLMD0	Output	Mode signal	FLMD0	4	FLMD0	4	FLMD0	4
FLMD1	Output	Mode signal	HS/P15/TOH0/FLMD1	14	HS/P15/TOH0/FLMD1	14	HS/P15/TOH0/FLMD1	14
H/S	Input	Handshake signal for CSI10 + HS signal	HS/P15/TOH0/FLMD1	14	Not required	Not required	Not required	Not required
V_{DD}	I/O	V_{DD} voltage generation	V_{DD}	5	V_{DD}	5	V_{DD}	5
			AV_{REF}	20	AV_{REF}	20	AV_{REF}	20
GND	–	Ground	V_{SS}	1	V_{SS}	1	V_{SS}	1

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 21-2. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode

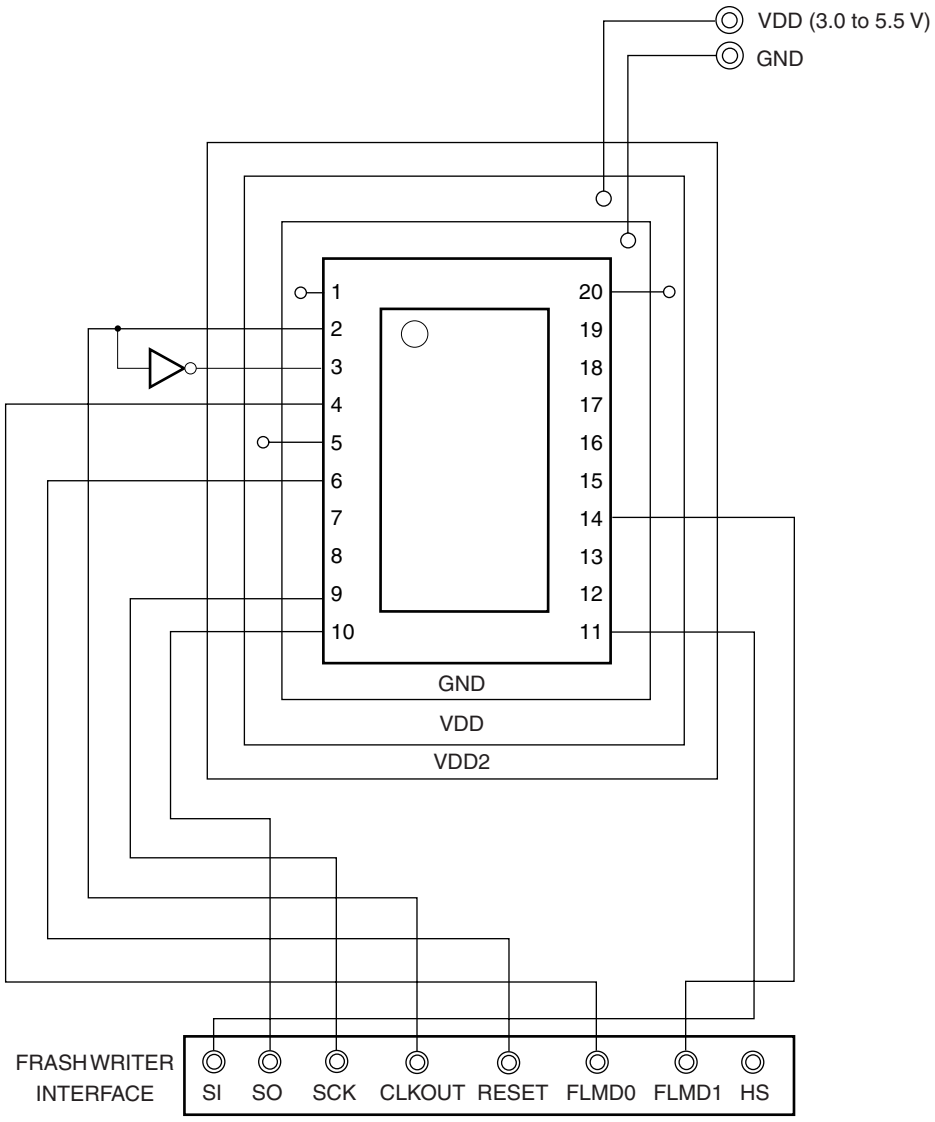


Figure 21-3. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10 + HS) Mode

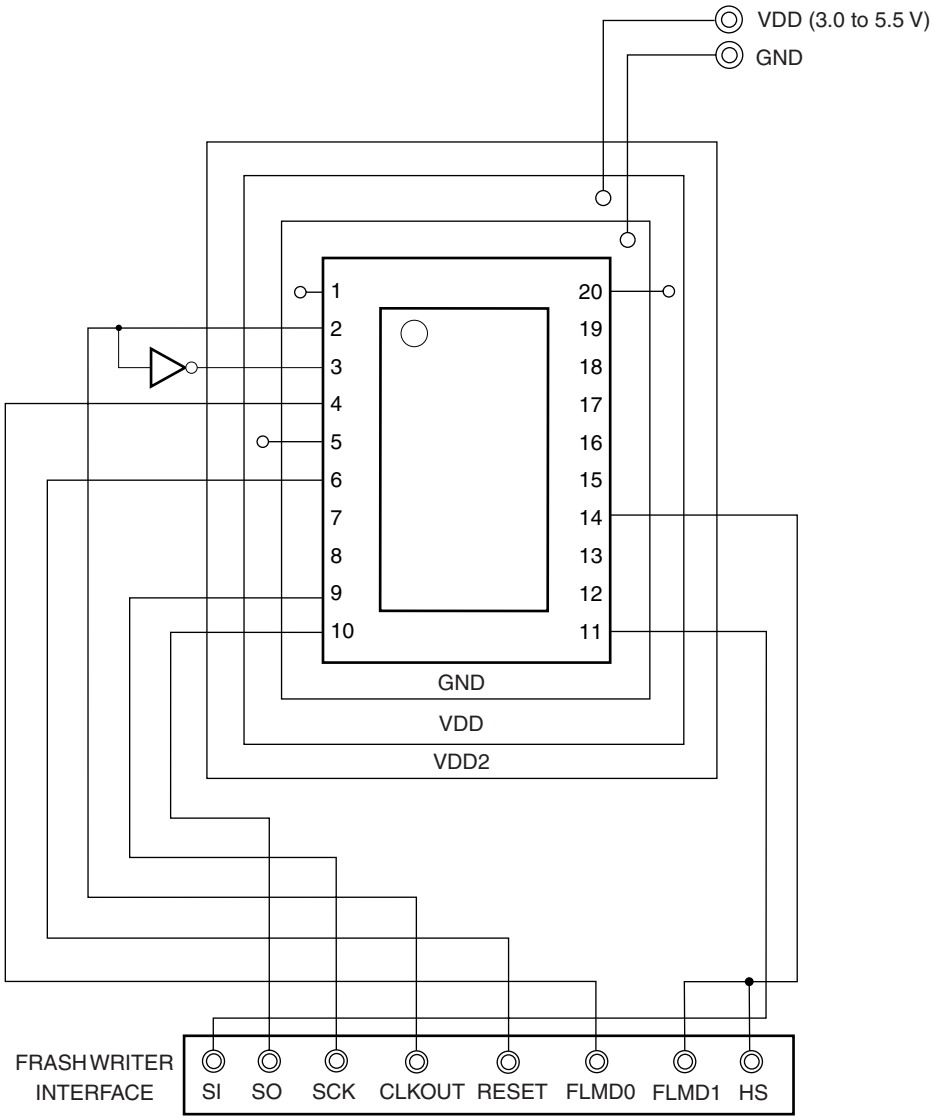
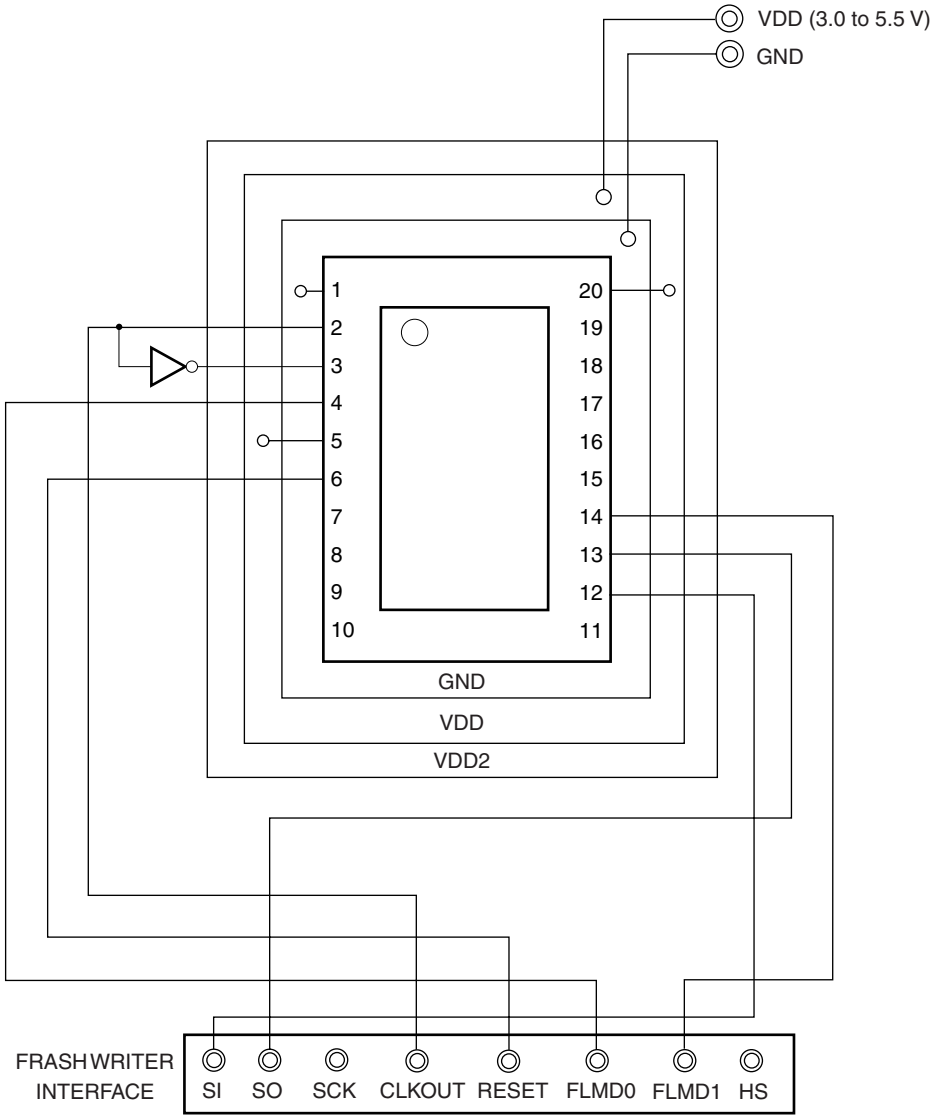


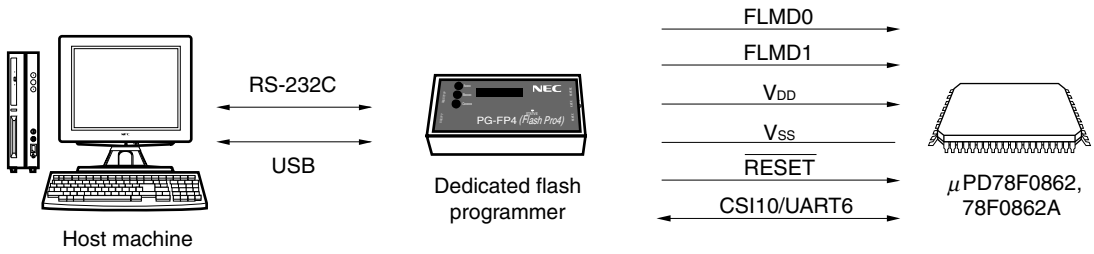
Figure 21-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART6) Mode



21.3 Programming Environment

The environment required for writing a program to the flash memory of the μPD78F0862 and 78F0862A illustrated below.

Figure 21-5. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash programmer is necessary.

CSI10 or UART6 is used for manipulation such as writing and erasing to interface between the dedicated flash programmer and the μPD78F0862, 78F0862A. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

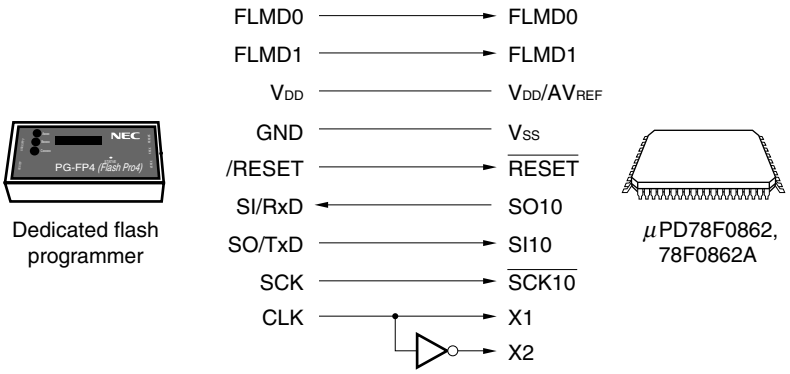
21.4 Communication Mode

Communication between the dedicated flash programmer and the μPD78F0862, 78F0862A are established by serial communication via CSI10 or UART6 of the μPD78F0862 and 78F0862A.

(1) CSI10

<R> Transfer rate: 2.4 kHz to 2.5 MHz

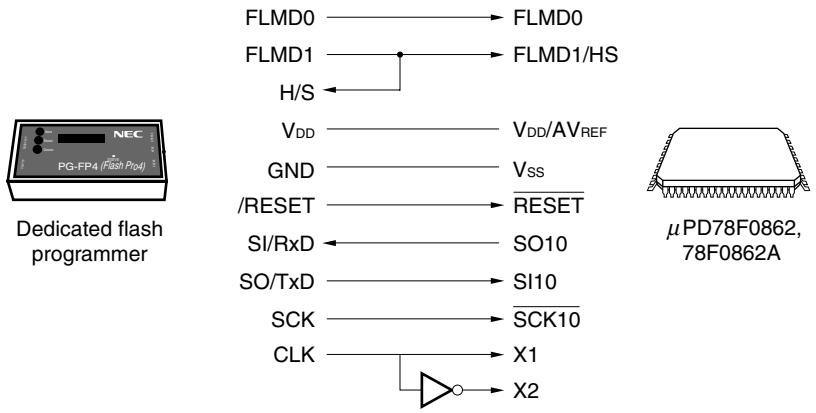
Figure 21-6. Communication with Dedicated Flash Programmer (CSI10)



(2) CSI communication mode supporting handshake

<R> Transfer rate: 2.4 kHz to 2.5 MHz

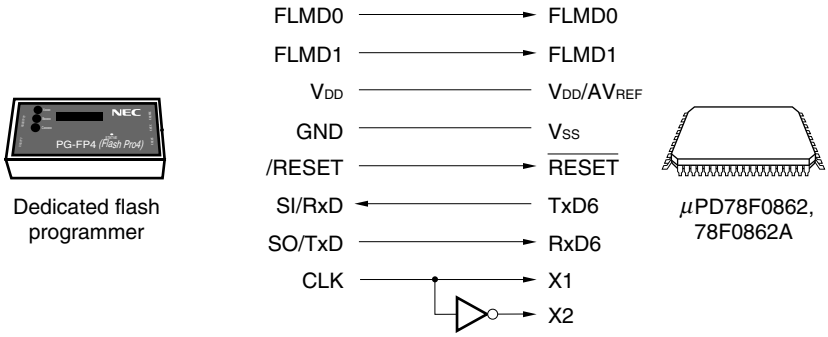
Figure 21-7. Communication with Dedicated Flash Programmer (CSI10 + HS)



(3) UART6

<R> Transfer rate: 9600, 19200, 31250, 38400, 76800 and 153600[Note] bps

Figure 21-8. Communication with Dedicated Flash Programmer (UART6)



<R> **Note** When peripheral hardware clock frequency is 2.5 MHz or less, 153600 bps cannot be selected.

If FlashPro4 is used as the dedicated flash programmer, FlashPro4 generates the following signal for the μPD78F0862 and 78F0862A. For details, refer to the FlashPro4 Manual.

Table 21-4. Pin Connection

FlashPro4			μPD78F0862, 78F0862A	Connection	
Signal Name	I/O	Pin Function	Pin Name	CSI10	UART6
FLMD0	Output	Mode signal	FLMD0	◎	◎
FLMD1	Output	Mode signal	FLMD1	◎	◎
V_{DD}	I/O	V_{DD} voltage generation	V_{DD}, AV_{REF}	◎	◎
GND	–	Ground	V_{SS}	◎	◎
CLK	Output	Clock output to μPD78F0862	X1, X2[Note]	○	○
/RESET	Output	Reset signal	$\overline{RESET}$	◎	◎
SI/RxD	Input	Receive signal	SO10/TxD6	◎	◎
SO/TxD	Output	Transmit signal	SI10/RxD6	◎	◎
SCK	Output	Transfer clock	$\overline{SCK10}$	◎	×
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

21.5 Handling of Pins on Board

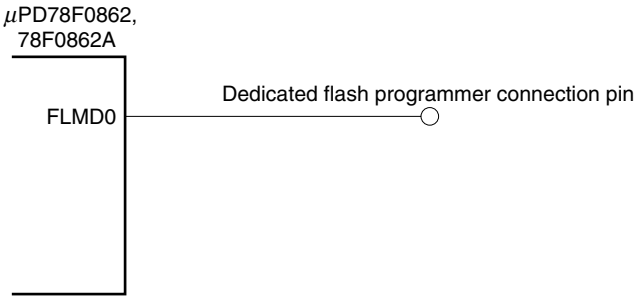
To write the flash memory on-board, connectors that connect the dedicated flash programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

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.

21.5.1 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. The following shows an example of the connection of the FLMD0 pin.

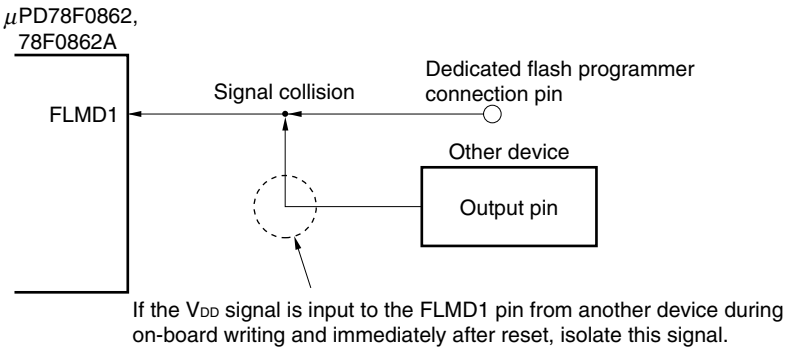
Figure 21-9. FLMD0 Pin Connection Example



21.5.2 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 21-10. FLMD1 Pin Connection Example



21.5.3 Serial interface pins

The pins used by each serial interface are listed below.

Table 21-5. Pins Used by Each Serial Interface

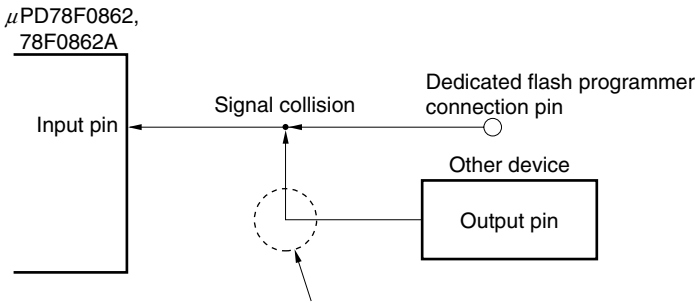
Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{SCK10}$
CSI10 + HS	SO10, SI10, $\overline{SCK10}$, HS
UART6	TxD6, RxD6

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 21-11. Signal Collision (Input Pin of Serial Interface)

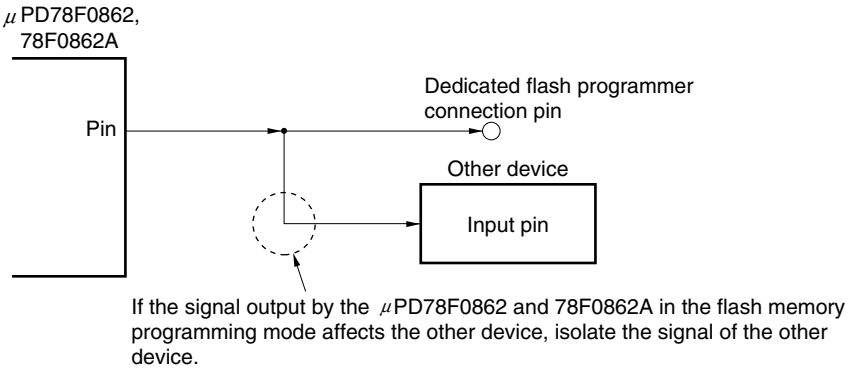


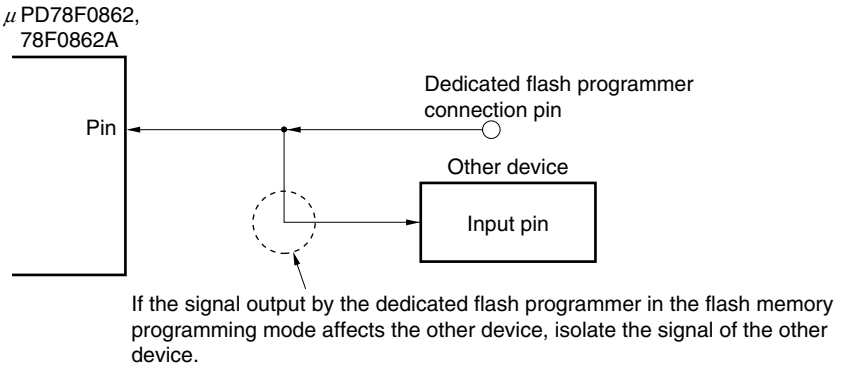
In the flash memory programming mode, the signal output by the device collides with the signal sent from the dedicated flash programmer. Therefore, isolate the signal of the other device.

(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 21-12. Malfunction of Other Device



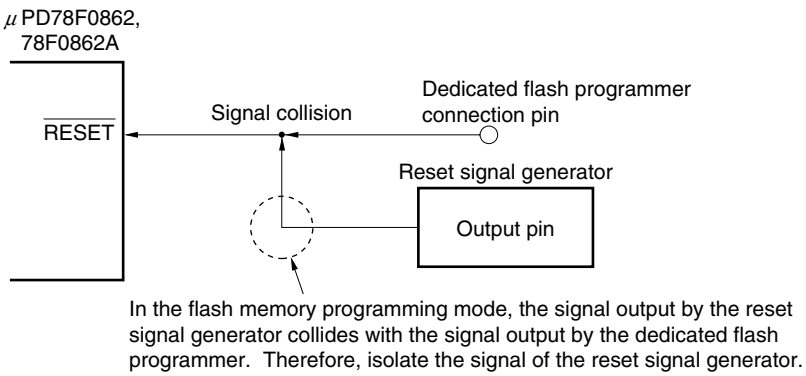


21.5.4 $\overline{\text{RESET}}$ pin

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 21-13. Signal Collision ($\overline{\text{RESET}}$ Pin)



21.5.5 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.

21.5.6 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.

21.5.7 Power supply

To use the power supply 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 power supply, connect in compliance with the normal operation mode.

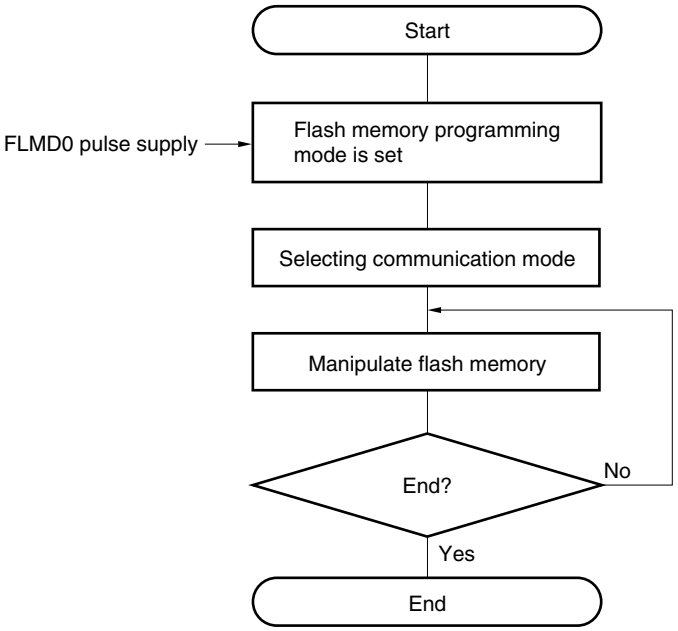
Supply the same other power supply (AV_{REF}) as those in the normal operation mode.

21.6 Programming Method

21.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 21-14. Flash Memory Manipulation Procedure



21.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash programmer, set the μPD78F0862 and 78F0862A 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 21-15. Flash Memory Programming Mode

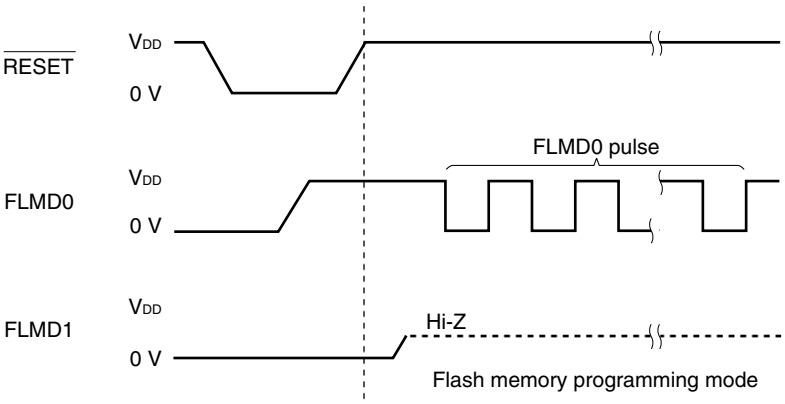


Table 21-6. Relationship of Operation Mode of FLMD0 and FLMD1 Pins

FLMD0	FLMD1	Operation Mode
0	×	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

21.6.3 Selecting communication mode

In the μPD78F0862 and 78F0862A, a communication mode is selected by inputting pulses (up to 11 pulses) to the FLMD0 pin after the flash memory programming mode is entered. These FLMD0 pulses are generated by the dedicated flash programmer.

The following table shows the relationship between the number of pulses and communication modes.

<R>

Table 21-7. 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	9600, 19200, 31250, 38400, 76800, and 153600 bps[Notes 3, 4]	Optional	2 M to 10 MHz[Note 2]	1.0	TxD6, RxD6	0
3-wire serial I/O (CSI10)	SIO-ch0	2.4 kHz to 2.5 MHz				SO10, SI10, $\overline{SCK10}$	8
3-wire serial I/O with handshake supported (CSI10 + HS)	SIO-H/S	2.4 kHz to 2.5 MHz				SO10, SI10, $\overline{SCK10}$, HS/P15	11

Notes 1. Selection items for Standard settings on FlashPro4.

2. The possible setting range differs depending on the voltage. For details, refer to the chapters of electrical specifications.

3. When peripheral hardware clock frequency is 2.5 MHz or less, 153600 bps cannot be selected.

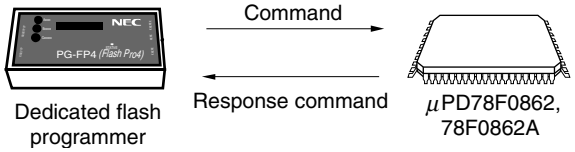
4. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

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.

21.6.4 Communication commands

The μPD78F0862 and 78F0862A communicate with the dedicated flash programmer by using commands. The signals sent from the flash programmer to the μPD78F0862 and 78F0862A are called commands, and the commands sent from the μPD78F0862 and 78F0862A to the dedicated flash programmer are called response commands.

Figure 21-16. Communication Commands



The flash memory control commands of the μPD78F0862 and 78F0862 are listed in the table below. All these commands are issued from the programmer and the μPD78F0862 and 78F0862A perform processing corresponding to the respective commands.

Table 21-8. 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 μPD78F0862 and 78F0862A return a response command for the command issued by the dedicated flash programmer. The response commands sent from the μPD78F0862 and 78F0862A are listed below.

Table 21-9. Response Commands

Command Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

CHAPTER 22 INSTRUCTION SET

This chapter lists each instruction set of the μPD780862 Subseries in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

22.1 Conventions Used in Operation List

22.1.1 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 22-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol[Note]
sfrp	Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note 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**.

22.1.2 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
( ): 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)
$\overline{\ \ }$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

22.1.3 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

22.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		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	Byte	Clock		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	Byte	Clock		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	Byte	Clock		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 (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		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, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1 time	×
	ROL	A, 1	1	2	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1 time	×
	RORC	A, 1	1	2	–	(CY ← A_0, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1 time	×
	ROLC	A, 1	1	2	–	(CY ← A_7, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1 time	×
	ROR4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{3-0}$, $(HL)_{7-4} \leftarrow A_{3-0}$, $(HL)_{3-0} \leftarrow (HL)_{7-4}$	
	ROL4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{7-4}$, $(HL)_{3-0} \leftarrow A_{3-0}$, $(HL)_{7-4} \leftarrow (HL)_{3-0}$	
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	Byte	Clock		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	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$	
	CALLF	!addr11	2	5	–	$(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP-2$	
	CALLT	[addr5]	1	6	–	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$	
	BRK		1	6	–	$(SP-1) \leftarrow PSW$, $(SP-2) \leftarrow (PC+1)_H$, $(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (003FH), $PC_L \leftarrow$ (003EH), $SP \leftarrow SP-3$, $IE \leftarrow 0$	
	RET		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
	RETI		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$,	R R R
	RETB		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$	R R R
Stack manipulate	**PUSH**	PSW	1	2	–	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$	
		rp	1	4	–	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$	
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R R R
		rp	1	4	–	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
	MOVW	SP, #word	4	–	10	$SP \leftarrow$ word	
		SP, AX	2	–	8	$SP \leftarrow AX$	
		AX, SP	2	–	8	$AX \leftarrow SP$	
Unconditional branch	**BR**	!addr16	3	6	–	$PC \leftarrow$ addr16	
		$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8	
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$	
Conditional branch	**BC**	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ 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	Byte	Clock		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.

22.3 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

First Operand \ Second 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

<R> Target products: μPD780861, 780862, 78F0862, 78F0862A, 780861(A), 780862(A), 78F0862(A), 78F0862A(A)

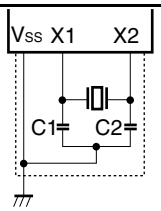
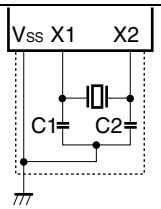
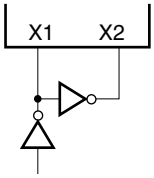
Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.3 to +6.5	V
	V_{SS}		−0.3 to +0.3	V
	AV_{REF}		−0.3 to V_{DD} + 0.3[Note]	V
Input voltage	V_{I1}	P00, P01, P10 to P15, P20 to P23, X1, X2, $\overline{RESET}$	−0.3 to V_{DD} + 0.3[Note]	V
Output voltage	V_O		−0.3 to V_{DD} + 0.3[Note]	V
Analog input voltage	V_{AN}		V_{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 P00, P01, P10 to P15, P130 pins	−30	mA
Output current, low	I_{OL}	Per pin	20	mA
		Total of P00, P01, P10 to P15, P130 pins	35	mA
Operating ambient temperature	T_A	In normal operation mode	−40 to +85	°C
		In flash memory programming	−40 to +85	
Storage temperature	T_{stg}	Mask ROM versions	−65 to +150	°C
		Flash memory versions	−40 to +150	

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.

Crystal/Ceramic Oscillator Characteristics (When Selecting Crystal/Ceramic Oscillation)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XH})Note	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	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	
Ceramic resonator		Oscillation frequency (f_{XH})Note	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	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_{XH})Note	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	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_{XH}, t_{XL})	4.0 V ≤ V_{DD} ≤ 5.5 V	46		250	ns
			3.3 V ≤ V_{DD} < 4.0 V	56		250	
			2.7 V ≤ V_{DD} < 3.3 V	96		250	

<R>
<R>
<R>

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

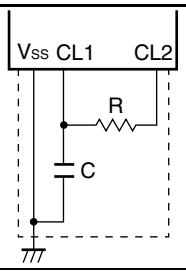
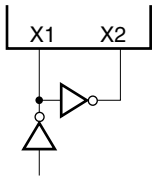
Caution When using the crystal/ceramic oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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.**

Remark For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

External RC Oscillator Characteristics (When Selecting External RC Oscillation)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
RC oscillation		Oscillation frequency (f_{XH})[Note]		3.0		4.0	MHz
External clock		X1 input frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	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	
<R>		X1 input high-/low-level width (t_{XH}, t_{XL})	4.0 V ≤ V_{DD} ≤ 5.5 V	46		250	ns
<R>			3.3 V ≤ V_{DD} < 4.0 V	56		250	
<R>			2.7 V ≤ V_{DD} < 3.3 V	96		250	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution When using the RC oscillator, wire as follows in the area enclosed by the broken lines in the above figure 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.**

External RC Oscillation Frequency Characteristics (When Selecting External RC Oscillation)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency (f_{XH})[Note]	R = 6.8 kΩ, C = 22 pF Target value: 3 MHz	2.5	3.0	3.5	MHz
	R = 4.7 kΩ, C = 22 pF Target value: 4 MHz	3.5	4.0	4.7	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution Set one of the above values to R and C.

Internal High-Speed Oscillator Characteristics (When Selecting Internal High-Speed Oscillation)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal high-speed oscillator	Oscillation frequency (f_{XH})[Note]		6.80	8.00	9.20	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Internal Low-Speed Oscillator Characteristics (T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal low-speed oscillator	Oscillation frequency (f_R)[Note]		120	240	480	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

DC Characteristics (T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V) (1/3)

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, P01, P10 to P15, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10	mA
Output current, low	I_{OL}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			10	mA
		Total of P00, P01, P10 to P15, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
			2.7 V ≤ V_{DD} < 4.0 V			10	mA
Input voltage, high	V_{IH1}	P02[Note 1], P12, P13, P15		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P10, P11, P14, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
	V_{IH3}	P20 to P23[Note 2]		0.7AV_{REF}		AV_{REF}	V
	V_{IH4}	X1, X2		V_{DD} – 0.5		V_{DD}	V
Input voltage, low	V_{IL1}	P02[Note 1], P12, P13, P15		0		0.3V_{DD}	V
	V_{IL2}	P00, P01, P10, P11, P14, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
	V_{IL3}	P20 to P23[Note 2]		0		0.3AV_{REF}	V
	V_{IL4}	X1, X2		0		0.4	V
Output voltage, high	V_{OH}	Total of P00, P01, P10 to P15, P130 pins 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	2.7 V ≤ V_{DD} < 4.0 V	V_{DD} – 0.5			V
Output voltage, low	V_{OL}	Total of P00, P01, P10 to P15, P130 pins 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
Input leakage current, high	I_{LIH1}	V_I = V_{DD}	P00, P01, P10 to P15, $\overline{\text{RESET}}$			3	μA
		V_I = AV_{REF}	P20 to P23			3	μA
	I_{LIH2}	V_I = V_{DD}	X1, X2[Note 3]			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	P00, P01, P10 to P15, P20 to P23, $\overline{\text{RESET}}$			–3	μA
	I_{LIL2}		X1, X2[Note 3]			–20	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	V_I = 0 V		10	30	100	kΩ
FLMD0 supply voltage (Flash memory versions only)	Flmd	In normal operation mode		0		0.2V_{DD}	V

<R> **Notes 1.** When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

2. When used as a digital input port, set AV_{REF} = V_{DD}.

3. When the inverse input level of X1 is input to X2.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/3): Flash Memory Versions

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Crystal/ ceramic oscillation operating mode[Notes 2, 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		7.8	15.4	mA
				When A/D converter is operating[Note 4]		8.8	17.4	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		2.4	5.1	mA
				When A/D converter is operating[Note 4]		3.0	6.3	mA
	I_{DD2}	Crystal/ ceramic oscillation HALT mode[Note 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.7	3.8	mA
				When peripheral functions are operating			6.7	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.48	1.0	mA
				When peripheral functions are operating			2.1	mA
	I_{DD3}	External RC oscillation operating mode[Notes 2, 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		4.5	9.5	mA
				When A/D converter is operating[Note 4]		5.5	11.5	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When A/D converter is stopped		2.4	5.1	mA
				When A/D converter is operating[Note 4]		3.0	6.3	mA
	I_{DD4}	External RC oscillation HALT mode[Note 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	3.5	mA
				When peripheral functions are operating			5.3	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.87	2.0	mA
				When peripheral functions are operating			3.0	mA
	I_{DD5}	Internal high-speed oscillation operating mode[Notes 2, 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		6.9	14.4	mA
				When A/D converter is operating[Note 4]		7.9	16.4	mA
	I_{DD6}	Internal high-speed oscillation HALT mode[Note 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.4	3.2	mA
				When peripheral functions are operating			5.9	mA
	I_{DD7}	Internal low-speed oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			1.8	7.2	mA
			V_{DD} = 3.0 V ±10%			0.88	3.5	mA
	I_{DD8}	Internal low-speed oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%			0.08	0.32	mA
			V_{DD} = 3.0 V ±10%			0.06	0.24	mA
	I_{DD9}	STOP mode	V_{DD} = 5.0 V ±10%	Internal low-speed oscillation: OFF		3.5	35.5	μA
				Internal low-speed oscillation: ON		17.5	63.5	μA
			V_{DD} = 3.0 V ±10%	Internal low-speed oscillation OFF		3.5	15.5	μA
				Internal low-speed oscillation: ON		11.0	30.5	μA

Notes 1. 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).

2. Peripheral operation current is included.

3. When PCC = 00H.

4. Total of the current that flows through the V_{DD} pin and AV_{REF} pin.

5. When high-speed system clock is stopped.

6. When crystal/ceramic oscillation is selected as the high-speed system clock using an option byte.

7. When an external RC is selected as the high-speed system clock using an option byte.

8. When an internal high-speed oscillation is selected as the high-speed system clock using an option byte.

DC Characteristics (3/3): Mask ROM Versions

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Crystal/ ceramic oscillation operating mode[Notes 2, 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		6.1	11.9	mA
				When A/D converter is operating[Note 4]		7.1	13.9	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		1.7	3.6	mA
				When A/D converter is operating[Note 4]		2.3	4.8	mA
	I_{DD2}	Crystal/ ceramic oscillation HALT mode[Note 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	3.6	mA
				When peripheral functions are operating			6.5	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.41	0.96	mA
				When peripheral functions are operating			2.1	mA
	I_{DD3}	External RC oscillation operating mode[Notes 2, 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		3.2	6.4	mA
				When A/D converter is operating[Note 4]		4.2	8.4	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When A/D converter is stopped		1.7	3.6	mA
				When A/D converter is operating[Note 4]		2.3	4.8	mA
	I_{DD4}	External RC oscillation HALT mode[Note 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	3.5	mA
				When peripheral functions are operating			5.3	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.87	2.0	mA
				When peripheral functions are operating			3.0	mA
	I_{DD5}	Internal high-speed oscillation operating mode[Notes 2, 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		4.98	10.1	mA
				When A/D converter is operating[Note 4]		5.98	12.1	mA
	I_{DD6}	Internal high-speed oscillation HALT mode[Note 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.24	2.8	mA
				When peripheral functions are operating			5.5	mA
	I_{DD7}	Internal low-speed oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			0.17	0.68	mA
			V_{DD} = 3.0 V ±10%			0.11	0.44	mA
	I_{DD8}	Internal low-speed oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%			0.04	0.16	mA
			V_{DD} = 3.0 V ±10%			0.03	0.12	mA
	I_{DD9}	STOP mode	V_{DD} = 5.0 V ±10%	Internal low-speed oscillator: OFF		3.5	35.5	μA
				Internal low-speed oscillator : ON		17.5	63.5	μA
			V_{DD} = 3.0 V ±10%	Internal low-speed oscillator: OFF		3.5	15.5	μA
				Internal low-speed oscillator: ON		11.0	30.5	μA

Notes 1. 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).

2. Peripheral operation current is included.

3. When PCC = 00H.

4. Total of the current that flows through the V_{DD} pin and AV_{REF} pin.

5. When high-speed system clock is stopped.

6. When crystal/ceramic oscillation is selected as the high-speed system clock using mask option.

7. When an external RC is selected as the high-speed system clock using mask option.

8. When an internal high-speed oscillation is selected as the high-speed system clock using mask option.

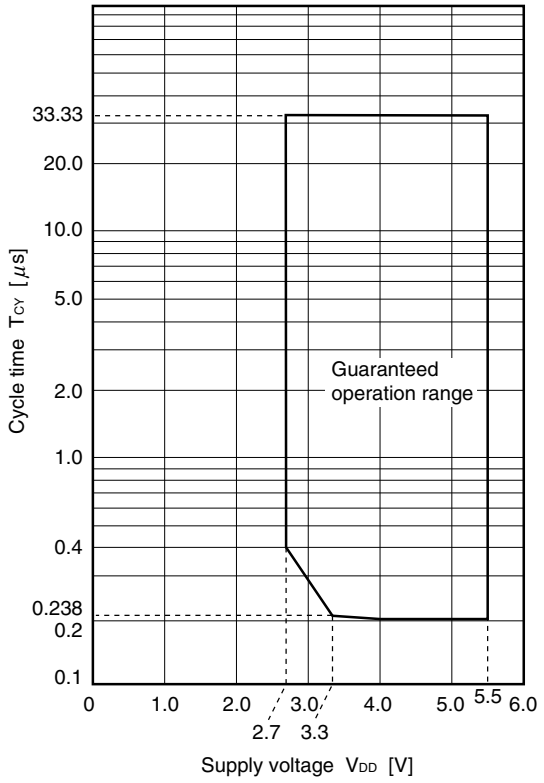
AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{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 ≤ V_{DD} ≤ 5.5 V	0.2		16	μs
					3.3 V ≤ V_{DD} < 4.0 V	0.238		16	μs
					2.7 V ≤ V_{DD} < 3.3 V	0.4		16	μs
				External RC oscillation clock	2.7 V ≤ V_{DD} ≤ 5.5 V	0.426		12.8	μs
				Internal high-speed oscillation clock	4.0 V ≤ V_{DD} ≤ 5.5 V	0.217	0.25	4.7	μs
			Internal low-speed oscillation clock		2.7 V ≤ V_{DD} ≤ 5.5 V	4.17	8.33	33.33	μs
TI00 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} ≤ 5.5 V				$2/f_{sam}$ + 0.1[Note]			μs
		2.7 V ≤ V_{DD} < 4.0 V				$2/f_{sam}$ + 0.2[Note]			μs
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}					1			μs
$\overline{RESET}$ low-level width	t_{RSL}					10			μs

Note Selection of f_{sam} = f_{XH}, $f_{XH}/4$, or $f_{XH}/256$ 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_{XH}.

T_{CY} vs. V_{DD} (Main System Clock Operation)



(2) Serial interface (T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{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) 3-wire serial I/O mode ($\overline{SCK10}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ 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
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}		t_{KCY1}/2 – 10			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK1}		30			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO1}	C = 100 pF[Note]			30	ns

Note C is the load capacitance of the $\overline{SCK10}$ and SO10 output lines.

(c) 3-wire serial I/O mode ($\overline{SCK10}$... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}		400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}		t_{KCY2}/2			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK2}		80			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO2}	C = 100 pF[Note]			120	ns

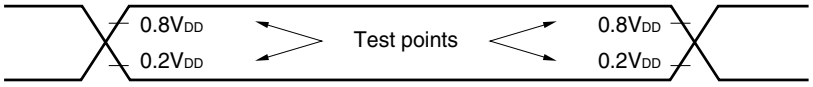
Note C is the load capacitance of the SO10 output line.

(3) Manchester code generator (T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

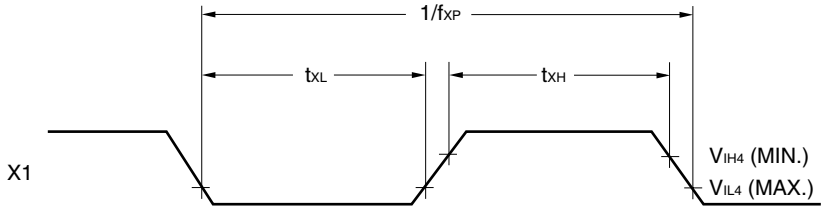
(a) Dedicated baud rate generator output

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					250.0	kbps

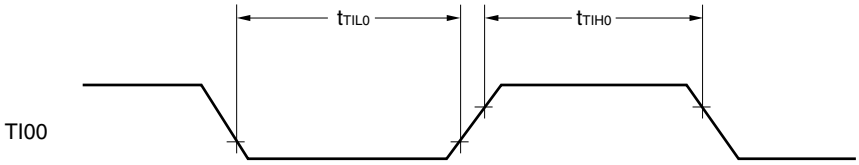
AC Timing Test Points (Excluding X1)



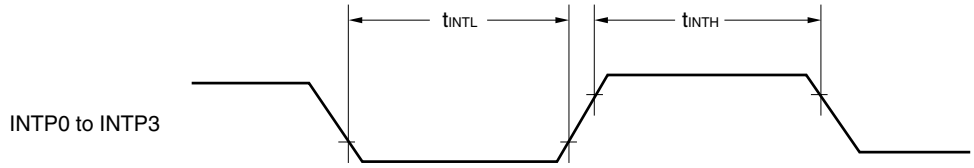
Clock Timing



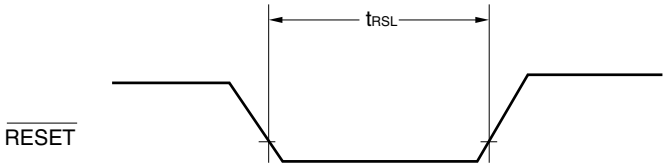
TI Timing



Interrupt Request Input Timing

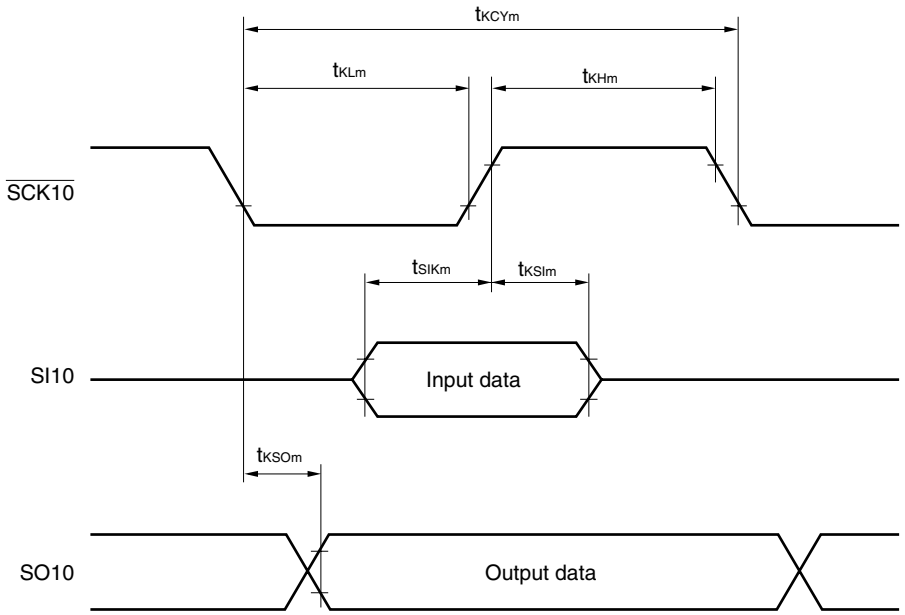


$\overline{\text{RESET}}$ Input Timing



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} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V[Note 1])

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Notes 2, 3]		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 2, 3]		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 2, 3]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Integral linearity error[Note 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
Differential linearity error[Note 2]		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}		V_{SS}[Note 1]		AV_{REF}	V

Notes 1. V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. The above specifications are for when only the A/D converter is operating.

2. Excludes quantization error (±1/2 LSB).

3. This value is indicated as a ratio (%FSR) to the full-scale value.

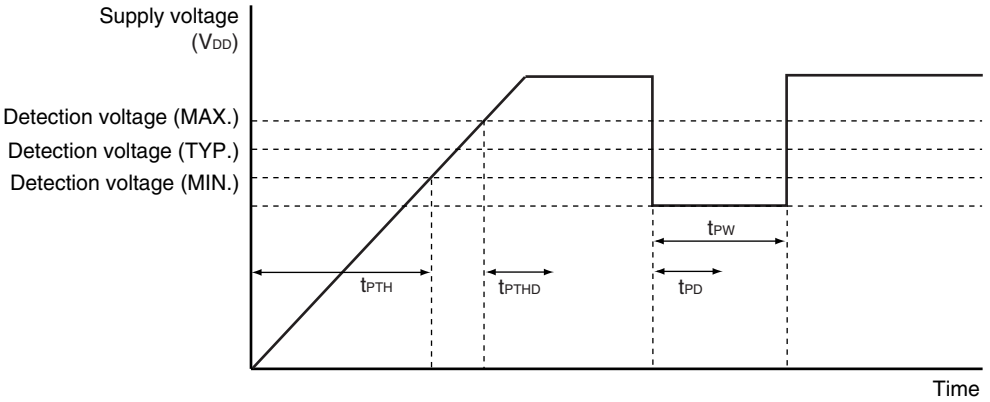
POC Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		2.7	2.85	3.0	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → 2.7 V	0.0015			ms
Response delay time 1[Note 1]	t_{PTHD}	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2[Note 2]	t_{PD}	When V_{DD} falls			1.0	ms
Minimum pulse width	t_{PW}		0.2			ms

Notes 1. Time required from voltage detection to reset release.

2. Time required from voltage detection to internal reset output.

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
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

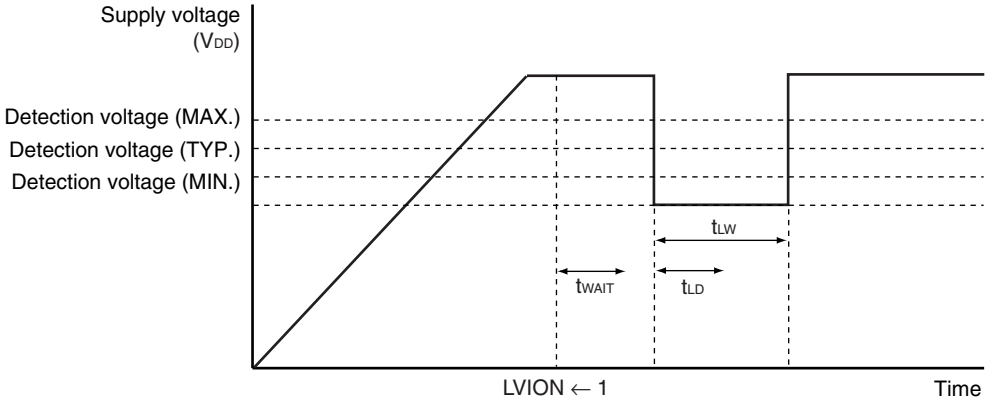
Notes 1. Time required from voltage detection to interrupt output or reset output.

2. Time required from setting LVION to 1 to operation stabilization.

Remarks 1. $V_{LVI0} > V_{LVI1} > V_{LVI2} > V_{LVI3} > V_{LVI4} > V_{LVI5} > V_{LVI6}$

2. $V_{POC} < V_{LVIm}$ (m = 0 to 6)

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
<R> Data retention supply voltage	V_{DDDR}		2.7		5.5	V
Release signal set time	t_{SREL}		0			μs

Flash Memory Programming Characteristics: Flash Memory Versions

(T_A = 10 to 65°C, 3.0 V ≤ V_{DD} ≤ 5.5 V, 3.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

(1) μPD78F0862, 78F0862 (A)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> V_{DD} supply current		I_{DD}	f_X = 10 MHz, V_{DD} = 5.5 V		20	45	mA
Step erase time	Chip unit	T_{erac}		100			ms
	Sector unit	T_{eras}		100			ms
Erase time[Note 1]	Chip unit	T_{eraca}				25.5	s
	Sector unit	T_{erasa}				25.5	s
Step write time		T_{wrw}		50			μs
Write time		T_{wrwa}				500	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 2]			100[Note 3]	Times

Notes 1. The prewrite time before erasure and the erase verify time (writeback time) are not included.

2. When a product is first written after shipment, "erase → write" and "write only" are both taken as one rewrite.

<R> **3.** μPD78F0862(A): 10 times (MAX.)

<R> **(2) μPD78F0862A, 78F0862A (A)**

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}	f_X = 10 MHz, V_{DD} = 5.5 V			30.5	mA
Step erase time	Chip unit	T_{erac}			10		ms
	Sector unit	T_{eras}			10		ms
Erase time[Note 1]	Chip unit	T_{eraca}				2.55	s
	Sector unit	T_{erasa}				2.55	s
Step write time		T_{wrw}				500	μs
Write time		T_{wrwa}				500	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 2]			100	Times

Notes 1. The prewrite time before erasure and the erase verify time (writeback time) are not included.

2. When a product is first written after shipment, "erase → write" and "write only" are both taken as one rewrite.

CHAPTER 24 ELECTRICAL SPECIFICATIONS ((A1) GRADE PRODUCTS)

<R> Target products: μPD780861(A1), 780862(A1), 78F0862A(A1)

Absolute Maximum Ratings (T_A = 25°C)

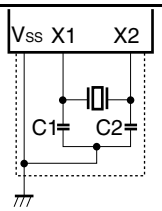
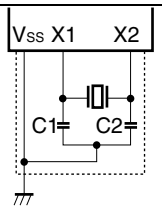
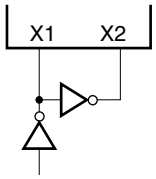
Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.3 to +6.5	V
	V_{SS}		−0.3 to +0.3	V
	AV_{REF}		−0.3 to V_{DD} + 0.3[Note]	V
Input voltage	V_{I1}	P00, P01, P10 to P15, P20 to P23, X1, X2, $\overline{RESET}$	−0.3 to V_{DD} + 0.3[Note]	V
Output voltage	V_O		−0.3 to V_{DD} + 0.3[Note]	V
Analog input voltage	V_{AN}		V_{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	−8	mA
		Total of P00, P01, P10 to P15, P130 pins	−24	mA
Output current, low	I_{OL}	Per pin	16	mA
		Total of P00, P01, P10 to P15, P130 pins	28	mA
Operating ambient temperature	T_A	In normal operation mode	−40 to +110	°C
<R>		In flash memory programming mode	−40 to +85	
Storage temperature	T_{stg}	Mask ROM versions	−65 to +150	°C
<R>		Flash memory version	−40 to +150	

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.

Crystal/Ceramic Oscillator Characteristics (When Selecting Crystal/Ceramic Oscillation)
(T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
Ceramic resonator		Oscillation frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
External clock		X1 input frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
<R>		X1 input high-/low-level width (t_{XH}, t_{XL})	4.0 V ≤ V_{DD} ≤ 5.5 V	46		250	ns
<R>			2.7 V ≤ V_{DD} < 4.0 V	96		250	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

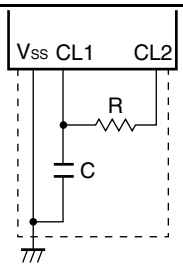
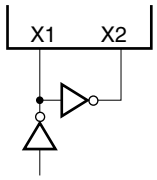
Caution When using the crystal/ceramic oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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.**

Remark For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

External RC Oscillator Characteristics (When Selecting External RC Oscillation)

(T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
RC oscillation		Oscillation frequency (f_{XH})Note		3.0		4.0	MHz
External clock		X1 input frequency (f_{XH})Note	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		10	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
<R>		X1 input high-/low-level width (t_{XH}, t_{XL})	4.0 V ≤ V_{DD} ≤ 5.5 V	46		250	ns
<R>			2.7 V ≤ V_{DD} < 4.0 V	96		250	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution When using the RC oscillator, wire as follows in the area enclosed by the broken lines in the above figure 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.**

External RC Oscillation Frequency Characteristics (When Selecting External RC Oscillation)

(T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency (f_{XH})Note	R = 6.8 kΩ, C = 22 pF Target value: 3 MHz	2.5	3.0	3.5	MHz
	R = 4.7 kΩ, C = 22 pF Target value: 4 MHz	3.5	4.0	4.7	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution Set one of the above values to R and C.

Internal High-Speed Oscillator Characteristics (When Selecting Internal High-Speed Oscillation)

(T_A = –40 to +110°C, 4.0 V ≤ V_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal high-speed oscillator	Oscillation frequency (f_{XH})Note		6.80	8.00	9.20	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Internal Low-Speed Oscillator Characteristics (T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal low-speed oscillator	Oscillation frequency (f_R)[Note]		120	240	490	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

DC Characteristics (T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V) (1/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			–4	mA
		Total of P00, P01, P10 to P15, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			–20	mA
			2.7 V ≤ V_{DD} < 4.0 V			–8	mA
Output current, low	I_{OL}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			8	mA
		Total of P00, P01, P10 to P15, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			24	mA
			2.7 V ≤ V_{DD} < 4.0 V			8	mA
Input voltage, high	V_{IH1}	P02[Note 1], P12, P13, P15		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P10, P11, P14, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
	V_{IH3}	P20 to P23[Note 2]		0.7AV_{REF}		AV_{REF}	V
	V_{IH4}	X1, X2		V_{DD} – 0.5		V_{DD}	V
Input voltage, low	V_{IL1}	P02[Note 1], P12, P13, P15		0		0.3V_{DD}	V
	V_{IL2}	P00, P01, P10, P11, P14, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
	V_{IL3}	P20 to P23[Note 2]		0		0.3AV_{REF}	V
	V_{IL4}	X1, X2		0		0.4	V
Output voltage, high	V_{OH}	Total of P00, P01, P10 to P15, P130 pins I_{OH} = –20 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –4 mA	V_{DD} – 1.0			V
		I_{OH} = –100 μA	2.7 V ≤ V_{DD} < 4.0 V	V_{DD} – 0.5			V
Output voltage, low	V_{OL}	Total of P00, P01, P10 to P15, P130 pins I_{OL} = 24 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 8 mA			1.3	V
		I_{OL} = 400 μA	2.7 V ≤ V_{DD} < 4.0 V			0.4	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD}	P00, P01, P10 to P15, $\overline{\text{RESET}}$			10	μA
		V_I = AV_{REF}	P20 to P23			10	μA
	I_{LIH2}	V_I = V_{DD}	X1, X2[Note 3]			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	P00, P01, P10 to P15, P20 to P23, $\overline{\text{RESET}}$			–10	μA
	I_{LIL2}		X1, X2[Note 3]			–20	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD}				10	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–10	μA
Pull-up resistance value	R	V_I = 0 V		10	30	120	kΩ
<R> FLMD0 supply voltage (Flash memory version only)	Flmd	In normal operation mode		0		0.2V_{DD}	V

<R> **Notes 1.** When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

2. When used as a digital input port, set AV_{REF} = V_{DD}.

3. When the inverse input level of X1 is input to X2.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

<R> **DC Characteristics (2/3): Flash Memory Version**

(T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Crystal/ ceramic oscillation operating mode[Notes 2, 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		7.8	16.2	mA
				When A/D converter is operating[Note 4]		8.8	18.2	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		2.4	5.5	mA
				When A/D converter is operating[Note 4]		3.0	6.7	mA
	I_{DD2}	Crystal/ ceramic oscillation HALT mode[Note 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.7	4.6	mA
				When peripheral functions are operating			7.5	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.48	1.4	mA
				When peripheral functions are operating			2.5	mA
	I_{DD3}	External RC oscillation operating mode[Notes 2, 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		4.5	10.3	mA
				When A/D converter is operating[Note 4]		5.5	12.3	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When A/D converter is stopped		2.4	5.5	mA
				When A/D converter is operating[Note 4]		3.0	6.7	mA
	I_{DD4}	External RC oscillation HALT mode[Note 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	4.3	mA
				When peripheral functions are operating			6.1	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.87	2.4	mA
				When peripheral functions are operating			3.4	mA
	I_{DD5}	Internal high-speed oscillation operating mode[Notes 2, 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		6.9	15.2	mA
				When A/D converter is operating[Note 4]		7.9	17.2	mA
	I_{DD6}	Internal high-speed oscillation HALT mode[Note 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.4	4.0	mA
				When peripheral functions are operating			6.7	mA
	I_{DD7}	Internal low-speed oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			1.8	8.0	mA
			V_{DD} = 3.0 V ±10%			0.88	3.9	mA
	I_{DD8}	Internal low-speed oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%			0.08	1.12	mA
			V_{DD} = 3.0 V ±10%			0.06	0.64	mA
	I_{DD9}	STOP mode	V_{DD} = 5.0 V ±10%	Internal low-speed oscillation: OFF		3.5	800	μA
				Internal low-speed oscillation: ON		17.5	900	μA
			V_{DD} = 3.0 V ±10%	Internal low-speed oscillation OFF		3.5	400	μA
				Internal low-speed oscillation: ON		11.0	500	μA

Notes 1. 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).

2. Peripheral operation current is included.

3. When PCC = 00H.

4. Total of the current that flows through the V_{DD} pin and AV_{REF} pin.

5. When high-speed system clock is stopped.

6. When crystal/ceramic oscillation is selected as the high-speed system clock using an option byte.

7. When an external RC is selected as the high-speed system clock using an option byte.

8. When an internal high-speed oscillation is selected as the high-speed system clock using an option byte.

DC Characteristics (3/3): Mask ROM Versions

(T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Crystal/ ceramic oscillation operating mode[Notes 2, 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		6.1	12.7	mA
				When A/D converter is operating[Note 4]		7.1	14.7	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		1.7	4.0	mA
				When A/D converter is operating[Note 4]		2.3	5.2	mA
	I_{DD2}	Crystal/ ceramic oscillation HALT mode[Note 6]	f_{XH} = 10 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	4.4	mA
				When peripheral functions are operating			7.3	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.41	1.36	mA
				When peripheral functions are operating			2.5	mA
	I_{DD3}	External RC oscillation operating mode[Notes 2, 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		3.2	7.2	mA
				When A/D converter is operating[Note 4]		4.2	9.2	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When A/D converter is stopped		1.7	4.0	mA
				When A/D converter is operating[Note 4]		2.3	5.2	mA
	I_{DD4}	External RC oscillation HALT mode[Note 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	4.3	mA
				When peripheral functions are operating			6.1	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.87	2.4	mA
				When peripheral functions are operating			3.4	mA
	I_{DD5}	Internal high-speed oscillation operating mode[Notes 2, 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		4.98	10.9	mA
				When A/D converter is operating[Note 4]		5.98	12.9	mA
	I_{DD6}	Internal high-speed oscillation HALT mode[Note 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.24	3.6	mA
				When peripheral functions are operating			6.3	mA
	I_{DD7}	Internal low-speed oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			0.17	1.48	mA
			V_{DD} = 3.0 V ±10%			0.11	0.84	mA
	I_{DD8}	Internal low-speed oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%			0.04	0.96	mA
			V_{DD} = 3.0 V ±10%			0.03	0.52	mA
	I_{DD9}	STOP mode	V_{DD} = 5.0 V ±10%	Internal low-speed oscillation: OFF		3.5	800	μA
				Internal low-speed oscillation: ON		17.5	900	μA
			V_{DD} = 3.0 V ±10%	Internal low-speed oscillation OFF		3.5	400	μA
				Internal low-speed oscillation: ON		11.0	500	μA

Notes 1. 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).

2. Peripheral operation current is included.

3. When PCC = 00H.

4. Total of the current that flows through the V_{DD} pin and AV_{REF} pin.

5. When high-speed system clock is stopped.

6. When crystal/ceramic oscillation is selected as the high-speed system clock using mask option.

7. When an external RC is selected as the high-speed system clock using mask option.

8. When an internal high-speed oscillation is selected as the high-speed system clock using mask option.

AC Characteristics

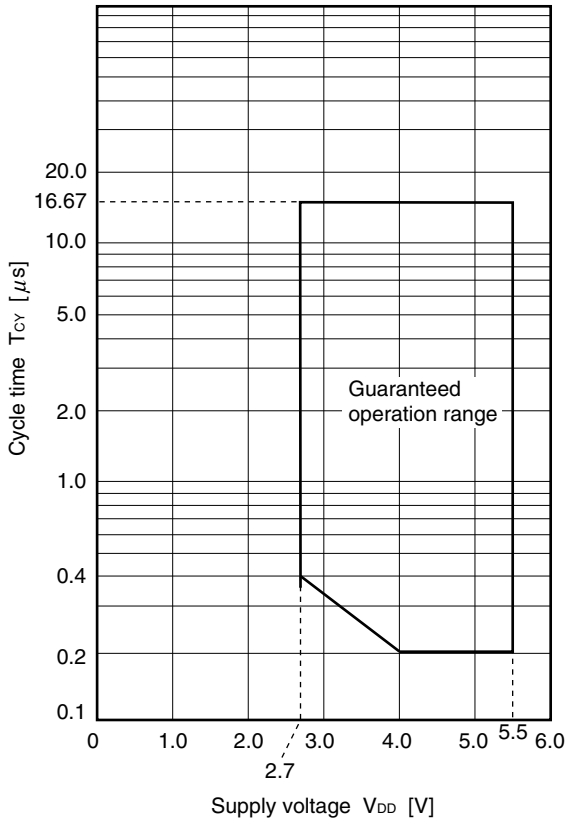
(1) Basic operation (T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

<R>

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 ≤ V_{DD} ≤ 5.5 V	0.2		16	μs
					2.7 V ≤ V_{DD} < 4.0 V	0.4		16	μs
				External RC oscillation clock	2.7 V ≤ V_{DD} ≤ 5.5 V	0.426		12.8	μs
				Internal high-speed oscillation clock	4.0 V ≤ V_{DD} ≤ 5.5 V	0.217	0.25	4.7	μs
			Internal low-speed oscillation clock		2.7 V ≤ V_{DD} ≤ 5.5 V	4.09	8.33	16.67	μs
TI00 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} ≤ 5.5 V				$2/f_{sam}$ + 0.1[Note]			μs
		2.7 V ≤ V_{DD} < 4.0 V				$2/f_{sam}$ + 0.2[Note]			μs
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}					1			μs
$\overline{RESET}$ low-level width	t_{RSL}					10			μs

Note Selection of f_{sam} = f_{XH}, $f_{XH}/4$, or $f_{XH}/256$ 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_{XH}.

T_{CY} vs. V_{DD} (Main System Clock Operation)



(2) Serial interface (T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{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) 3-wire serial I/O mode ($\overline{SCK10}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.0 V	400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}		t_{KCY1}/2 – 10			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK1}		30			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO1}	C = 100 pF[Note]			30	ns

Note C is the load capacitance of the $\overline{SCK10}$ and SO10 output lines.

(c) 3-wire serial I/O mode ($\overline{SCK10}$... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}		400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}		t_{KCY2}/2			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK2}		80			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO2}	C = 100 pF[Note]			120	ns

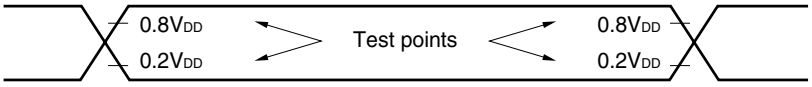
Note C is the load capacitance of the SO10 output line.

(3) Manchester code generator (T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

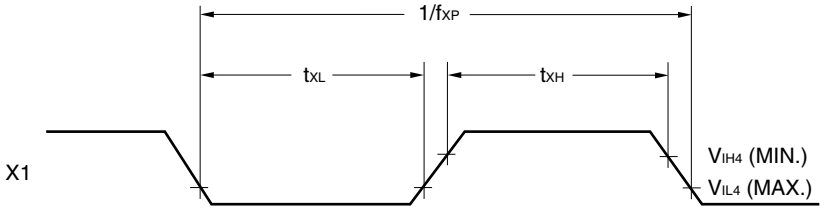
(a) Dedicated baud rate generator output

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					250.0	kbps

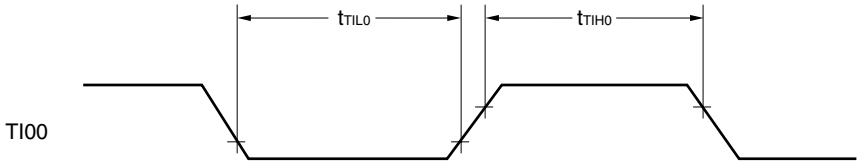
AC Timing Test Points (Excluding X1)



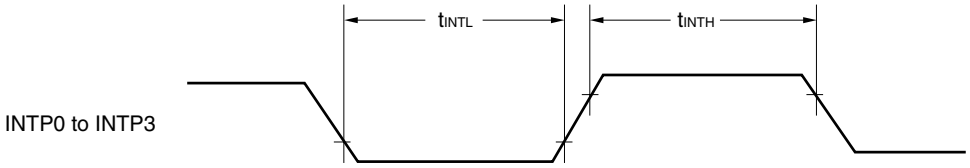
Clock Timing



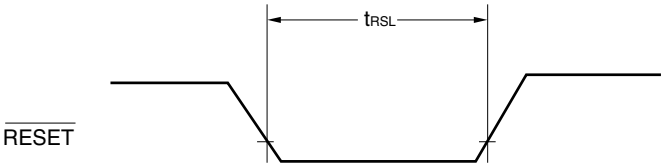
TI Timing



Interrupt Request Input Timing

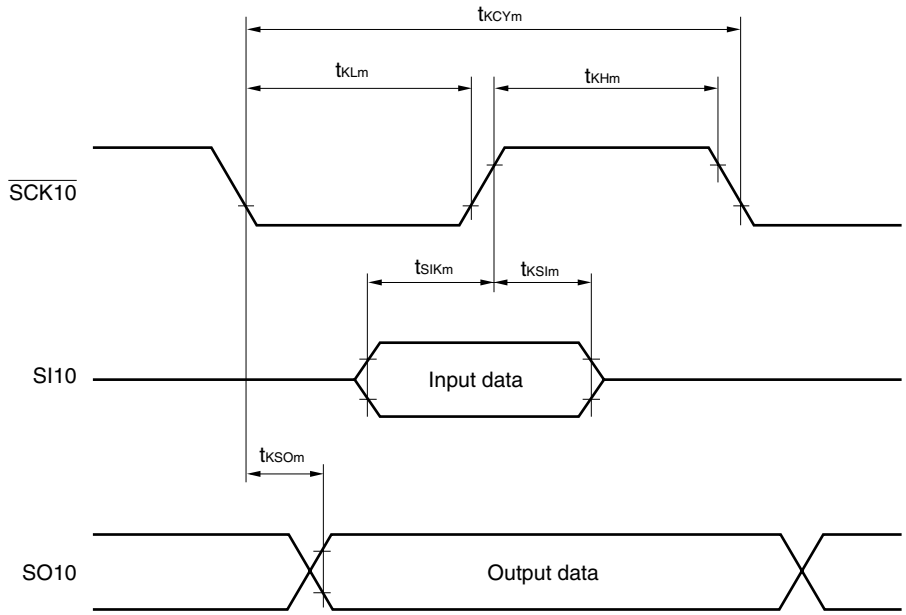


$\overline{\text{RESET}}$ Input Timing



Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

A/D Converter Characteristics (T_A = –40 to +110°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V[Note 1])

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Notes 2, 3]		4.0 V ≤ AV_{REF} ≤ 5.5 V		±0.2	±0.6	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V		±0.3	±0.8	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	14		60	μs
		2.7 V ≤ AV_{REF} < 4.0 V	19		60	μs
Zero-scale error[Notes 2, 3]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.8	%FSR
Full-scale error[Notes 2, 3]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.8	%FSR
Integral linearity error[Note 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±4.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±6.5	LSB
Differential linearity error[Note 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.0	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.5	LSB
Analog input voltage	V_{AIN}		V_{SS}[Note 1]		AV_{REF}	V

Notes 1. V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. The above specifications are for when only the A/D converter is operating.

2. Excludes quantization error (±1/2 LSB).

3. This value is indicated as a ratio (%FSR) to the full-scale value.

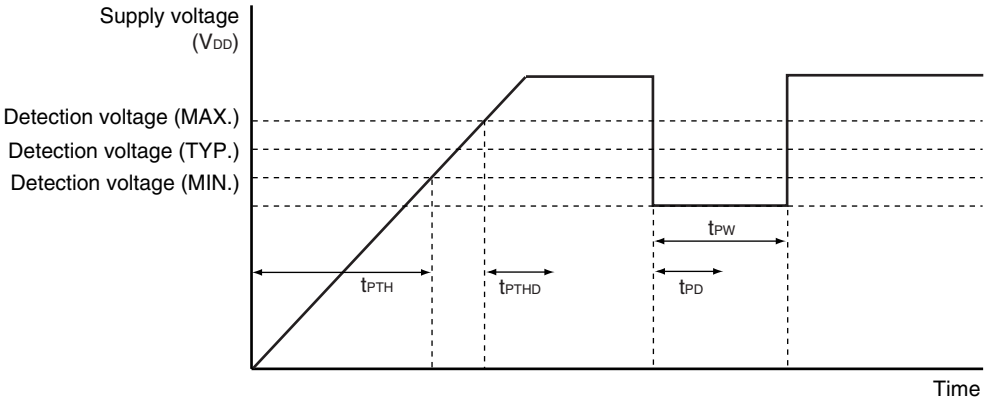
POC Circuit Characteristics (T_A = –40 to +110°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		2.7	2.85	3.02	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → 2.7 V	0.0015			ms
Response delay time 1[Note 1]	t_{PTHD}	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2[Note 2]	t_{PD}	When V_{DD} falls			1.0	ms
Minimum pulse width	t_{PW}		0.2			ms

Notes 1. Time required from voltage detection to reset release.

2. Time required from voltage detection to internal reset output.

POC Circuit Timing



LVI Circuit Characteristics (T_A = –40 to +110°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		4.1	4.3	4.52	V
	V_{LVI1}		3.9	4.1	4.32	V
	V_{LVI2}		3.7	3.9	4.12	V
	V_{LVI3}		3.5	3.7	3.92	V
	V_{LVI4}		3.3	3.5	3.72	V
	V_{LVI5}		3.15	3.3	3.47	V
	V_{LVI6}		2.95	3.1	3.27	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

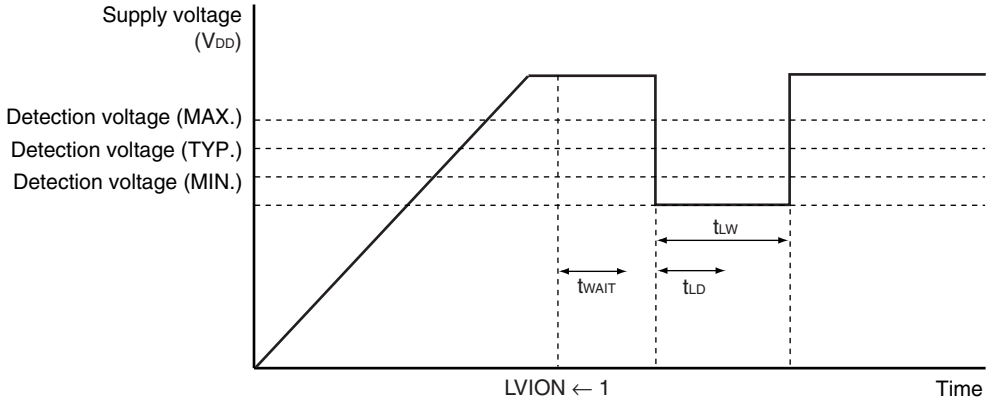
Notes 1. Time required from voltage detection to interrupt output or reset output.

2. Time required from setting LVION to 1 to operation stabilization.

Remarks 1. $V_{LVI0} > V_{LVI1} > V_{LVI2} > V_{LVI3} > V_{LVI4} > V_{LVI5} > V_{LVI6}$

2. $V_{POC} < V_{LVIm}$ (m = 0 to 6)

LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +110°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> Data retention supply voltage	V_{DDDR}		2.7		5.5	V
Release signal set time	t_{SREL}		0			μs

<R> **Flash Memory Programming Characteristics: Flash Memory Version**

(T_A = 10 to 65°C, 3.0 V ≤ V_{DD} ≤ 5.5 V, 3.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}	f_X = 10 MHz, V_{DD} = 5.5 V			30.5	mA
Step erase time	Chip unit	T_{erac}			10		ms
	Sector unit	T_{eras}			10		ms
Erase time[Note 1]	Chip unit	T_{eraca}				2.55	s
	Sector unit	T_{erasa}				2.55	s
Step write time		T_{wrw}				500	μs
Write time		T_{wrwa}				500	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 2]			100	Times

Notes 1. The prewrite time before erasure and the erase verify time (writeback time) are not included.

2. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

CHAPTER 25 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)

<R> Target products: μPD780861(A2), 780862(A2), 78F0862A(A2)

Absolute Maximum Ratings (T_A = 25°C)

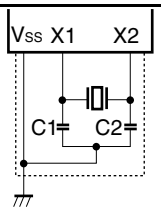
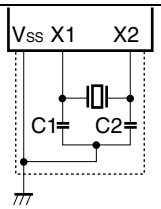
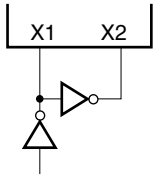
Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.3 to +6.5	V
	V_{SS}		−0.3 to +0.3	V
	AV_{REF}		−0.3 to V_{DD} + 0.3[Note]	V
Input voltage	V_{I1}	P00, P01, P10 to P15, P20 to P23, X1, X2, $\overline{RESET}$	−0.3 to V_{DD} + 0.3[Note]	V
Output voltage	V_O		−0.3 to V_{DD} + 0.3[Note]	V
Analog input voltage	V_{AN}		V_{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	−7	mA
		Total of P00, P01, P10 to P15, P130 pins	−21	mA
Output current, low	I_{OL}	Per pin	14	mA
		Total of P00, P01, P10 to P15, P130 pins	24.5	mA
<R> Operating ambient temperature	T_A	In normal operation mode	−40 to +125	°C
		In flash memory programming mode	−40 to +85	
<R> Storage temperature	T_{stg}	Mask ROM versions	−65 to +150	°C
		Flash memory version	−40 to +150	

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.

Crystal/Ceramic Oscillator Characteristics (When Selecting Crystal/Ceramic Oscillation)
(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		9.2	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
Ceramic resonator		Oscillation frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		9.2	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
External clock		X1 input frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		9.2	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
<R>		X1 input high-/low-level width (t_{XH}, t_{XL})	4.0 V ≤ V_{DD} ≤ 5.5 V	51		250	ns
<R>			2.7 V ≤ V_{DD} < 4.0 V	96		250	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

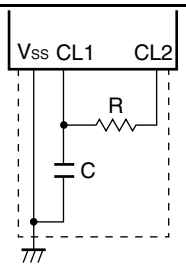
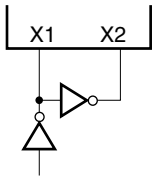
Caution When using the crystal/ceramic oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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.**

Remark For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

External RC Oscillator Characteristics (When Selecting External RC Oscillation)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
RC oscillation		Oscillation frequency (f_{XH})[Note]		3.0		4.0	MHz
External clock		X1 input frequency (f_{XH})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		9.2	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		5.0	
		X1 input high-/low-level width (t_{XH}, t_{XL})	4.0 V ≤ V_{DD} ≤ 5.5 V	51		250	ns
			2.7 V ≤ V_{DD} < 4.0 V	96		250	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution When using the RC oscillator, wire as follows in the area enclosed by the broken lines in the above figure 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.**

External RC Oscillation Frequency Characteristics (When Selecting External RC Oscillation)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency (f_{XH})[Note]	R = 6.8 kΩ, C = 22 pF Target value: 3 MHz	2.5	3.0	3.5	MHz
	R = 4.7 kΩ, C = 22 pF Target value: 4 MHz	3.5	4.0	4.7	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution Set one of the above values to R and C.

Internal High-Speed Oscillator Characteristics (When Selecting Internal High-Speed Oscillation)

(T_A = –40 to +125°C, 4.0 V ≤ V_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal high-speed oscillator	Oscillation frequency (f_{XH})[Note]		6.80	8.00	9.20	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Internal Low-Speed Oscillator Characteristics (T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal low-speed oscillator	Oscillation frequency (f_R)[Note]		120	240	495	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

DC Characteristics (T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V) (1/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.5	mA
		Total of P00, P01, P10 to P15, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			–17.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			–7	mA
Output current, low	I_{OL}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			7	mA
		Total of P00, P01, P10 to P15, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			21	mA
			2.7 V ≤ V_{DD} < 4.0 V			7	mA
Input voltage, high	V_{IH1}	P02[Note 1], P12, P13, P15		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P10, P11, P14, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
	V_{IH3}	P20 to P23[Note 2]		0.7AV_{REF}		AV_{REF}	V
	V_{IH4}	X1, X2		V_{DD} – 0.5		V_{DD}	V
Input voltage, low	V_{IL1}	P02[Note 1], P12, P13, P15		0		0.3V_{DD}	V
	V_{IL2}	P00, P01, P10, P11, P14, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
	V_{IL3}	P20 to P23[Note 2]		0		0.3AV_{REF}	V
	V_{IL4}	X1, X2		0		0.4	V
Output voltage, high	V_{OH}	Total of P00, P01, P10 to P15, P130 pins I_{OH} = –17.5 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –3.5 mA	V_{DD} – 1.0			V
		I_{OH} = –100 μA	2.7 V ≤ V_{DD} < 4.0 V	V_{DD} – 0.5			V
Output voltage, low	V_{OL}	Total of P00, P01, P10 to P15, P130 pins I_{OL} = 21 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 7 mA			1.3	V
		I_{OL} = 400 μA	2.7 V ≤ V_{DD} < 4.0 V			0.4	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD}	P00, P01, P10 to P15, $\overline{\text{RESET}}$			10	μA
		V_I = AV_{REF}	P20 to P23			10	μA
	I_{LIH2}	V_I = V_{DD}	X1, X2[Note 3]			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	P00, P01, P10 to P15, P20 to P23, $\overline{\text{RESET}}$			–10	μA
	I_{LIL2}		X1, X2[Note 3]			–20	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD}				10	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–10	μA
<R> Pull-up resistance value	R	V_I = 0 V		10	30	120	kΩ
FLMD0 supply voltage (Flash memory version only)	Flmd	In normal operation mode		0		0.2V_{DD}	V

<R> **Notes 1.** When the internal high-speed oscillation clock is selected as the high-speed system clock, P02 can be used as a port input pin.

2. When used as a digital input port, set AV_{REF} = V_{DD}.

3. When the inverse input level of X1 is input to X2.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

<R> **DC Characteristics (2/3) : Flash Memory Version**

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Crystal/ ceramic oscillation operating mode[Notes 2, 6]	f_{XH} = 9.2 MHz, V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		7.2	15.9	mA
				When A/D converter is operating[Note 4]		8.2	17.9	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		2.4	5.7	mA
				When A/D converter is operating[Note 4]		3.0	6.9	mA
	I_{DD2}	Crystal/ ceramic oscillation HALT mode[Note 6]	f_{XH} = 9.2 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.7	4.7	mA
				When peripheral functions are operating			7.4	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.48	1.6	mA
				When peripheral functions are operating			2.7	mA
	I_{DD3}	External RC oscillation operating mode[Notes 2, 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		4.5	10.7	mA
				When A/D converter is operating[Note 4]		5.5	12.7	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When A/D converter is stopped		2.4	5.7	mA
				When A/D converter is operating[Note 4]		3.0	6.9	mA
	I_{DD4}	External RC oscillation HALT mode[Note 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	4.7	mA
				When peripheral functions are operating			6.5	mA
			f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.87	2.6	mA
				When peripheral functions are operating			3.6	mA
	I_{DD5}	Internal high-speed oscillation operating mode[Notes 2, 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		6.9	15.6	mA
				When A/D converter is operating[Note 4]		7.9	17.6	mA
	I_{DD6}	Internal high-speed oscillation HALT mode[Note 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.4	4.4	mA
				When peripheral functions are operating			7.1	mA
	I_{DD7}	Internal low-speed oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			1.8	8.4	mA
			V_{DD} = 3.0 V ±10%			0.88	4.1	mA
	I_{DD8}	Internal low-speed oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%			0.08	1.52	mA
			V_{DD} = 3.0 V ±10%			0.06	0.84	mA
	I_{DD9}	STOP mode	V_{DD} = 5.0 V ±10%	Internal low-speed oscillation: OFF		3.5	1200	μA
				Internal low-speed oscillation: ON		17.5	1300	μA
			V_{DD} = 3.0 V ±10%	Internal low-speed oscillation OFF		3.5	600	μA
				Internal low-speed oscillation: ON		11.0	700	μA

Notes 1. 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).

2. Peripheral operation current is included.

3. When PCC = 00H.

4. Total of the current that flows through the V_{DD} pin and AV_{REF} pin.

5. When high-speed system clock is stopped.

6. When crystal/ceramic oscillation is selected as the high-speed system clock using an option byte.

7. When an external RC is selected as the high-speed system clock using an option byte.

8. When an internal high-speed oscillation is selected as the high-speed system clock using an option byte.

DC Characteristics (3/3): Mask ROM Versions

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
<R>	Supply current[Note 1]	I_{DD1}	Crystal/ ceramic oscillation operating mode[Notes 2, 6]	f_{XH} = 9.2 MHz, V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		5.3	11.6	mA
<R>					When A/D converter is operating[Note 4]		6.3	13.6	mA
				f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		1.7	4.2	mA
					When A/D converter is operating[Note 4]		2.3	5.4	mA
<R>		I_{DD2}	Crystal/ ceramic oscillation HALT mode[Note 6]	f_{XH} = 9.2 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.5	4.3	mA
<R>					When peripheral functions are operating			7.0	mA
				f_{XH} = 5 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.41	1.56	mA
					When peripheral functions are operating			2.7	mA
		I_{DD3}	External RC oscillation operating mode[Notes 2, 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		3.2	7.6	mA
					When A/D converter is operating[Note 4]		4.2	9.6	mA
				f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When A/D converter is stopped		1.7	4.2	mA
					When A/D converter is operating[Note 4]		2.3	5.4	mA
		I_{DD4}	External RC oscillation HALT mode[Note 7]	f_X = 4 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.6	4.7	mA
					When peripheral functions are operating			6.5	mA
				f_X = 4 MHz, V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.87	2.6	mA
					When peripheral functions are operating			3.6	mA
		I_{DD5}	Internal high-speed oscillation operating mode[Notes 2, 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When A/D converter is stopped		4.98	11.3	mA
					When A/D converter is operating[Note 4]		5.98	13.3	mA
		I_{DD6}	Internal high-speed oscillation HALT mode[Note 8]	f_{XH} = 8 MHz, V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		1.24	4.0	mA
					When peripheral functions are operating			6.7	mA
		I_{DD7}	Internal low-speed oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			0.17	1.88	mA
				V_{DD} = 3.0 V ±10%			0.11	1.04	mA
		I_{DD8}	Internal low-speed oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%			0.04	1.36	mA
				V_{DD} = 3.0 V ±10%			0.03	0.72	mA
		I_{DD9}	STOP mode	V_{DD} = 5.0 V ±10%	Internal low-speed oscillation: OFF		3.5	1200	μA
					Internal low-speed oscillation: ON		17.5	1300	μA
				V_{DD} = 3.0 V ±10%	Internal low-speed oscillation OFF		3.5	600	μA
					Internal low-speed oscillation: ON		11.0	700	μA

Notes 1. 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).

2. Peripheral operation current is included.

3. When PCC = 00H.

4. Total of the current that flows through the V_{DD} pin and AV_{REF} pin.

5. When high-speed system clock is stopped.

6. When crystal/ceramic oscillation is selected as the high-speed system clock using mask option.

7. When an external RC is selected as the high-speed system clock using mask option.

8. When an internal high-speed oscillation is selected as the high-speed system clock using mask option.

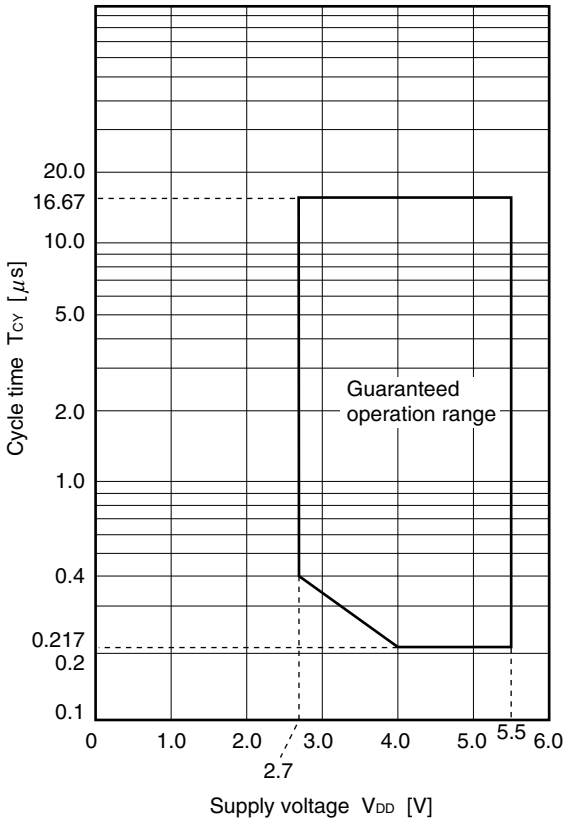
AC Characteristics

(1) Basic operation (T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{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 ≤ V_{DD} ≤ 5.5 V	0.217		16	μs
					2.7 V ≤ V_{DD} < 4.0 V	0.4		16	μs
				External RC oscillation clock	2.7 V ≤ V_{DD} ≤ 5.5 V	0.426		12.8	μs
				Internal high-speed oscillation clock	4.0 V ≤ V_{DD} ≤ 5.5 V	0.217	0.25	4.7	μs
<R>			Internal low-speed oscillation clock		2.7 V ≤ V_{DD} ≤ 5.5 V	4.04	8.33	16.67	μs
TI00 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} ≤ 5.5 V				$2/f_{sam}$ + 0.1[Note]			μs
		2.7 V ≤ V_{DD} < 4.0 V				$2/f_{sam}$ + 0.2[Note]			μs
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}					1			μs
$\overline{RESET}$ low-level width	t_{RSL}					10			μs

Note Selection of f_{sam} = f_{XH}, $f_{XH}/4$, or $f_{XH}/256$ 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_{XH}.

T_{CY} vs. V_{DD} (Main System Clock Operation)



(2) Serial interface (T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{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) 3-wire serial I/O mode ($\overline{SCK10}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.0 V	400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}		t_{KCY1}/2 – 10			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK1}		30			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO1}	C = 100 pF[Note]			30	ns

Note C is the load capacitance of the $\overline{SCK10}$ and SO10 output lines.

(c) 3-wire serial I/O mode ($\overline{SCK10}$... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}		400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}		t_{KCY2}/2			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK2}		80			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO2}	C = 100 pF[Note]			120	ns

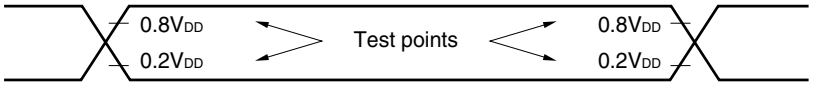
Note C is the load capacitance of the SO10 output line.

(3) Manchester code generator (T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

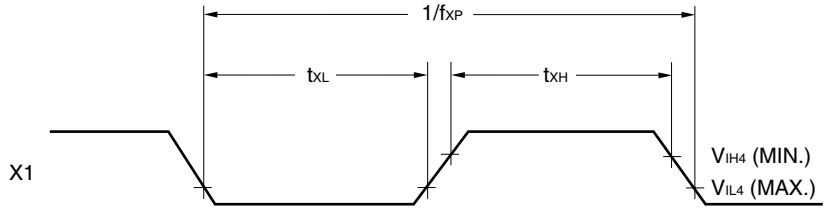
(a) Dedicated baud rate generator output

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					250.0	kbps

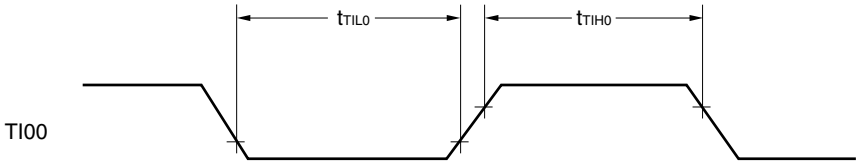
AC Timing Test Points (Excluding X1)



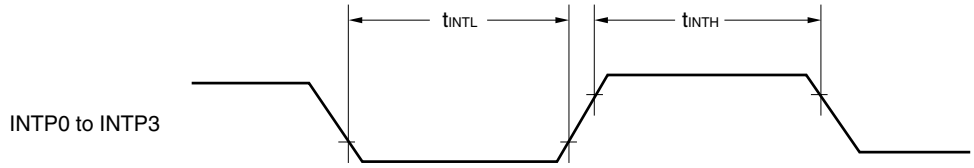
Clock Timing



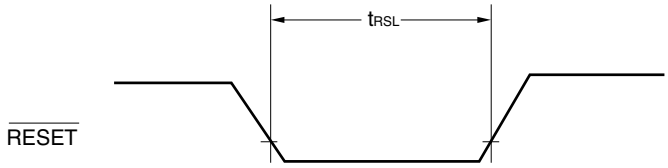
TI Timing



Interrupt Request Input Timing

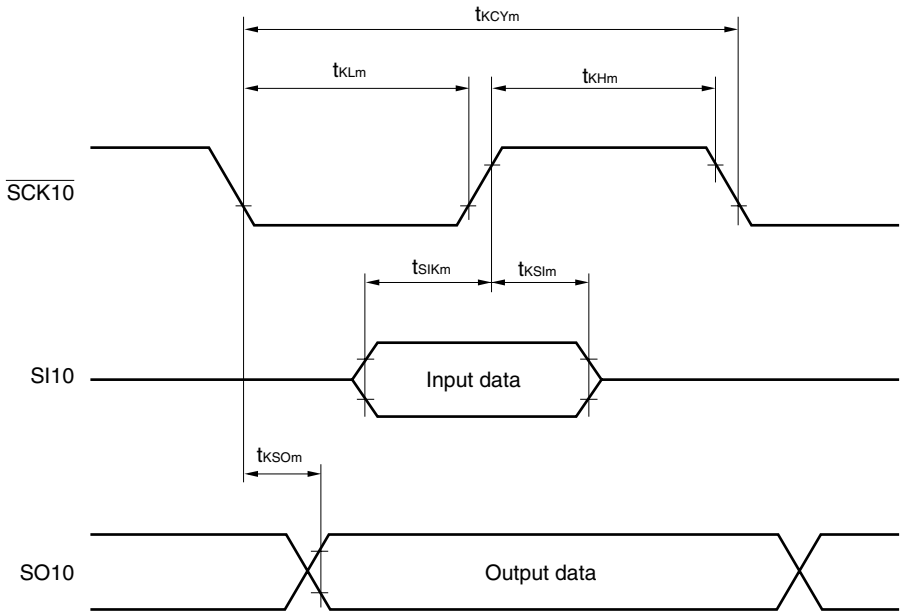


$\overline{\text{RESET}}$ Input Timing



Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

A/D Converter Characteristics (T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V[Note 1])

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Notes 2, 3]		4.0 V ≤ AV_{REF} ≤ 5.5 V		±0.2	±0.7	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V		±0.3	±0.9	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	16		48	μs
		2.7 V ≤ AV_{REF} < 4.0 V	19		48	μs
Zero-scale error[Notes 2, 3]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.7	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.9	%FSR
Full-scale error[Notes 2, 3]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.7	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.9	%FSR
Integral linearity error[Note 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±5.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±7.5	LSB
Differential linearity error[Note 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±3.0	LSB
Analog input voltage	V_{AIN}		V_{SS}[Note 1]		AV_{REF}	V

Notes 1. V_{SS} and AV_{SS} are internally connected in the μPD780862 Subseries. The above specifications are for when only the A/D converter is operating.

2. Excludes quantization error (±1/2 LSB).

3. This value is indicated as a ratio (%FSR) to the full-scale value.

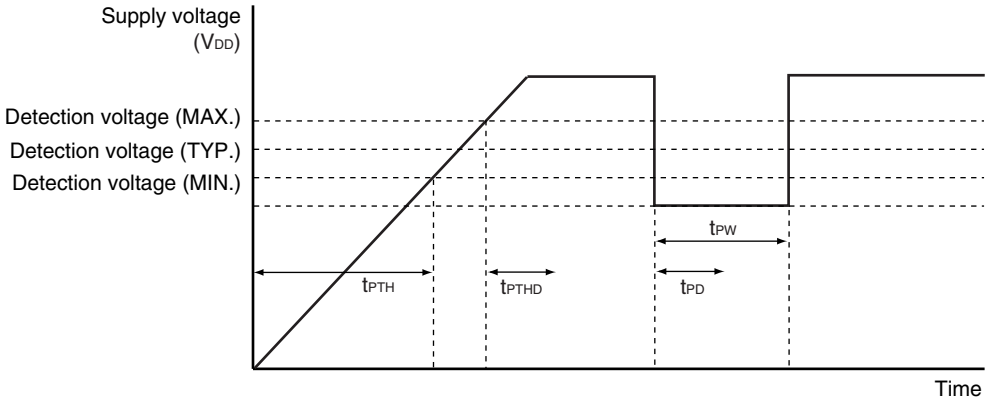
POC Circuit Characteristics (T_A = –40 to +125°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		2.7	2.85	3.06	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → 2.7 V	0.0015			ms
Response delay time 1[Note 1]	t_{PTHD}	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2[Note 2]	t_{PD}	When V_{DD} falls			1.0	ms
Minimum pulse width	t_{PW}		0.2			ms

Notes 1. Time required from voltage detection to reset release.

2. Time required from voltage detection to internal reset output.

POC Circuit Timing



LVI Circuit Characteristics (T_A = –40 to +125°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		4.1	4.3	4.56	V
	V_{LVI1}		3.9	4.1	4.36	V
	V_{LVI2}		3.7	3.9	4.16	V
	V_{LVI3}		3.5	3.7	3.96	V
	V_{LVI4}		3.3	3.5	3.76	V
	V_{LVI5}		3.15	3.3	3.51	V
	V_{LVI6}		2.95	3.1	3.31	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

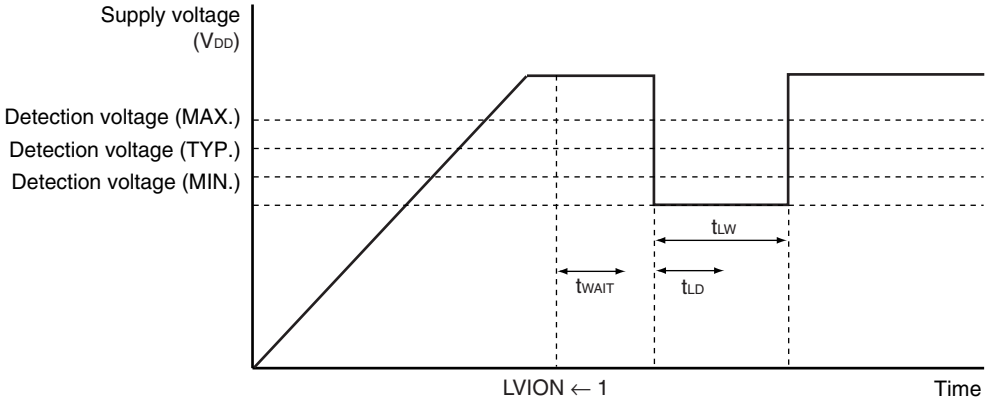
Notes 1. Time required from voltage detection to interrupt output or reset output.

2. Time required from setting LVION to 1 to operation stabilization.

Remarks 1. $V_{LVI0} > V_{LVI1} > V_{LVI2} > V_{LVI3} > V_{LVI4} > V_{LVI5} > V_{LVI6}$

2. $V_{POC} < V_{LVIm}$ (m = 0 to 6)

LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +125°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> Data retention supply voltage	V_{DDDR}		2.7		5.5	V
Release signal set time	t_{SREL}		0			μs

<R> **Flash Memory Programming Characteristics: Flash Memory Version**

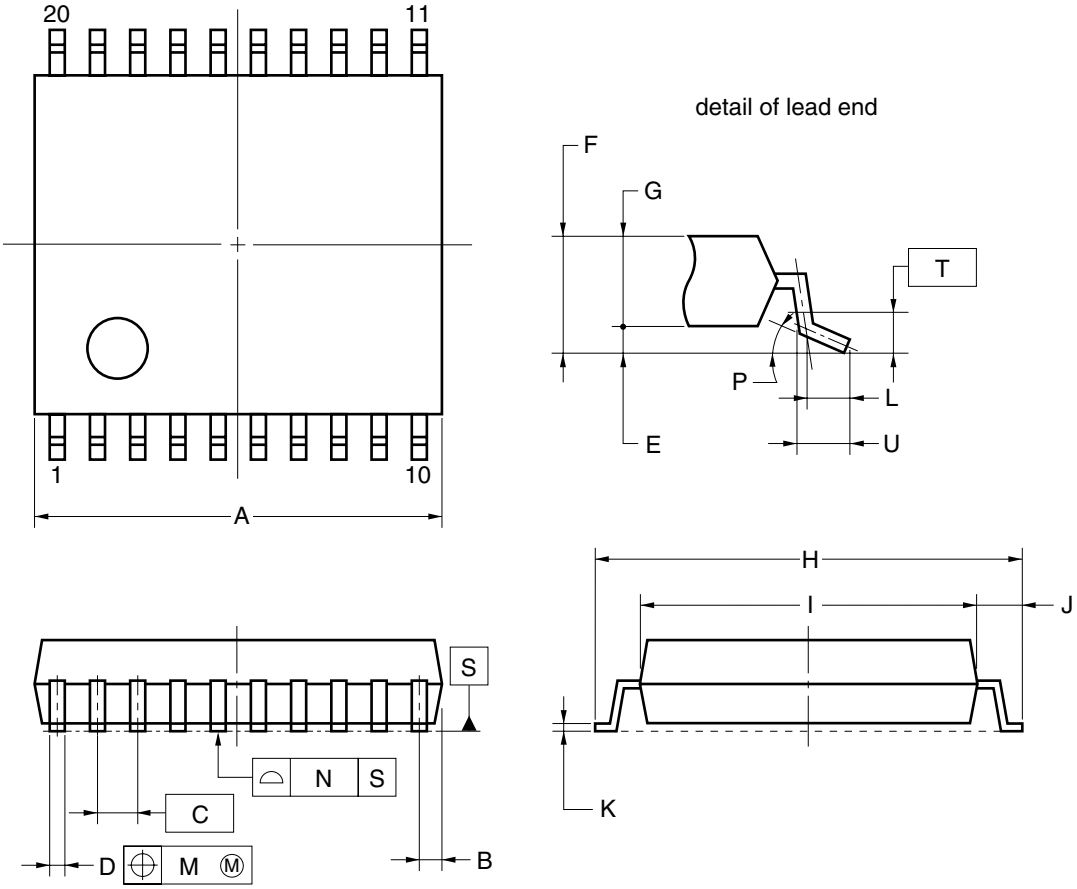
(T_A = 10 to 65°C, 3.0 V ≤ V_{DD} ≤ 5.5 V, 3.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}	f_X = 10 MHz, V_{DD} = 5.5 V			30.5	mA
Step erase time	Chip unit	T_{erac}			10		ms
	Sector unit	T_{eras}			10		ms
Erase time[Note 1]	Chip unit	T_{eraca}				2.55	s
	Sector unit	T_{erasa}				2.55	s
Step write time		T_{wrw}				500	μs
Write time		T_{wrwa}				500	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 2]			100	Times

Notes 1. The prewrite time before erasure and the erase verify time (writeback time) are not included.

2. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

20-PIN PLASTIC SSOP (7.62 mm (300))



NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	6.65±0.15
B	0.475 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

S20MC-65-5A4-2

CHAPTER 27 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, please contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 27-1. Surface Mounting Type Soldering Conditions (1/2)

(1) 20-pin plastic SSOP (7.62 mm (300))

μPD780861MC-×××-5A4, 780862MC-×××-5A4
μPD780861MC(A)-×××-5A4, 780862MC(A)-×××-5A4,
μPD780861MC(A1)-×××-5A4, 780862MC(A1)-×××-5A4,
μPD780861MC(A2)-×××-5A4, 780862MC(A2)-×××-5A4,
<R> μPD78F0862MC-5A4, 78F0862AMC-5A4, 78F0862MC(A)-5A4, 78F0862AMC(A)-5A4,
<R> μPD78F0862AMC(A1)-5A4, 78F0862AMC(A2)-5A4

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR35-207-3
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	VP15-207-3
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature), Exposure limit: 7 days[Note 2](after that, prebake at 125°C for 20 to 72 hours)	WS60-207-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Table 27-1. Surface Mounting Type Soldering Conditions (2/2)

<R> **(2) 20-pin plastic SSOP (7.62 mm (300))**

μPD780861MC-×××-5A4-A, 780862MC-×××-5A4-A
μPD780861MC(A)-×××-5A4-A, 780862MC(A)-×××-5A4-A,
μPD780861MC(A1)-×××-5A4-A, 780862MC(A1)-×××-5A4-A,
μPD780861MC(A2)-×××-5A4-A, 780862MC(A2)-×××-5A4-A,
μPD78F0862MC-5A4-A, 78F0862AMC-5A4-A, 78F0862MC(A)-5A4-A, 78F0862AMC(A)-5A4-A,
μPD78F0862AMC(A1)-5A4-A, 78F0862AMC(A2)-5A4-A

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 30 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Wave soldering	For details, contact an NEC Electronics sales representative.	–
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remark Products with -A at the end of the part number are lead-free products.

CHAPTER 28 CAUTIONS FOR WAIT

28.1 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 28-1**). This must be noted when real-time processing is performed.

28.2 Peripheral Hardware That Generates Wait

Table 28-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

Table 28-1. 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 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> $\cdot \frac{2\ f_{CPU}}{f_{MACRO}} + 1$ *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.

Remark The clock is the CPU clock (f_{CPU}).

28.3 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 28-2. 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 μPD780862 Subseries.

Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**

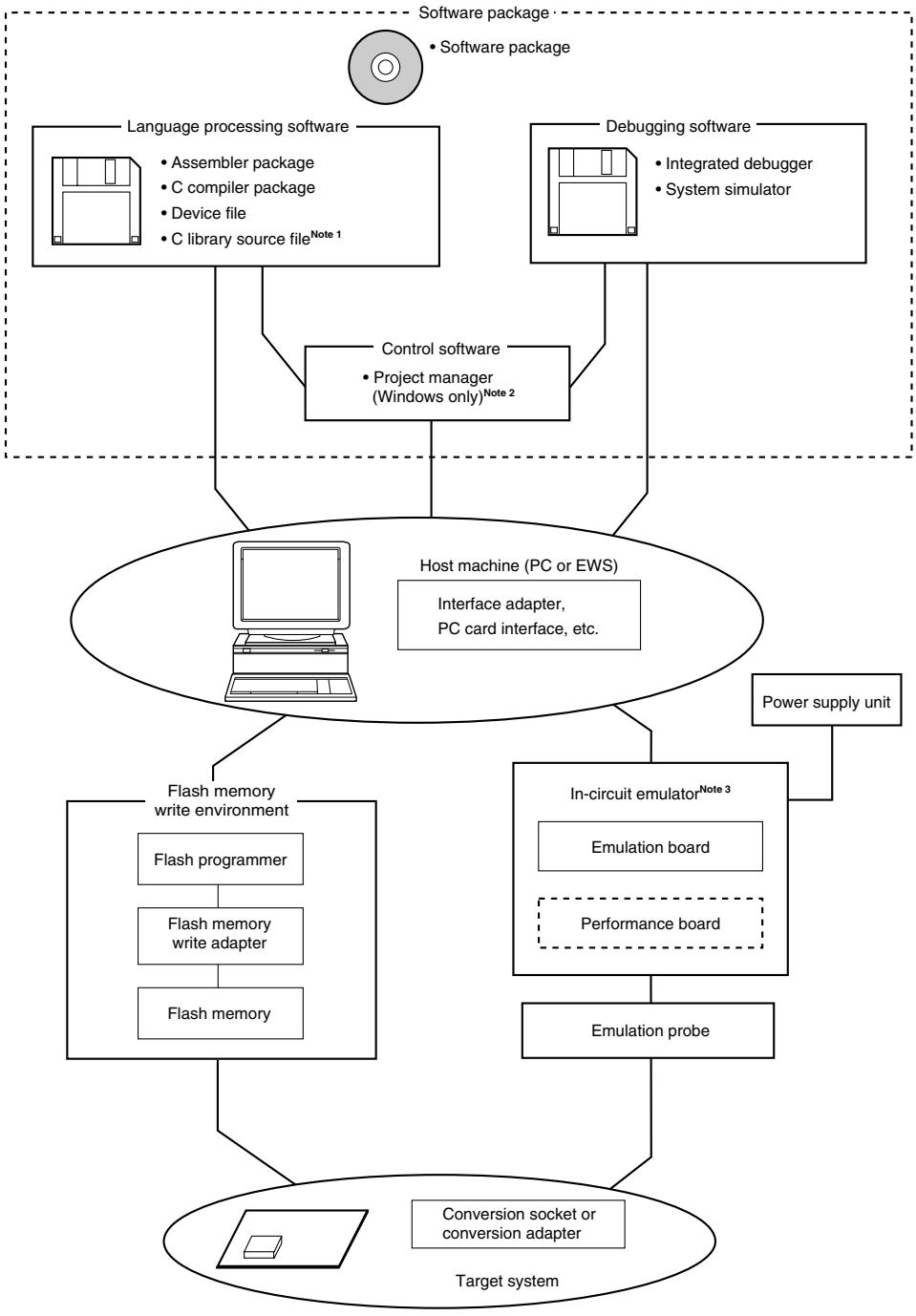
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows**

  Unless otherwise specified, “Windows” means the following OSs.

  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows NT™
  - Windows 2000
  - Windows XP

Figure A-1. Development Tool Configuration



Notes 1. The C library source file is not included in the software package.

2. The project manage PM plus is included in the assembler package.
PM plus is only used for Windows.

3. Products other than in-circuit emulators IE-78K0-NS and IE-78K0-NS-A are all sold separately.

A.1 Software Package

SP78K0 78K/0 Series software package	Development tools (software) common to the 78K/0 Series are combined in this package.
	Part number: μS××××SP78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

RA78K0 Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF780862) (sold separately). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Precaution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××CC78K0
DF780862[Note 1] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used.
	Part number: μS××××DF780862
CC78K0-L[Note 2] C library source file	This is a source file of the functions that configure the object library included in the C compiler package. This file is required to match the object library included in the C compiler package to the user's specifications. Since this is a source file, its operation environment does not depend on any particular operating system.
	Part number: μS××××CC78K0-L

Notes 1. The DF780862 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0.

2. The CC78K0-L is not included in the software package (SP78K0).

Remark ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4) Solaris™ (Rel. 2.5.1)	

μS××××DF780862
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4) Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

A.3 Control Software

<R>

PM plus Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from PM plus. **<Caution>** The project manager is included in the assembler package (RA78K0). It can only be used in Windows.

A.4 Flash Memory Writing Tools

FlashPro4 (part number: FL-PR4, PG-FP4) Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
<R> FA-20MC-5A4-A Flash memory writing adapter	Flash memory writing adapter used connected to the FlashPro4. 20-pin plastic SSOP (MC-5A4 type)

Remark FL-PR4 and FA-20MC-5A4-A are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

A.5 Debugging Tools (Hardware)

IE-78K0-NS In-circuit emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to the integrated debugger (ID78K0-NS). This emulator should be used in combination with a power supply unit, emulation probe, and the interface adapter required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance board		This board is connected to the IE-78K0-NS to expand its functions. Adding this board adds a coverage function and enhances debugging functions such as tracer and timer functions.
IE-78K0-NS-A In-circuit emulator		Product that combines the IE-78K0-NS and IE-78K0-NS-PA
IE-70000-MC-PS-B Power supply unit		This adapter is used for supplying power from a 100 V to 240 V AC outlet.
IE-70000-98-IF-C Interface adapter		This adapter is required when using a PC-9800 series computer (except notebook type) as the IE-78K0-NS(-A) host machine (C bus compatible).
IE-70000-CD-IF-A PC card interface		This is PC card and interface cable required when using a notebook-type computer as the IE-78K0-NS(-A) host machine (PCMCIA socket compatible).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using an IBM PC/AT compatible computer as the IE-78K0-NS(-A) host machine (ISA bus compatible).
IE-70000-PCI-IF-A Interface adapter		This adapter is required when using a computer with a PCI bus as the IE-78K0-NS(-A) host machine.
IE-780862-NS-EM1 Emulation board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
NP-30MC Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for use with a 30-pin plastic SSOP (MC-5A4 type).
	NSPACK20BK YSPACK30BK HSPACK30BK YQ-Guide Conversion socket	This conversion socket connects the NP-30MC to a target system board designed to mount a 20-pin plastic SSOP (MC-5A4 type). • NSPACK20BK: Socket for connecting target • YSPACK30BK: Socket for connecting emulator • HSPACK30BK: Cover for mounting device • YQ-Guide: Guide pin

Remarks 1. NP-30MC is a product of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

2. NSPACK20BK, YSPACK30BK, HSPACK30BK, and YQ-Guide are products of TOKYO ELETECH CORPORATION.
For further information, contact Daimaru Kogyo Co., Ltd.
Tokyo Electronics Department (TEL: +81-3-3820-7112)
Osaka Electronics Department (TEL: +81-6-6244-6672)

A.6 Debugging Tools (Software)

SM78K0 System simulator	This system simulator is used to perform debugging at C source level or assembler level while simulating the operation of the target system on a host machine. This simulator runs on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with a device file (DF780862) (sold separately).
	Part number: μS××××SM78K0
ID78K0-NS Integrated debugger (supporting in-circuit emulator IE-78K0-NS, IE-78K0-NS-A)	This debugger is a control program used to debug 78K/0 Series microcontrollers. The ID78K0-NS is Windows-based software. It has an enhanced debugging function for C language programs, and thus trace results can be displayed on screen at C-language level by using the windows integration function which links a trace result with its source program, disassembled display, and memory display. It should be used in combination with a device file (sold separately).
	Part number: μS××××ID78K0-NS

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0
μS××××ID78K0-NS

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX B NOTES ON TARGET SYSTEM DESIGN

The following shows the conditions when connecting the emulation probe to the conversion adapter. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

Among the products described in this appendix, NP-30MC is a product of Naito Densei Machida Mfg. Co., Ltd., and YSPACK30BK, NSPACK20BK, and YQ-Guide are products of TOKYO ELETECH CORPORATION.

Table B-1. Distance Between IE System and Conversion Adapter

Emulation Probe	Conversion Adapter	Distance Between IE System and Conversion Adapter
NP-30MC	YSPACK30BK NSPACK20BK YQ-Guide	150 mm

Figure B-1. Distance Between In-Circuit Emulator and Conversion Adapter

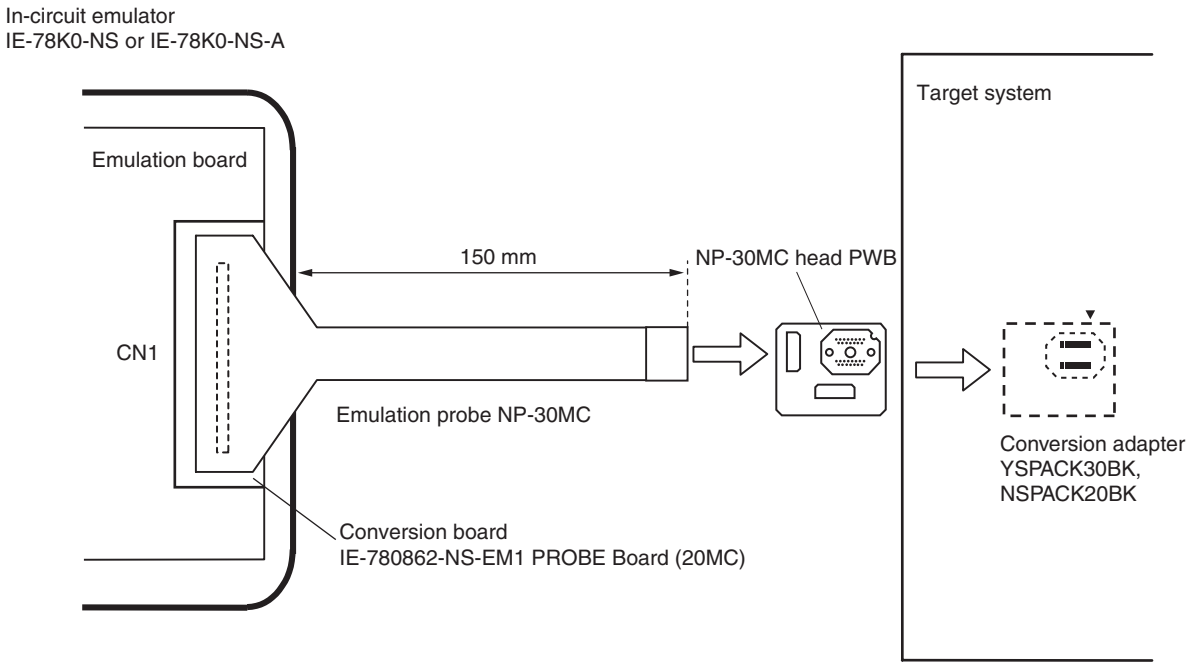
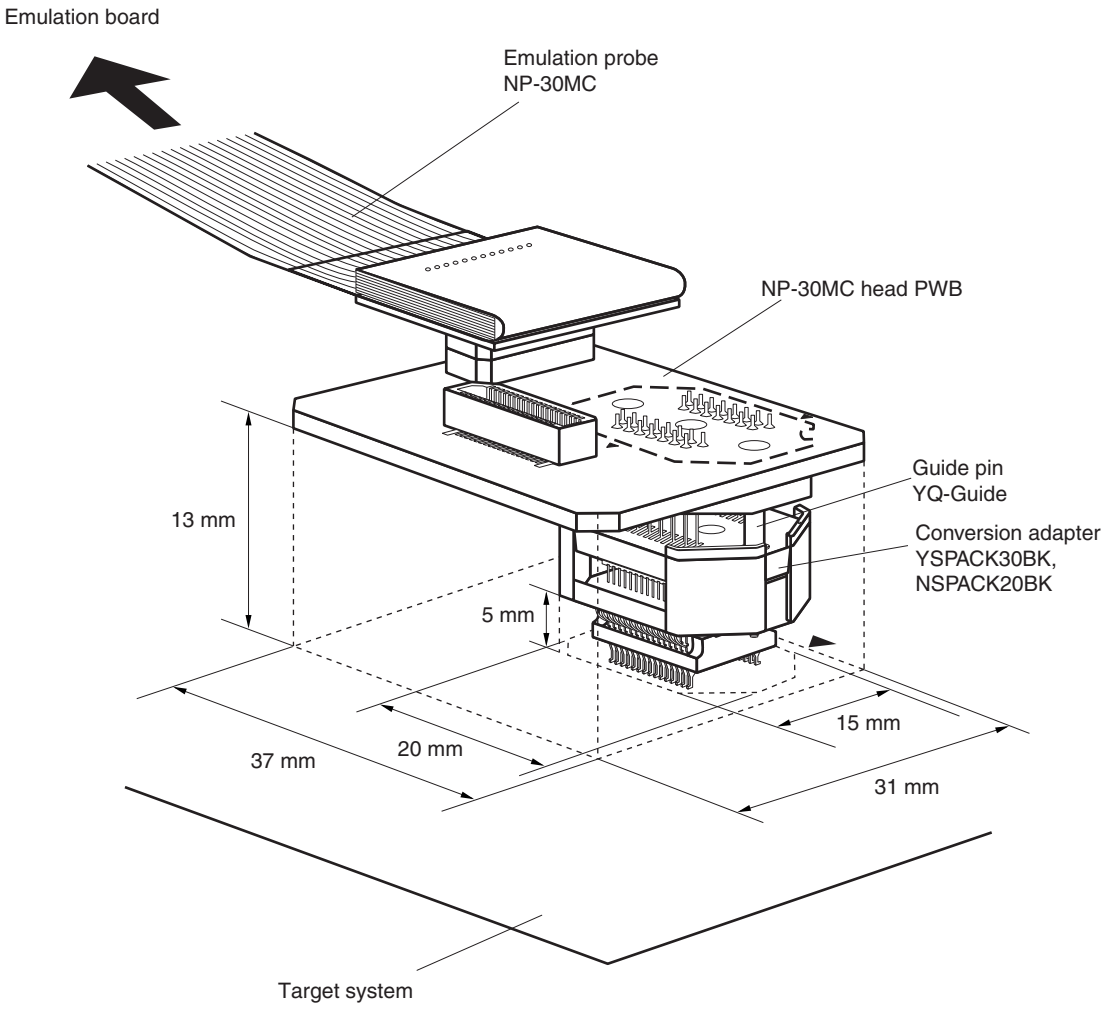


Figure B-2. Connection Conditions of Target System



APPENDIX C REGISTER INDEX

C.1 Register Index (In Alphabetical Order with Respect to Register Names)

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

APPENDIX D REVISION HISTORY

D.1 Major Revisions in This Edition

(1/3)

Page	Description
Throughout	Addition of the following part numbers μPD780861MC-×××-5A4-A, 780862MC-×××-5A4-A, 780861MC(A)-×××-5A4-A, 780862MC(A)-×××-5A4, 780861MC(A1)-×××-5A4-A, 780862MC(A1)-×××-5A4-A, 780861MC(A2)-×××-5A4-A, 780862MC(A2)-×××-5A4-A, 78F0862MC-5A4-A, 78F0862AMC-5A4, 78F0862AMC-5A4-A, 78F0862MC(A)-5A4-A, 78F0862AMC(A)-5A4, 78F0862AMC(A)-5A4-A, 78F0862AMC(A1)-5A4, 78F0862AMC(A1)-5A4-A, 78F0862AMC(A2)-5A4, 78F0862AMC(A2)-5A4-A
p.15	Addition of **Note** to **1.1 Features**
p.21	Addition of description of (A1) grade products and (2) grade products, and **Note 2** to High-speed system clock (oscillation frequency) in **1.6 Outline of Functions**
p.45	Modification of description on Symbol in **3.2.3 Special function registers (SFRs)**
p.65	Modification of **Caution** in **4.2.2 Port 1**
p.76	Addition of **4.3 (5) Input switch control register (ISC)**
p.85	Addition of **Cautions 1 and 2** to **Figure 5-7 Format of Oscillation Stabilization Time Select Register (OSTS)**
p.106	Addition of description of **<When used as capture register>** to Interrupt request generation in **Figure 6-5 Format of 16-Bit Timer Mode Control Register 00 (TMC00).**
p.109	Modification of **Caution 4** in **Figure 6-8 Format of Prescaler Mode Register 00 (PRM00)**
pp.130, 132	**6.4.6 One-shot pulse output operation** • Modification of **Caution 1** in **(1) One-shot pulse output with software trigger** • Modification of **Caution** in **(2) One-shot pulse output with external trigger**
p.135	Modification of **(a) One-shot pulse output by software** and **(b) One-shot pulse output with external trigger** in **(5) Re-triggering one-shot pulse** in **6.5 Cautions for 16-Bit Timer/Event Counter 00**
p.137	Modification of description on <1> in **(11) Edge detection** in **6.5 Cautions for 16-Bit Timer/Event Counter 00**
p.141	Addition of **Remark 2** to **Figure 7-5 Format of Timer Clock Switch Control Register (CSEL)**
p.156	Addition of **Remark 3** to **Figure 8-7 Format of Timer Clock Switch Control Register (CSEL)**
p.157	Modification of description on RMC1 bit and NRZB bit in **Figure 8-8 Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)**
p.161	Modification of **(c) Operation when CMP0n = 00H** in **Figure 8-12 Timing of Interval Timer/Square-Wave Output Operation**
p.168	Modification of description on RMC1 bit and NRZB bit in **8.4.3 (2) Carrier output control**
p.175	Modification of **Table 9-1 Loop Detection Time of Watchdog Timer**
p.178	Modification of description on the overflow time setting in **Figure 9-2 Format of Watchdog Timer Mode Register (WDTM)**
p.193	Modification of **10.4.1 Basic operations of A/D converter**
p.219	Modification of **Caution 1** in **Figure 11-10 Format of Asynchronous Serial Interface Control Register 6 (ASICL6)**
p.246	Modification of **Note 2** in **Figure 12-2 Format of Serial Operation Mode Register 10 (CSIM10)**
p.247	Modification of **Caution 3** in **Figure 12-3 Format of Serial Clock Selection Register 10 (CSIC10)**
p.249	Modification of **Note 1** in **12.4.1 (1) (a) Serial operation mode register 10 (CSIM10)**
p.253	Modification of **(b) Type 2** and **(d) Type 4** in **Figure 12-6 Timing of Clock/Data Phase**

(2/3)

Page	Description
p.258	Addition of **Remark** to **13.2 (2) MCG transmit bit count specification register (MC0BIT)**
p.268	Addition of **14.4.2 (3) Format of "0" and "1" of Manchester code output**
p.271, 272	Modification of **(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)** and **(4) Transmit timing (MC0OLV = 0, total transmit bit length = 13 bits)** in **Figure 13-8 Timing of Manchester Code Generator Mode (LSB First)**
p.280, 281	Modification of **(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)** and **(4) Transmit timing (MC0OLV = 0, total transmit bit length = 13 bits)** in **Figure 13-9 Timing of Bit Sequential Buffer Mode (LSB First)**
p.283	Modification of description on INTTM00 and INTTM01 in **Table 14-1 Interrupt Source List**
p.299	Modification of **Table 15-1 Relationship Between Operation Clocks in Each Operation Status**
p.302	Addition of **Cautions 1 and 2** to **Figure 15-2 Format of Oscillation Stabilization Time Select Register (OSTS)**
p.305	Modification of **(2) (b) Release by reset signal (reset by $\overline{\text{RESET}}$ input, POC, LVI, clock monitor, or WDT)** in **15.2.1 HALT mode**
p.308	Modification of **(2) (b) Release by reset signal (reset by $\overline{\text{RESET}}$ input, POC, LVI, clock monitor, or WDT)** in **15.2.2 STOP mode**
p.315	Addition of description of WDTRF, CLMRF, and LVIRF to the table of **Note** in **Table 16-1 Hardware Statuses After Reset**
p.341	Modification of **Caution 2** to **Table 21-1 Differences Between μPD78F0862, 78F0862A and Mask ROM Versions**
pp.347, 348	Modification of Transfer rate in **21.4 (1) CSI10**, **(2) CSI communication mode supporting handshake**, and **(3) UART6 Figure,** and addition of **Note** to **21.4 (3) UART6**
p.355	Modification of **Table 21-7 Communication Modes**
pp.370 to 373, 383, 384	Modification or addition of the following contents in or to **CHAPTER 23 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS, (A) GRADE PRODUCTS)** • Addition of μPD78F0862A and 78F0862A(A) in Target products • Modification of Max. value of X1 input high-/low-level width (t_{XH}, t_{XL}) of the external clock • Addition of **Note 1** to **DC Characteristics (1/3)** • Modification of Min. value of Data retention supply voltage • Flash Memory Programming Characteristics Modification of V_{DD} supply current (I_{DD}) in, and addition of **Note 3** to (1) μPD78F0862, 78F0862(A) Addition of (2) μPD78F0862A, 78F0862A(A)
pp.385 to 389, 393, 398, 399	Modification or addition of the following contents in or to **CHAPTER 24 ELECTRICAL SPECIFICATIONS ((A1) GRADE PRODUCTS)** • Addition of μPD78F0862A(A1) in Target products, and the item of Flash memory version • Modification of Max. value of X1 input high-/low-level width (t_{XH}, t_{XL}) of the external clock • Addition of FLMD0 supply voltage and **Note 1** to **DC Characteristics (1/3)** • Modification of Instruction cycle when Internal low-speed oscillation clock is operating as Main system clock in **AC Characteristics** • Modification of Min. value of Data retention supply voltage • Addition of **Flash Memory Programming**

(3/3)

Page	Description
pp.400 to 404, 406, 408, 413, 414	Modification or addition of the following contents in or to **CHAPTER 25 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)** • Addition of μPD78F0862A(A2) in Target products, and the item of Flash memory version • Modification of Max. value of X1 input high-/low-level width (t_{XH}, t_{XL}) of the external clock • Addition of FLMD0 supply voltage and **Note 1** to **DC Characteristics (1/3)** • Addition of the value of I_{DD1} and I_{DD2} to **DC Characteristics (3/3)** • Modification of Instruction cycle when Internal low-speed oscillation clock is operating as Main system clock in **AC Characteristics** • Modification of Min. value of Data retention supply voltage • Addition of **Flash Memory Programming**
pp.416, 417	Modification of **Table 27-1 Surface Mounting Type Soldering Conditions**
p.424	Addition of “PM plus” to **A.3 Control Software**, and modification of the part number of the flash memory writing adapter in **A.4 Flash Memory Writing Tools**
p.438	Addition of **D.2 Revision History of Preceding Editions**

<R> D.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/7)

Edition	Description	Chapter
2nd edition	The following packages have been changed from under development to in mass-production. μPD780861MC(A)-×××-5A4, 780862MC(A)-×××-5A4, 780861MC(A1)-×××-5A4, μPD780862MC(A1)-×××-5A4, 780861MC(A2)-×××-5A4, 780862MC(A2)-×××-5A4 Addition of μPD78F0862MC-5A4	Throughout
	Addition of **Note** and description on operating ambient temperature to **1.1 Features**	**CHAPTER 1 OUTLINE**
	Addition of **Note 3** to **1.4 Pin Configuration (Top View)**	
	Addition of **Note 3** to **1.5 Block Diagram**	
	Addition of **Note** to **1.6 Outline of Functions**	
	Addition of **Table 2-1 Pin I/O Buffer Power Supplies**	**CHAPTER 2 PIN FUNCTIONS**
	Addition of **Note 1** to **2.1 (1) Port pins**	
	Modification of description on AV_{REF} in **2.1 (2) Non-port pins**	
	Addition of **Caution** to **2.2.1 P00 to P02 (port 0)**	
	Addition of description to **2.2.11 FLMD0 and FLMD1 (flash memory version only)**	
	Addition of **Note 1** to **Table 2-2 Pin I/O Circuit Types**	
	Modification of figure in **Figure 3-3 Memory Map (μPD78F0862)**	**CHAPTER 3 CPU ARCHITECTURE**
	Addition of **(3) Option byte area (flash memory version only)** to **3.1.1 Internal program memory space**	
	Modification of figure in **Figure 3-10 Data to Be Saved to Stack Memory**	
	Modification of figure in **Figure 3-11 Data to Be Restored from Stack Memory**	
	Modification of [Description example] in **3.4.4 Short direct addressing**	
	Addition of [Illustration] to **3.4.7 Based addressing**	
	Addition of [Illustration] to **3.4.8 Based indexed addressing**	
	Addition of [Illustration] to **3.4.9 Stack addressing**	
	Addition of **Table 4-1 Pin I/O Buffer Power Supplies**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **Note 1** to **Table 4-2 Port Functions**	
	Addition of **Caution** to **4.2.1 Port 0**	
	Modification of **Figure 4-6 Block Diagram of P12**	
	Modification of **Figure 4-7 Block Diagram of P13**	
	Modification of **Cautions 2 and 3** in **4.3 (1) Port mode registers (PM0 and PM1)**	
	Addition of **4.3 (2) Port registers (P0 to P2, P13)**	
	Addition of description to **4.4.1 (1) Output mode**	
	Addition of description to **4.4.3 (1) Output mode** and modification of description in **(2) Input mode**	
	Modification of **Figure 5-2 Format of Processor Clock Control Register (PCC)**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **Table 5-2 Relationship Between CPU Clock and Minimum Instruction Execution Time**	

(2/7)

Edition	Description	Chapter
2nd edition	Addition of **Cautions 2 and 3** to **Figure 5-6 Format of Oscillation Stabilization Time Counter Status Register (OSTC)**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **Table 5-5 Maximum Time Required to Switch Between Internal Low-Speed Oscillation Clock and High-Speed System Clock**	
	Addition of **5.7 Time Required for CPU Clock Switchover**	
	Modification of **Table 6-1 Configuration of 16-Bit Timer/Event Counter 00**	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00**
	Modification of **Figure 6-1 Block Diagram of 16-Bit Timer/Event Counter 00**	
	Addition of **Figure 6-2 Format of 16-Bit Timer Counter 00 (TM00)**	
	Modification of description in **6.2 (2) 16-bit timer capture/compare register 000 (CR000)**	
	Addition of **Figure 6-3 Format of 16-Bit Timer Capture/Compare Register 000 (CR000)**	
	Modification of **Table 6-2 CR000 Capture Trigger and Valid Edges of TI000 and TI010 Pins**	
	Modification of description in **6.2 (3) 16-bit timer capture/compare register 010 (CR010)**	
	Modification of **Table 6-3 CR010 Capture Trigger and Valid Edge of TI000 Pin (CRC002 = 1)**	
	Addition of **Caution 3** to **Figure 6-6 Format of Capture/Compare Control Register 00 (CRC00)**	
	Modification of **Caution 5** and addition of **Cautions 6 and 7** in **Figure 6-7 Format of 16-Bit Timer Output Control Register 00 (TOC00)**	
	Addition of **Caution 1** to **Figure 6-8 Format of Prescaler Mode Register 00 (PRM00)**	
	Addition of description to **6.3 (5) Port mode register 0 (PM0)**	
	Modification of description in **6.4.1 Interval timer operation**	
	Addition of **Figure 6-10 (c) Prescaler mode register 00 (PRM00)**	
	Modification of **Figure 6-12 Timing of Interval Timer Operation**	
	Modification of description in **6.4.2 PPG output operation**	
	Addition of **Figure 6-13 (d) Prescaler mode register 00 (PRM00)**	
	Modification of **Figure 6-15 PPG Output Operation Timing**	
	Modification of description in **6.4.3 Pulse width measurement operation**	
	Modification of description in **6.4.3 (1) Pulse width measurement with free-running counter and one capture register**	
	Addition of **Figure 6-17 (c) Prescaler mode register 00 (PRM00)**	
	Addition of **Note** to **Figure 6-19 Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)**	
	Modification of description in **6.4.3 (2) Measurement of two pulse widths with free-running counter**	
	Addition of **Figure 6-20 (c) Prescaler mode register 00 (PRM00)**	
	Addition of **Note** to **Figure 6-21 Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)**	
	Addition of **Figure 6-22 (c) Prescaler mode register 00 (PRM00)**	

(3/7)

Edition	Description	Chapter
2nd edition	Addition of **Note** to **Figure 6-23 Timing of Pulse Width Measurement Operation with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)**	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00**
	Addition of **Figure 6-24 (c) Prescaler mode register 00 (PRM00)**	
	Modification of description in **6.4.4 External event counter operation**	
	Addition of **Figure 6-26 (c) Prescaler mode register 00 (PRM00)**	
	Modification of **Figure 6-27 Configuration Diagram of External Event Counter**	
	Modification of description in **6.4.5 Square-wave output operation**	
	Addition of **Figure 6-29 (d) Prescaler mode register 00 (PRM00)**	
	Modification of description in **6.4.6 One-shot pulse output operation**	
	Modification of **Note** in **6.4.6 (1) One-shot pulse output with software trigger**	
	Addition of **Figure 6-31 (d) Prescaler mode register 00 (PRM00)**	
	Modification of **Note** in **6.4.6 (2) One-shot pulse output with external trigger**	
	Addition of **Figure 6-33 (d) Prescaler mode register 00 (PRM00)**	
	Modification of **Figure 6-34 Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)**	
	Modification of **Figure 7-1 Block Diagram of 8-Bit Timer 50**	**CHAPTER 7 8-BIT TIMER 50**
	Addition of **Figure 7-2 Format of 8-Bit Timer Counter 50 (TM50)**	
	Addition of **Figure 7-3 Format of 8-Bit Timer Compare Register 50 (CR50)**	
	Modification of **Figure 7-6 Format of 8-Bit Timer Mode Control Register 50 (TMC50)**	
	Modification of **Figure 7-7 (a) Basic operation**	
	Addition of **7.4.2 Operation as operating clock of TMH0 and UART6**	
	Modification of **Figure 8-2 Block Diagram of 8-Bit Timer H1**	**CHAPTER 8 8-BIT TIMERS H0 AND H1**
	Addition of **Note 1** and **Caution 1** to **Figure 8-5 Format of 8-Bit Timer H Mode Register 0 (TMHMD0)**	
	Addition of **Caution 1** to **Figure 8-6 Format of 8-Bit Timer H Mode Register 1 (TMHMD1)**	
	Addition of description to **8.4.1 Operation as interval timer**	
	Modification of **Figure 8-12 Timing of Interval Timer/Square-Wave Output Operation**	
	Modification of description on duty in **8.4.2 (1) Usage**	
	Modification of **Figure 8-14 Operation Timing in PWM Output Mode**	
	Modification of description on carrier clock output cycle and duty in **8.4.3 (3) Usage**	
	Modification of figures (a) and (b) in **Figure 8-17 Carrier Generator Mode Operation Timing**	
	Modification of **Caution 3** and addition of **Caution 5** in **Figure 9-2 Format of Watchdog Timer Mode Register (WDTM)**	**CHAPTER 9 WATCHDOG TIMER**
	Modification of **Cautions 1 and 2** in **Figure 9-3 Format of Watchdog Timer Enable Register (WDTE)**	
	Addition of **Table 9-4 Relationship Between Watchdog Timer Operation and Internal Reset Signal Generated by Watchdog Timer**	
	Modification of **Caution** in **9.4.1 Watchdog timer operation when "internal low-speed oscillation clock cannot be stopped" is selected by mask option**	

(4/7)

Edition	Description	Chapter
2nd edition	Modification of **Figure 10-1 Block Diagram of A/D Converter**	**CHAPTER 10 A/D CONVERTER**
	Modification of **10.2 Configuration of A/D Converter**	
	Modification of **Note 1** in **Figure 10-3 Format of A/D Converter Mode Register (ADM)**	
	Modification of **Note** in **Figure 10-4 Timing Chart When Boost Reference Voltage Generator Is Used**	
	Addition of **10.3 (3) A/D conversion result register (ADCR)**	
	Modification of description in **10.3 (4) Power-fail comparison mode register (PFM)**	
	Modification of description in **10.4.1 Basic operations of A/D converter**	
	Addition of description to **10.4.2 Input voltage and conversion results**	
	Modification of **Figure 10-10 Relationship Between Analog Input Voltage and A/D Conversion Result**	
	Modification of description in **10.4.3 (1) A/D conversion operation (when PFEN = 0)**	
	Modification of description in **10.4.3 (2) Power-fail detection function (when PFEN = 1)**	
	Modification of **Caution 3** in **10.4.3 • When used as power-fail function**	
	Modification of description in **10.6 (6) Input impedance of ANI0 to ANI3 pins**	
	Modification of description in **10.6 (9) Conversion results just after A/D conversion start**	
	Modification of **Figure 10-21 Timing of A/D Converter Sampling and A/D Conversion Start Delay**	
	Addition of **10.6 (12) Internal equivalent circuit**	
	Modification of **Cautions 1 and 3** in **11.1 (2) Asynchronous serial interface (UART) mode**	**CHAPTER 11 SERIAL INTERFACE UART6**
	Modification of **Figure 11-1 LIN Transmission Operation**	
	Modification of **Figure 11-2 LIN Reception Operation**	
	Modification of **Figure 11-3 Port Configuration for LIN Reception Operation**	
	Modification of **Caution 2** in **11.2 (3) Transmit buffer register 6 (TXB6)**	
	Addition of **Note 2** and modification of **Note 3** in **Figure 11-5 Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)**	
	Addition of **Cautions 1 and 3** and modification of **Caution 2** in **Figure 11-5 Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (2/2)**	
	Addition of **Note** and **Caution 1** in **Figure 11-8 Format of Clock Selection Register 6 (CKSR6)**	
	Modification of **Figure 11-10 Format of Asynchronous Serial Interface Control Register 6 (ASICL6)**	
	Addition of **11.3 (7) Input switch control register (ISC)**	
	Addition of **11.3 (8) Port mode register 1 (PM1)**	
	Modification of **Note 2** in **11.4.1 (1) Register used**	
	Modification of description in **11.4.2 (1) Registers used**	
	Modification of description in **11.4.2 (2) (c) Normal transmission**	
	Modification of description in **11.4.2 (2) (d) Continuous transmission**	

(5/7)

Edition	Description	Chapter
2nd edition	Modification of **Figure 11-16 Example of Continuous Transmission Processing Flow**	**CHAPTER 11 SERIAL INTERFACE UART6**
	Modification of description in **11.4.2 (2) (e) Normal reception**	
	Modification of description in **11.4.2 (2) (h) SBF transmission**	
	Modification of example in **11.4.3 (2) (b) Error of baud rate**	
	Modification of **Figure 12-2 Format of Serial Operation Mode Register 10 (CSIM10)**	**CHAPTER 12 SERRAL INTERFACE CSI10**
	Modification of **Figure 12-3 Format of Serial Clock Selection Register 10 (CSIC10)**	
	Modification of **12.3 (3) Port mode register 1 (PM1)**	
	Modification of description in **12.4.1 (1) Register used**	
	Modification of description in **12.4.2 (1) Registers used**	
	Addition of **Table 12-2 Relationship Between Register Settings and Pins**	
	Addition of **12.4.2 (5) SO10 output**	
	Addition of **13.4.2 (1) (c) <1> Baud rate**, **<2> Error of baud rate**, and **<3> Example of setting baud rate**	**CHAPTER 13 MENCHESTER CODE GENERATOR**
	Addition of **13.4.2 (1) (e) Port mode registers 0, 1 (PM0, PM1)**	
	Modification of description in **13.4.2 (2) (b) When P13/TxD6/INTP1/(TOH1)/(MCGO) is set as Manchester code output**	
	Addition of **13.4.3 (1) (c) <1> Baud rate**, **<2> Error of baud rate**, and **<3> Example of setting baud rate**	
	Addition of **13.4.3 (1) (e) Port mode registers 0, 1 (PM0, PM1)**	
	Modification of **Figure 14-1 Basic Configuration of Interrupt Function**	**CHAPTER 14 INTERRUPT FUNCTIONS**
	Addition of **Caution 3** in **Figure 14-2 Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L)**	
	Modification of **Figure 14-5 Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)**	
	Addition of **Table 14-3 Ports Corresponding to EGPn and EGNn**	
	Modification of **Table 14-5 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**	
	Modification of **Table 15-1 Relationship Between Operation Clocks in Each Operation Status**	**CHAPTER 15 STANDBY FUNCTION**
	Addition of **Cautions 2 and 3** to **Figure 15-1 Format of Oscillation Stabilization Time Counter Status Register (OSTC)**	
	Modification of **Table 15-2 Operating Statuses in HALT Mode**	
	Modification of UART6 in **Table 15-4 Operating Statuses in STOP Mode**	
	Modification of **Figure 16-1 Block Diagram of Reset Function**	**CHAPTER 16 RESET FUNCTION**
	Modification of **Figure 16-2 Timing of Reset by $\overline{\text{RESET}}$ Input**	
	Modification of **Figure 16-3 Timing of Reset Due to Watchdog Timer Overflow**	
	Modification of **Figure 16-4 Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	
	Modification of description in **17.1 Functions of Clock Monitor**	**CHAPTER 17 CLOCK MONITOR**
	Modification of **Figure 17-1 Block Diagram of Clock Monitor**	

(6/7)

Edition	Description	Chapter
2nd edition	Addition of **Caution 3** to **Figure 17-2 Format of Clock Monitor Mode Register (CLM)**	**CHAPTER 17 CLOCK MONITOR**
	Addition of **Figure 17-3 (6) Clock monitor status after high-speed system clock oscillation is stopped by software**	
	Addition of **Figure 17-3 (7) Clock monitor status after internal low-speed oscillation clock oscillation is stopped by software**	
	Addition of **Caution 2** to **18.1 Functions of Power-on-Clear Circuit**	**CHAPTER 18 POWER-ON-CLEAR CIRCUIT**
	Modification of **Figure 18-1 Block Diagram of Power-on-Clear Circuit**	
	Modification of **Figure 18-3 Example of Software Processing After Release of Reset**	
	Modification of **Figure 19-1 Block Diagram of Low-Voltage Detector**	**CHAPTER 19 LOW-VOLTAGE DETECTOR**
	Addition of **Caution** to **Figure 19-3 Format of Low-Voltage Detection Level Selection Register (LVIS)**	
	Modification of **Figure 19-4 Timing of Low-Voltage Detector Internal Reset Signal Generation**	
	Modification of **Figure 19-5 Timing of Low-Voltage Detector Interrupt Signal Generation**	
	Modification of **Figure 19-6 Example of Software Processing After Release of Reset**	
	Modification of description in **19.5 Cautions for Low-Voltage Detector <Action> (2) When used as interrupt**	
	Modification of **Figure 20-2 Format of Option Bytes (Flash Memory Version)**	**CHAPTER 20 MASK OPTIONS / OPTION BYTE**
	Modification of **Table 21-3 Wiring Between μPD78F0862 and Dedicated Flash Programmer**	**CHAPTER 21 FLASH MEMORY**
	Modification of **Figure 21-2 Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode**	
	Modification of **Figure 21-3 Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10 + HS) Mode**	
	Modification of **Figure 21-4 Example of Wiring Adapter for Flash Memory Writing in UART (UART6) Mode**	
	Addition of **21.3 Programming Environment**	
	Addition of **21.4 Communication Mode**	
	Addition of **21.5 Handling of Pins on Board**	
	Addition of **21.6 Programming Method**	
	Modification of **CHAPTER 23 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS, (A) GRADE PRODUCTS)**	**CHAPTER 23 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS, (A) GRADE PRODUCTS**
	Addition of **CHAPTER 24 ELECTRICAL SPECIFICATIONS ((A1) GRADE PRODUCTS)**	**CHAPTER 24 ELECTRICAL SPECIFICATIONS ((A1) GRADE PRODUCTS**

(7/7)

Edition	Description	Chapter
2nd edition	Addition of **CHAPTER 25 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**	**CHAPTER 25 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**
	Addition of **CHAPTER 27 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 27 RECOMMENDED SOLDERING CONDITIONS**
	Modification of **Figure A-1 Development Tool Configuration**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of **A.3 Control Software**	
	Modification of **A.5 Debugging Tools (Hardware)**	
	Addition of **APPENDIX B NOTES ON TARGET SYSTEM DESIGN**	**APPENDIX B NOTES ON TARGET SYSTEM DESIGN**
	Addition of **APPENDIX D REVISION HISTORY**	**APPENDIX D REVISION HISTORY**

[MEMO]

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielski Strasse 166 B
30177 Hanover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52180
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Limburglaan 5
5616 HR Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
TEL: 010-8235-1155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

Seoul Branch
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/