NEC

Preliminary User's Manual

78K0/KF1

8-Bit Single-Chip Microcontrollers

μPD780143
μPD780144
μPD780146
μPD780148
μPD78F0148

Document No. U15947EJ1V1UD00 (1st edition)
Date Published July 2002 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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 once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build 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 bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

License not needed: μPD78F0148

The customer must judge the need for a license: μPD780143, 780144, 780146, and 780148

- **The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Corporation, at its own discretion, may withdraw the device prior to its production.**
- **Not all devices/types available in every country. Please check with local NEC 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 Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
- NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation 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 the customer's equipment shall be done under the full responsibility of the customer. NEC Corporation assumes no responsibility for any losses incurred by the customer or third parties arising from the use of these circuits, software, and information.
- While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
- NEC devices are classified into the following three quality grades:
  "Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device 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 or medical equipment for life support, etc.

  The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

M5D 98. 12

Regional Information

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

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

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

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP, Brasil
Tel: 11-6462-6810
Fax: 11-6462-6829

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
  Fax: 091-504 28 60

- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
  Fax: 01-30-67 58 99

- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
  Fax: 02-66 75 42 99

- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
  Fax: 040-244 45 80

- **Branch Sweden**
  Taeby, Sweden
  Tel: 08-63 80 820
  Fax: 08-63 80 388

- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133
  Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138
Fax: 021-6841-1137

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 253-8311
Fax: 250-3583

J02.4

Major Revisions in This Edition

Page	Description
 p.73 p.74 p.76	Modification of reset value of the following registers in **Table 3-5 Special Function Register List** • Serial I/O shift register 10 (SIO10) • Serial I/O shift register 11 (SIO11) • Interrupt mask flag register 1H (MK1H)
 p.75	Modification of manipulatable bit unit of the following register in **Table 3-5 Special Function Register List** • Oscillation stabilization time counter status register (OSTC)
p.139	Modification of manipulatable bit unit and clear condition in **6.3 (5) Oscillation stabilization time counter status register (OSTC)**
pp.146 to 149	Modification of **Figure 6-13 Status Transition Diagram**
p.150	Modification of **Table 6-4 Oscillation Control Flags and Clock Oscillation Status**
p.162	Modification of reset value in **7.2 (2) 16-bit timer capture/compare register 00n (CR00n)** and **(3) 16-bit timer capture/compare register 01n (CR01n)**
p.170	Modification of manipulatable bit unit in **7.3 (4) Prescaler mode register 0n (PRM0n)**
p.276	Addition of caution description in **13.6 (10) A/D conversion result register (ADCR) read operation**
p.347	Modification of reset value in **16.2 (2) Serial I/O shift register 1n (SIO1n)**
p.435	Modification of reset value in **19.3 (2) Interrupt mask flag register (MK1H)**
p.450	Modification of manipulatable bit unit and clear condition in **21.1.2 (1) Oscillation stabilization time counter status register (OSTC)**
pp.452, 453	Modification of A/D converter item in **Table 21-2 Operating Statuses in HALT Mode**
pp.466, 468	Modification of stop condition of clock monitor in **23.1 Functions of Clock Monitor** and **23.4 Operation of Clock Monitor**
pp.474, 475	Addition of **24.4 Cautions for Power-on-Clear Circuit**
p.479	Modification of **Figure 25-3 Format of Low-Voltage Detection Level Selection Register (LVIS)**
pp.484 to 487	Addition of **25.5 Cautions for Low-Voltage Detector**
p.488	Modification of description in **26.1 Outline of Regulator**
  pp.515, 516 p.517 p.518 pp.519 to 522 p.534 p.534 p.535 p.535 p.535 pp.536, 537	Modification of the following contents in **CHAPTER 30 ELECTRICAL SPECIFICATIONS (TARGET VALUES)** • **Absolute Maximum Ratings** • **X1 Oscillator Characteristics** • **Subsystem Clock Oscillator Characteristics** • **DC Characteristics** • **A/D Converter Characteristics** • **POC Circuit Characteristics** • **LVI Circuit Characteristics** • **Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics** (deletion of data retention supply current) • Deletion of **Ring-OSC Characteristics** • **Flash Memory Programming Characteristics**
pp.540 to 542	Modification from **CHAPTER 32 RETRY** to **CHAPTER 32 CAUTIONS FOR WAIT**

The mark ★ shows major revised points.

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0/KF1 Series and design and develop application systems and programs for these devices.
The target products are as follows.

78K0/KF1 Series: μPD780143, 780144, 780146, 780148, and 78F0148

Purpose

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

Organization

The 78K0/KF1 Series manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

78K0/KF1 User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupts	• Explanation of each instruction
• Other on-chip peripheral functions	
• Electrical specifications	

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) products and (A1) products:
  → Only the quality grade differs between standard products and (A) and (A1) products. Read the part number as follows.
  - μPD780143 → μPD780143(A), 780143(A1)
  - μPD780144 → μPD780144(A), 780144(A1)
  - μPD780146 → μPD780146(A), 780146(A1)
  - μPD780148 → μPD780148(A), 780148(A1)
  - μPD78F0148 → μPD78F0148(A)

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**.
- How to interpret the register format:
  → For a bit number enclosed in square, the bit name is defined as a reserved word in the assembler, and is already defined in the header file named sfrbit.h in the C compiler.
- To check the details of a register when you know the register name.
  → Refer to **APPENDIX C REGISTER INDEX**.

- To know details of the 78K/0 Series instructions.
  → Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

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.
78K0/KF1 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 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
SM78K0S, SM78K0 System Simulator Ver. 2.10 or Later	Operation (Windows™ Based)	U14611E
SM78K Series System Simulator Ver. 2.10 or Later	External Part User Open Interface Specifications	U15006E
ID78K0-NS Integrated Debugger Ver. 2.00 or Later	Operation (Windows Based)	U14379E
ID78K0 Integrated Debugger Windows Based	Reference	U11539E
	Guide	U11649E
RX78K0 Real-Time OS	Fundamentals	U11537E
	Installation	U11536E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document 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-780148-NS-EM1 Emulation Board	To be prepared

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP3 Flash Memory Programmer User’s Manual	U13502E

Other Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE – Product & Packages –	X13769E
Semiconductor Device Mounting Technology Manual	C10535E
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

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

LIST OF FIGURES (1/8)

LIST OF FIGURES (2/8)

LIST OF FIGURES (3/8)

LIST OF FIGURES (4/8)

Figure No. Title Page

LIST OF FIGURES (5/8)

Figure No. Title Page

LIST OF FIGURES (6/8)

LIST OF FIGURES (7/8)

Figure No. Title Page

LIST OF FIGURES (8/8)

Figure No. Title Page

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

CHAPTER 1 OUTLINE

1.1 Features

- ROM, RAM capacities

Item Part Number	Program Memory (ROM)		Data Memory	
			Internal High-Speed RAM	Internal Expansion RAM
μPD780143	Mask ROM	24 KB	1024 bytes	–
μPD780144		32 KB		
μPD780146		48 KB		1024 bytes
μPD780148		60 KB		
μPD78F0148	Flash memory	60 KB[Note]		1024 bytes[Note]

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

- Buffer RAM: 32 bytes
- External memory expansion space: 64 KB
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- Short startup is possible via the CPU default start using the on-chip Ring-OSC
- On-chip clock monitor function using on-chip Ring-OSC
- On-chip watchdog timer (operable with Ring-OSC clock)
- On-chip UART supporting LIN (Local Interconnect Network) bus
- On-chip multiplier/divider
- On-chip key interrupt function
- On-chip external bus interface function
- On-chip clock output/buzzer output controller
- On-chip regulator
- Minimum instruction execution time can be changed from high speed (0.2 μs: @ 10 MHz operation with X1 input clock) to ultra low-speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- I/O ports: 67 (N-ch open drain: 4)
- Timer
  μPD780143, 780144: 7 channels
  μPD780146, 780148, 78F0148: 8 channels
- Serial interface
  μPD780143, 780144: 3 channels
  (UART: 1 channel, CSI/UART[Note]: 1 channel, CSI with automatic transmit/receive function: 1 channel)
  μPD780146, 780148, 78F0148: 4 channels
  (UART: 1 channel, CSI: 1 channel, CSI/UART[Note]: 1 channel, CSI with automatic transmit/receive function: 1 channel)
- 10-bit resolution A/D converter: 8 channels
- Supply voltage: V_{DD} = 2.7 to 5.5 V

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

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

1.3 Ordering Information

(1) Mask ROM version

Part Number	Package
μPD780143GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780143GC-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780144GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780144GC-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780146GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780146GC-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780148GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780148GC-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780143GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780143GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780144GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780144GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780146GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780146GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780148GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780148GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780143GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780143GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780144GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780144GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780146GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780146GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)
μPD780148GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD780148GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)

Remark ××× indicates ROM code suffix.

(2) Flash memory

Part Number	Package
μPD78F0148M1GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M1GC-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M2GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M2GC-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M3GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M3GC-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M4GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M4GC-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M5GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M5GC-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M6GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M6GC-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M1GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M1GC(A)-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M2GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M2GC(A)-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M3GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M3GC(A)-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M4GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M4GC(A)-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M5GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M5GC(A)-8BT	80-pin plastic QFP (14 × 14)
μPD78F0148M6GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F0148M6GC(A)-8BT	80-pin plastic QFP (14 × 14)

Mask ROM versions (μPD780143, 780144, 780146, and 780148) include mask options. When ordering, it is possible to select “Power-on-clear (POC) circuit can be used/cannot be used”, “Ring-OSC clock can be stopped/cannot be stopped by software” and “Pull-up resistor incorporated/not incorporated in 1-bit units (P60 to P63)”.

Flash memory versions corresponding to the mask options of the mask ROM versions are as follows.

Table 1-1. Flash Memory Versions Corresponding to Mask Options of Mask ROM Versions

Mask Option		Flash Memory Versions (Part Number)
POC Circuit	Ring-OSC	
POC cannot be used	Cannot be stopped	μPD78F0148M1GK-9EU μPD78F0148M1GC-8BT μPD78F0148M1GK(A)-9EU μPD78F0148M1GC(A)-8BT
	Can be stopped by software	μPD78F0148M2GK-9EU μPD78F0148M2GC-8BT μPD78F0148M2GK(A)-9EU μPD78F0148M2GC(A)-8BT
POC used (V_{POC} = 2.85 V ±0.15 V)	Cannot be stopped	μPD78F0148M3GK-9EU μPD78F0148M3GC-8BT μPD78F0148M3GK(A)-9EU μPD78F0148M3GC(A)-8BT
	Can be stopped by software	μPD78F0148M4GK-9EU μPD78F0148M4GC-8BT μPD78F0148M4GK(A)-9EU μPD78F0148M4GC(A)-8BT
POC used (V_{POC} = 3.5 V ±0.2 V)	Cannot be stopped	μPD78F0148M5GK-9EU μPD78F0148M5GC-8BT μPD78F0148M5GK(A)-9EU μPD78F0148M5GC(A)-8BT
	Can be stopped by software	μPD78F0148M6GK-9EU μPD78F0148M6GC-8BT μPD78F0148M6GK(A)-9EU μPD78F0148M6GC(A)-8BT

1.4 Quality Grade

(1) Mask ROM version

Part Number	Package	Quality Grade
μPD780143GK-××× -9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD780143GC-×××-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD780144GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD780144GC-×××-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD780146GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD780146GC-×××-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD780148GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD780148GC-×××-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD780143GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780143GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)	Special
μPD780144GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780144GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)	Special
μPD780146GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780146GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)	Special
μPD780148GK(A)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780148GC(A)-×××-8BT	80-pin plastic QFP (14 × 14)	Special
μPD780143GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780143GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)	Special
μPD780144GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780144GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)	Special
μPD780146GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780146GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)	Special
μPD780148GK(A1)-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD780148GC(A1)-×××-8BT	80-pin plastic QFP (14 × 14)	Special

Remark ××× indicates ROM code suffix.

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

(2) Flash memory version

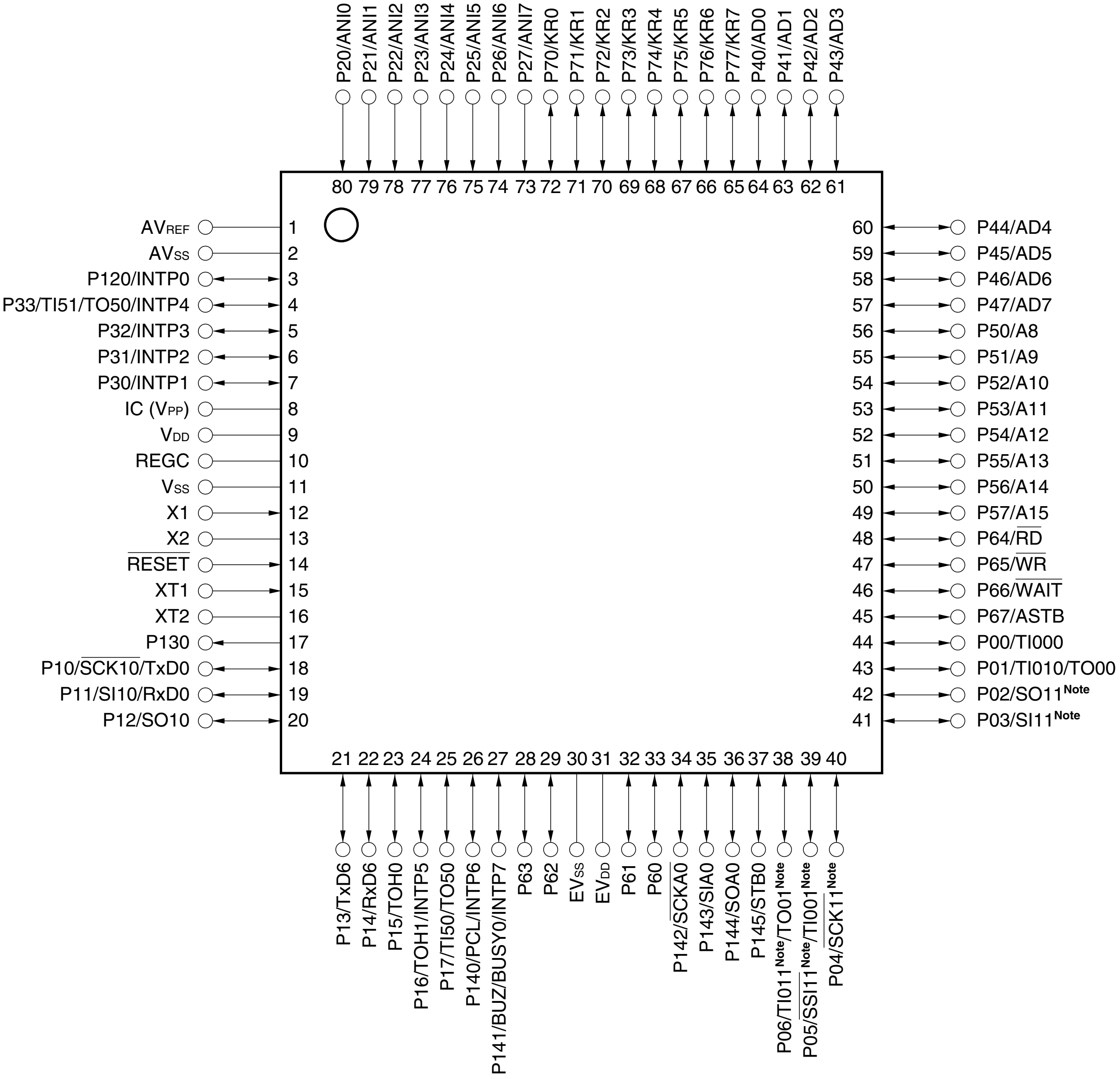
Part Number	Package	Quality Grade
μPD78F0148M1GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD78F0148M1GC-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD78F0148M2GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD78F0148M2GC-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD78F0148M3GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD78F0148M3GC-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD78F0148M4GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD78F0148M4GC-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD78F0148M5GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD78F0148M5GC-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD78F0148M6GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Standard
μPD78F0148M6GC-8BT	80-pin plastic QFP (14 × 14)	Standard
μPD78F0148M1GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD78F0148M1GC(A)-8BT	80-pin plastic QFP (14 × 14)	Special
μPD78F0148M2GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD78F0148M2GC(A)-8BT	80-pin plastic QFP (14 × 14)	Special
μPD78F0148M3GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD78F0148M3GC(A)-8BT	80-pin plastic QFP (14 × 14)	Special
μPD78F0148M4GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD78F0148M4GC(A)-8BT	80-pin plastic QFP (14 × 14)	Special
μPD78F0148M5GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD78F0148M5GC(A)-8BT	80-pin plastic QFP (14 × 14)	Special
μPD78F0148M6GK(A)-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Special
μPD78F0148M6GC(A)-8BT	80-pin plastic QFP (14 × 14)	Special

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

1.5 Pin Configuration (Top View)

- 80-pin plastic TQFP (fine pitch) (12 × 12)
  μPD780143GK-×××-9EU, 780144GK-×××-9EU, 780146GK-×××-9EU, 780148GK-×××-9EU,
  μPD780143GK(A)-×××-9EU, 780144GK(A)-×××-9EU, 780146GK(A)-×××-9EU,
  μPD780148GK(A)-×××-9EU, 780143GK(A1)-×××-9EU, 780144GK(A1)-×××-9EU,
  μPD780146GK(A1)-×××-9EU, 780148GK(A1)-×××-9EU,
  μPD78F0148M1GK-9EU, 78F0148M2GK-9EU, 78F0148M3GK-9EU, 78F0148M4GK-9EU,
  μPD78F0148M5GK-9EU, 78F0148M6GK-9EU, 78F0148M1GK(A)-9EU, 78F0148M2GK(A)-9EU,
  μPD78F0148M3GK(A)-9EU, 78F0148M4GK(A)-9EU, 78F0148M5GK(A)-9EU, 78F0148M6GK(A)-9EU

- 80-pin plastic QFP (14 × 14)
  μPD780143GC-×××-8BT, 780144GC-×××-8BT, 780146GC-×××-8BT, 780148GC-×××-8BT,
  μPD780143GC(A)-×××-8BT, 780144GC(A)-×××-8BT, 780146GC(A)-×××-8BT,
  μPD780148GC(A)-×××-8BT, 780143GC(A1)-×××-8BT, 780144GC(A1)-×××-8BT,
  μPD780146GC(A1)-×××-8BT, 780148GC(A1)-×××-8BT,
  μPD78F0148M1GC-8BT, 78F0148M2GC-8BT, 78F0148M3GC-8BT, 78F0148M4GC-8BT,
  μPD78F0148M5GC-8BT, 78F0148M6GC-8BT, 78F0148M1GC(A)-8BT, 78F0148M2GC(A)-8BT,
  μPD78F0148M3GC(A)-8BT, 78F0148M4GC(A)-8BT, 78F0148M5GC(A)-8BT, 78F0148M6GC(A)-8BT



Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

Cautions 1. Connect the IC (Internally Connected) pin directly to V_{SS}.

2. Connect the AV_{REF} pin to V_{DD}.

3. Connect the AV_{SS} pin to V_{SS}.

4. When using the regulator, connect the REGC pin to V_{SS} via 0.1 μF capacitor. When the regulator is not used, connect the REGC pin directly to V_{DD}.

Remark Figures in parentheses apply only to the μPD78F0148.

Pin Identification

A8 to A15:	Address bus	REGC:	Regulator capacitance
AD0 to AD7:	Address/data bus	$\overline{\text{RESET}}$:	Reset
ANI0 to ANI7:	Analog input	RxD0, RxD6:	Receive data
ASTB:	Address strobe	$\overline{\text{RD}}$:	Read strobe
AV_{REF}:	Analog reference voltage	$\overline{\text{SCK10}}$, $\overline{\text{SCK11}}$[Note],	
AV_{SS}:	Analog ground	$\overline{\text{SCKA0}}$:	Serial clock input/output
BUSY0:	Serial busy input	SI10, SI11[Note], SIA0:	Serial data input
BUZ:	Buzzer output	SO10, SO11[Note],	
EV_{DD}:	Power supply for port	SOA1:	Serial data output
EV_{SS}:	Ground for port	$\overline{\text{SSI11}}$[Note]:	Serial interface chip select input
IC:	Internally connected	STB0:	Serial strobe
INTP0 to INTP7:	External interrupt input	TI000, TI010,	
KR0 to KR7:	Key return	TI001[Note], TI011[Note],	
P00 to P06:	Port 0	TI50, TI51:	Timer input
P10 to P17:	Port 1	TO00, TO01[Note],	
P20 to P27:	Port 2	TO50, TO51,	
P30 to P33:	Port 3	TOH0, TOH1:	Timer output
P40 to P47:	Port 4	TxD0, TxD6:	Transmit data
P50 to P57:	Port 5	V_{DD}:	Power supply
P60 to P67:	Port 6	V_{PP}:	Programming power supply
P70 to P77:	Port 7	V_{SS}:	Ground
P120:	Port 12	$\overline{\text{WAIT}}$:	Wait
P130:	Port 13	$\overline{\text{WR}}$:	Write strobe
P140 to P145:	Port 14	X1, X2:	Crystal (X1 input clock)
PCL:	Programmable clock output	XT1, XT2:	Crystal (Subsystem clock)

Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

1.6 78K0/Kxx Series Lineup

The lineup of products in the 78K0/Kxx Series (under development or in planning) is shown below.

78K0/KB1 Series: 30-pin (7.62 mm 0.65 mm pitch)

μPD78F0103	Flash memory: 24 KB, RAM: 768 bytes
μPD780103	Mask ROM: 24 KB, RAM: 768 bytes
μPD780102	Mask ROM: 16 KB, RAM: 768 bytes
μPD780101	Mask ROM: 8 KB, RAM: 512 bytes

78K0/KC1 Series: 44-pin (10 × 10 mm 0.8 mm pitch)

μPD78F0114	Flash memory: 32 KB, RAM: 1 KB
μPD780114	Mask ROM: 32 KB, RAM: 1 KB
μPD780113	Mask ROM: 24 KB, RAM: 1 KB
μPD780112	Mask ROM: 16 KB, RAM: 512 bytes
μPD780111	Mask ROM: 8 KB, RAM: 512 bytes

78K0/KD1 Series: 52-pin (10 × 10 mm 0.65 mm pitch)

μPD78F0124	Flash memory: 32 KB, RAM: 1 KB
μPD780124	Mask ROM: 32 KB, RAM: 1 KB
μPD780123	Mask ROM: 24 KB, RAM: 1 KB
μPD780122	Mask ROM: 16 KB, RAM: 512 bytes
μPD780121	Mask ROM: 8 KB, RAM: 512 bytes

78K0/KE1 Series: 64-pin (10 × 10 mm 0.5 mm pitch, 12 × 12 mm 0.65 mm pitch, 14 × 14 mm 0.8 mm pitch)

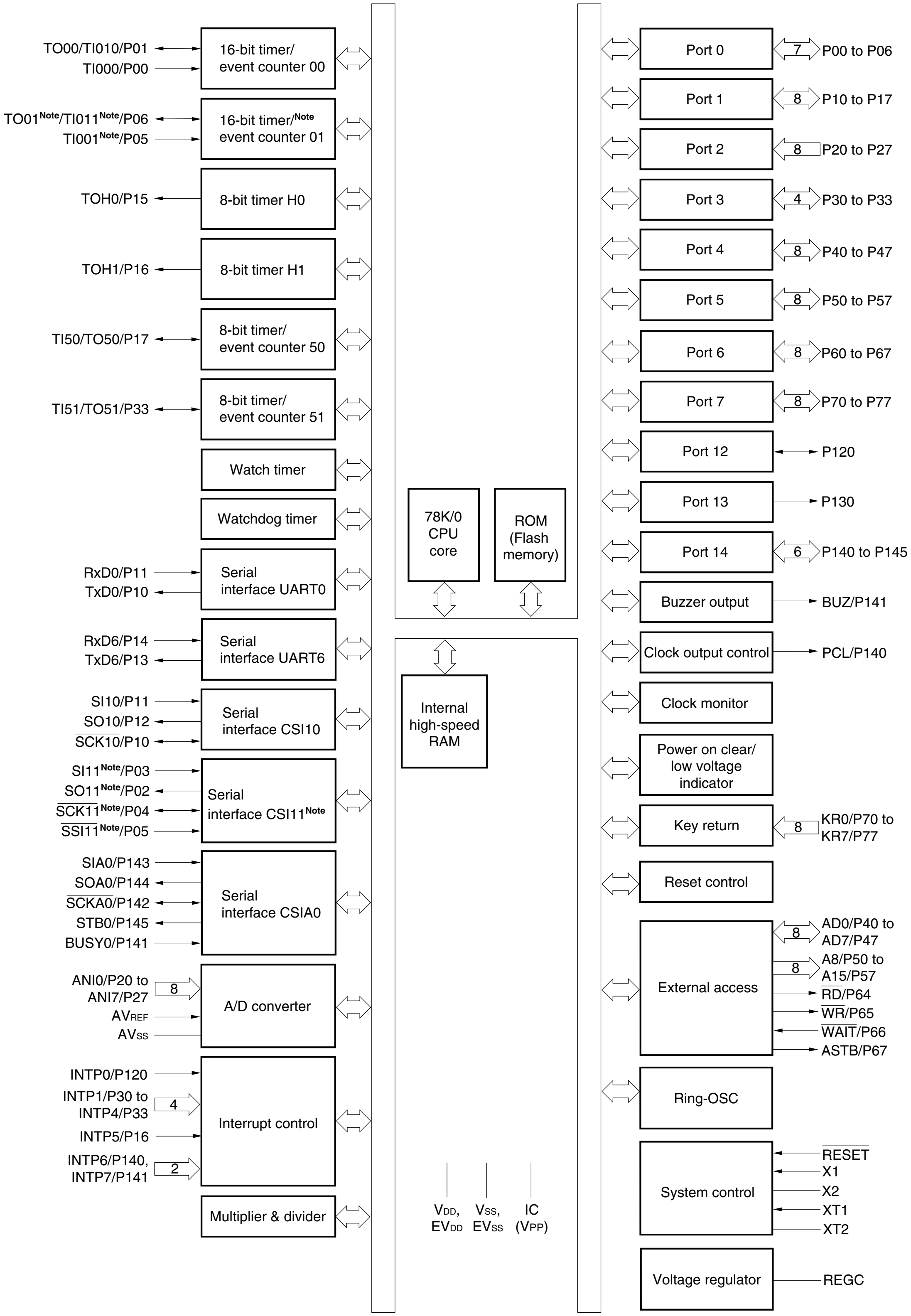
μPD78F0134	Flash memory: 32 KB, RAM: 1 KB
μPD780134	Mask ROM: 32 KB, RAM: 1 KB
μPD780133	Mask ROM: 24 KB, RAM: 1 KB
μPD780132	Mask ROM: 16 KB, RAM: 512 bytes
μPD780131	Mask ROM: 8 KB, RAM: 512 bytes

μPD78F0138	Flash memory: 60 KB, RAM: 2 KB
μPD780138	Mask ROM: 60 KB, RAM: 2 KB
μPD780136	Mask ROM: 48 KB, RAM: 2 KB

78K0/KF1 Series: 80-pin (12 × 12 mm 0.5 mm pitch, 14 × 14 mm 0.65 mm pitch)

μPD78F0148	Flash memory: 60 KB, RAM: 2 KB
μPD780148	Mask ROM: 60 KB, RAM: 2 KB
μPD780146	Mask ROM: 48 KB, RAM: 2 KB
μPD780144	Mask ROM: 32 KB, RAM: 1 KB
μPD780143	Mask ROM: 24 KB, RAM: 1 KB

1.7 Block Diagram



Note μPD780146, 780148, and 78F0148 only.

Remark Items in parentheses are available only in the μPD78F0148.

1.8 Outline of Functions

(1/2)

Item		μPD780143	μPD780144	μPD780146	μPD780148	μPD78F0148
Internal memory	ROM	24 KB	32 KB	48 KB	60 KB	60 KB[Note] (flash memory)
	High-speed RAM	1 KB				
	Expansion RAM	–		1 KB		1 KB[Note]
	Buffer RAM	32 bytes				
Memory space		64 KB				
X1 input clock (oscillation frequency)		Ceramic/crystal/external clock oscillation				
	REGC pin is directly connected to V_{DD}	10 MHz: V_{DD} = 4.0 to 5.5 V, 8.38 MHz: V_{DD} = 3.3 to 5.5 V, 5 MHz: V_{DD} = 2.7 to 5.5 V				
	0.1 μF capacitor is connected to REGC pin	8.38 MHz: V_{DD} = 3.3 to 5.5 V, 5 MHz: V_{DD} = 2.7 to 5.5 V				
Ring-OSC clock (oscillation frequency)		On-chip Ring oscillation (240 kHz (TYP.))				
Subsystem clock (oscillation frequency)		Crystal/external clock oscillation (32.768 kHz)				
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 (X1 input clock: @ f_{XP} = 10 MHz operation)				
		8.3 μs/16.6 μs/33.2 μs/66.4 μs/132.8 μs (TYP.) (Ring-OSC clock: @ f_R = 240 kHz (TYP.) operation)				
		122 μs (subsystem clock: @ f_{XT} = 32.768 kHz operation)				
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits × 4 banks) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.				
I/O ports		Total: 67 CMOS I/O 54 CMOS input 8 CMOS output 1 N-ch open-drain output 4				
Timers		• 16-bit timer/event counter: 2 channels (1 channel only in the μPD780143, 780144) • 8-bit timer/event counter: 2 channels • 8-bit timer: 2 channels • Watch timer 1 channel • Watchdog timer: 1 channel				
Timer outputs		5 (PWM output: 3)		6 (PWM output: 3)		
Clock output		• 78.125 kHz, 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (X1 input clock: 10 MHz) • 32.768 kHz (subsystem clock: 32.768 kHz)				
Buzzer output		1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 kHz (X1 input clock: 10 MHz)				
A/D converter		10-bit resolution × 8 channels				

Note The internal flash memory capacity and internal expansion RAM capacity can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

(2/2)

Item		μPD780143	μPD780144	μPD780146	μPD780148	μPD78F0148
Serial interface		• UART mode supporting LIN-bus: 1 channel • 3-wire serial I/O mode: 1 channel (No channel in the μPD780143, 780144) • 3-wire serial I/O mode with automatic transmit/receive function: 1 channel • 3-wire serial I/O mode/UART mode[Note]: 1 channel				
Multiplier/divider		• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 16 bits = 32 bits remainder of 16 bits (division)				
Vectored interrupt sources	Internal	17		20		
	External	9				
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of key input pins (KR0 to KR7).				
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by clock monitor • Internal reset by power-on-reset • Internal reset by low-voltage detector				
Supply voltage		V_{DD} = 2.7 to 5.5 V				
Operating ambient temperature		Standard products, (A) products: T_A = –40 to +85°C (A1) products: T_A = –40 to +110°C (μPD780143, 780144, 780146, and 780148 only)				
Package		• 80-pin plastic QFP (14 × 14) • 80-pin plastic TQFP (fine pitch) (12 × 12)				

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

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters 00 and 01[Note 1]	8-Bit Timer/ Event Counters 50 and 51	8-Bit Timers H0 and H1	Watch Timer	Watchdog Timer
Operation mode	Interval timer	2 channels	2 channels	2 channels	1 channel[Note 2]	1 channel
	External event counter	2 channels	2 channels	–	–	–
Function	Timer output	2 outputs	2 outputs	2 outputs	–	–
	PPG output	2 outputs	–	–	–	–
	PWM output	–	2 outputs	2 outputs	–	–
	Pulse width measurement	4 inputs	–	–	–	–
	Square-wave output	2 outputs	2 outputs	–	–	–
	Interrupt source	4	2	2	1	–

Notes 1. 16-bit timer/event counter 01 is available only in the μPD780146, 780148, and 78F0148.

2. In the watch timer, the watch timer function and interval timer function can be used simultaneously.

2.1 Pin Function List

(1) Port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 7-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI000
P01				TI010/TO00
P02				SO11$^{\text{Note}}$
P03				SI11$^{\text{Note}}$
P04				$\overline{\text{SCK11}}$$^{\text{Note}}$
P05				$\overline{\text{SSI11}}$$^{\text{Note}}$/TI001$^{\text{Note}}$
P06				TI011$^{\text{Note}}$/TO01$^{\text{Note}}$
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	$\overline{\text{SCK10}}$/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 to P27	Input	Port 2. 8-bit input-only port.	Input	ANI0 to ANI7
P30 to P32	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP1 to INTP3
P33				INTP4/TI51/TO51
P40 to P47	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	AD0 to AD7
P50 to P57	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	A8 to A15

Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

(1) Port pins (2/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P60 to P63	I/O	Port 6. 8-bit I/O port. Input/output can be specified in 1-bit units.	N-ch open-drain I/O port. Use of an on-chip pull-up resistor can be specified by a mask option only for mask ROM versions.	Input	–
P64			Use of an on-chip pull-up resistor can be specified by a software setting.		$\overline{\text{RD}}$
P65					$\overline{\text{WR}}$
P66					$\overline{\text{WAIT}}$
P67					ASTB
P70 to P77	I/O	Port 7. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	KR0 to KR7
P120	I/O	Port 12. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	INTP0
P130	Output	Port 13. 1-bit output-only port.		Output	–
P140	I/O	Port 14. 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	PCL/INTP6
P141					BUZ/BUSY0/ INTP7
P142					$\overline{\text{SCKA0}}$
P143					SIA0
P144					SOA0
P145					STB0

(2) Non-port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P120
INTP1 to INTP3				P30 to P32
INTP4				P33/TI51/TO51
INTP5				P16/TOH1
INTP6				P140/PCL
INTP7				P141/BUZ/BUSY0
SI10	Input	Serial data input to serial interface	Input	P11/RxD0
SI11[Note]				P03
SIA0				P143
SO10	Output	Serial data output from serial interface	Input	P12
SO11[Note]				P02
SOA0				P144
$\overline{\text{SCK10}}$	I/O	Clock input/output for serial interface	Input	P10/TxD0
$\overline{\text{SCK11}}$[Note]				P04
$\overline{\text{SCKA0}}$				P142
$\overline{\text{SSI11}}$[Note]	Input	Serial interface chip select input	Input	P05/TI001
BUSY0	Input	Serial interface busy input	Input	P141/BUZ/INTP7
STB0	Output	Serial interface strobe output	Input	P145
RxD0	Input	Serial data input to asynchronous serial interface	Input	P11/SI10
RxD6				P14
TxD0	Output	Serial data output from asynchronous serial interface	Input	P10/$\overline{\text{SCK10}}$
TxD6				P13
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input	P00
TI001[Note]		External count clock input to 16-bit timer/event counter 01 Capture trigger input to capture registers (CR001, CR011) of 16-bit timer/event counter 01		P05/$\overline{\text{SSI11}}$[Note]
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TI011[Note]		Capture trigger input to capture register (CR001) of 16-bit timer/event counter 01		P06/TO01[Note]
TO00	Output	16-bit timer/event counter 00 output	Input	P01/TI010
TO01[Note]		16-bit timer/event counter 01 output		P06/TI011[Note]
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P17/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P33/TO51/INTP4
TO50	Output	8-bit timer/event counter 50 output	Input	P17/TI50
TO51		8-bit timer/event counter 51 output		P33/TI51/INTP4
TOH0		8-bit timer H0 output		P15
TOH1		8-bit timer H1 output		P16/INTP5

Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

(2) Non-port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
PCL	Output	Clock output (for trimming of X1 input clock, subsystem clock)	Input	P140/INTP6
BUZ	Output	Buzzer output	Input	P141/INTP7/BUSY0
AD0 to AD7	I/O	Lower address/data bus for external memory expansion	Input	P40 to P47
A8 to A15	Output	Higher address bus for external memory expansion	Input	P50 to P57
$\overline{\text{RD}}$	Output	Strobe signal output for external memory read operation	Input	P64
$\overline{\text{WR}}$	Output	Strobe signal output for external memory write operation	Input	P65
$\overline{\text{WAIT}}$	Input	Wait insertion on external memory access	Input	P66
ASTB	Output	Strobe output that externally latches address information output to ports 4 and 5 for access to external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P20 to P27
AV_{REF}	Input	A/D converter reference voltage input	–	–
AV_{SS}	–	A/D converter ground potential. Make the same potential as EV_{SS} or V_{SS}.	–	–
KR0 to KR7	Input	Key interrupt input	Input	P70 to P77
REGC	–	Connecting regulator output stabilization capacitor. When using the regulator, connect to V_{SS} via a 0.1 μF capacitor. When the regulator is not used, connect directly to V_{DD}.	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
X1	Input	Connecting crystal resonator for X1 input clock oscillation	–	–
X2	–		–	–
XT1	Input	Connecting crystal resonator for subsystem clock oscillation	–	–
XT2	–		–	–
V_{DD}	–	Positive power supply (except for ports)	–	–
EV_{DD}	–	Positive power supply for ports	–	–
V_{SS}	–	Ground potential (except for ports)	–	–
EV_{SS}	–	Ground potential for ports	–	–
IC	–	Internally connected. Connect directly to EV_{SS} or V_{SS}.	–	–
V_{PP}	–	Flash memory programming mode setting. High-voltage application for program write/verify. Connect directly to EV_{SS} or V_{SS} in normal operation mode.	–	–

2.2 Description of Pin Functions

2.2.1 P00 to P06 (port 0)

P00 to P06 function as a 7-bit I/O port. These pins also function as timer I/O, serial interface data I/O, clock I/O, and chip select input.

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

(1) Port mode

P00 to P06 function as a 7-bit I/O port. P00 to P06 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 P06 function as timer I/O.

(a) TI000, TI001[Note]

These are the pins for inputting an external count clock to 16-bit timer/event counters 00 and 01 and are also for inputting a capture trigger signal to the capture registers (CR000, CR010 or CR001, CR011) of 16-bit timer/event counters 00 and 01.

(b) TI010, TI011[Note]

These are the pins for inputting a capture trigger signal to the capture register (CR000 or CR001) of 16-bit timer/event counters 00 and 01.

(c) TO00, TO01[Note]

These are timer output pins.

(d) SI11[Note], SO11[Note]

These are serial interface serial data I/O pins.

(e) $\overline{\text{SCK11}}$[Note]

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

(f) $\overline{\text{SSI11}}$[Note]

This is the serial interface chip select input pin.

Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

2.2.2 P10 to P17 (port 1)

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

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

(1) Port mode

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

(2) Control mode

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

(a) SI10, SO10

These are serial interface serial data I/O pins.

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

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

(c) RxD0, RxD6, TxD0, and TxD6

These are the serial data I/O pins of the asynchronous serial interface.

(d) TI50

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

(e) TO50, TOH0, and TOH1

These are timer output pins.

(f) INTP5

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

2.2.3 P20 to P27 (port 2)

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

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

(1) Port mode

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

(2) Control mode

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7).

2.2.4 P30 to P33 (port 3)

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

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

(1) Port mode

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

(2) Control mode

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

(a) INTP1 to INTP4

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

(b) TI51

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

(c) TO51

This is a timer output pin.

2.2.5 P40 to P47 (port 4)

P40 to P47 function as an 8-bit I/O port. These pins also function as address/data bus pins.

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

(1) Port mode

P40 to P47 function as an 8-bit I/O port. P40 to P47 can be set to input or output in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

(2) Control mode

P40 to P47 function as the pins for the lower address/data bus (AD0 to AD7) in external memory expansion mode.

2.2.6 P50 to P57 (port 5)

P50 to P57 function as an 8-bit I/O port. These pins also function as address bus pins.

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

(1) Port mode

P50 to P57 function as an 8-bit I/O port. P50 to P57 can be set to input or output in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

(2) Control mode

P50 to P57 function as the pins for the higher address bus (A8 to A15) in external memory expansion mode.

2.2.7 P60 to P67 (port 6)

P60 to P67 function as an 8-bit I/O port. These pins also function as control pins in external memory expansion mode.

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

(1) Port mode

P60 to P67 function as an 8-bit I/O port. P60 to P67 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

P60 to P63 are N-ch open-drain pins. Use of an on-chip pull-up resistor can be specified by a mask option only for mask ROM versions.

Use of an on-chip pull-up resistor can be specified for P64 to P67 by pull-up resistor option register 6 (PU6).

(2) Control mode

P64 to P67 function as control signal output pins ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB) in external memory expansion mode.

Caution P66 functions as an I/O port if the external wait is not used in external memory expansion mode.

2.2.8 P70 to P77 (port 7)

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

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

(1) Port mode

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

(2) Control mode

P70 to P77 function as key interrupt input pins.

2.2.9 P120 (port 12)

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

The following operation modes can be specified.

(1) Port mode

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

(2) Control mode

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

2.2.10 P130 (port 13)

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

2.2.11 P140 to P145 (port 14)

P140 to P145 function as a 6-bit I/O port. These pins also function as external interrupt request input, clock output, buzzer output, serial interface data I/O, clock I/O, busy input, and strobe output pins.

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

(1) Port mode

P140 to P145 function as a 6-bit I/O port. P140 to P145 can be set to input or output in 1-bit units using port mode register 14 (PM14). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

(2) Control mode

P140 to P145 function as external interrupt request input, clock output, buzzer output, serial interface data I/O, clock I/O, busy input, and strobe output pins.

(a) INTP6, INTP7

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

(b) PCL

This is a clock output pin.

(c) BUZ

This is a buzzer output pin.

(d) SIA0, SOA

These are serial interface serial data I/O pins.

(e) $\overline{\text{SCKA0}}$

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

(f) BUSY0

This is the serial interface busy input pin.

(g) STB0

This is the serial interface strobe output pin.

2.2.12 AV_{REF}

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

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

2.2.13 AV_{SS}

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

2.2.14 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

2.2.15 REGC

This is the pin for connecting the capacitor for the regulator. When using the regulator, connect this pin to V_{SS} via a 0.1 μF capacitor. When the regulator is not used, connect this pin directly to V_{DD} and use it with the same potential as V_{DD} pin.

2.2.16 X1 and X2

These are the pins for connecting a crystal resonator for X1 input clock oscillation.

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

2.2.17 XT1 and XT2

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

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

2.2.18 V_{DD} and EV_{DD}

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

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

2.2.19 V_{SS} and EV_{SS}

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

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

2.2.20 V_{PP} (flash memory versions only)

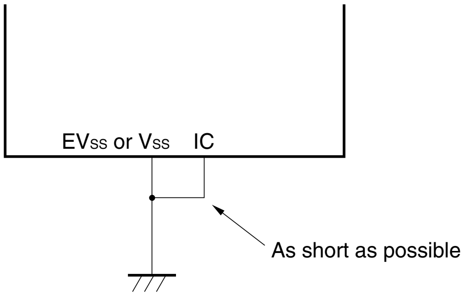
This is a pin for flash memory programming mode setting and high-voltage application for program write/verify. Connect directly to EV_{SS} or V_{SS} in the normal operation mode.

2.2.21 IC (mask ROM versions only)

The IC (Internally Connected) pin is provided to set the test mode to check the 78K0/KF1 Series at shipment. Connect it directly to EV_{SS} or V_{SS} pin with the shortest possible wire in the normal operation mode.

When a potential difference is produced between the IC pin and the EV_{SS} or 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 EV_{SS} or V_{SS}.**



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

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

Table 2-1. Pin I/O Circuit Types (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000	8-A	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01/TI010/TO00			
P02/SO11[Note]			
P03/SI11[Note]			
P04/$\overline{\text{SCK11}}$[Note]			
P05/$\overline{\text{SSI11}}$[Note]/TI001[Note]			
P06/TI011[Note]/TO01[Note]			
P10/$\overline{\text{SCK10}}$/TxD0[Note]			
P11/SI10/RxD0[Note]			
P12/SO10	5-A		
P13/TxD6			
P14/RxD6	8-A		
P15/TOH0	5-A		
P16/TOH1/INTP5	8-A		
P17/TI50/TO50			
P20/ANI0 to P27/ANI7	9-C	Input	Connect to EV_{DD} or EV_{SS}.
P30/INTP1 to P32/INTP3	8-A	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P33/TI51/TO51/INTP4			
P40/AD0 to P47/AD7	5-A		
P50/A8 to P57/A15			
P60, P61 (Mask ROM version)	13-S		Input: Independently connect to EV_{DD} via a resistor. Output: Leave open.
P60, P61 (Flash memory version)	13-R		
P62, P63 (Mask ROM version)	13-W		
P62, P63 (Flash memory version)	13-V		
P64/$\overline{\text{WD}}$	5-A		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P65/$\overline{\text{WR}}$			
P66/$\overline{\text{WAIT}}$			
P67/ASTB			
P70/KR0 to P77/KR7	8-A		
P120/INTP0			

Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

Table 2-1. Pin I/O Circuit Types (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P130	3-C	Output	Leave open.
P140/PCL/INTP6	8-A	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P141/BUZ/BUSY0/INTP7			
P142/$\overline{\text{SCKA0}}$			
P143/SIA0			
P144/SOA0	5-A		
P145/STB			
$\overline{\text{RESET}}$	2	Input	–
XT1	16		Connect directly to EV_{DD} or V_{DD}.
XT2		–	Leave open.
AV_{REF}	–	–	Connect directly to EV_{DD} or V_{DD}.
AV_{SS}	–	–	Connect directly to EV_{SS} or V_{SS}.
IC	–	–	Connect directly to EV_{SS} or V_{SS}.
V_{PP}			

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

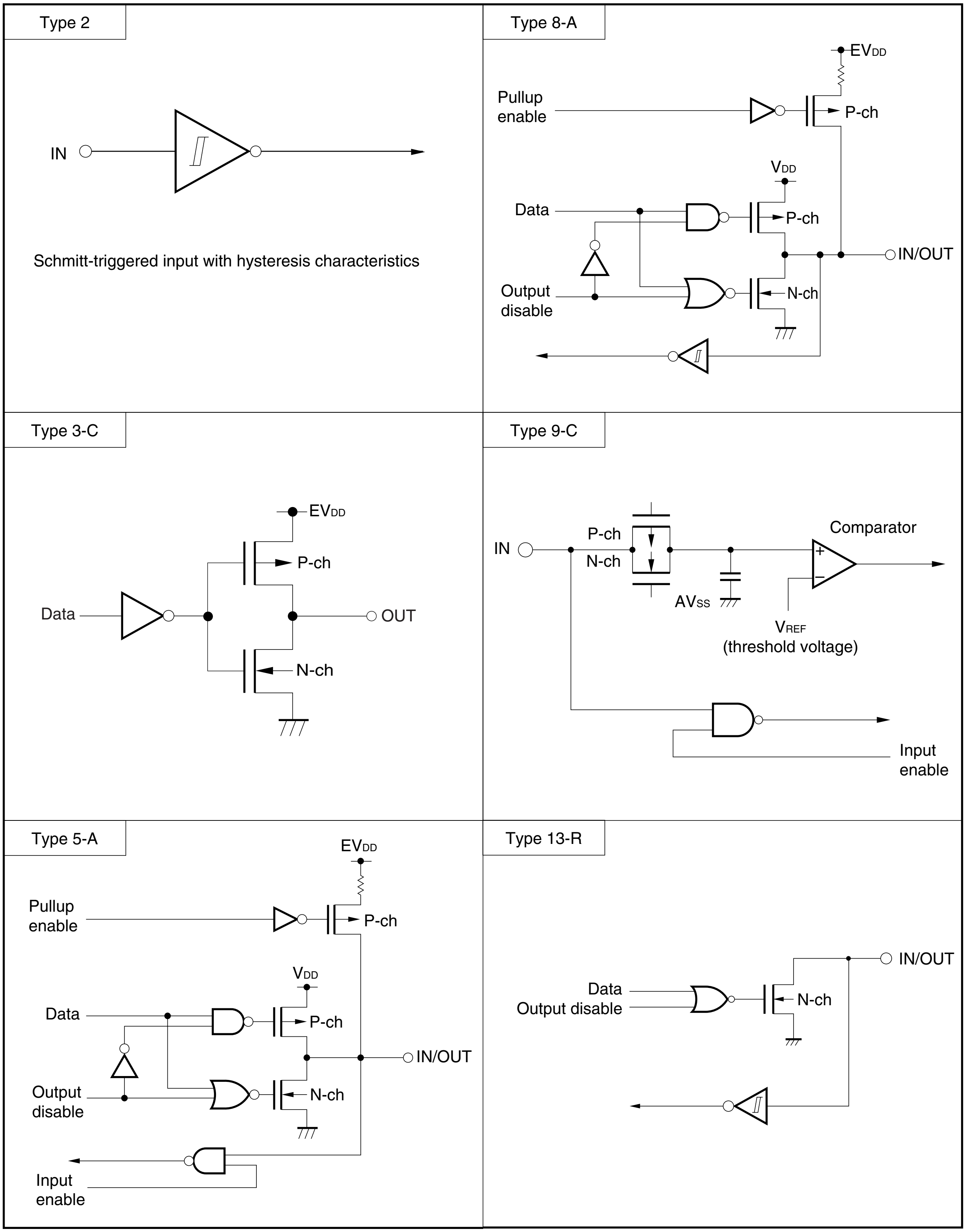
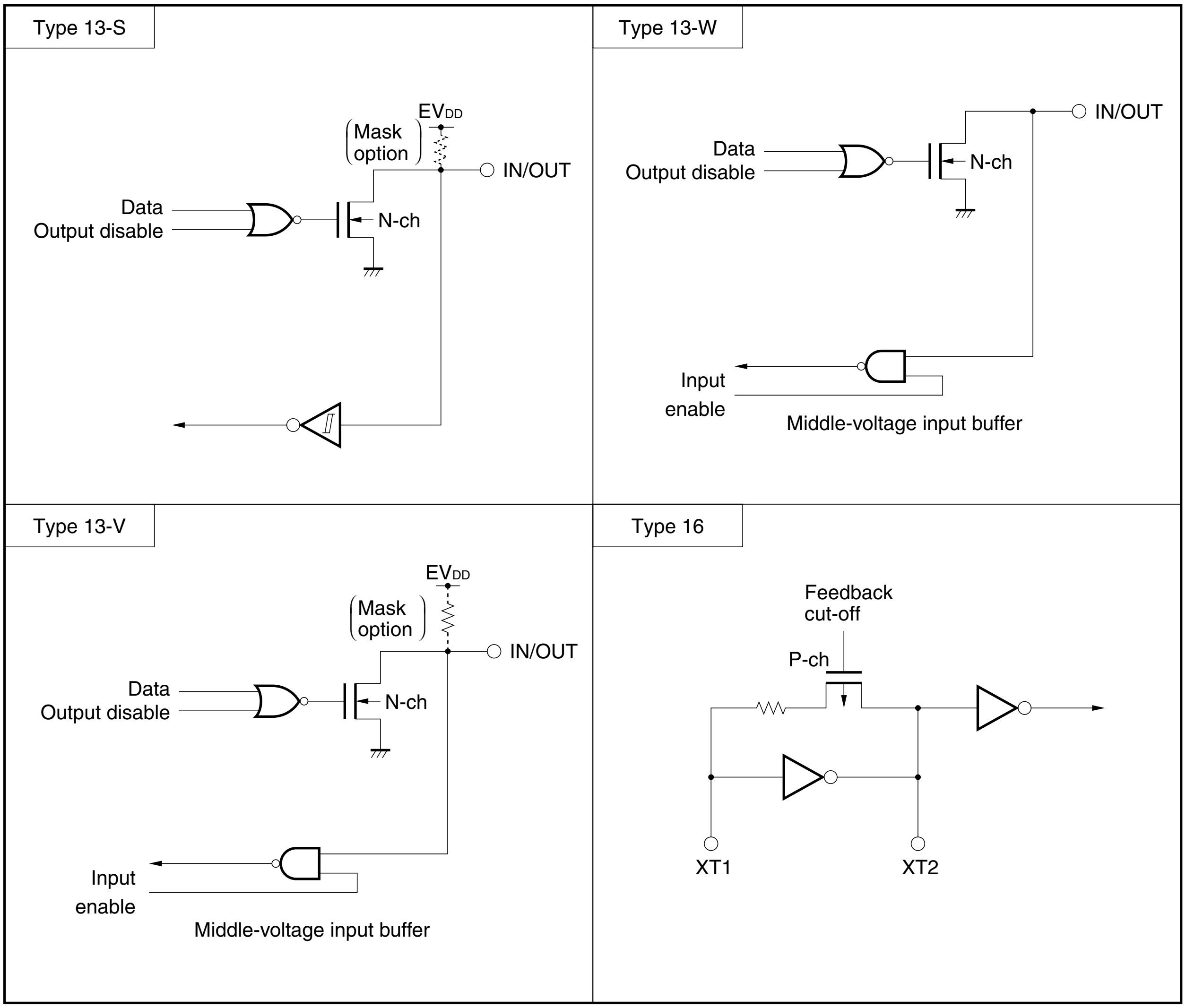


Figure 2-1. Pin I/O Circuit List (2/2)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the 78K0/KF1 Series can each access a 64 KB memory space. Figures 3-1 to 3-5 show the memory maps.

Caution Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products in the 78K0/KF1 Series are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

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

	IMS	IXS
μPD780143	C6H	0CH
μPD780144	C8H	
μPD780146	CCH	0AH
μPD780148	CFH	
μPD78F0148	Value corresponding to mask ROM version	

Figure 3-1. Memory Map (μPD780143)

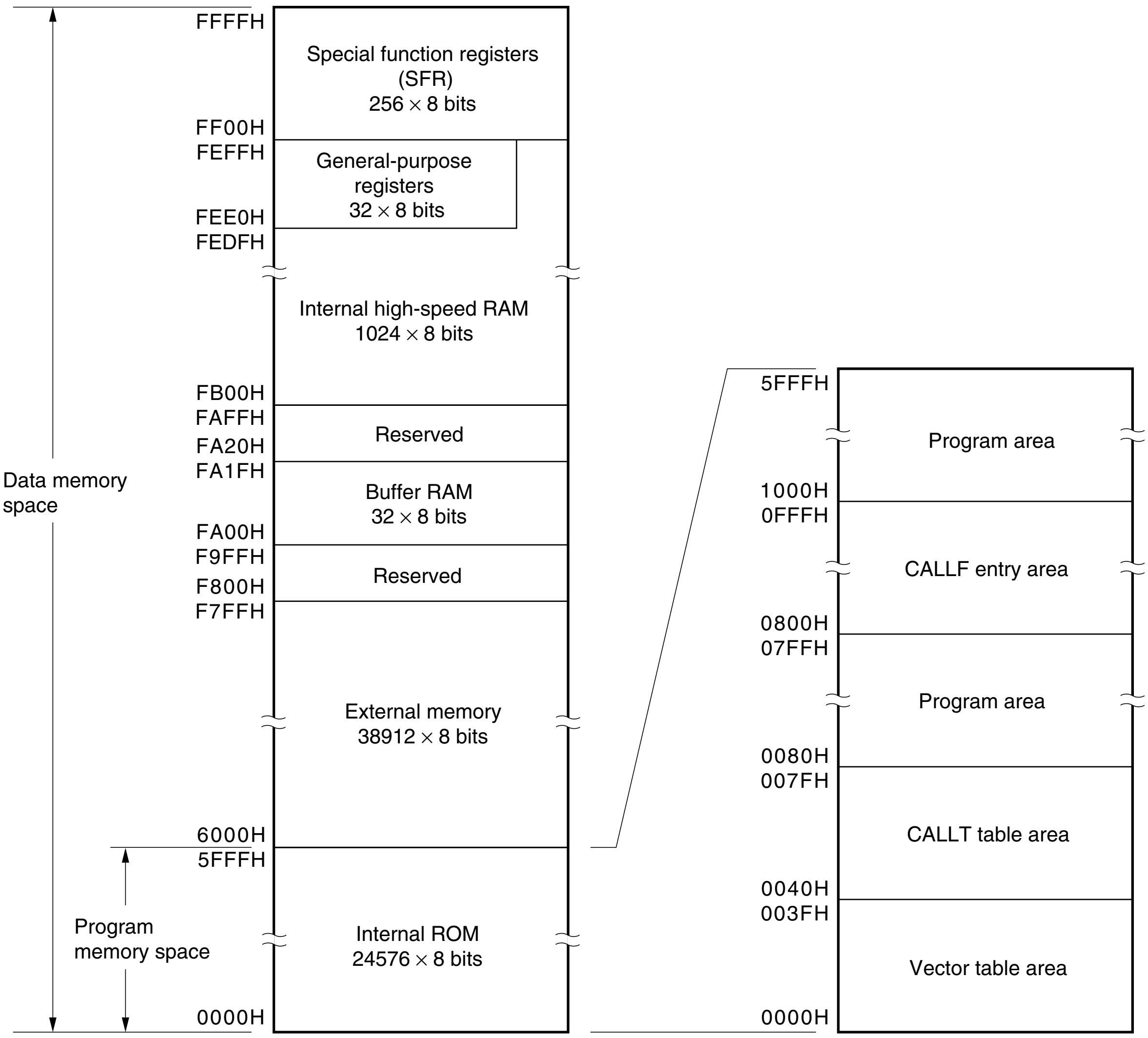


Figure 3-2. Memory Map (μPD780144)

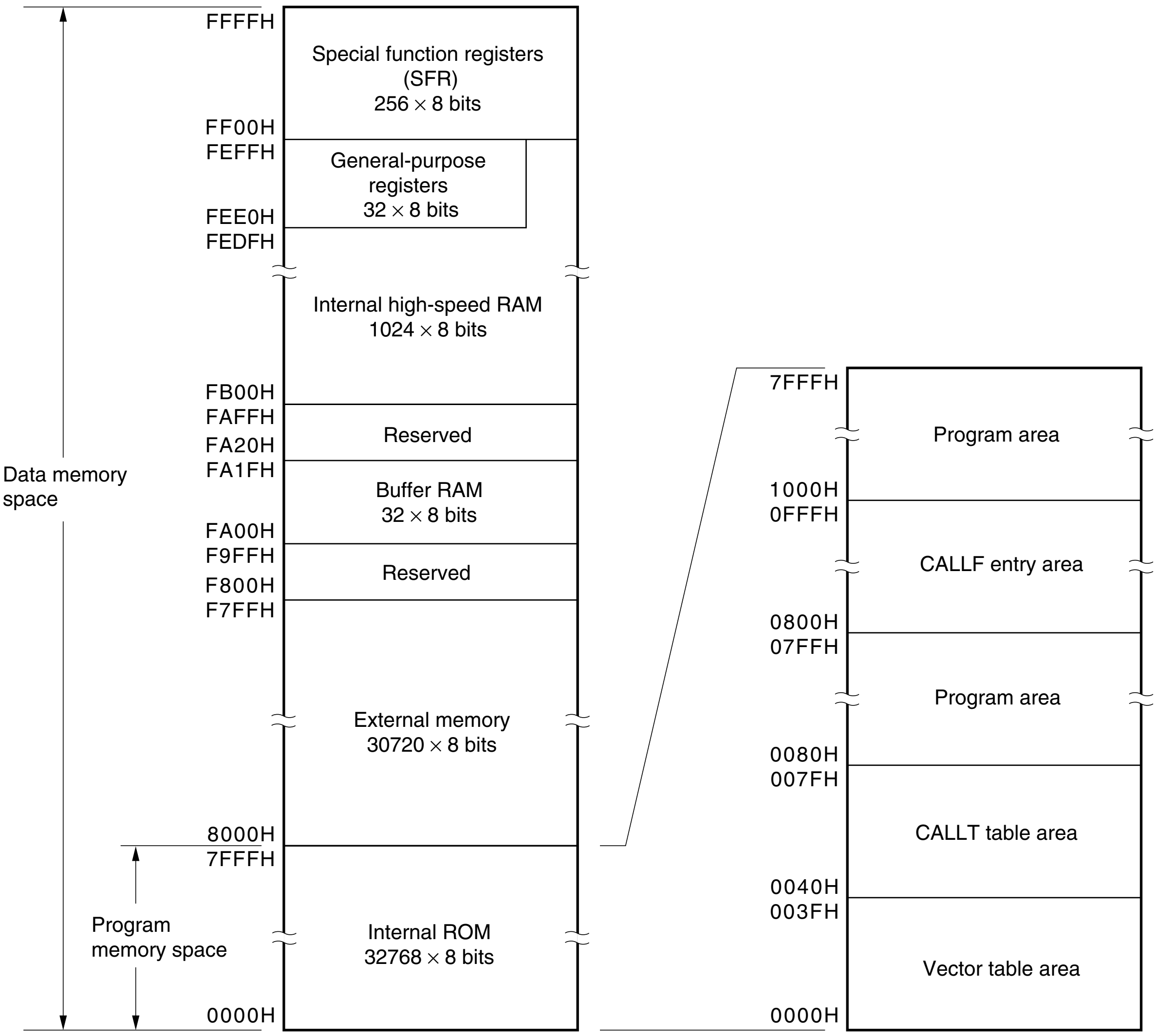


Figure 3-3. Memory Map (μPD780146)

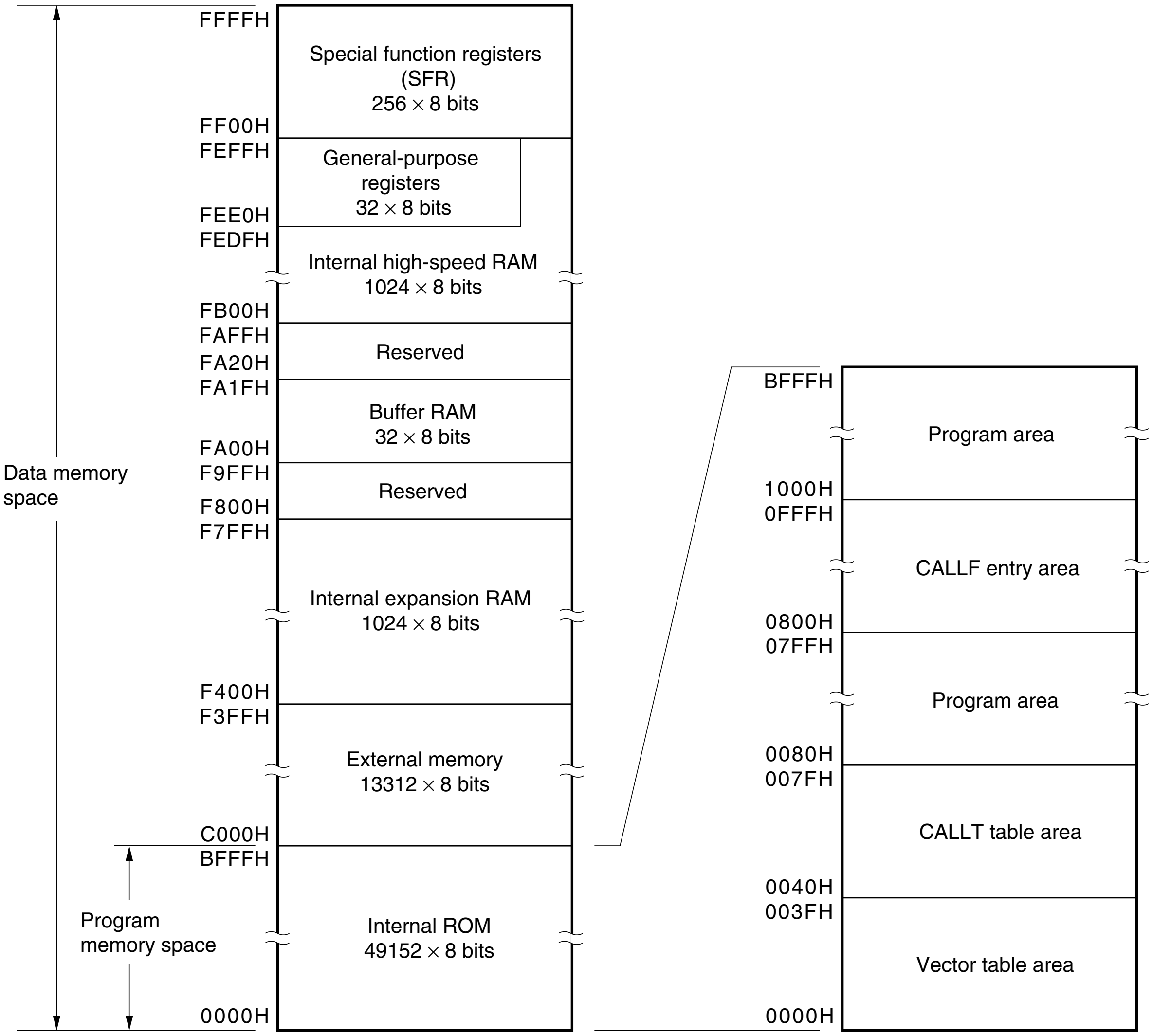


Figure 3-4. Memory Map (μPD780148)

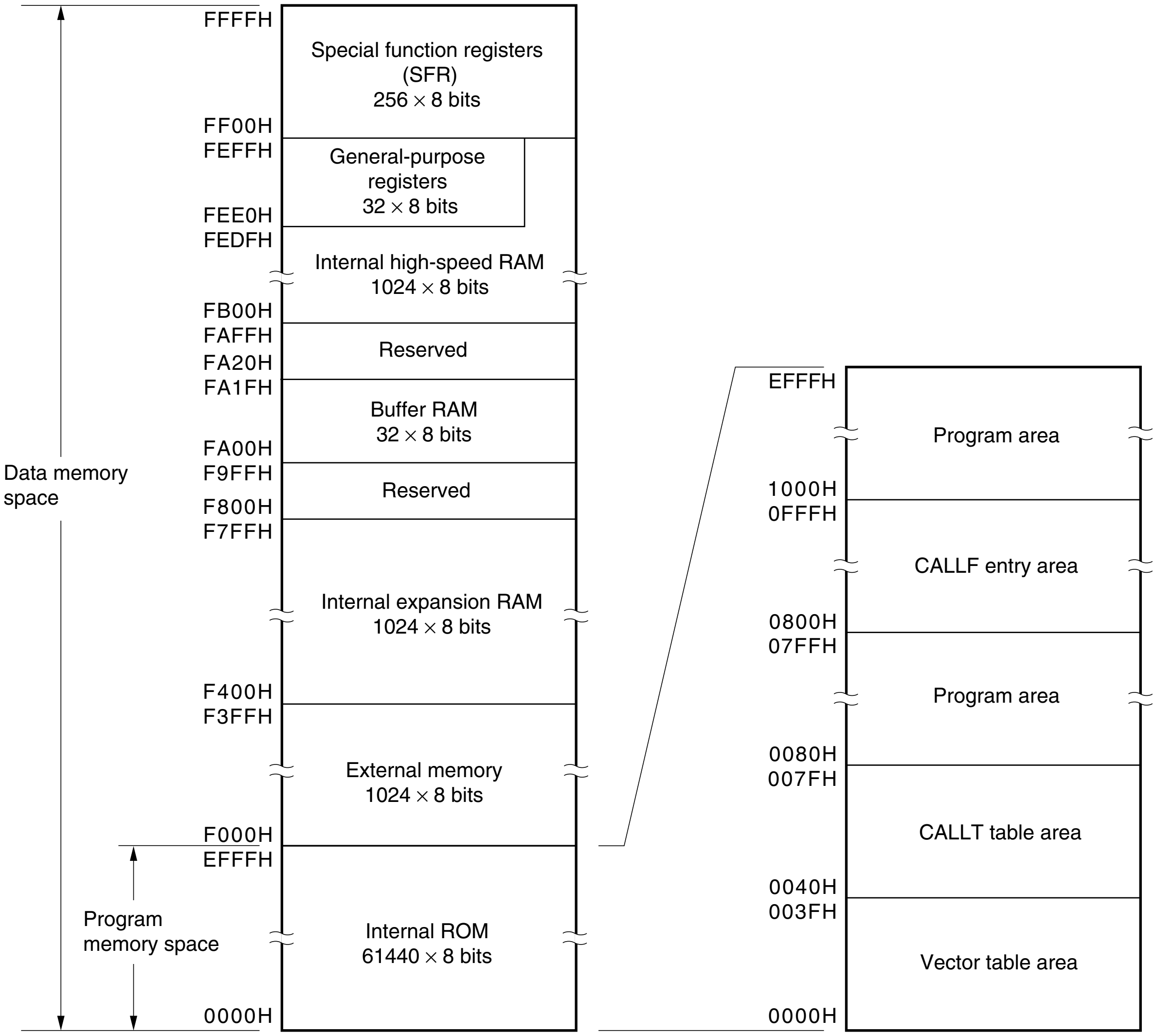
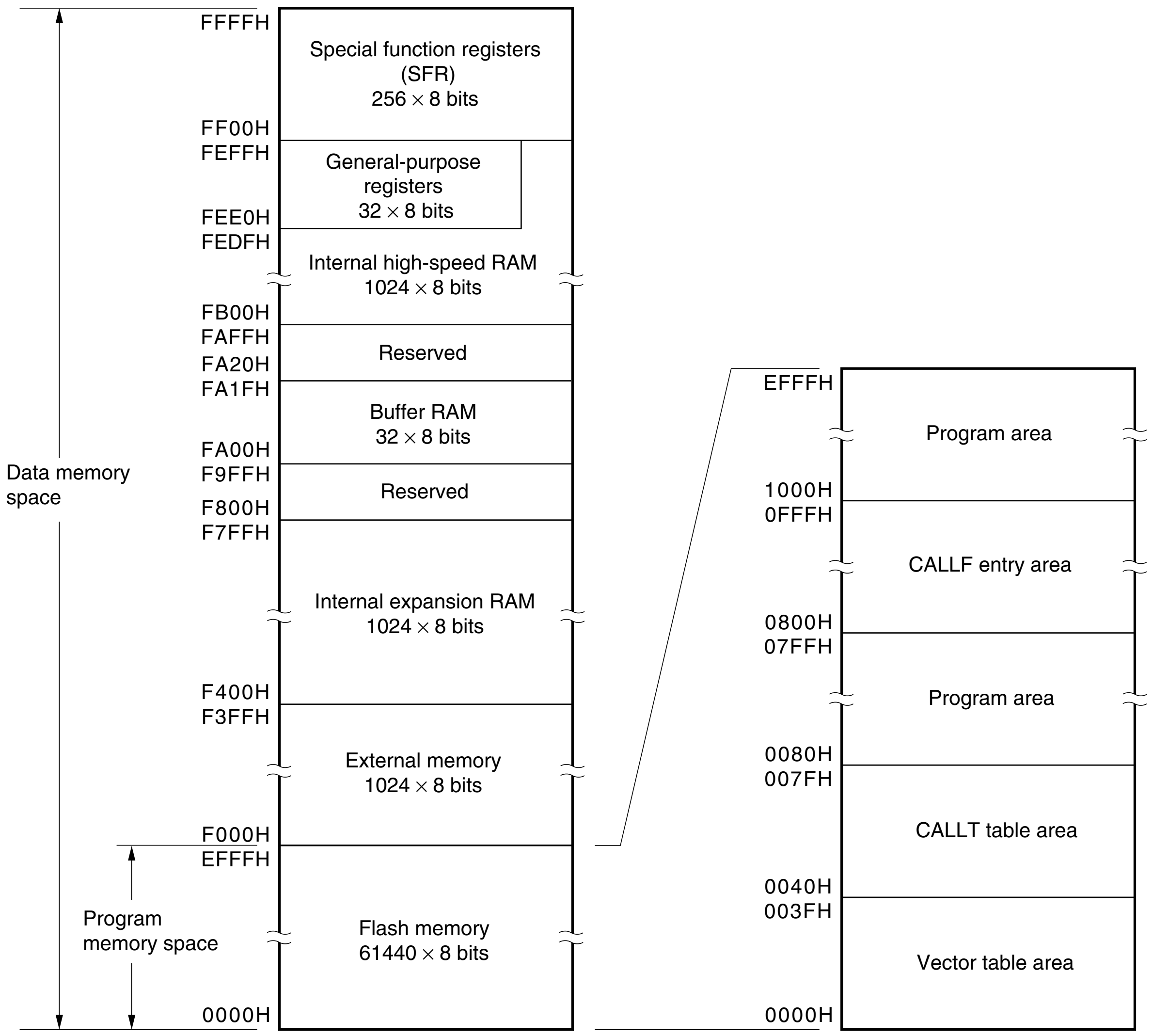


Figure 3-5. Memory Map (μPD78F0148)



3.1.1 Internal program memory space

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

78K0/KF1 Series products incorporate internal ROM (or flash memory), as shown below.

Table 3-2. Internal Memory Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD780143	Mask ROM	24576 × 8 bits (0000H to 5FFFH)
μPD780144		32768 × 8 bits (0000H to 7FFFH)
μPD780146		49152 × 8 bits (0000H to BFFFH)
μPD780148		61440 × 8 bits (0000H to EFFFH)
μPD78F0148	Flash memory	61440 × 8 bits (0000H to EFFFH)

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 $\overline{\text{RESET}}$ 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	0020H	INTTM000
		0022H	INTTM010
0004H	INTLVI	0024H	INTAD
0006H	INTP0	0026H	INTSR0
0008H	INTP1	0028H	INTWTI
000AH	INTP2	002AH	INTTM51
000CH	INTP3	002CH	INTKR
000EH	INTP4	002EH	INTWT
0010H	INTP5	0030H	INTP6
0012H	INTSRE6	0032H	INTP7
0014H	INTSR6	0034H	INTDMU
0016H	INTST6	0036H	INTCSI11[Note]
0018H	INTCSI10/INTST0	0038H	INTTM001[Note]
001AH	INTTMH1	003AH	INTTM011[Note]
001CH	INTTMH0	003CH	INTACSI
001EH	INTTM50		

Note Available only in the μPD780146, 780148, and 78F0148.

(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) CALLF instruction entry area

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

3.1.2 Internal data memory space

78K0/KF1 Series products incorporate the following RAMs.

(1) Internal high-speed RAM

The internal high-speed RAM is allocated to the area FB00H to FEFFH in a 1024 × 8 bits configuration.

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

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

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

(2) Internal expansion RAM

Table 3-4. Internal Expansion RAM Capacity

Part Number	Internal Expansion RAM
μPD780143	–
μPD780144	
μPD780146	1024 × 8 bits (F400H to F7FFH)
μPD780148	
μPD78F0148	

The internal expansion RAM can also be used as a normal data area similar to the internal high-speed RAM, as well as a program area in which instructions can be written and executed.

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 78K0/KF1 Series, 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 3-6 to 3-10. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

Figure 3-6. Data Memory Addressing (μPD780143)

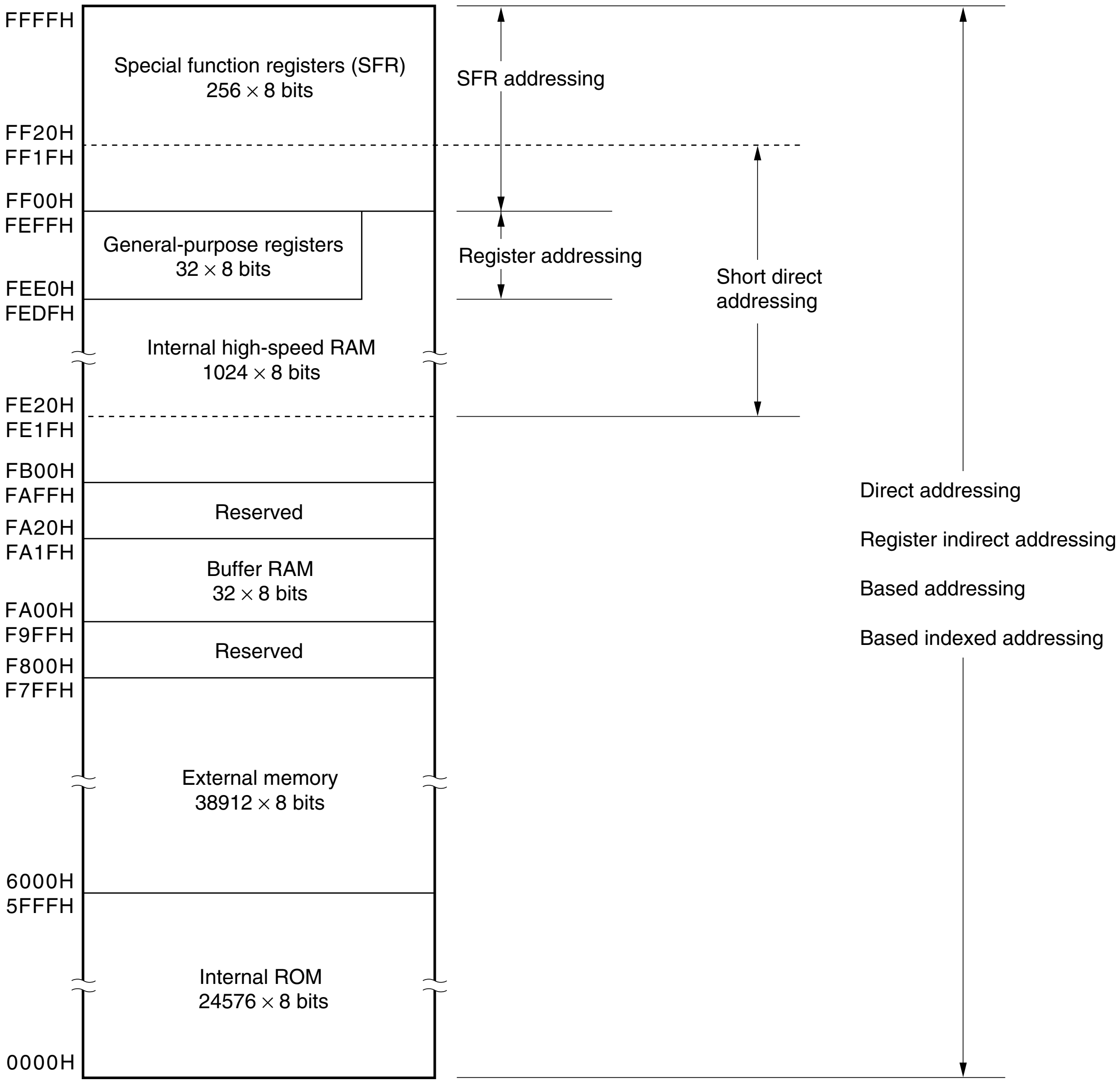


Figure 3-7. Data Memory Addressing (μPD780144)

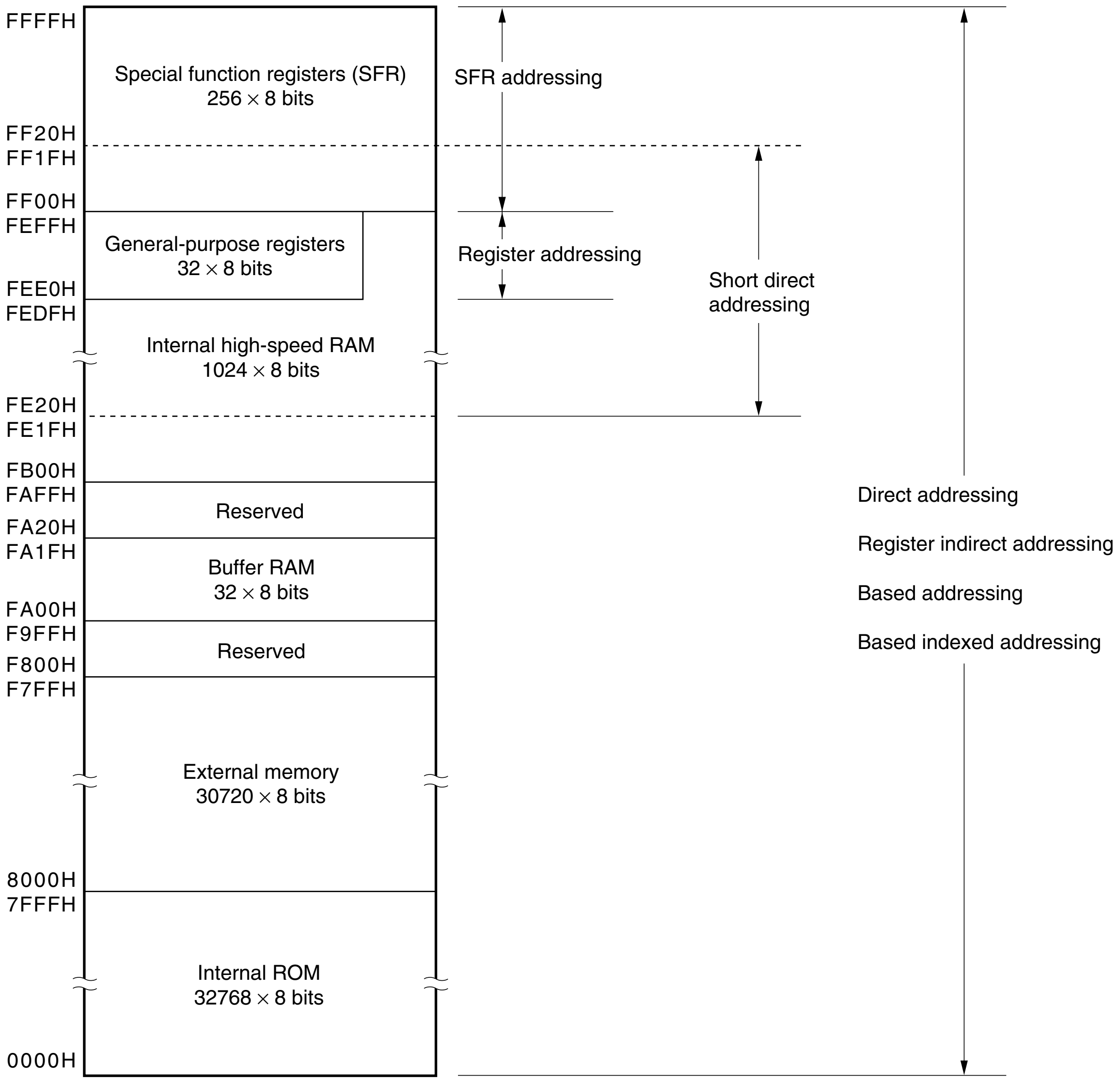


Figure 3-8. Data Memory Addressing (μPD780146)

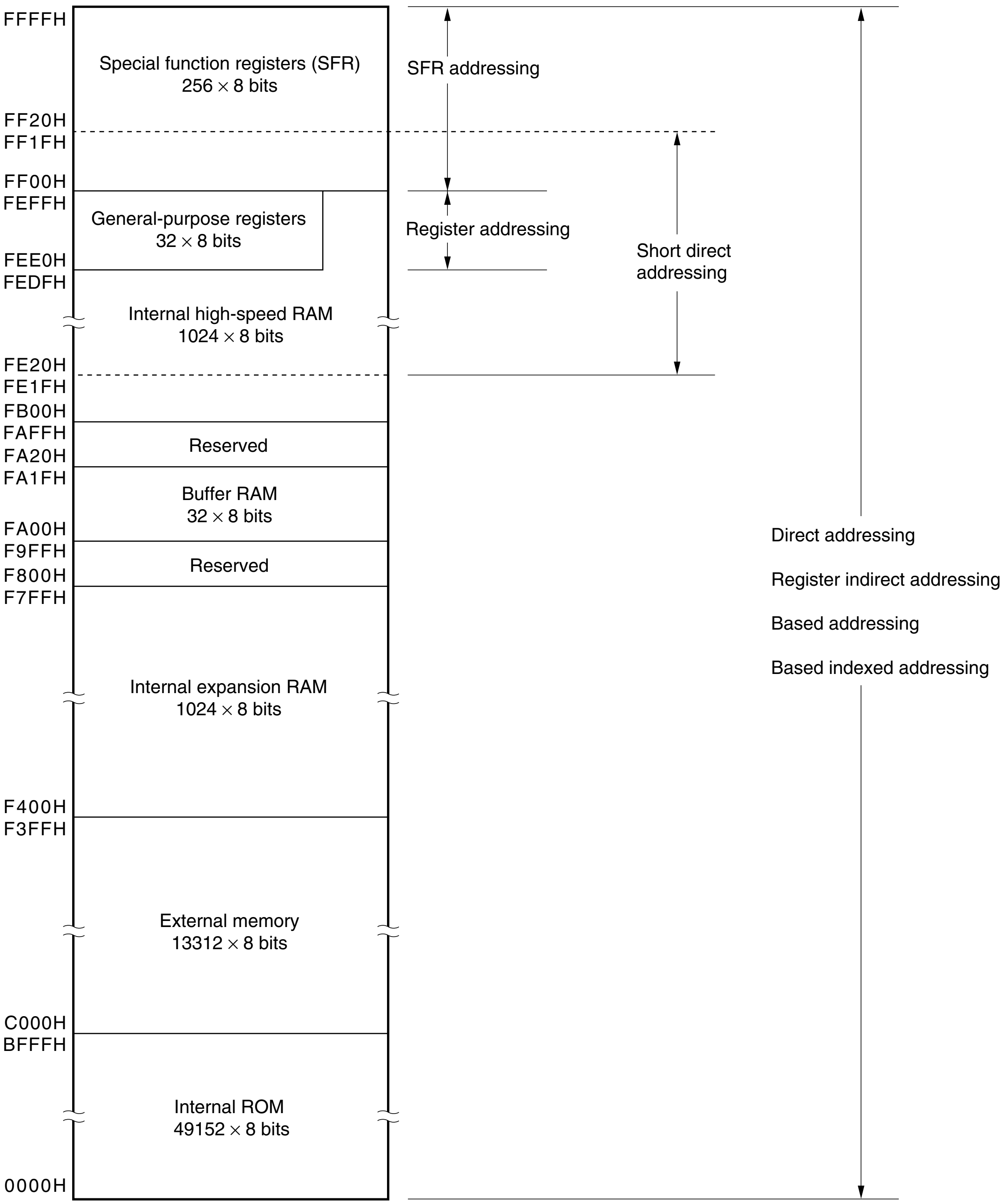


Figure 3-9. Data Memory Addressing (μPD780148)

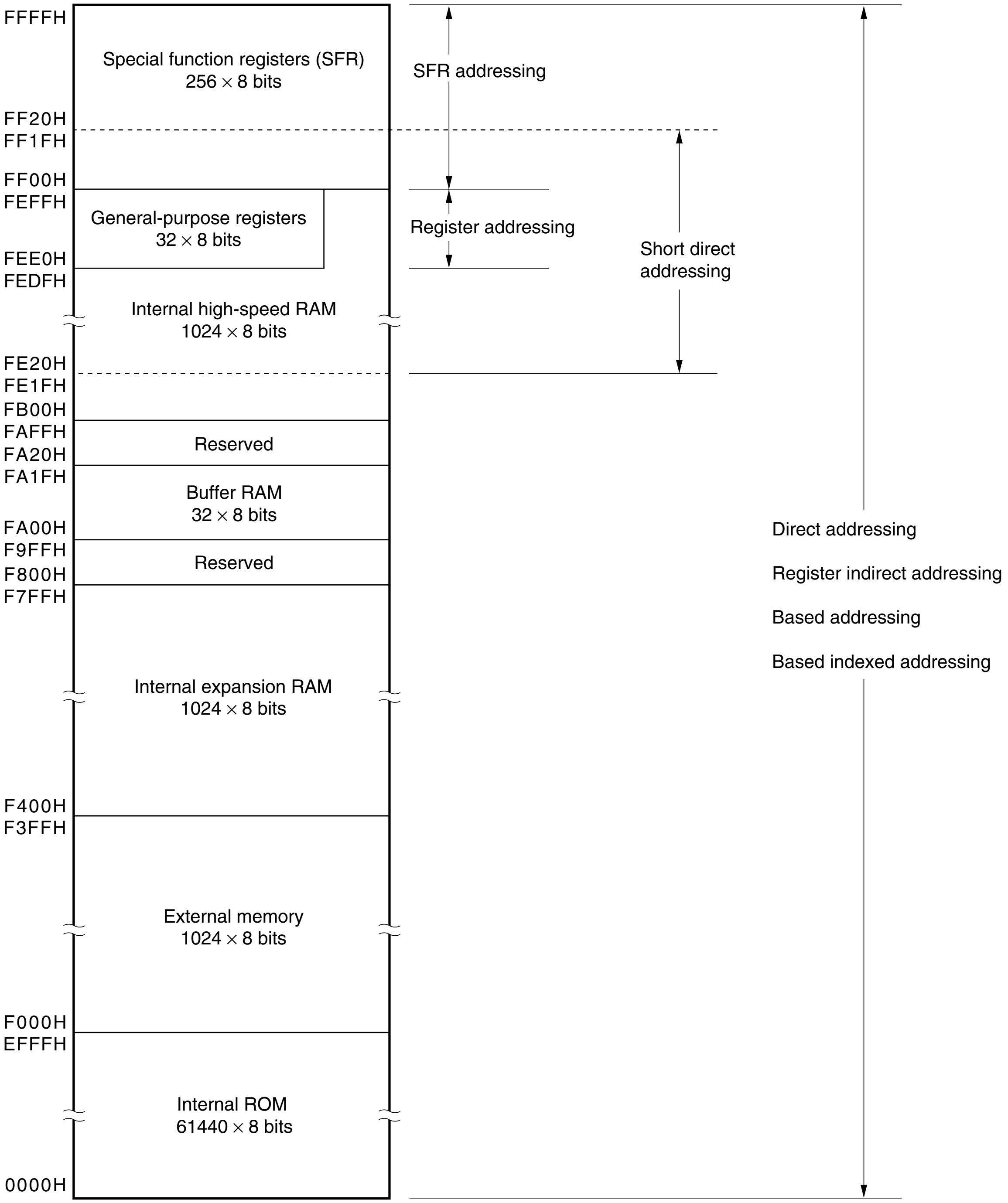
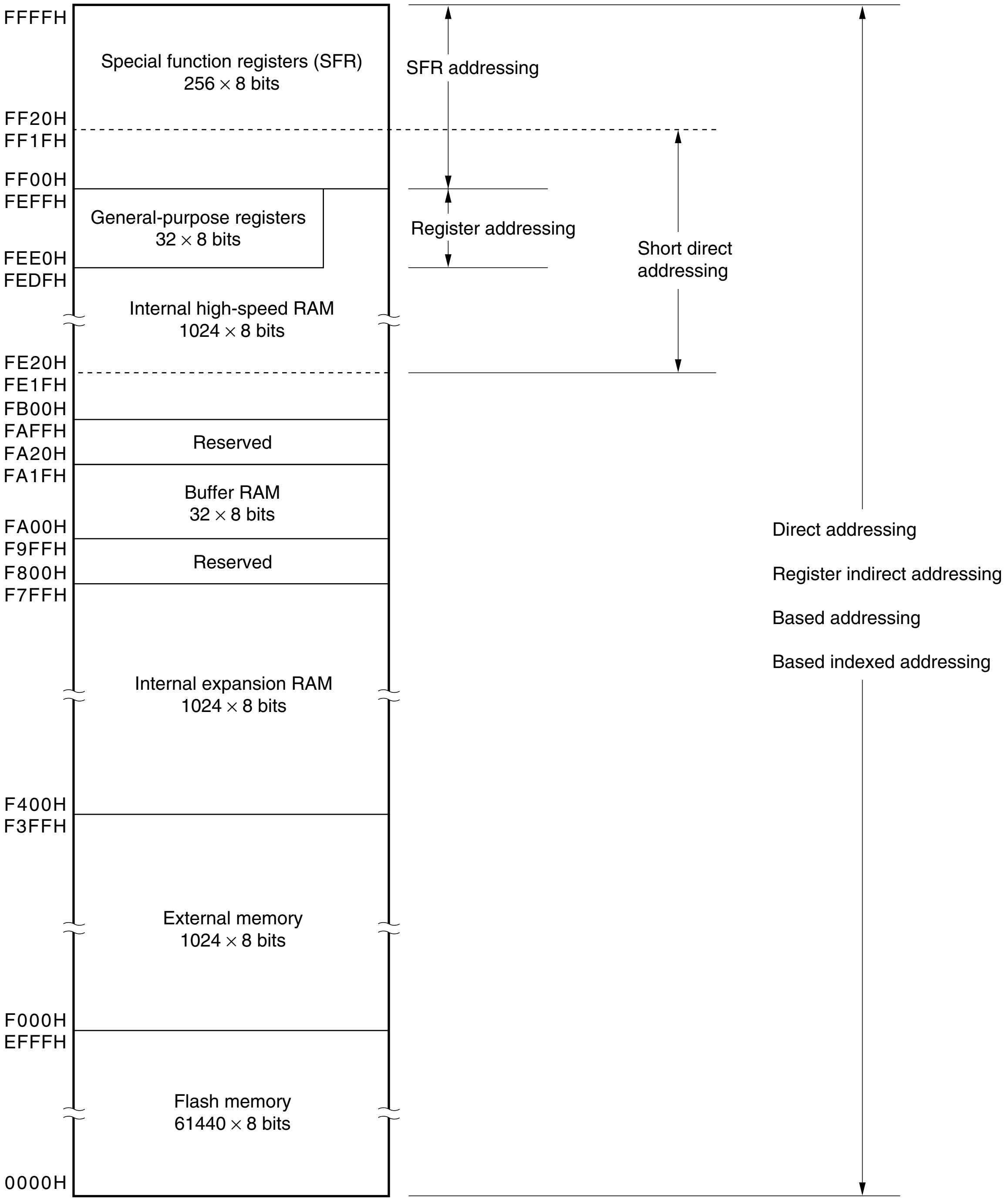


Figure 3-10. Data Memory Addressing (μPD78F0148)



3.2 Processor Registers

The 78K0/KF1 Series 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 3-11. Format of Program Counter

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

(2) Program status word (PSW)

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

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

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

Figure 3-12. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

When 0, the IE flag is set to the interrupt disabled (DI) state, and only non-maskable interrupt requests become acknowledgeable. Other interrupt requests are all disabled.

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L, PR1H) (refer to **19.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)**) can not be acknowledged. Actual request acknowledgement is controlled by the interrupt enable flag (IE).

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-13. Format of Stack Pointer

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

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

Each stack operation saves/restores data as shown in Figures 3-14 and 3-15.

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

Figure 3-14. Data to Be Saved to Stack Memory

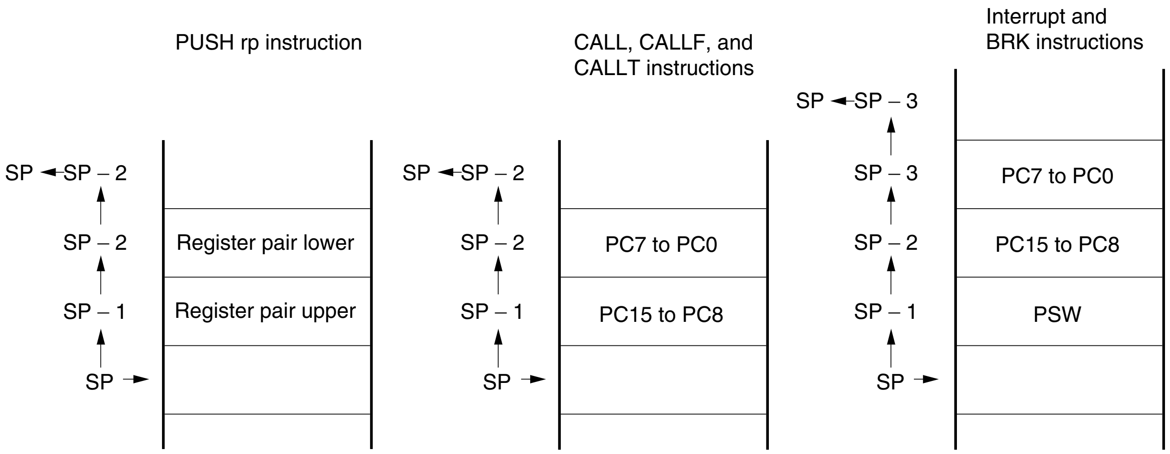
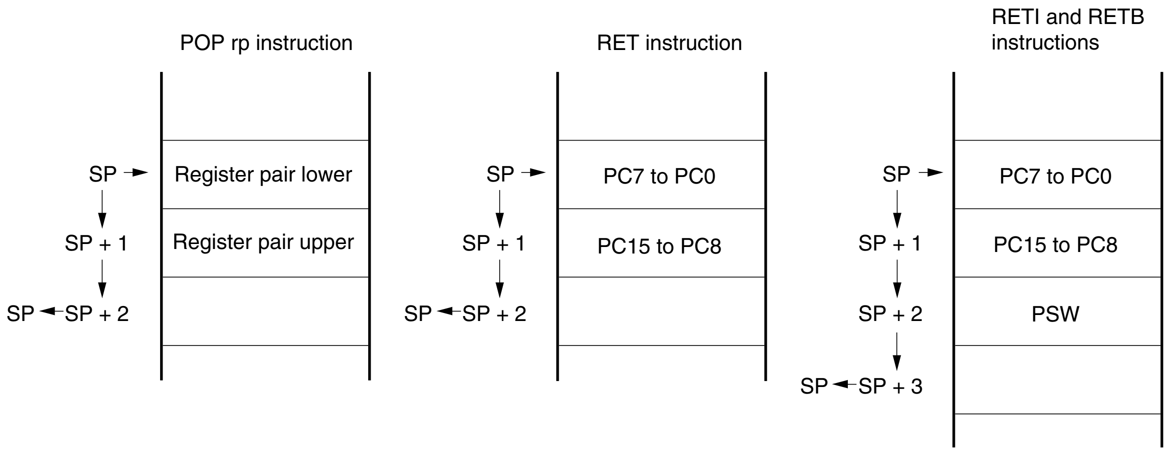


Figure 3-15. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

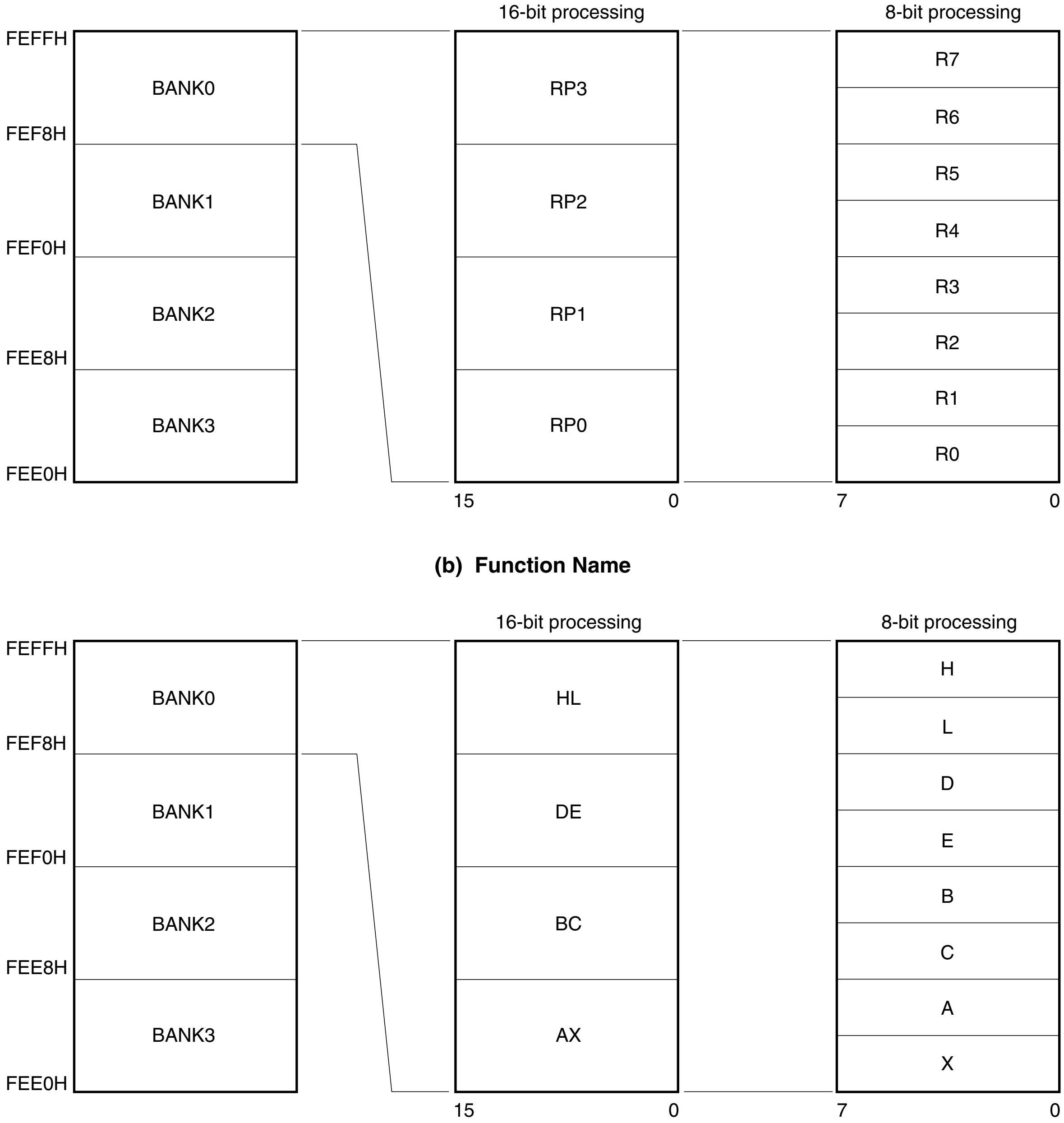
General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

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

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

Figure 3-16. Configuration of General-Purpose Registers



(a) Absolute Name

(b) Function Name

3.2.3 Special Function Registers (SFRs)

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

SFRs are allocated to the FF00H to FFFFH area.

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit).
  This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr).
  This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (sfrp).
  When specifying an address, describe an even address.

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

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

Table 3-5. Special Function Register List (1/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	–	00H
FF01H	Port 1	P1	R/W	√	√	–	00H
FF02H	Port 2	P2	R	√	√	–	00H
FF03H	Port 3	P3	R/W	√	√	–	00H
FF04H	Port 4	P4	R/W	√	√	–	00H
FF05H	Port 5	P5	R/W	√	√	–	00H
FF06H	Port 6	P6	R/W	√	√	–	00H
FF07H	Port 7	P7	R/W	√	√	–	00H
FF08H	A/D conversion result register	ADCR	R	–	–	√	Undefined
FF09H							
FF0AH	Receive buffer register 6	RXB6	R	–	√	–	FFH
FF0BH	Transmit buffer register 6	TXB6	R/W	–	√	–	FFH
FF0CH	Port 12	P12	R/W	√	√	–	00H
FF0DH	Port 13	P13	R/W	√	√	–	00H
FF0EH	Port 14	P14	R/W	√	√	–	00H
★ FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H
FF10H	16-bit timer counter 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16-bit timer capture/compare register 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H
FF17H	8-bit timer compare register 50	CR50	R/W	–	√	–	00H
FF18H	8-bit timer H compare register 00	CMP00	R/W	–	√	–	00H
FF19H	8-bit timer H compare register 10	CMP10	R/W	–	√	–	00H
FF1AH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	00H
FF1BH	8-bit timer H compare register 11	CMP11	R/W	–	√	–	00H
FF1FH	8-bit timer counter 51	TM51	R	–	√	–	00H
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1	R/W	√	√	–	FFH
FF23H	Port mode register 3	PM3	R/W	√	√	–	FFH
FF24H	Port mode register 4	PM4	R/W	√	√	–	FFH
FF25H	Port mode register 5	PM5	R/W	√	√	–	FFH
FF26H	Port mode register 6	PM6	R/W	√	√	–	FFH
FF27H	Port mode register 7	PM7	R/W	√	√	–	FFH
FF28H	A/D converter mode register	ADM	R/W	√	√	–	00H
FF29H	Analog input channel specification register	ADS	R/W	√	√	–	00H
FF2AH	Power-fail comparison mode register	PFM	R/W	√	√	–	00H
FF2BH	Power-fail comparison threshold register	PFT	R/W	–	√	–	00H

Table 3-5. Special Function Register List (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FF30H	Pull-up resistor option register 0	PU0		R/W	√	√	–	00H
FF31H	Pull-up resistor option register 1	PU1		R/W	√	√	–	00H
FF33H	Pull-up resistor option register 3	PU3		R/W	√	√	–	00H
FF34H	Pull-up resistor option register 4	PU4		R/W	√	√	–	00H
FF35H	Pull-up resistor option register 5	PU5		R/W	√	√	–	00H
FF36H	Pull-up resistor option register 6	PU6		R/W	√	√	–	00H
FF37H	Pull-up resistor option register 7	PU7		R/W	√	√	–	00H
FF3CH	Pull-up resistor option register 12	PU12		R/W	√	√	–	00H
FF3EH	Pull-up resistor option register 14	PU14		R/W	√	√	–	00H
FF40H	Clock output selection register	CKS		R/W	√	√	–	00H
FF41H	8-bit timer compare register 51	CR51		R/W	–	√	–	00H
FF43H	8-bit timer mode control register 51	TMC51		R/W	√	√	–	00H
FF47H	Memory expansion mode register	MEM		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
★ FF4AH	Serial I/O shift register 11[Note]	SIO11		R	–	√	–	00H
FF4CH	Transmit buffer register 11[Note]	SOTB11		R/W	–	√	–	Undefined
FF4FH	Input switch control register	ISC		R/W	√	√	–	00H
FF50H	Asynchronous serial interface operation mode register 6	ASIM6		R/W	√	√	–	01H
FF53H	Asynchronous serial interface reception error status register 6	ASIS6		R	–	√	–	00H
FF55H	Asynchronous serial interface transmission status register 6	ASIF6		R	–	√	–	00H
FF56H	Clock selection register 6	CKSR6		R/W	–	√	–	00H
FF57H	Baud rate generator control register 6	BRGC6		R/W	–	√	–	FFH
FF58H	Asynchronous serial interface control register 6	ASICL6		R/W	√	√	–	16H
FF60H	Remainder data register 0	SDR0	SDR0L	R	–	√	√	00H
FF61H			SDR0H		–	√		00H
FF62H	Multiplication/division data register A0	MDA0L	MDA0LL	R/W	–	√	√	00H
FF63H			MDA0LH		–	√		00H
FF64H		MDA0H	MDA0HL	R/W	–	√	√	00H
FF65H			MDA0HH		–	√		00H
FF66H	Multiplication/division data register B0	MDB0	MDB0L	R/W	–	√	√	00H
FF67H			MDB0H		–	√		00H
FF68H	Multiplier/divider control register 0	DMUC0		R/W	√	√	–	00H
FF69H	8-bit timer H mode register 0	TMHMD0		R/W	√	√	–	00H
FF6AH	Timer clock selection register 50	TCL50		R/W	–	√	–	00H

Note μPD780146, 780148, and 78F0148 only.

Table 3-5. Special Function Register List (3/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF6BH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF6CH	8-bit timer H mode register 1	TMHMD1	R/W	√	√	–	00H
FF6DH	8-bit timer H carrier control register 1	TMCYC1	R/W	√	√	–	00H
FF6EH	Key return mode register	KRM	R/W	√	√	–	00H
FF6FH	Watch timer operation mode register	WTM	R/W	√	√	–	00H
FF70H	Asynchronous serial interface operation mode register 0	ASIM0	R/W	√	√	–	01H
FF71H	Baud rate generator control register 0	BRGC0	R/W	–	√	–	1FH
FF72H	Receive buffer register 0	RXB0	R	–	√	–	FFH
FF73H	Asynchronous serial interface reception error status register 0	ASIS0	R	–	√	–	00H
FF74H	Transmit shift register 0	TXS0	W	–	√	–	FFH
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	Undefined
FF88H	Serial operation mode register 11[Note 1]	CSIM11	R/W	√	√	–	00H
FF89H	Serial clock selection register 11[Note 1]	CSIC11	R/W	√	√	–	00H
FF8CH	Timer clock selection register 51	TCL51	R/W	–	√	–	00H
FF90H	Serial operation mode specification register 0	CSIMA0	R/W	√	√	–	00H
FF91H	Serial status register 0	CSIS0	R/W	√	√	–	00H
FF92H	Serial trigger register 0	CSIT0	R/W	√	√	–	00H
FF93H	Divisor selection register 0	BRGCA0	R/W	–	√	–	03H
FF94H	Automatic data transfer address point specification register 0	ADTP0	R/W	–	√	–	00H
FF95H	Automatic data transfer interval specification register 0	ADTI0	R/W	–	√	–	00H
FF96H	Serial I/O shift register 0	SIOA0	R/W	–	√	–	00H
FF97H	Automatic data transfer address count register 0	ADTC3	R	–	√	–	00H
FF98H	Watchdog timer mode register	WDTM	R/W	–	√	–	67H
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	9AH
FFA0H	Ring-OSC mode register	RCM	R/W	√	√	–	00H
FFA1H	Main clock mode register	MCM	R/W	√	√	–	00H
FFA2H	Main OSC control register	MOC	R/W	√	√	–	00H
★ FFA3H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H
FFA4H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	05H
FFA9H	Clock monitor mode register	CLM	R/W	√	√	–	00H
FFACH	Reset control flag register	RESF	R	–	√	–	00H[Note 2]
FFB0H	16-bit timer counter 01[Note 1]	TM01	R	–	–	√	0000H
FFB1H							

Notes 1. μPD780146, 780148, and 78F0148 only.

2. This value varies depending on the reset source.

Table 3-5. Special Function Register List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFB2H	16-bit timer capture/compare register 001[Note 1]	CR001		R/W	–	–	√	0000H
FFB3H								
FFB4H	16-bit timer capture/compare register 011[Note 1]	CR011		R/W	–	–	√	0000H
FFB5H								
FFB6H	16-bit timer mode control register 01[Note 1]	TMC01		R/W	√	√	–	00H
FFB7H	Prescaler mode register 01[Note 1]	PRM01		R/W	√	√	–	00H
FFB8H	Capture/compare control register 01[Note 1]	CRC01		R/W	√	√	–	00H
FFB9H	16-bit timer output control register 01[Note 1]	TOC01		R/W	√	√	–	00H
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
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	IF1	IF1L	R/W	√	√	√	00H
FFE3H	Interrupt request flag register 1H		IF1H	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	MK1	MK1L	R/W	√	√	√	FFH
★ FFE7H	Interrupt mask flag register 1H		MK1H	R/W	√	√		DFH
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	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	Priority specification flag register 1H		PR1H	R/W	√	√		FFH
FFF0H	Internal memory size switching register[Note 2]	IMS		R/W	–	√	–	CFH
FFF4H	Internal expansion RAM size switching register[Note 2]	IXS		R/W	–	√	–	0CH
FFF8H	Memory expansion wait setting register	MM		R/W	√	√	–	10H
FFFBH	Processor clock control register	PCC		R/W	√	√	–	00H

Notes 1. μPD780146, 780148, and 78F0148 only.

2. The default value of IMS and IXS are fixed (IMS = CFH, IXS = 0CH) in all products in the 78K0/KF1 Series regardless of the internal memory capacity.

Therefore, set the following value to each product.

	IMS	IXS
μPD780143	C6H	0CH
μPD780144	C8H	
μPD780146	CCH	0AH
μPD780148	CFH	
μPD78F0148	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

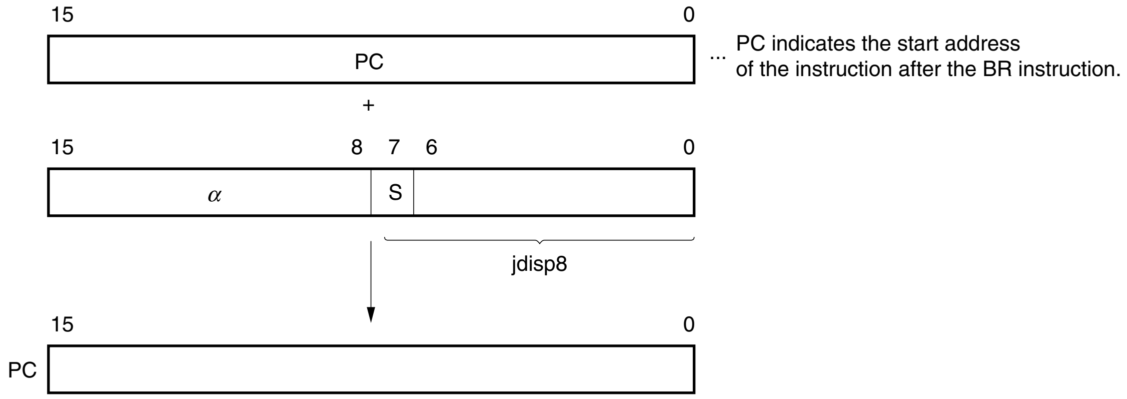
[Function]

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]


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

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

3.3.2 Immediate addressing

[Function]

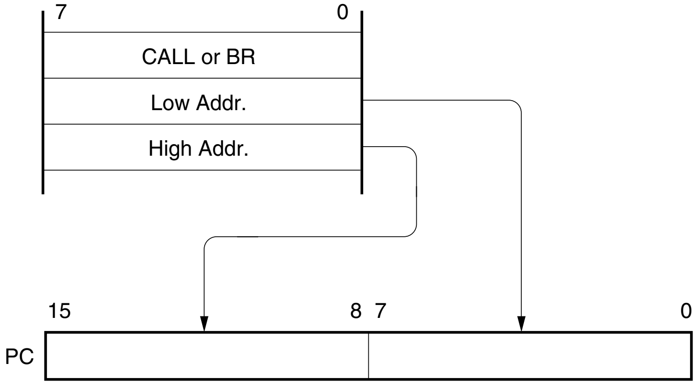
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

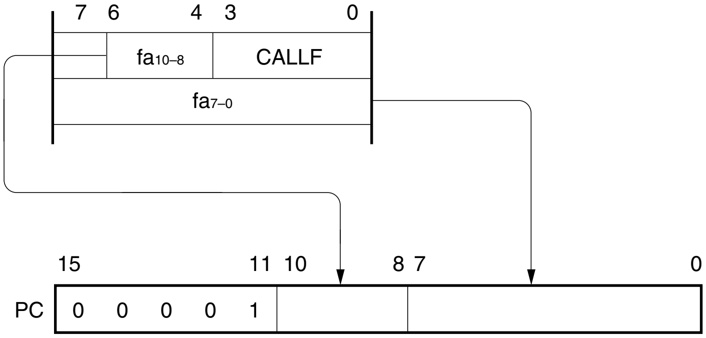
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

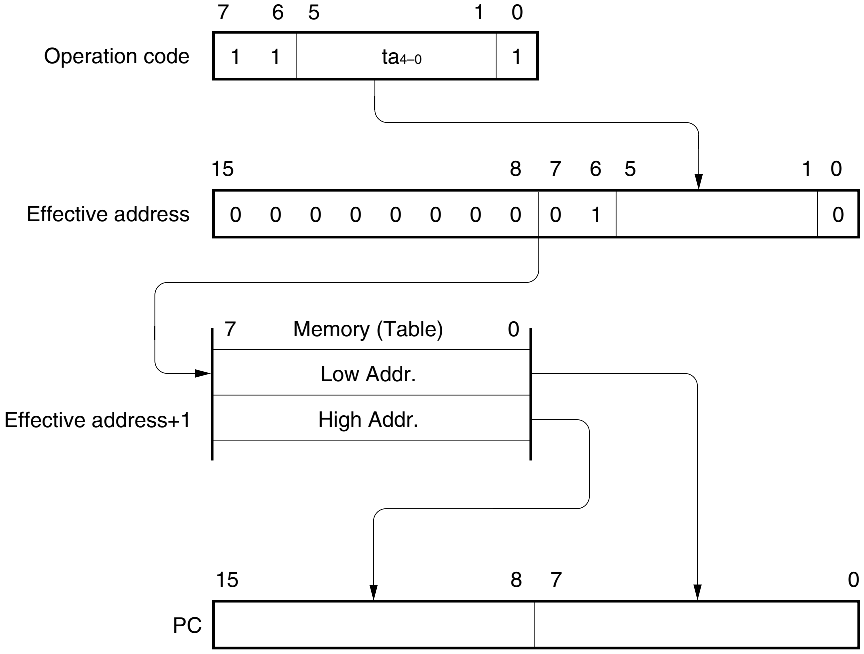
[Function]

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

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

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

[Illustration]



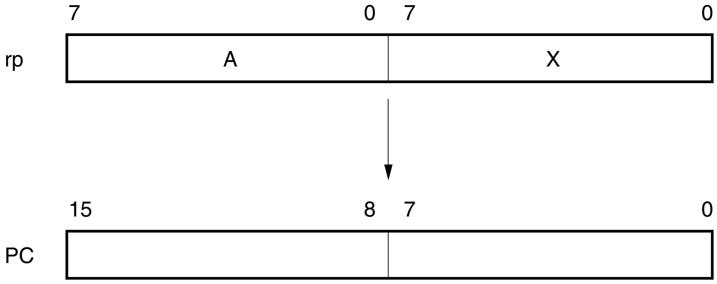
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the 78K0/KF1 Series instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

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

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

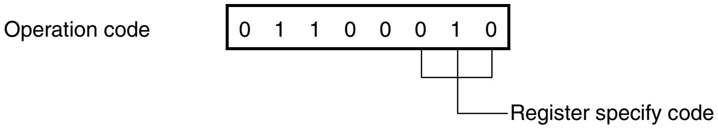
[Operand format]

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

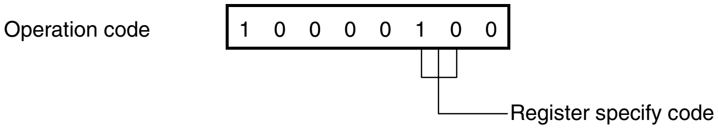
‘r’ and ‘rp’ can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

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

[Operand format]

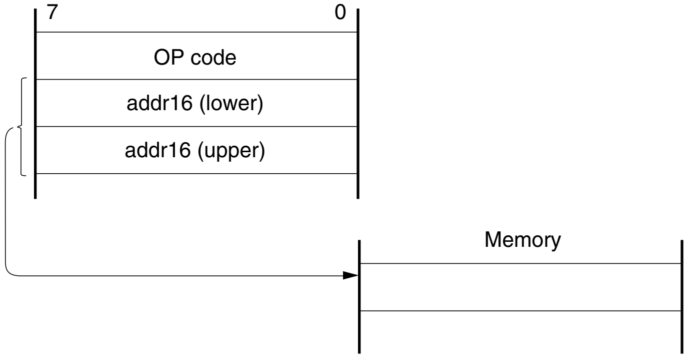
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

This addressing is applied to the 256-byte space FE20H to FF1FH. Internal RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to be manipulated with a small number of bytes and clocks.

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

[Operand format]

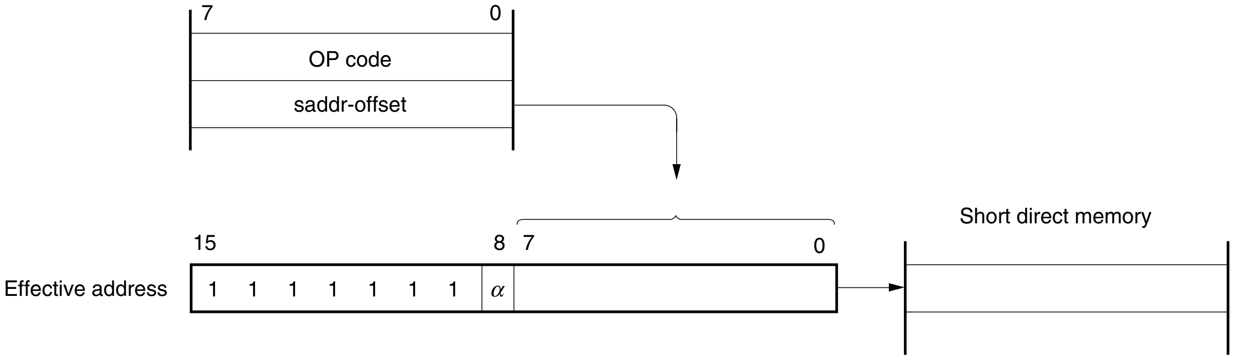
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

MOV 0FE30H, #50H; when setting saddr to FE30H and immediate data to 50H

Operation code	0 0 0 1 0 0 0 1	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)
	0 1 0 1 0 0 0 0	50H (immediate data)

[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]

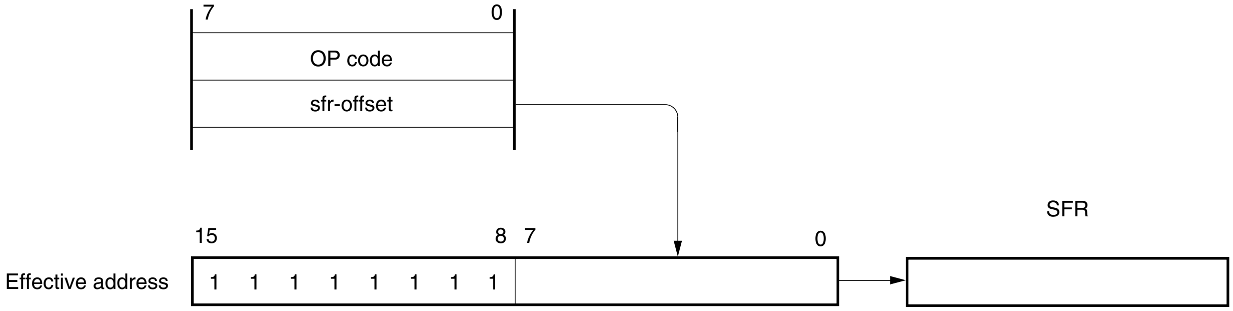
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

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

[Illustration]



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

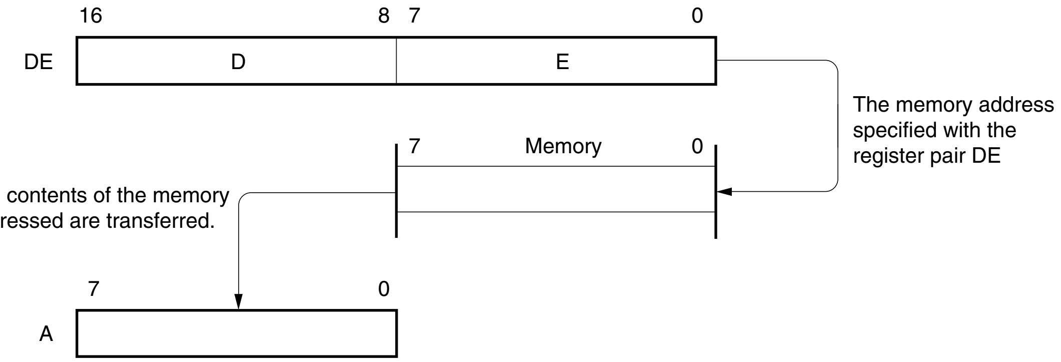
Identifier	Description
–	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]


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

3.4.7 Based addressing

[Function]

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

[Operand format]

Identifier	Description
–	[HL + byte]

[Description example]

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

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

3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

[Description example]

In the case of MOV A, [HL + B]

Operation code 1 0 1 0 1 0 1 1

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Operation code 1 0 1 1 0 1 0 1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

78K0/KF1 Series 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 4-1.

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

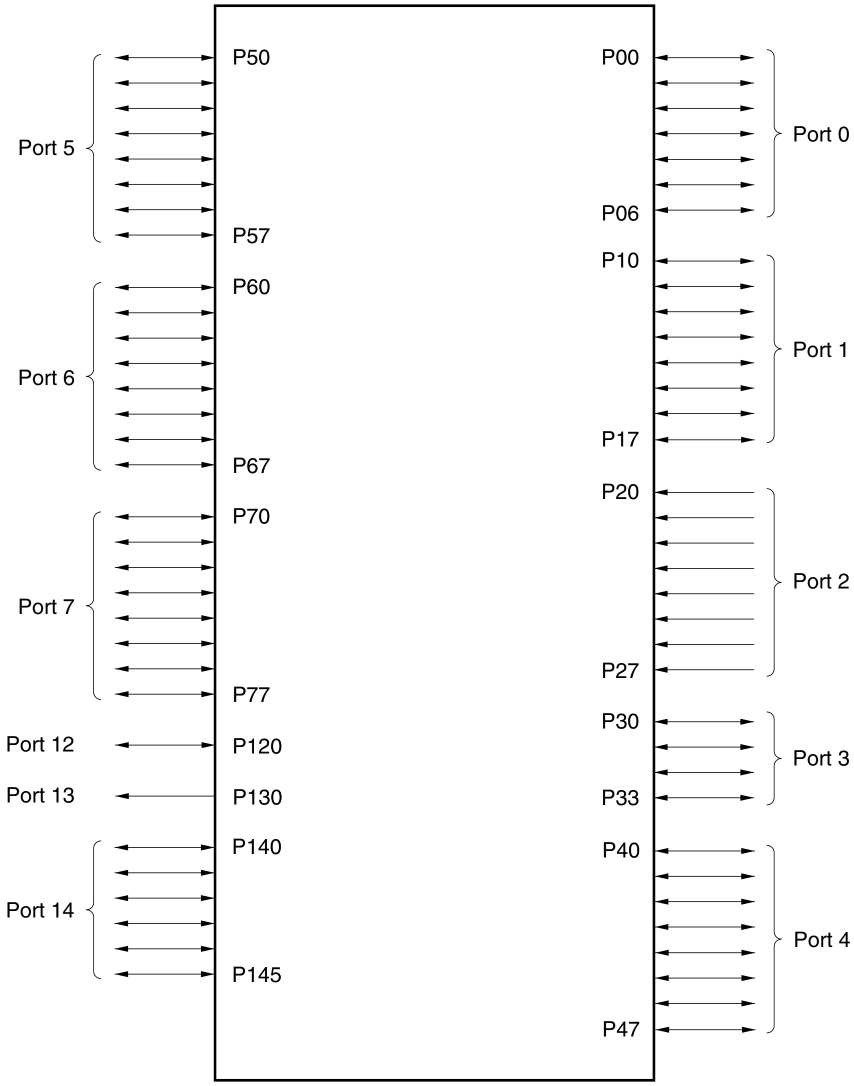


Table 4-1. Port Functions (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 7-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI000
P01				TI010/TO00
P02				SO11[Note]
P03				SI11[Note]
P04				$\overline{\text{SCK11}}$[Note]
P05				$\overline{\text{SSI11}}$[Note]/TI001[Note]
P06				TI011[Note]/TO01[Note]
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	$\overline{\text{SCK10}}$/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 to P27	Input	Port 2. 8-bit input-only port.	Input	ANI0 to ANI7
P30 to P32	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP1 to INTP3
P33				INTP4/TI51/TO51
P40 to P47	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	AD0 to AD7
P50 to P57	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	A8 to A15

Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

Table 4-1. Port Functions (2/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P60 to P63	I/O	Port 6. 8-bit I/O port. Input/output can be specified in 1-bit units.	N-ch open-drain I/O port. Use of an on-chip pull-up resistor can be specified by a mask option only for mask ROM versions.	Input	–
P64			Use of an on-chip pull-up resistor can be specified by a software setting.		$\overline{\text{RD}}$
P65					$\overline{\text{WR}}$
P66					$\overline{\text{WAIT}}$
P67					ASTB
P70 to P77	I/O	Port 7. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	KR0 to KR7
P120	I/O	Port 12. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	INTP0
P130	Output	Port 13. 1-bit output-only port.		Output	–
P140	I/O	Port 14. 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	PCL/INTP6
P141					BUZ/BUSY0/ INTP7
P142					$\overline{\text{SCKA0}}$
P143					SIA0
P144					SOA0
P145					STB0

4.2 Port Configuration

Ports consist of the following hardware.

Table 4-2. Port Configuration

Item	Configuration
Control registers	Port mode register (PM0, PM1, PM3 to PM7, PM12, PM14) Pull-up resistor option register (PU0, PU1, PU3 to PU7, PU12, PU14) Input switch control register (ISC)
Port	Total: 67 (CMOS I/O: 54, CMOS input: 8, CMOS output: 1, N-ch open drain output: 4)
Pull-up resistor	• Mask ROM version Total: 58 (software control: 54, mask option specification: 4) • Flash memory version: Total: 54

4.2.1 Port 0

Port 0 is a 7-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P06 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).

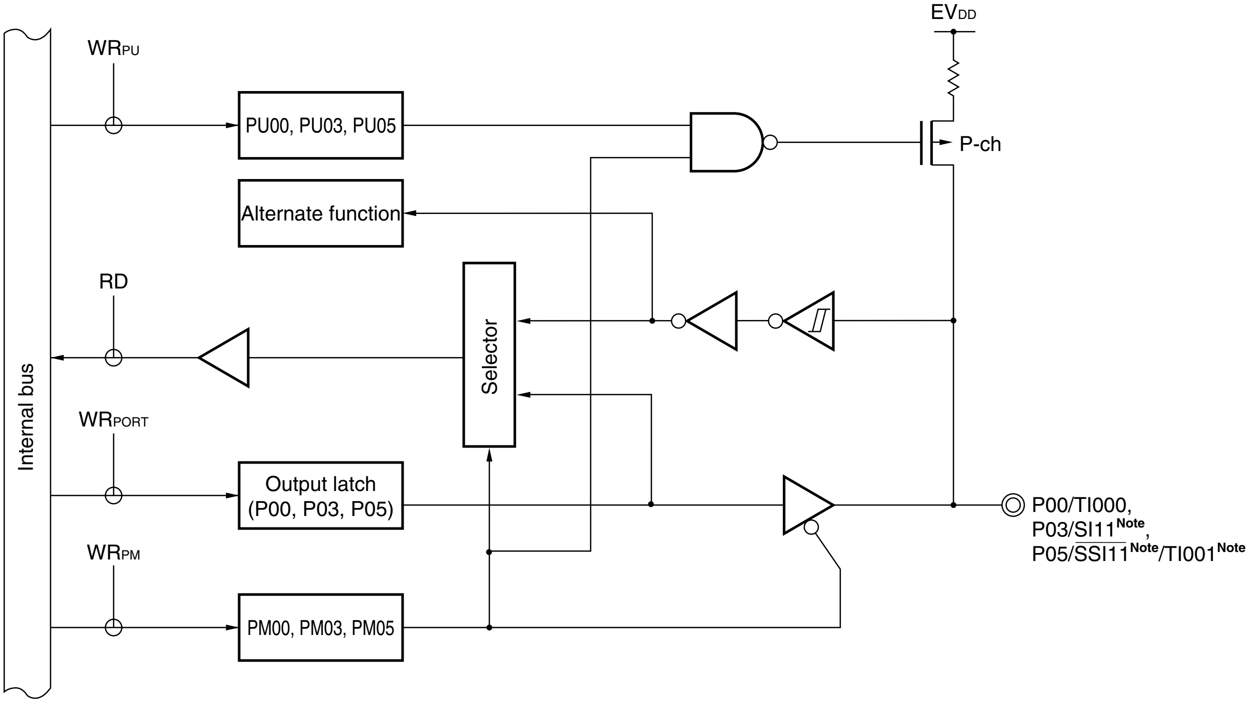
This port can also be used for timer I/O, serial interface data I/O, and clock I/O.

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

Figures 4-2 to 4-5 show block diagrams of port 0.

Caution When P02/SO11[Note], P03/SI11[Note], and P04/$\overline{\text{SCK11}}$[Note] are used as general-purpose ports, do not write to serial clock selection register 11 (CSIC11).

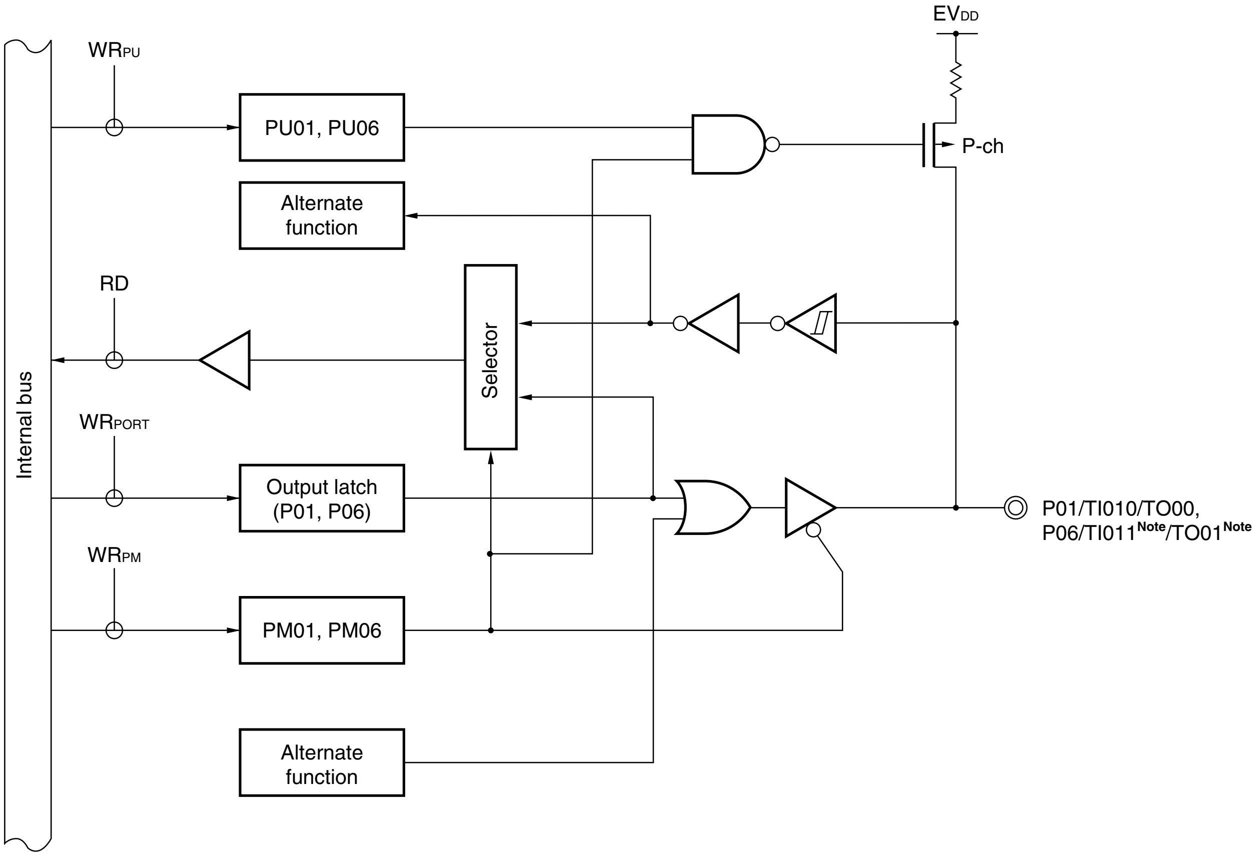
Figure 4-2. Block Diagram of P00, P03, and P05



Note Available only in the μPD780146, 780148, and 78F0148.

PU0:	Pull-up resistor option register 0
PM:	Port mode register
RD:	Port 0 read signal
WR:	Port 0 write signal

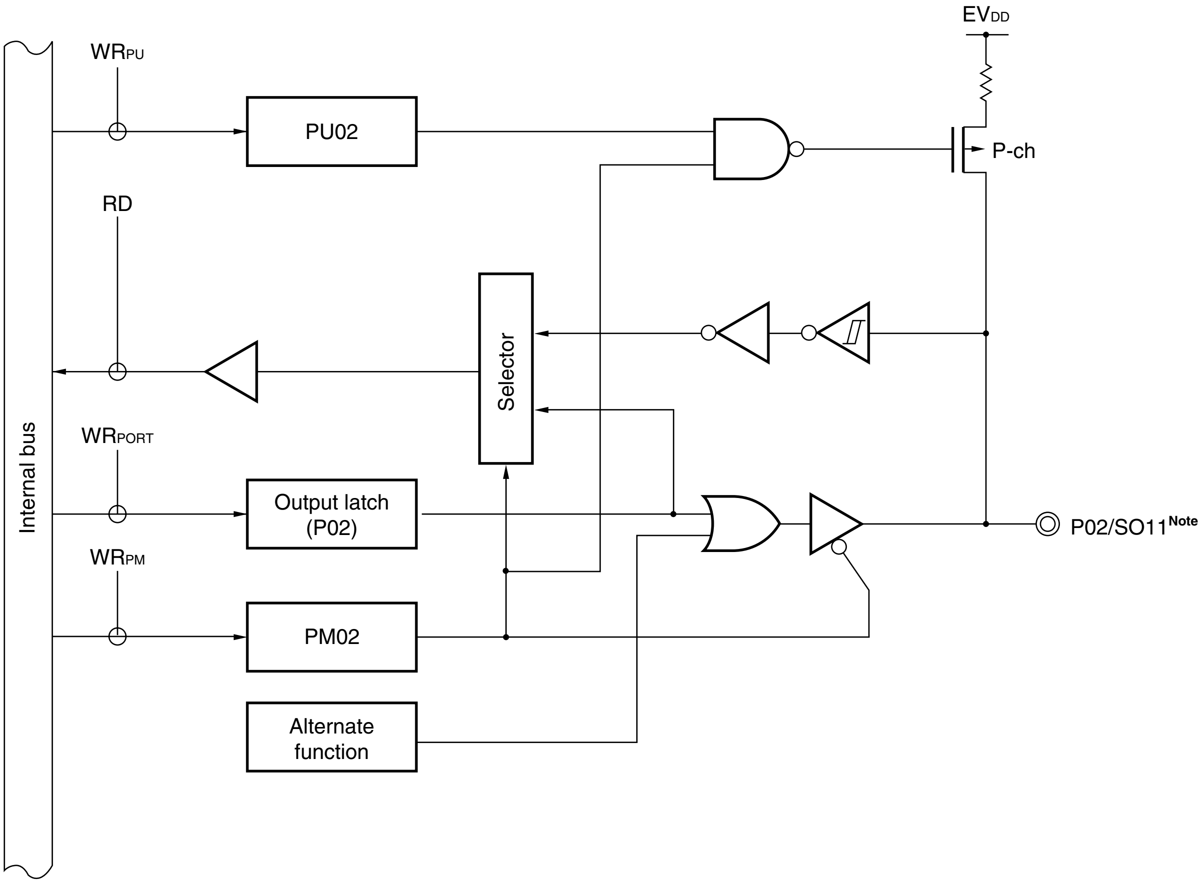
Figure 4-3. Block Diagram of P01 and P06



Note Available only in the μPD780146, 780148, and 78F0148.

PU0:	Pull-up resistor option register 0
PM:	Port mode register
RD:	Port 0 read signal
WR:	Port 0 write signal

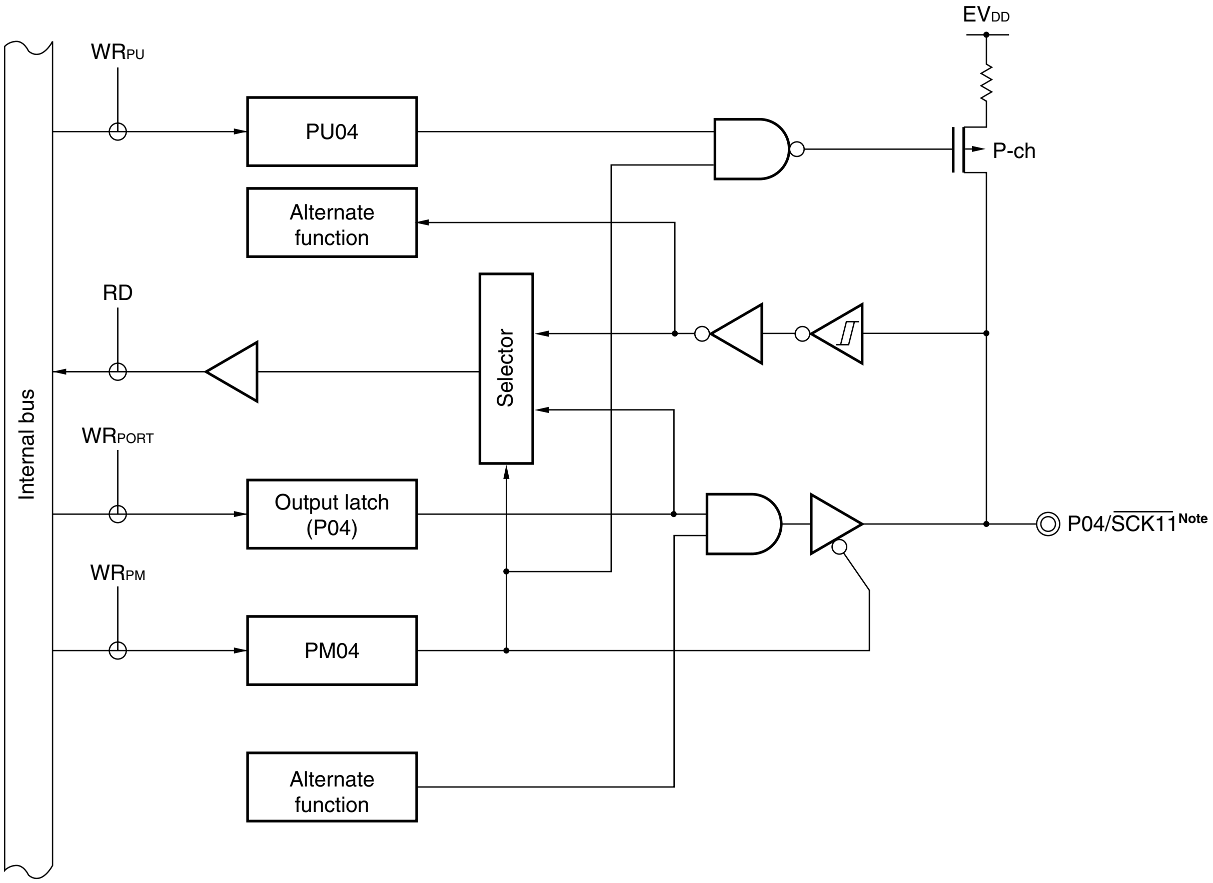
Figure 4-4. Block Diagram of P02



Note Available only in the μPD780146, 780148, and 78F0148.

PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

Figure 4-5. Block Diagram of P04



Note Available only in the μPD780146, 780148, and 78F0148.

PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

4.2.2 Port 1

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

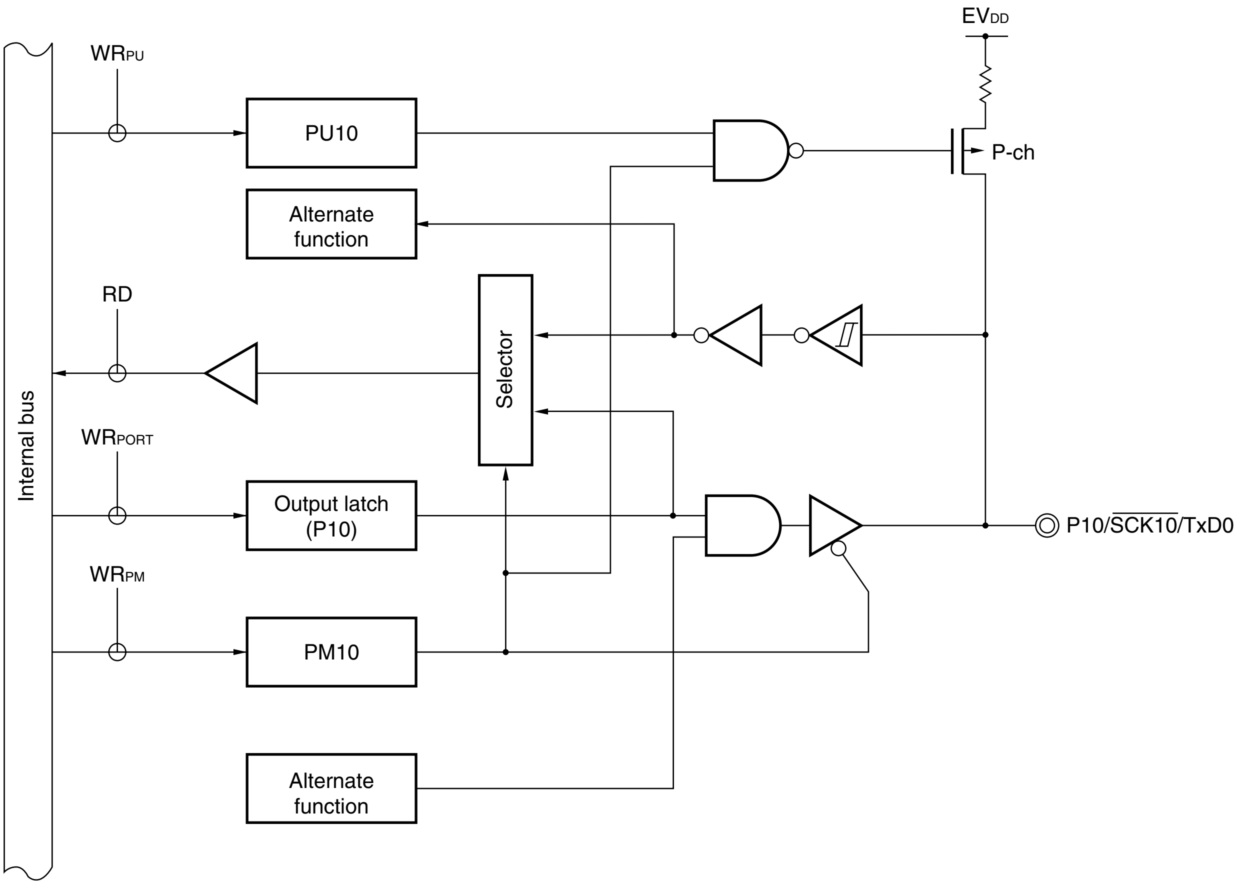
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

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

Figures 4-6 to 4-11 show block diagrams of port 1.

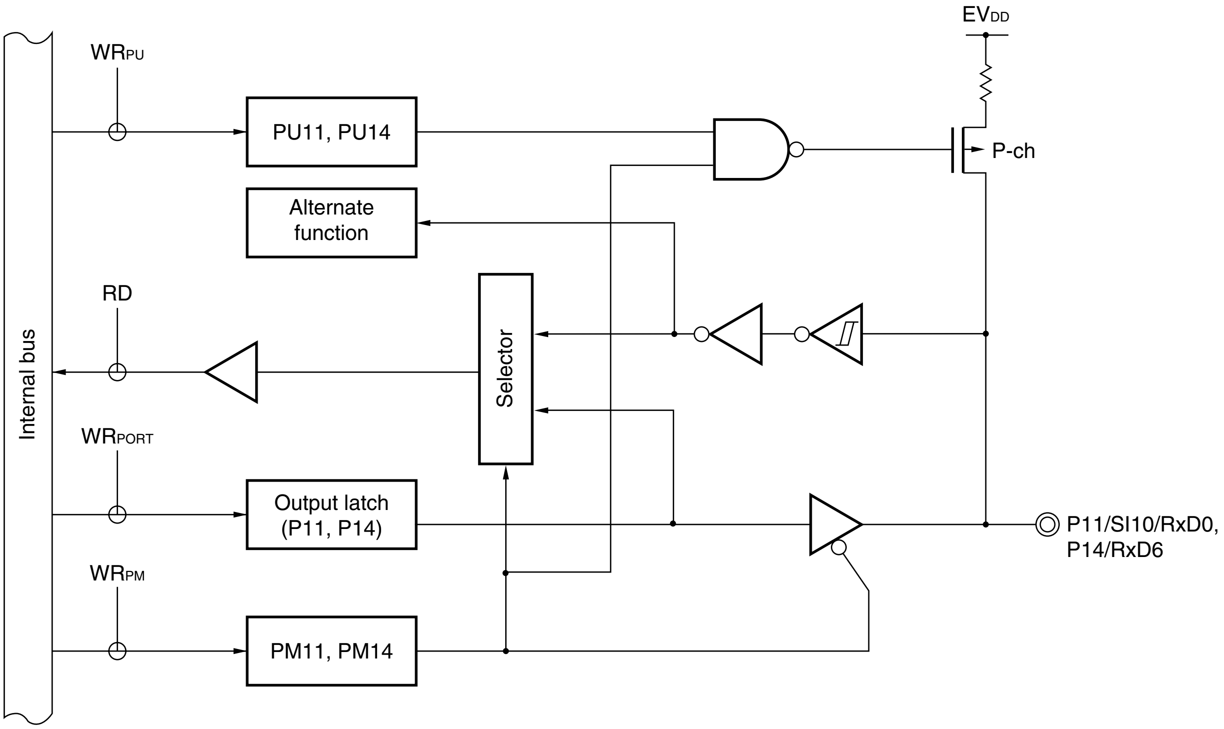
Caution When P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, and P12/SO10 are used as general-purpose ports, do not write to serial clock selection register 10 (CSIC10).

Figure 4-6. Block Diagram of P10



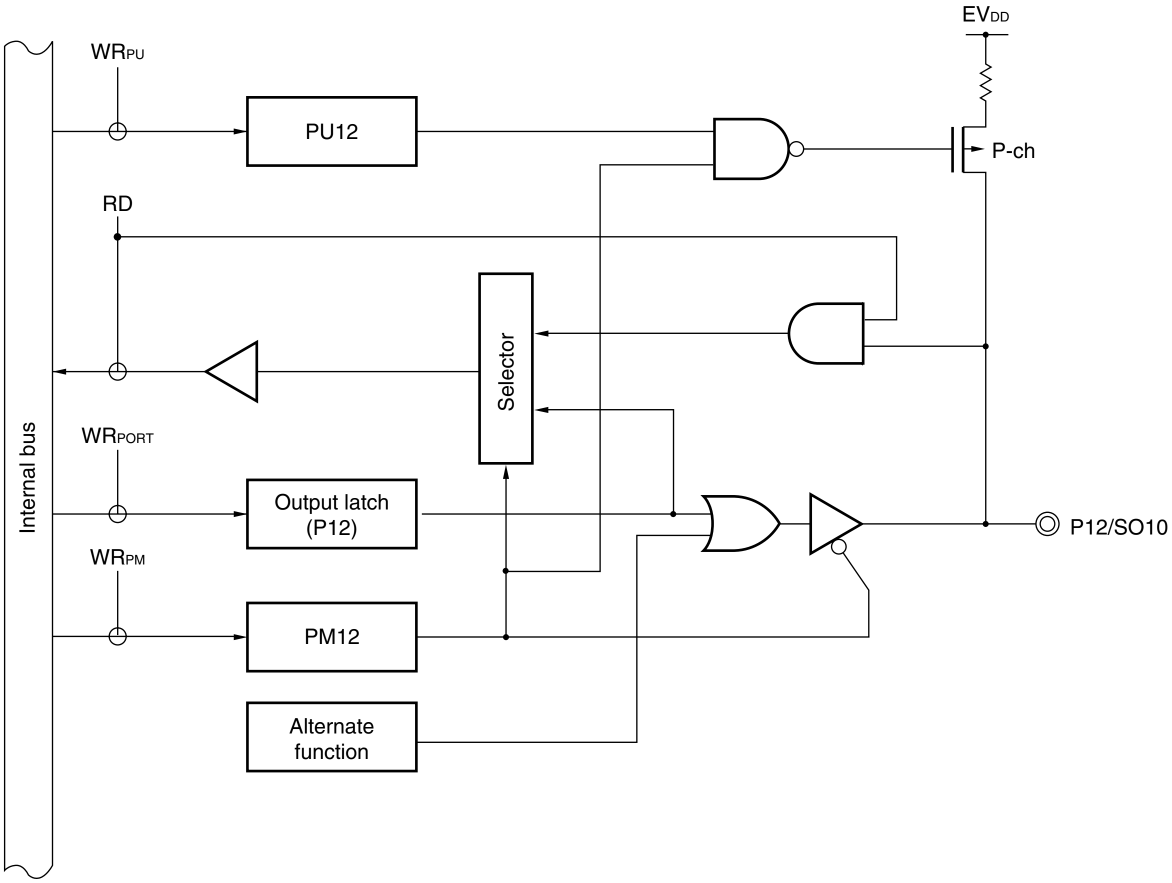
PU1:	Pull-up resistor option register 1
PM:	Port mode register
RD:	Port 1 read signal
WR:	Port 1 write signal

Figure 4-7. Block Diagram of P11 and P14



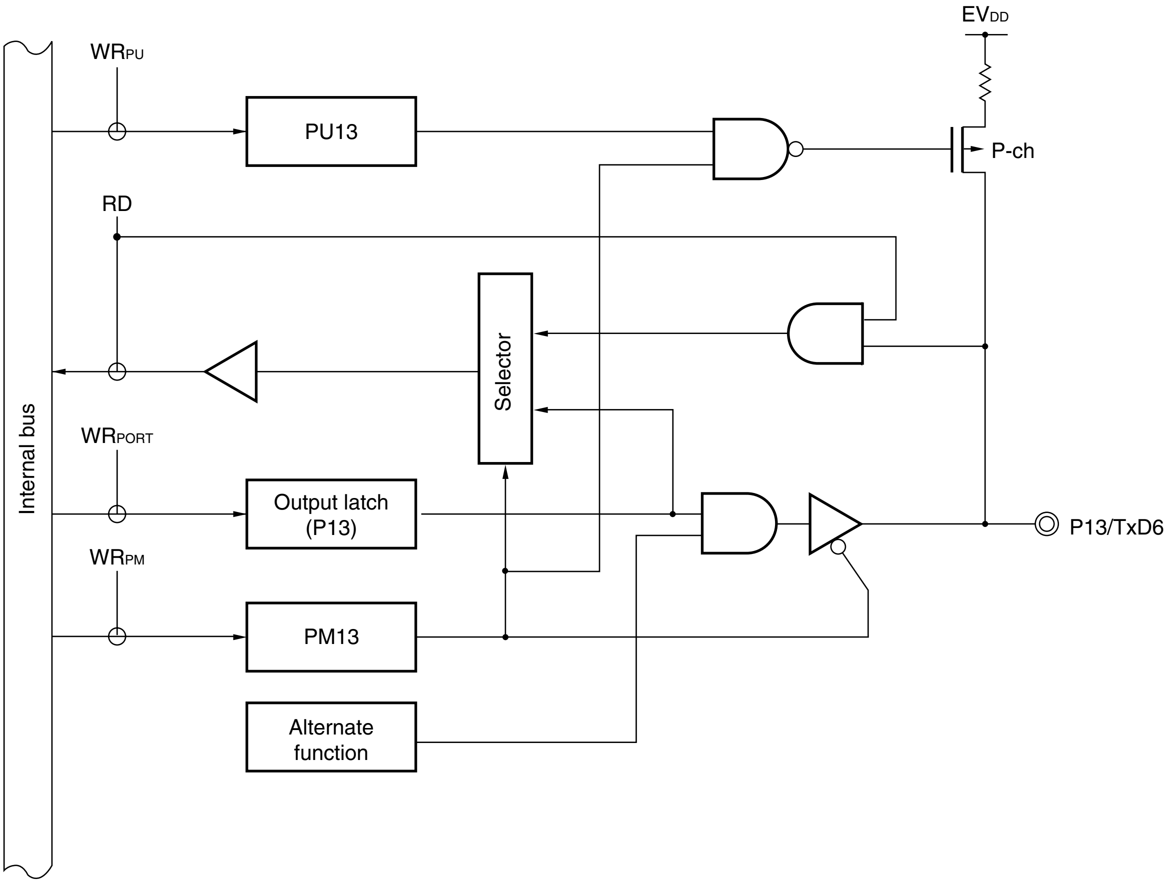
PU1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

Figure 4-8. Block Diagram of P12



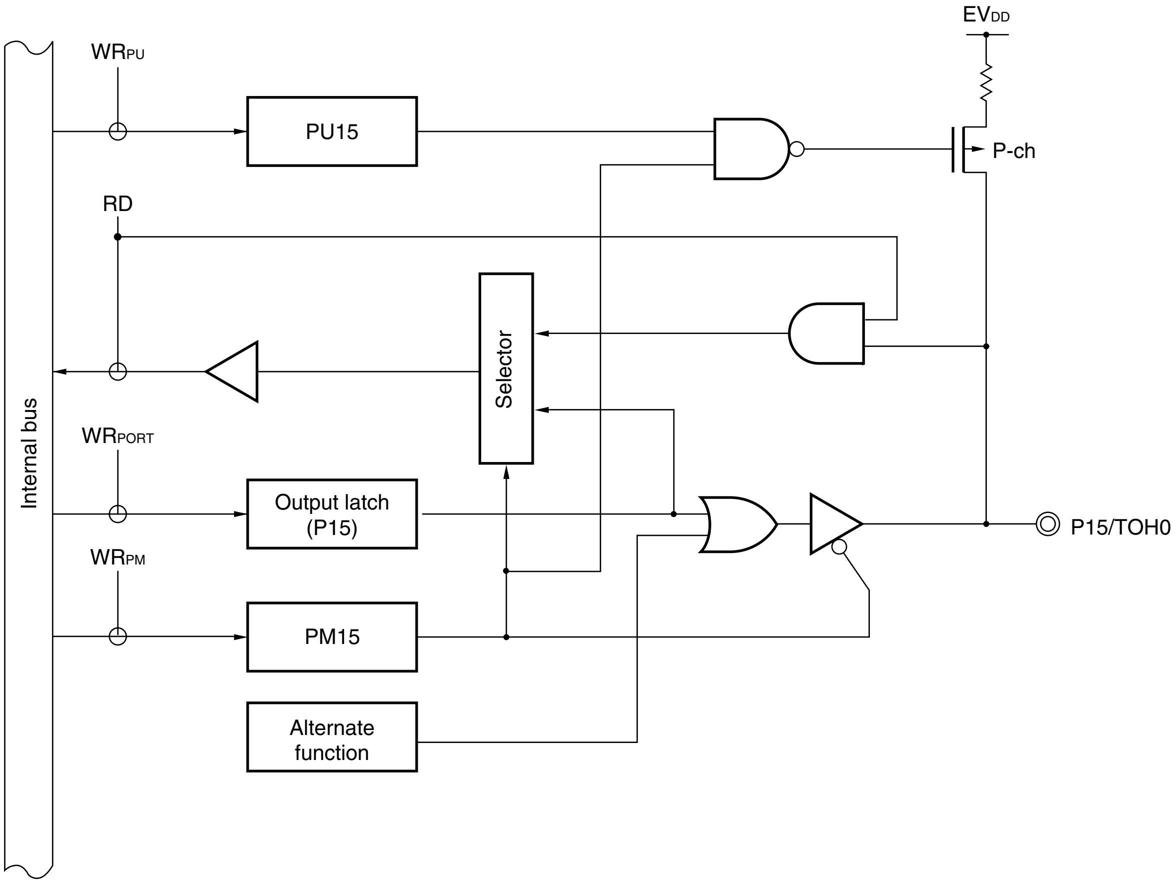
PU1:	Pull-up resistor option register 1
PM:	Port mode register
RD:	Port 1 read signal
WR:	Port 1 write signal

Figure 4-9. Block Diagram of P13



PU1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

Figure 4-10. Block Diagram of P15



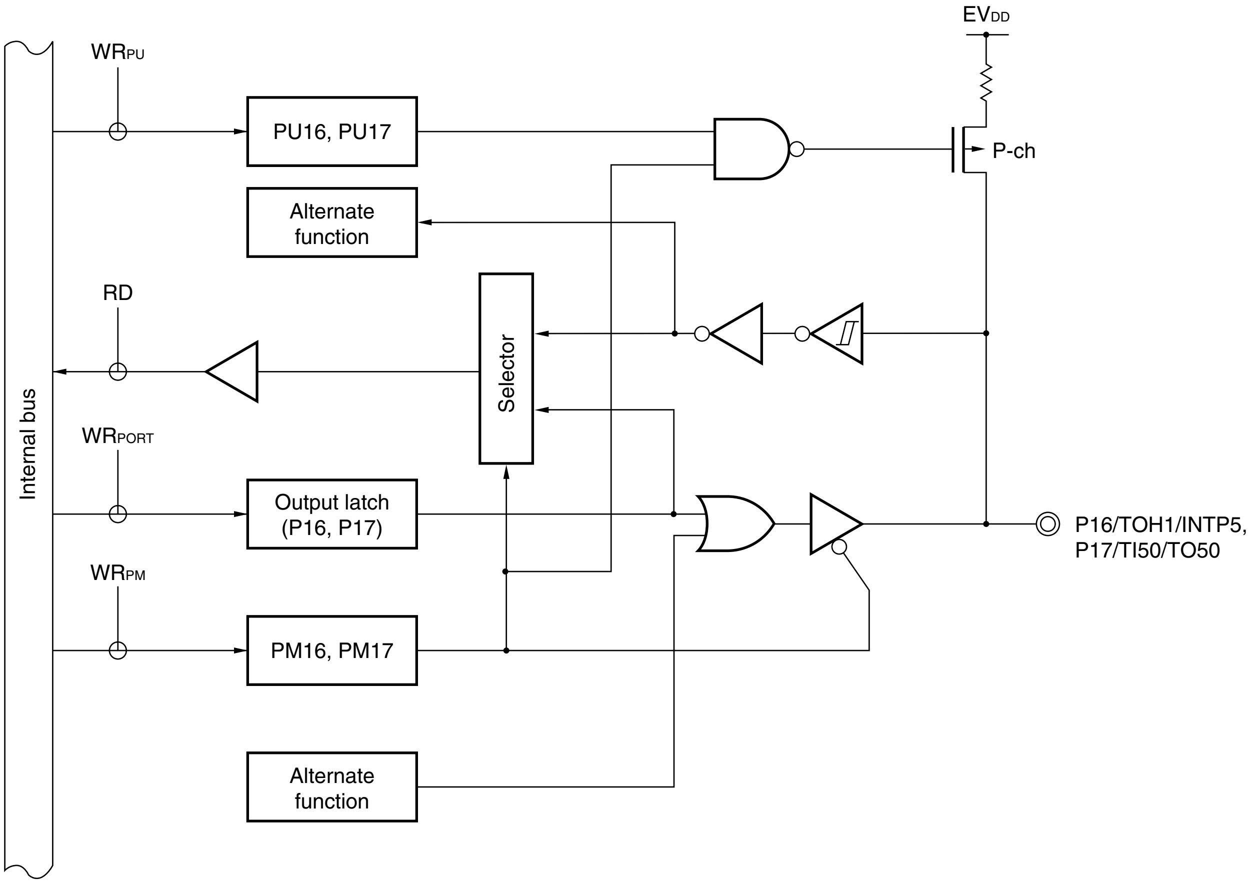
PU1: Pull-up resistor option register 1

PM: Port mode register

RD: Port 1 read signal

WR: Port 1 write signal

Figure 4-11. Block Diagram of P16 and P17



PU1:	Pull-up resistor option register 1
PM:	Port mode register
RD:	Port 1 read signal
WR:	Port 1 write signal

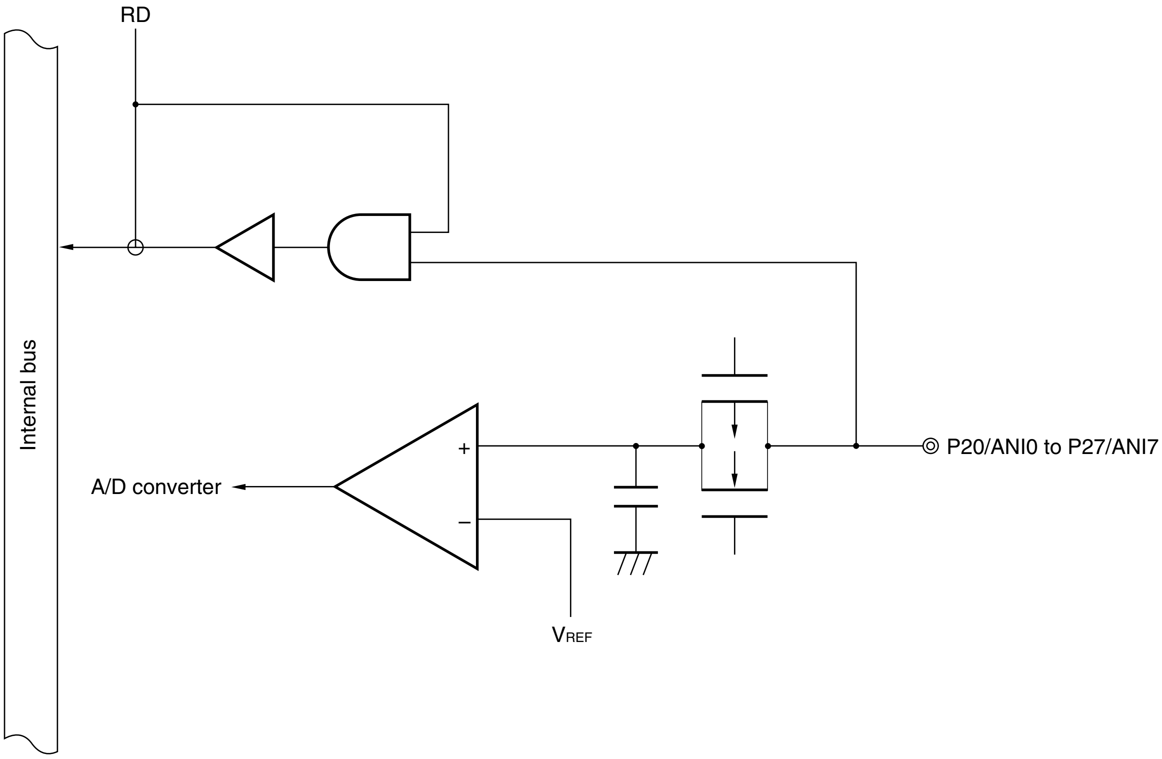
4.2.3 Port 2

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

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

Figure 4-12 shows a block diagram of port 2.

Figure 4-12. Block Diagram of P20 to P27



RD: Port 2 read signal

4.2.4 Port 3

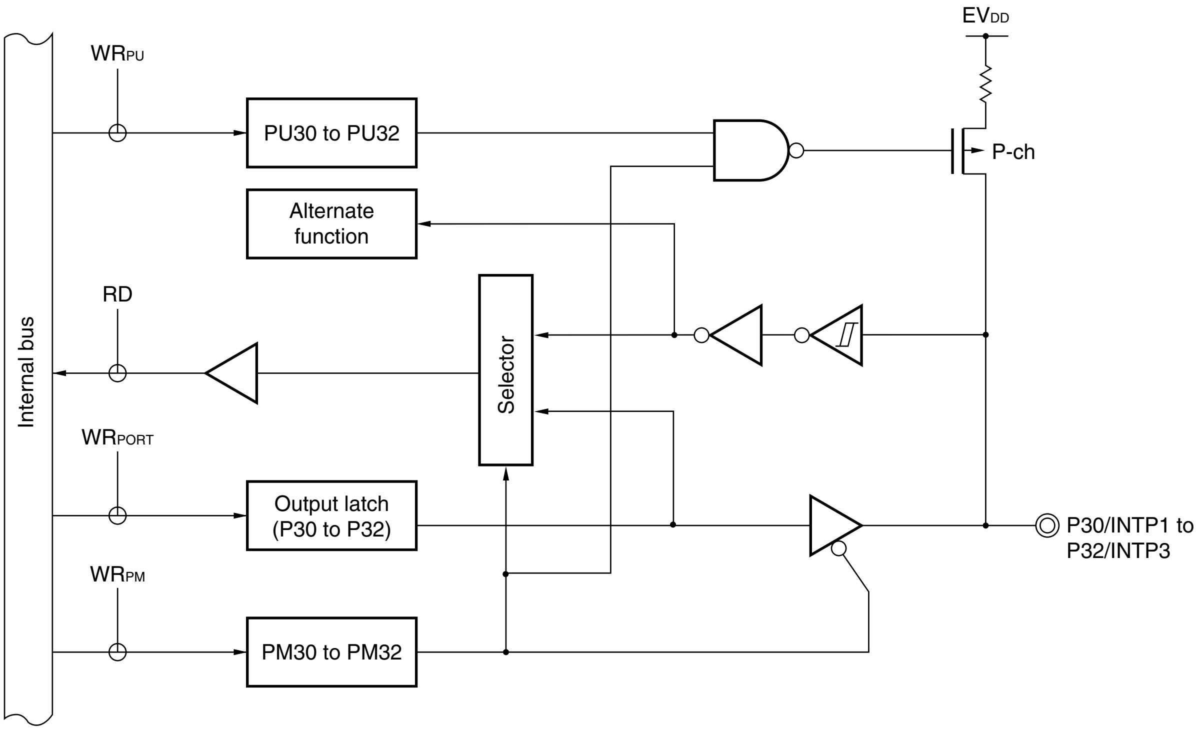
Port 3 is a 4-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

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

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

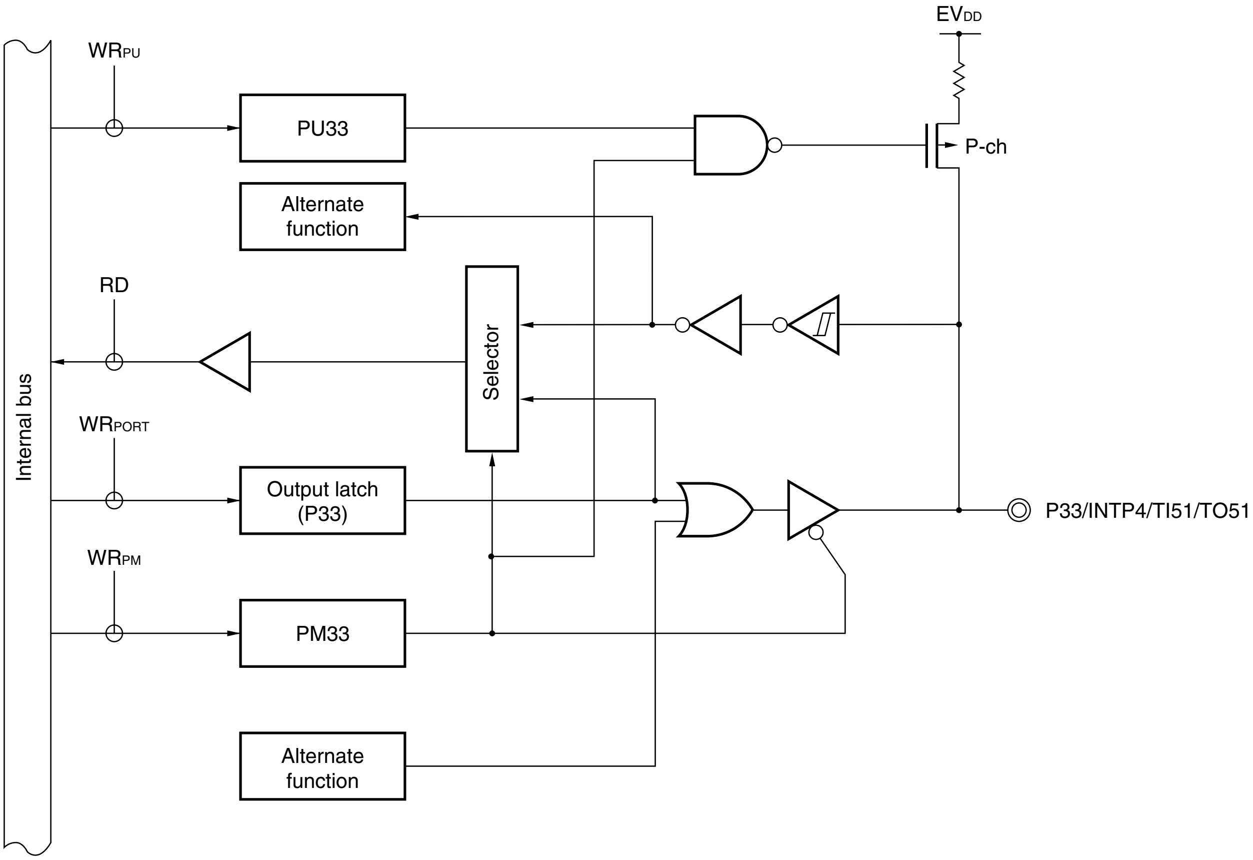
Figures 4-13 and 4-14 show block diagrams of port 3.

Figure 4-13. Block Diagram of P30 to P32



PU3:	Pull-up resistor option register 3
PM:	Port mode register
RD:	Port 3 read signal
WR:	Port 3 write signal

Figure 4-14. Block Diagram of P33



PU0:	Pull-up resistor option register 3
PM:	Port mode register
RD:	Port 3 read signal
WR:	Port 3 write signal

4.2.5 Port 4

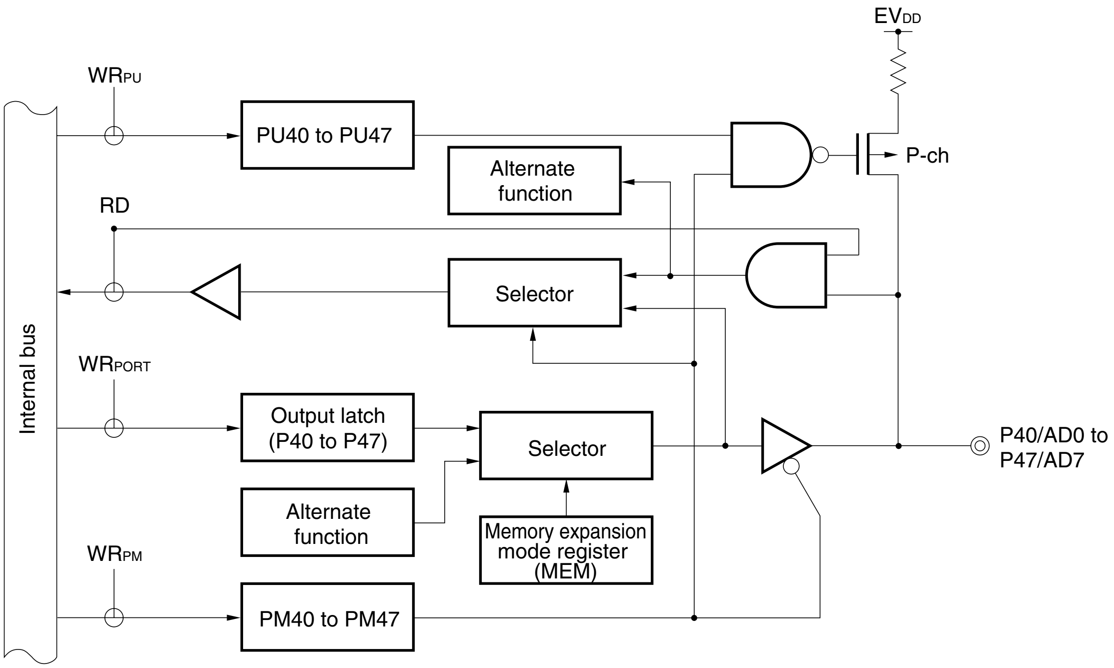
Port 4 is an 8-bit I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 4 (PU4).

This port can also be used as an address/data bus in external memory expansion mode.

$\overline{\text{RESET}}$ input sets port 4 to input mode.

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

Figure 4-15. Block Diagram of P40 to P47



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 4 read signal
WR: Port 4 write signal

4.2.6 Port 5

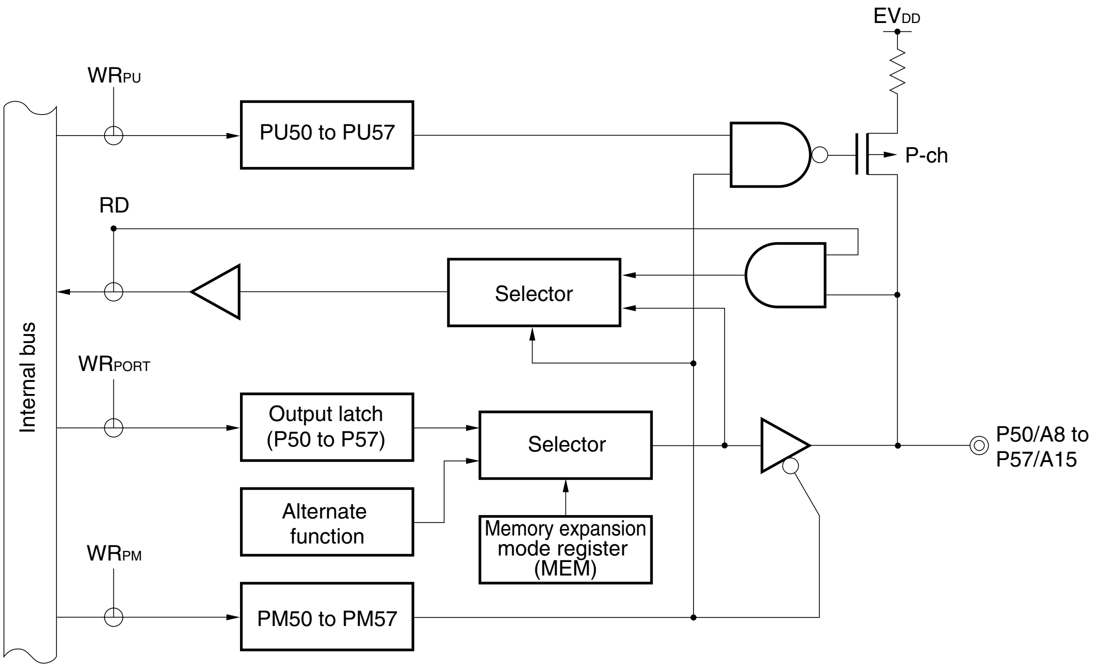
Port 5 is an 8-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified in 1-bit units using pull-up resistor option register 5 (PU5).

This port can also be used as an address bus in external memory expansion mode.

$\overline{\text{RESET}}$ input sets port 5 to input mode.

Figure 4-16 shows a block diagram of port 5.

Figure 4-16. Block Diagram of P50 to P57



PU:	Pull-up resistor option register
PM:	Port mode register
RD:	Port 5 read signal
WR:	Port 5 write signal

4.2.7 Port 6

Port 6 is an 8-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

This port has the following functions for pull-up resistors. These functions differ depending on the higher 4 bits/lower 4 bits of the port, and whether the product is a mask ROM version or a flash memory version.

Table 4-3. Pull-up Resistor of Port 6

	Higher 4 Bits (Pins P64 to P67)	Lower 4 Bits (Pins P60 to P63)
Mask ROM version	An on-chip pull-up resistor can be connected in 1-bit units by PU6	An on-chip pull-up resistor can be specified in 1-bit units by mask option
Flash memory version		On-chip pull-up resistors are not provided

PU6: Pull-up resistor option register 6

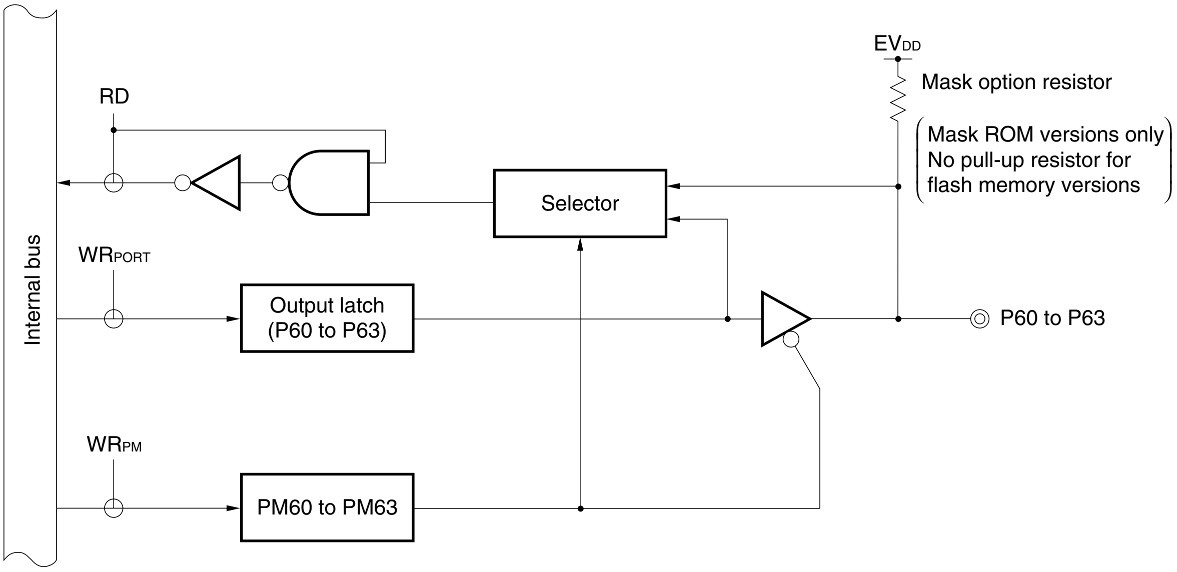
The P64 to P67 pins can also be used for the control signal output function in external memory expansion mode.

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

Figures 4-17 to 4-19 show block diagrams of port 6.

Caution P66 can be used as an I/O port when an external wait is not used in external memory expansion mode.

Figure 4-17. Block Diagram of P60 to P63

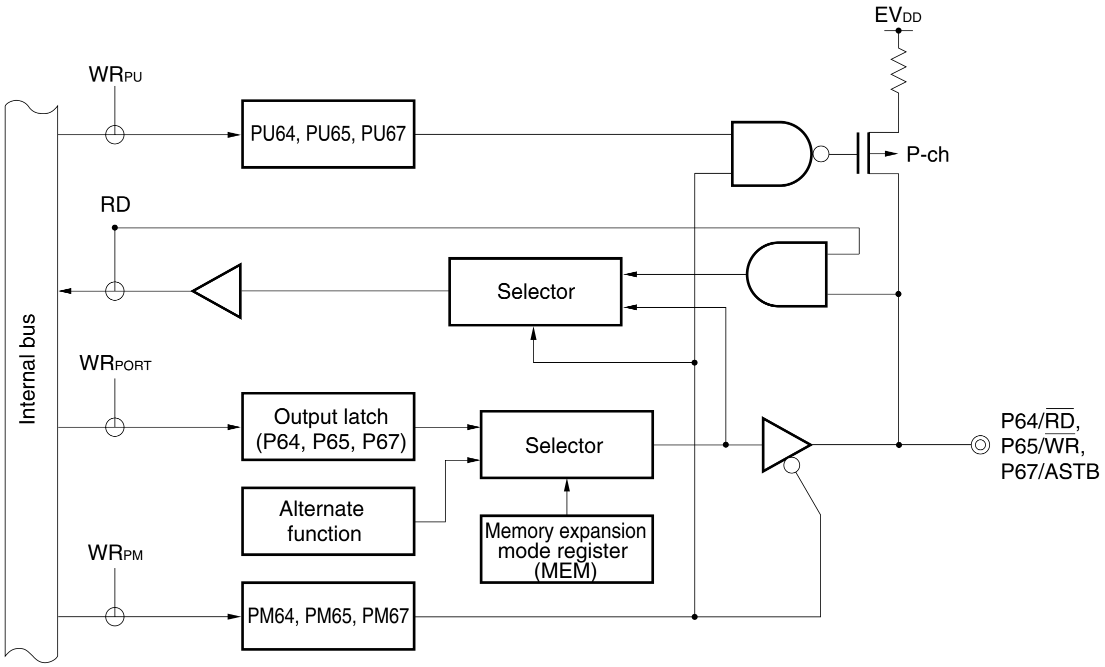


PM: Port mode register

RD: Port 6 read signal

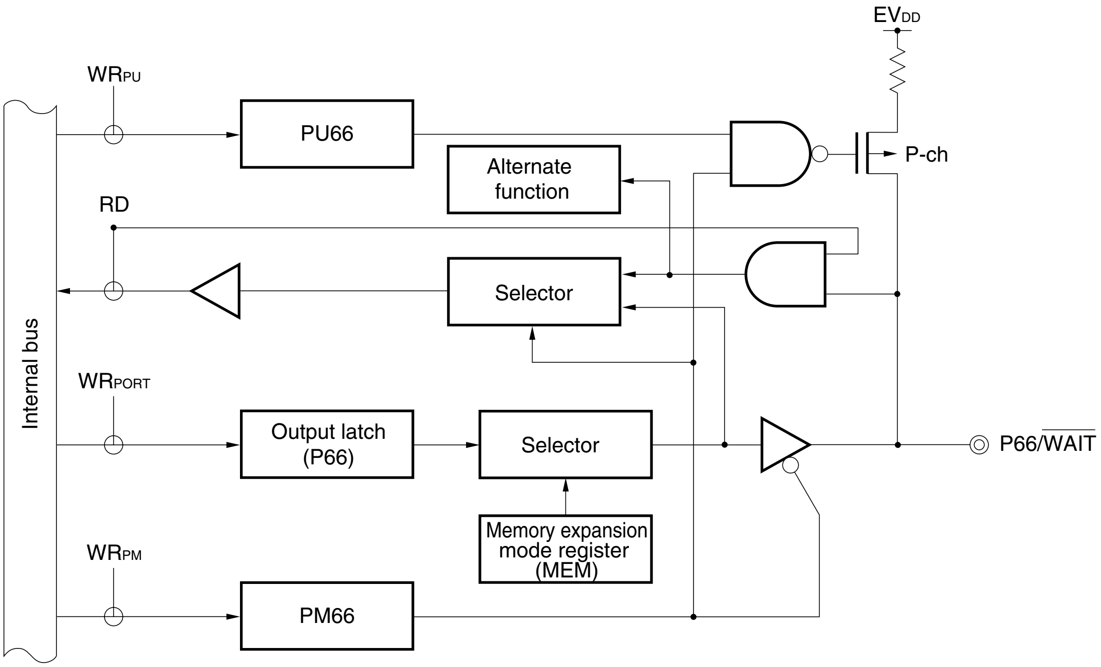
WR: Port 6 write signal

Figure 4-18. Block Diagram of P64, P65, and P67



PU6:	Pull-up resistor option register 6
PM:	Port mode register
RD:	Port 6 read signal
WR:	Port 6 write signal

Figure 4-19. Block Diagram of P66



PU6:	Pull-up resistor option register 6
PM:	Port mode register
RD:	Port 6 read signal
WR:	Port 6 write signal

4.2.8 Port 7

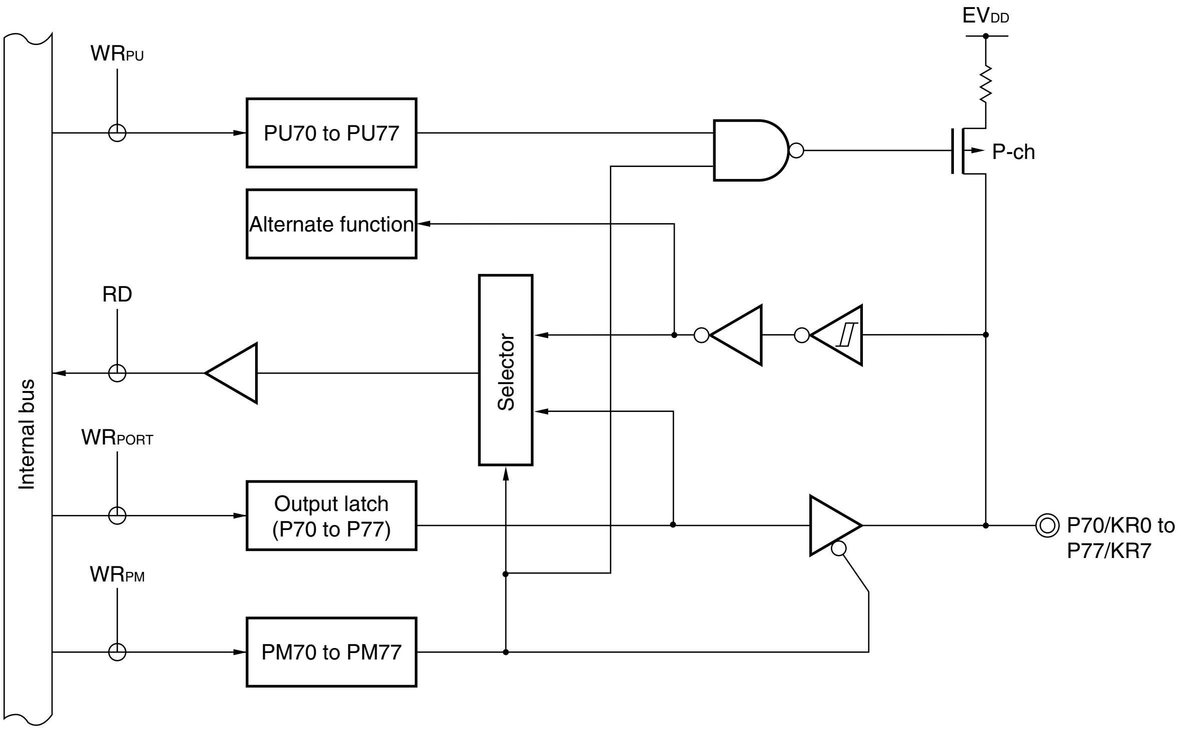
Port 7 is an 8-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P77 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

This port can also be used for key return input.

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

Figure 4-20 shows a block diagram of port 7.

Figure 4-20. Block Diagram of P70 to P77



PU7:	Pull-up resistor option register 7
PM:	Port mode register
RD:	Port 7 read signal
WR:	Port 7 write signal

4.2.9 Port 12

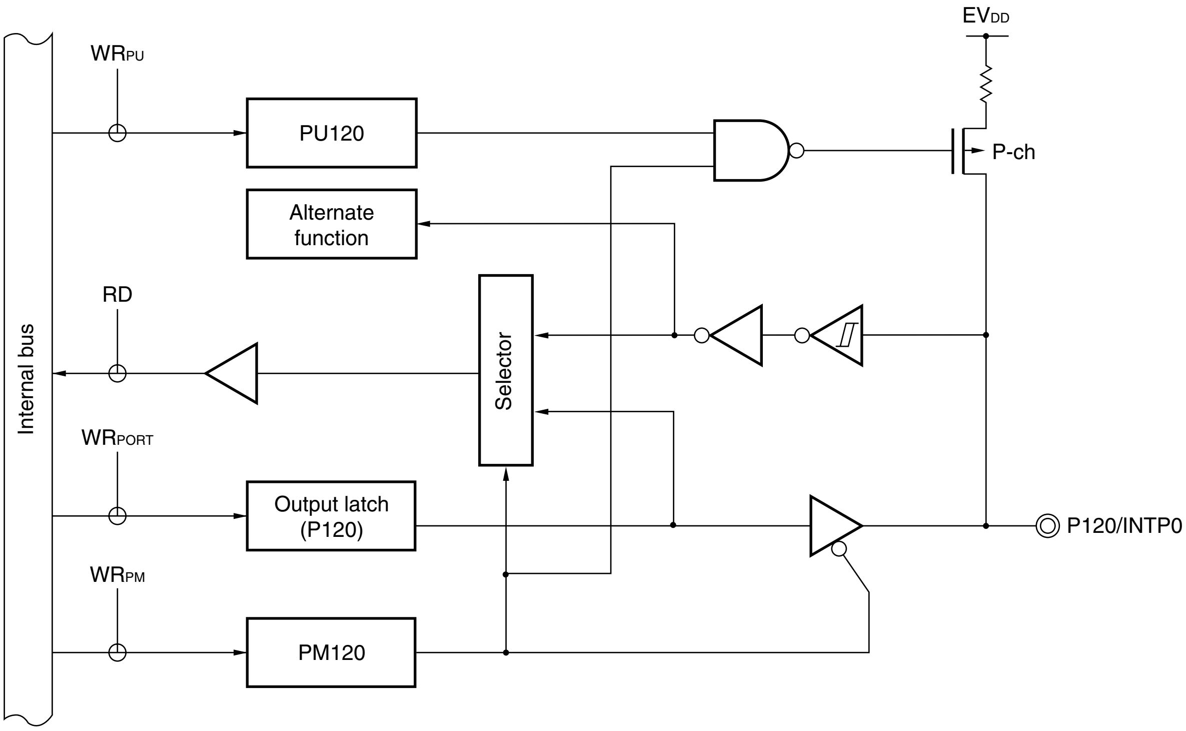
Port 12 is a 1-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

This port can also be used for external interrupt input.

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

Figure 4-21 shows a block diagram of port 12.

Figure 4-21. Block Diagram of P120



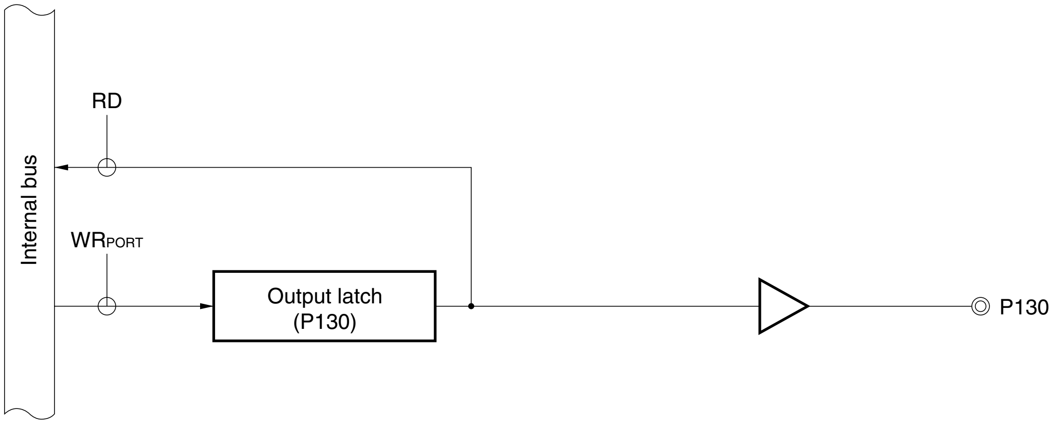
PU12:	Pull-up resistor option register 12
PM:	Port mode register
RD:	Port 12 read signal
WR:	Port 12 write signal

4.2.10 Port 13

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

Figure 4-22 shows a block diagram of port 13.

Figure 4-22. Block Diagram of P130



RD: Port 13 read signal

WD: Port 13 write signal

4.2.11 Port 14

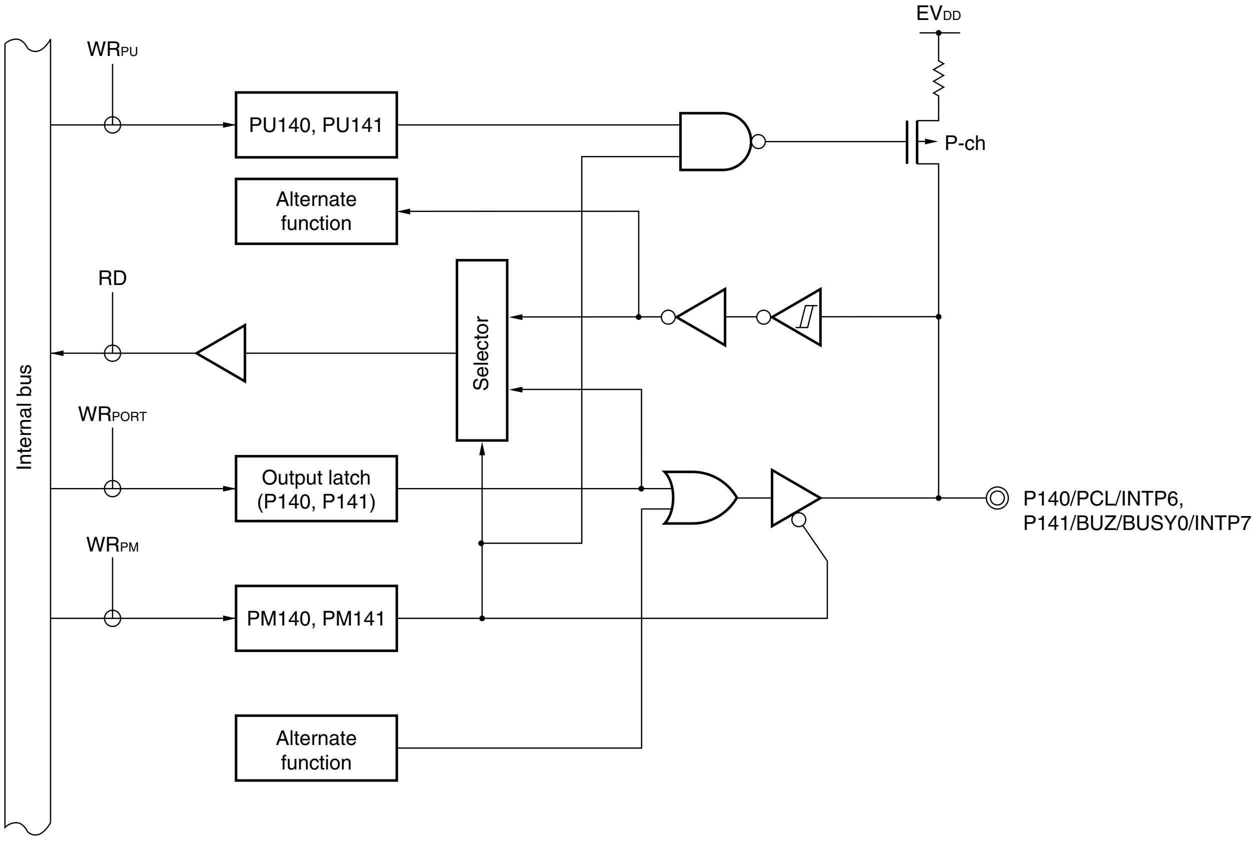
Port 14 is a 6-bit I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 to P145 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, busy input, buzzer output, and clock output.

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

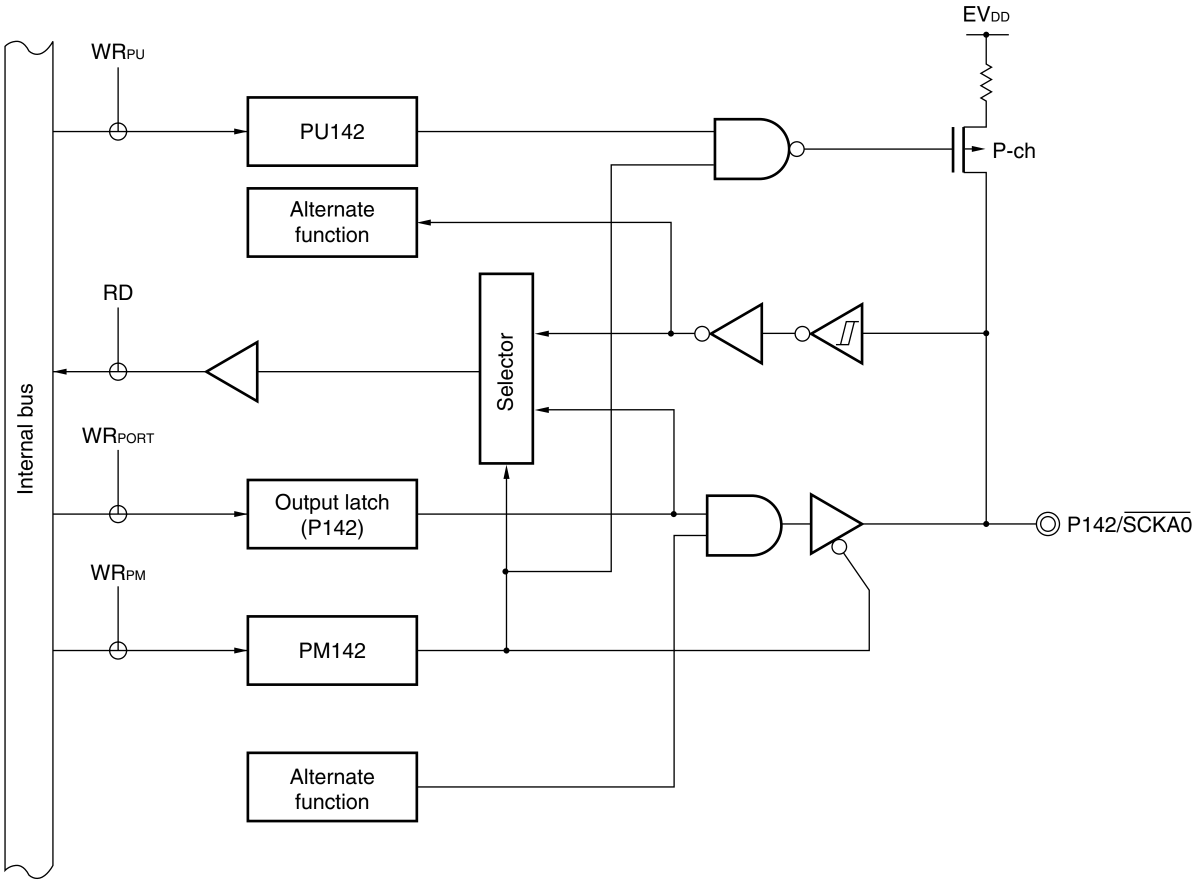
Figures 4-23 to 4-26 show block diagrams of port 14.

Figure 4-23. Block Diagram of P140 and P141



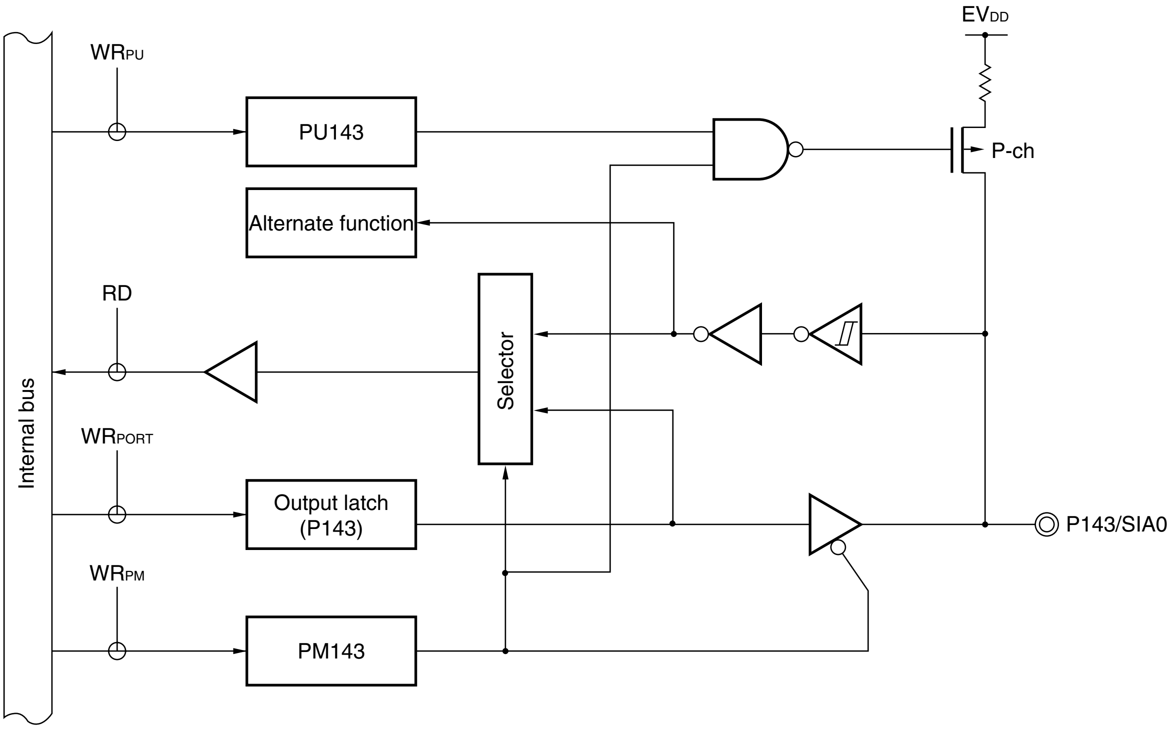
PU14:	Pull-up resistor option register 14
PM:	Port mode register
RD:	Port 14 read signal
WR:	Port 14 write signal

Figure 4-24. Block Diagram of P142



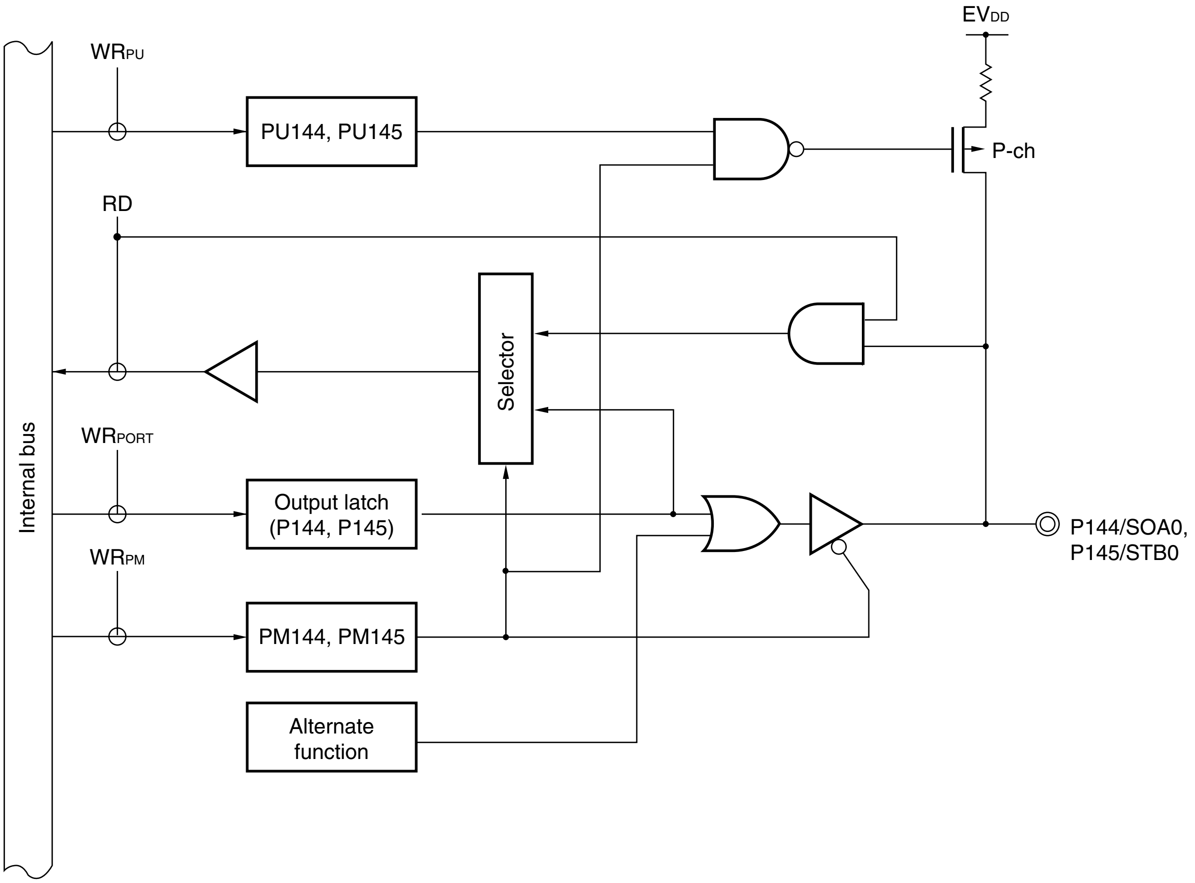
PU14: Pull-up resistor option register 14
PM: Port mode register
RD: Port 14 read signal
WR: Port 14 write signal

Figure 4-25. Block Diagram of P143



PU14:	Pull-up resistor option register 14
PM:	Port mode register
RD:	Port 14 read signal
WR:	Port 14 write signal

Figure 4-26. Block Diagram of P144 and P145



PU14: Pull-up resistor option register 14

PM: Port mode register

RD: Port 14 read signal

WR: Port 14 write signal

4.3 Registers Controlling Port Function

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

- Port mode registers (PM0, PM1, PM3 to PM7, PM12, PM14)
- Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU12, PU14)
- Input switch control register (ISC)

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

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

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

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

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

Figure 4-27. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FF2CH	FFH	R/W
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140	FF2EH	FFH	R/W

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

Table 4-4. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI000	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P02	SO11[Note]	Output	0	0
P03	SI11[Note]	Input	1	×
P04	$\overline{\text{SCK11}}$[Note]	Input	1	×
		Output	0	1
P05	$\overline{\text{SSI11}}$[Note]	Input	1	×
	TI001[Note]	Input	1	×
P06	TI011[Note]	Input	1	×
	TO01[Note]	Output	0	0
P10	$\overline{\text{SCK10}}$	Input	1	×
		Output	0	1
	TxD0	Output	0	1
P11	SI10	Input	1	×
	RxD0	Input	1	×
P12	SO10	Output	0	0
P13	TxD6	Output	0	1
P14	RxD6	Input	1	×
P15	TOH0	Output	0	0
P16	TOH1	Output	0	0
	INTP5	Input	1	×
P17	TI50	Input	1	×
	TO50	Output	0	0
P30 to P32	INTP1 to INTP3	Input	1	×
P33	INTP4	Input	1	×
	TI51	Input	1	×
	TO51	Output	0	0
P40 to P47	AD0 to AD7	Input	1	×
		Output	0	0
P50 to P57	A8 to A15	Output	0	0

Note SO11, SI11, $\overline{\text{SCK11}}$, $\overline{\text{SSI11}}$, TI001, TI011, and TO01 are available only in the μPD780146, 780148, and 78F0148.

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

Table 4-4. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P64	$\overline{\mathrm{RD}}$	Output	0	0
P65	$\overline{\mathrm{WR}}$	Output	0	0
P66	$\overline{\mathrm{WAIT}}$	Input	1	×
P67	ASTB	Output	0	0
P70 to P77	KR0 to KR7	Input	1	×
P120	INTP0	Input	1	×
P140	PCL	Output	0	0
	INTP6	Input	1	×
P141	BUZ	Output	0	0
	BUSY0	Input	1	×
	INTP7	Input	1	×
P142	$\overline{\mathrm{SCKA0}}$	Input	1	×
		Output	0	1
P143	SIA0	Input	1	×
P144	SOA0	Output	0	0
P145	STB0	Output	0	0

Remark ×: Don't care

PM××: Port mode register

P××: Port output latch

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

These registers specify whether the on-chip pull-up resistors of P00 to P06, P10 to P17, P30 to P33, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, or P140 to P145 are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified. 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, PU1, PU3 to PU7, PU12, and PU14.

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

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

Caution Use of a pull-up resistor can be specified for P60 to P63 pins by a mask option only in the mask ROM versions.

Figure 4-28. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	FF34H	00H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	FF35H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	FF36H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	FF37H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
PU14	0	0	PU145	PU144	PU143	PU142	PU141	PU140	FF3EH	00H	R/W

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

(3) Input switch control register (ISC)

This register is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception. The input signal is switched by setting ISC.

For the port configuration during LIN reception, refer to **Figure 15-3 Port Configuration for LIN Reception Operation** in **CHAPTER 15 SERIAL INTERFACE UART6**.

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 4-29. 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	Input signal selection
0	TI000 input
1	RxD6 input

ISC0	Input signal selection
0	INTP0 input
1	RxD6 input

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.

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

(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.

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.

4.4.2 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.

(2) Input mode

The output latch contents are undefined, but since the output buffer is off, the pin status does not change.

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.

CHAPTER 5 EXTERNAL BUS INTERFACE

5.1 External Bus Interface

The external bus interface connects external devices to areas other than the internal ROM, RAM, and SFR areas. Connection of external devices uses ports 4 to 6. Ports 4 to 6 control address/data, read/write strobe, wait, address strobe, etc.

The external bus interface is usable only when the X1 clock is selected as the CPU clock.

Table 5-1. Pin Functions in External Memory Expansion Mode

Pin Function When External Device Is Connected		Alternate Function
Name	Function	
AD0 to AD7	Multiplexed address/data bus	P40 to P47
A8 to A15	Address bus	P50 to P57
$\overline{\text{RD}}$	Read strobe signal	P64
$\overline{\text{WR}}$	Write strobe signal	P65
$\overline{\text{WAIT}}$	Wait signal	P66
ASTB	Address strobe signal	P67

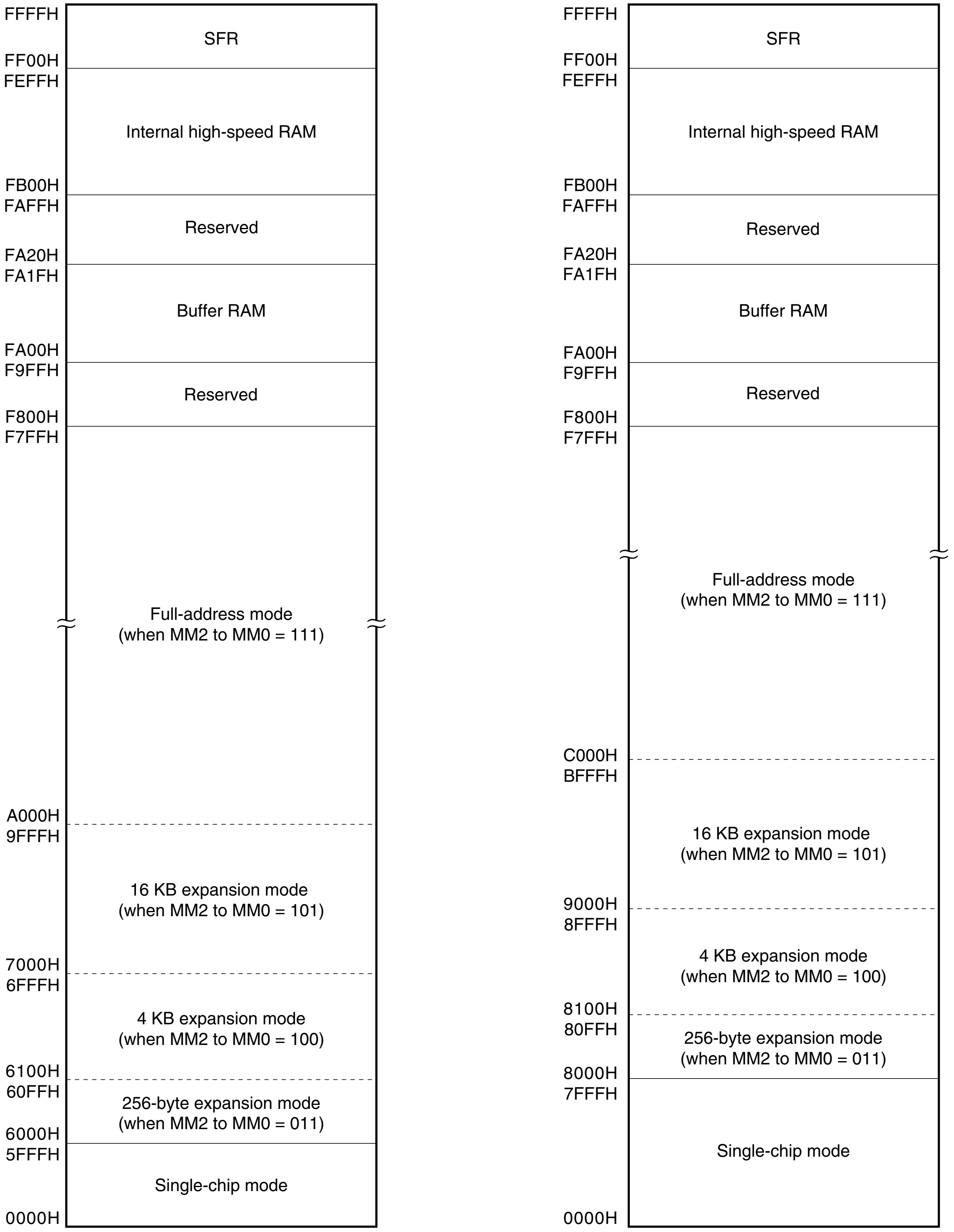
Table 5-2. State of Ports 4 to 6 Pins in External Memory Expansion Mode

Port External Expansion Mode	Port 4	Port 5								Port 6							
	0 to 7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Single-chip mode	Port	Port								Port							
256-byte expansion mode	Address/data	Port								Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			
4 KB expansion mode	Address/data	Address				Port				Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			
16 KB expansion mode	Address/data	Address						Port		Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			
Full-address mode	Address/data	Address								Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			

Caution When the external wait function is not used, the $\overline{\text{WAIT}}$ pin can be used as a port in all modes.

The memory maps when the external bus interface is used are as follows.

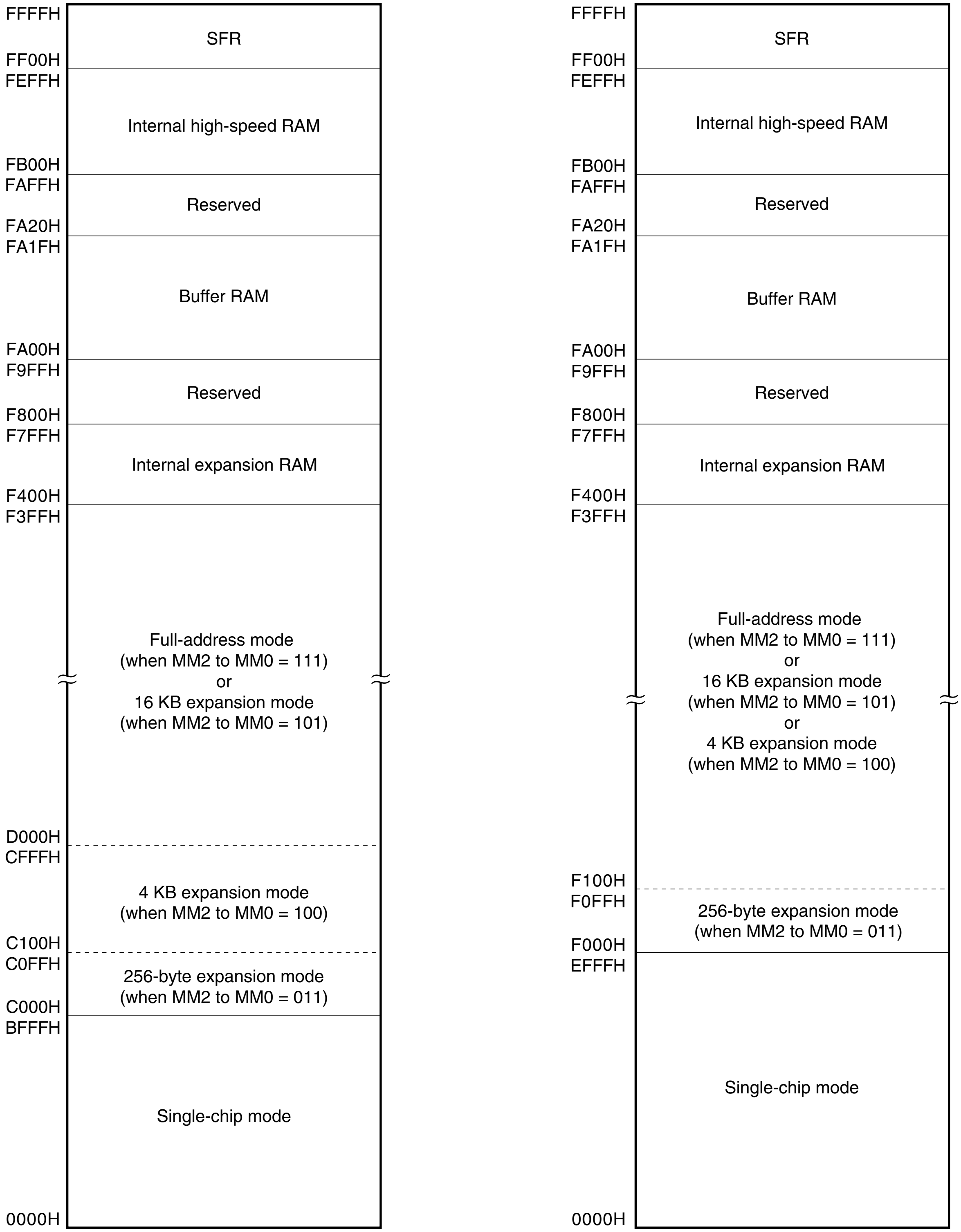
Figure 5-1. Memory Map When Using External Bus Interface (1/2)



(a) Memory map of μPD780143 and of μPD78F0148 when internal ROM (flash memory) size is 24 KB

(b) Memory map of μPD780144 and of μPD78F0148 when internal ROM (flash memory) size is 32 KB

Figure 5-1. Memory Map When Using External Bus Interface (2/2)



(c) Memory map of μPD780146 and of μPD78F0148 when internal ROM (flash memory) size is 48 KB

(d) Memory map of μPD780148 and of μPD78F0148 when internal ROM (flash memory) size is 60 KB

5.2 Registers Controlling External Bus Interface

The external bus interface is controlled by the following two registers.

- Memory expansion mode register (MEM)
- Memory expansion wait setting register (MM)

(1) Memory expansion mode register (MEM)

MEM sets the external expansion area.

MEM is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MEM to 00H.

Figure 5-2. Format of Memory Expansion Mode Register (MEM)

Address: FF47H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MEM	0	0	0	0	0	MM2	MM1	MM0

MM2	MM1	MM0	Single-chip/memory expansion mode selection		P40 to P47, P50 to P57, P64 to P67 pin state				
					P40 to P47	P50 to P53	P54, P55	P56, P57	P64 to P67
0	0	0	Single-chip mode		Port mode				
0	1	1	Memory expansion mode	256-byte mode	AD0 to AD7	Port mode			P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = $\overline{\text{WAIT}}$ P67 = ASTB
1	0	0		4 KB mode		A8 to A11	Port mode		
1	0	1		16 KB mode			A12, A13	Port mode	
1	1	1		Full-address mode[Note]				A14, A15	
Other than above			Setting prohibited						

Note The full-address mode allows external expansion to the entire 64 KB address space except for the internal ROM, RAM, SFR areas and the reserved areas.

(2) Memory expansion wait setting register (MM)

MM sets the number of waits.

MM is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MM to 10H.

Figure 5-3. Format of Memory Expansion Wait Setting Register (MM)

Address: FFF8H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
MM	0	0	PW1	PW0	0	0	0	0

PW1	PW0	Wait control
0	0	No wait
0	1	Wait (one wait state inserted)
1	0	Setting prohibited
1	1	Wait control by external wait pin

Caution To control wait with external wait pin, be sure to set $\overline{\text{WAIT}}$/P66 pin to input mode (set bit 6 (PM66) of port mode register 6 (PM6) to 1).

5.3 External Bus Interface Function Timing

Timing control signal output pins in the external memory expansion mode are as follows.

(1) $\overline{\text{RD}}$ pin (Alternate function: P64)

Read strobe signal output pin. The read strobe signal is output in data access and instruction fetch from external memory.

During internal memory access, the read strobe signal is not output (maintains high level).

(2) $\overline{\text{WR}}$ pin (Alternate function: P65)

Write strobe signal output pin. The write strobe signal is output in data access to external memory.

During internal memory access, the write strobe signal is not output (maintains high level).

(3) $\overline{\text{WAIT}}$ pin (Alternate function: P66)

External wait signal input pin.

When the external wait is not used, the $\overline{\text{WAIT}}$ pin can be used as an I/O port.

During internal memory access, the external wait signal is ignored.

(4) ASTB pin (Alternate function: P67)

Address strobe signal output pin. The address strobe signal is output regardless of data access and instruction fetch from external memory.

During internal memory access, the address strobe signal is output.

(5) AD0 to AD7, A8 to A15 pins (Alternate function: P40 to P47, P50 to P57)

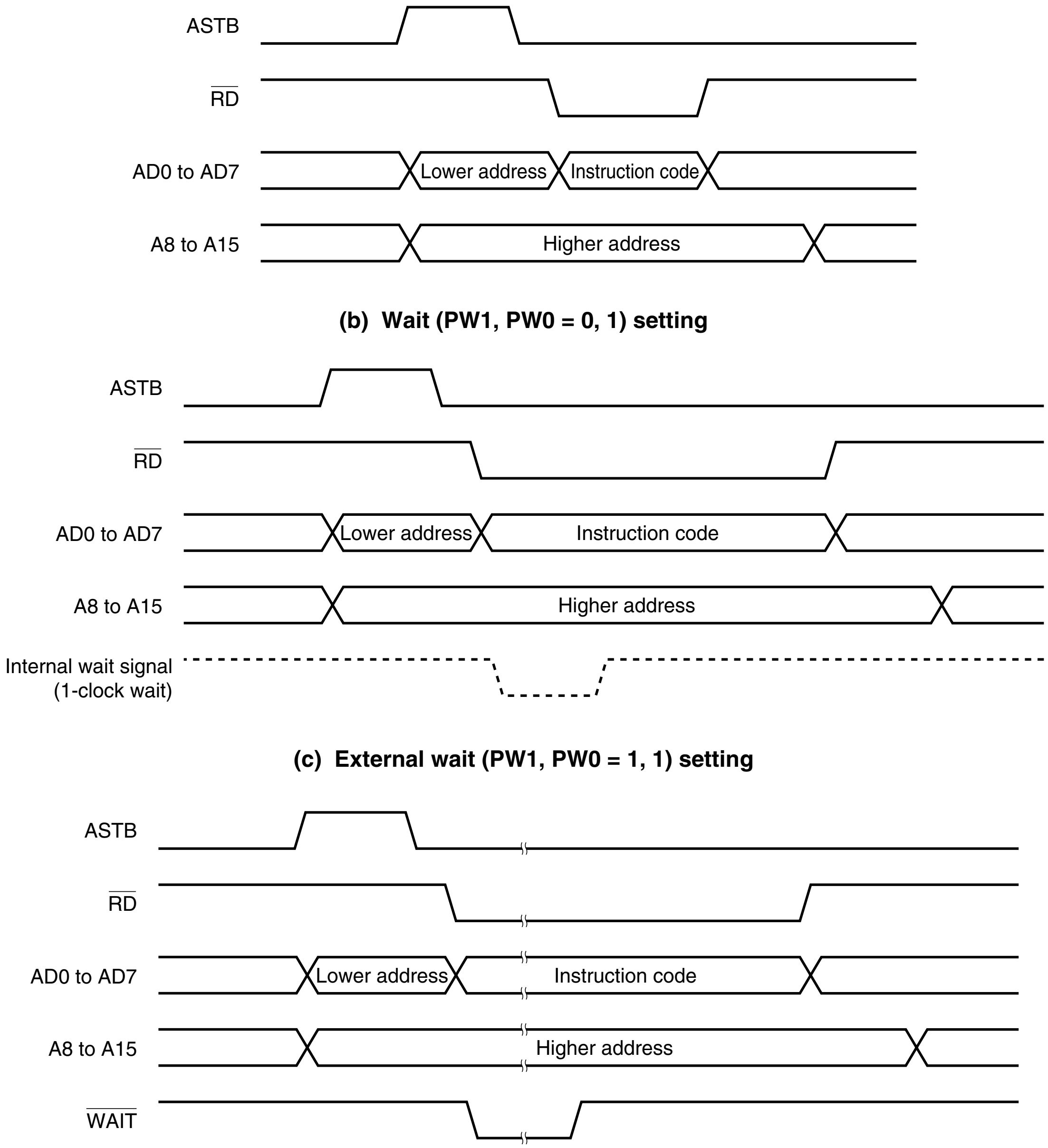
Address/data signal output pins. Valid signal is output or input during data accesses and instruction fetches from external memory.

These signals change even during internal memory access (output values are undefined).

The timing charts are shown in Figures 5-4 to 5-7.

Figure 5-4. Instruction Fetch from External Memory

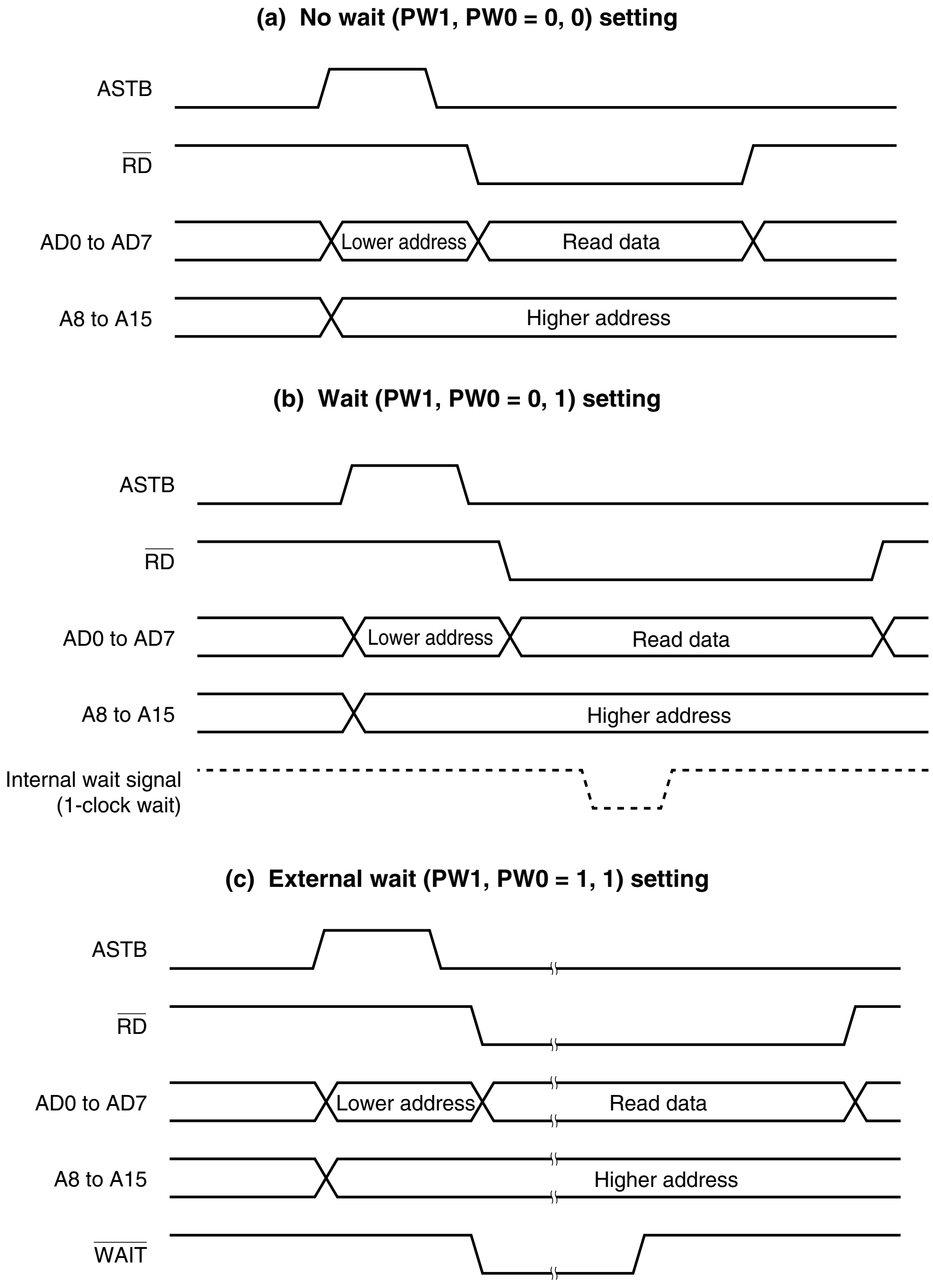
(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

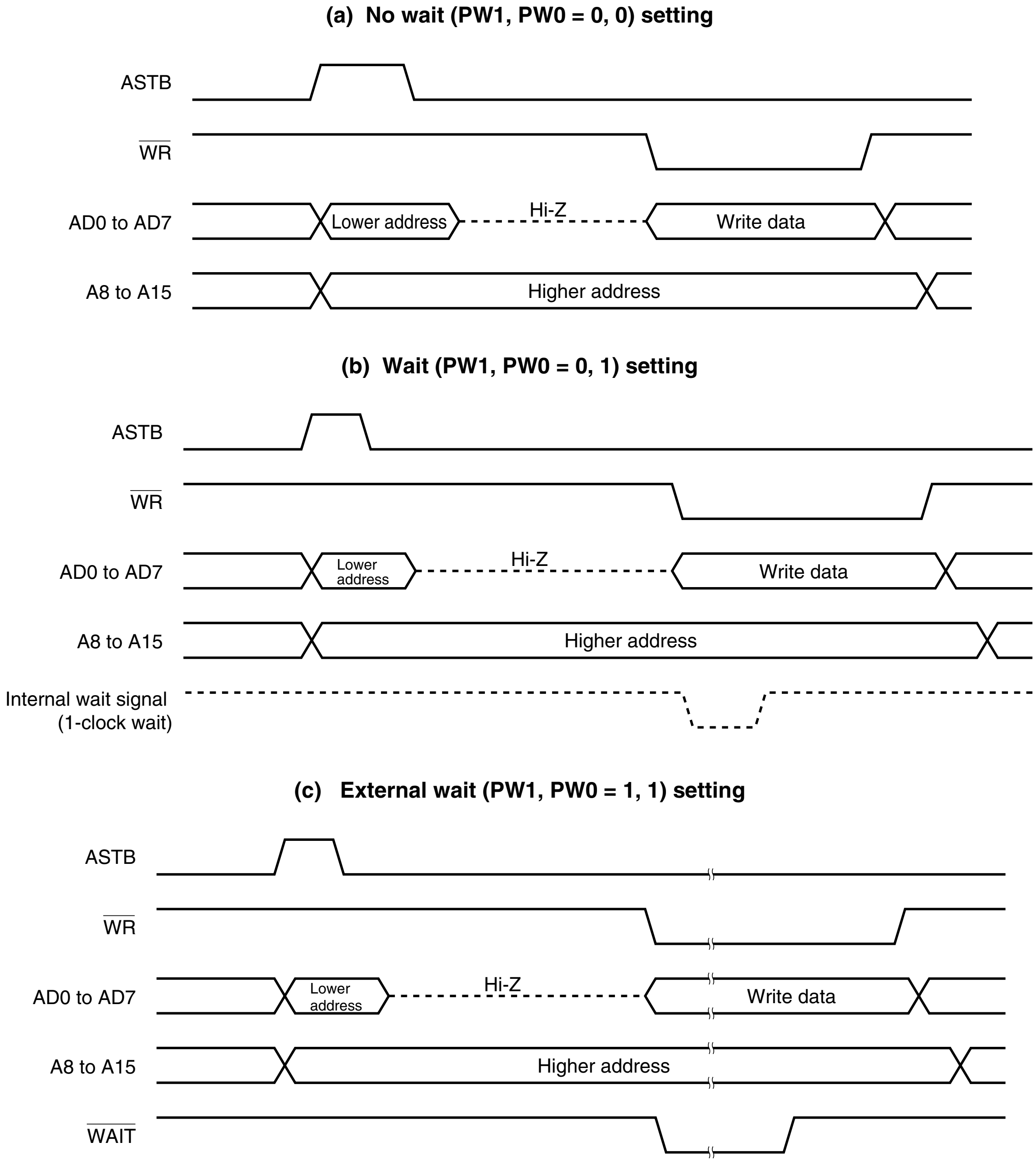
Figure 5-5. External Memory Read Timing


(a) No wait (PW1, PW0 = 0, 0) setting

(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

Figure 5-6. External Memory Write Timing

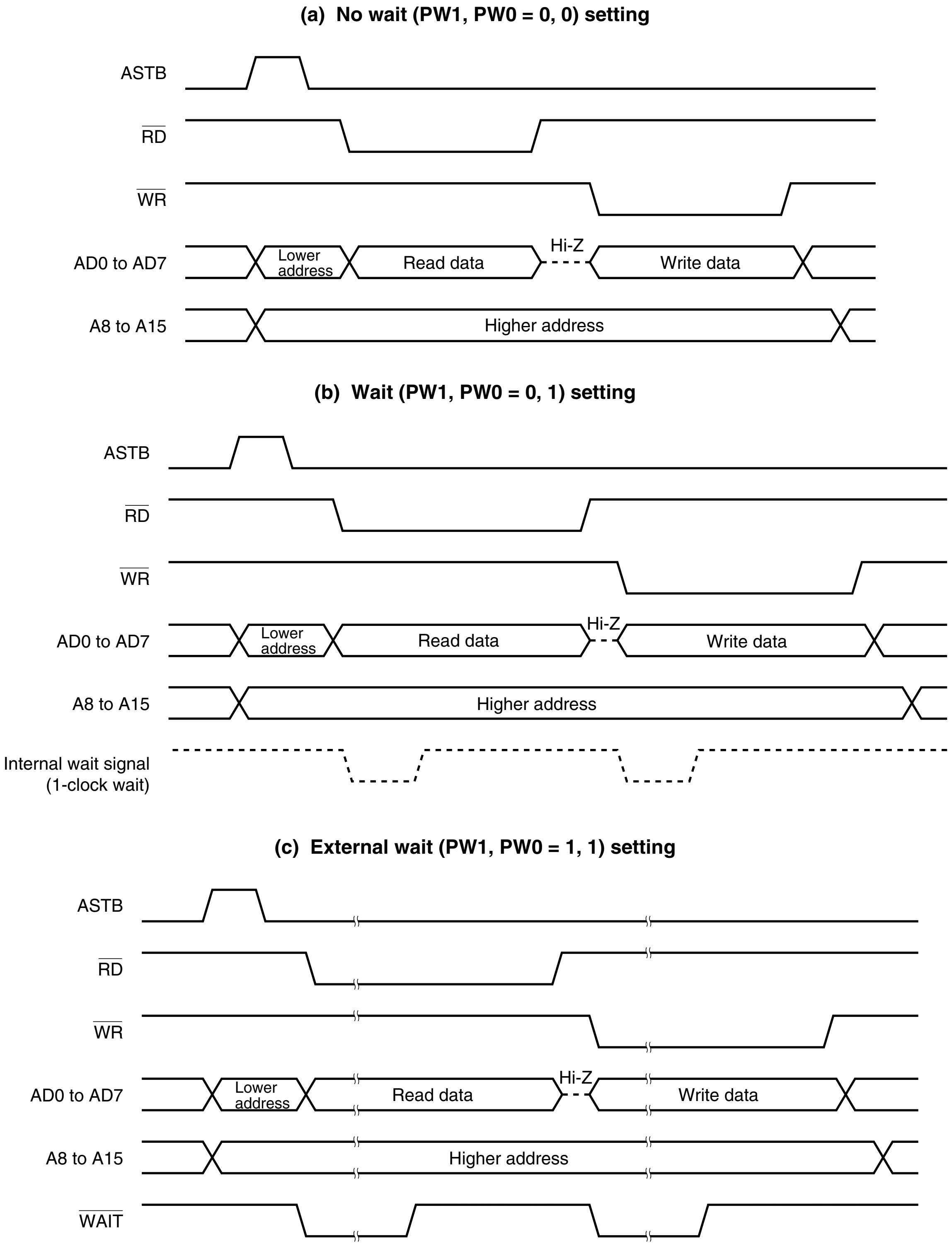


(a) No wait (PW1, PW0 = 0, 0) setting

(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

Figure 5-7. External Memory Read Modify Write Timing



(a) No wait (PW1, PW0 = 0, 0) setting

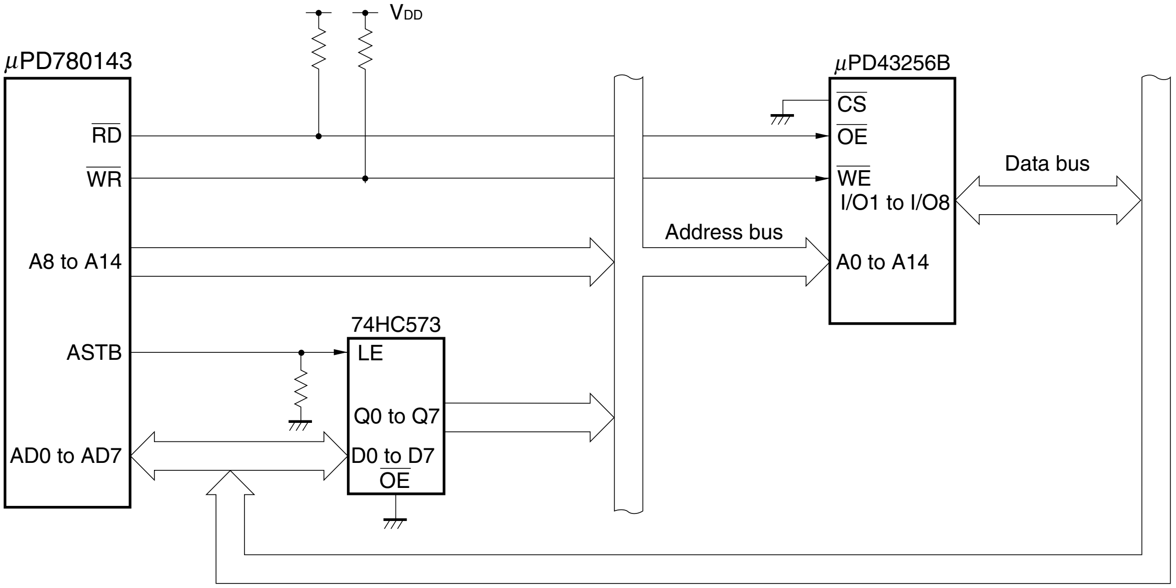
(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

5.4 Example of Connection with Memory

An example of connecting the μPD780143 with external memory (in this example, SRAM) is shown in Figure 5-8. In addition, the external bus interface function is used in the full-address mode, and the addresses from 0000H to 7FFFH (32 KB) are allocated to internal ROM, and the addresses after 8000H to SRAM.

Figure 5-8. Connection Example of μPD780143 and Memory



CHAPTER 6 CLOCK GENERATOR

6.1 Functions of Clock Generator

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

- X1 oscillator
  The X1 oscillator oscillates a clock of 2.0 to 10.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the main OSC control register (MOC) and processor clock control register (PCC).
- Ring-OSC oscillator
  The Ring-OSC oscillator oscillates a clock of 240 kHz (TYP.). Oscillation can be stopped by setting the Ring-OSC mode register (RCM) when "Can be stopped by software" is set by a mask option and the X1 input clock is used as the CPU clock.
- Subsystem clock oscillator
  The subsystem clock oscillator oscillates a clock of 32.768 kHz. Oscillation cannot be stopped. When subsystem clock oscillator is not used, setting not to use the on-chip feedback resistor is possible using the processor clock control register (PCC), and the power consumption can be reduced in the STOP mode.

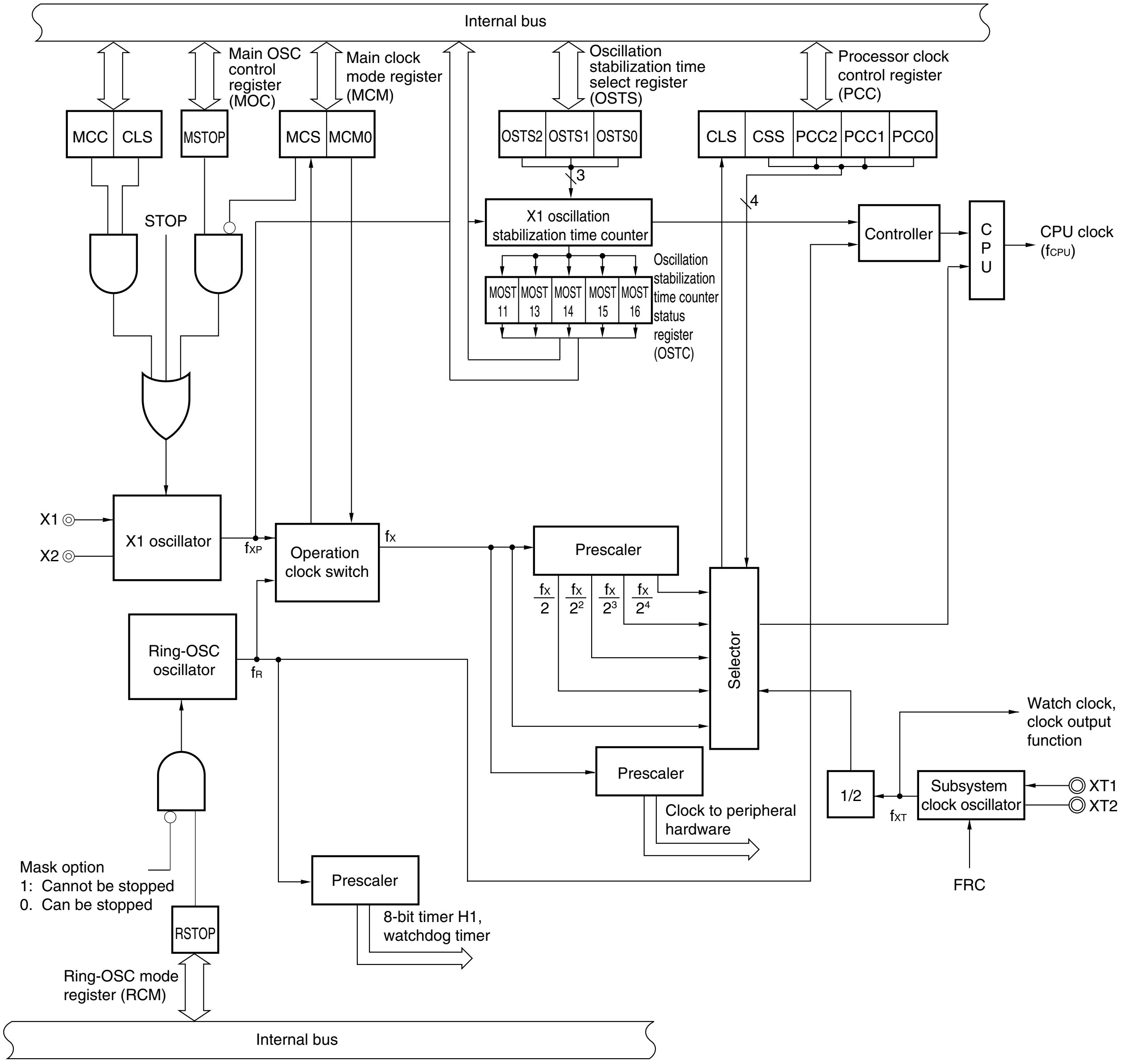
6.2 Configuration of Clock Generator

The clock generator consists of the following hardware.

Table 6-1. Configuration of Clock Generator

Item	Configuration
Control registers	Processor clock control register (PCC) Ring-OSC mode register (RCM) Main clock mode register (MCM) Main OSC control register (MOC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS)
Oscillator	X1 oscillator Ring-OSC oscillator Subsystem clock oscillator

Figure 6-1. Block Diagram of Clock Generator



6.3 Registers Controlling Clock Generator

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

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

(1) Processor clock control register (PCC)

The PCC register is used to select the CPU clock, the division ratio, main system clock oscillator operation/stop and whether to use the on-chip feedback resistor of the subsystem clock oscillator.

The PCC is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 6-2. Subsystem Clock Feedback Resistor

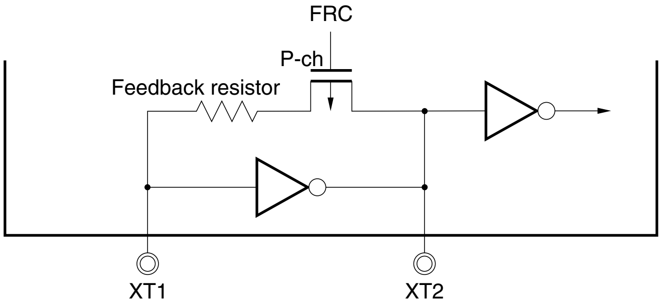


Figure 6-3. Format of Processor Clock Control Register (PCC)

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

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

MCC	Control of X1 oscillator operation[Note 2]
0	Oscillation possible
1	Oscillation stopped

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

CLS	CPU clock status
0	X1 input clock or Ring-OSC clock
1	Subsystem clock

CSS[Note 4]	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_X
	0	0	1	$f_X/2$
	0	1	0	$f_X/2^2$
	0	1	1	$f_X/2^3$
	1	0	0	$f_X/2^4$
1	0	0	0	$f_{XT}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

Notes 1. Bit 5 is read-only.

2. When the CPU is operating on the subsystem clock, MCC should be used to stop the X1 oscillator operation. When the CPU is operating on the Ring-OSC clock, use bit 7 (MSTOP) of the main OSC control register (MOC) to stop the X1 oscillator operation (this cannot be set by MCC). A STOP instruction should not be used.

3. The feedback resistor is required to adjust the bias point of the oscillation waveform to close to the middle of the power supply voltage. Setting FRC to 1 can further reduce the current consumption in the STOP mode, but only when the subsystem clock is not used.

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

Caution Be sure to set bit 3 to 0.

Remarks 1. f_X: Main system clock oscillation frequency (X1 input clock oscillation frequency or Ring-OSC clock oscillation frequency)

2. f_{XT}: Subsystem clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/KF1 Series. Therefore, the relationship between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in the Table 6-2.

Table 6-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$		
	X1 Input Clock[Note] (at 10 MHz Operation)	Ring-OSC Clock[Note] (at 240 kHz (TYP.) Operation)	Subsystem Clock (at 32.768 kHz Operation)
f_X	0.2 μs	8.3 μs (TYP.)	–
$f_X/2$	0.4 μs	16.6 μs (TYP.)	–
$f_X/2^2$	0.8 μs	33.2 μs (TYP.)	–
$f_X/2^3$	1.6 μs	66.4 μs (TYP.)	–
$f_X/2^4$	3.2 μs	132.8 μs (TYP.)	–
$f_{XT}/2$	–	–	122.1 μs

Note The main clock mode register (MCM) is used to set the CPU clock (X1 input clock/Ring-OSC clock) (see **Figure 6-5**).

(2) Ring-OSC mode register (RCM)

This register sets the operation mode of Ring-OSC.

This register is valid when “Can be stopped by software” is set for Ring-OSC by a mask option, and the X1 input clock or subsystem clock is selected as the CPU clock. If “Cannot be stopped” is selected for Ring-OSC 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 6-4. Format of Ring-OSC Mode Register (RCM)

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	$\boxed{0}$
RCM	0	0	0	0	0	0	0	RSTOP

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

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

(3) Main clock mode register (MCM)

This register sets the CPU clock (X1 input clock/Ring-OSC 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 6-5. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
MCM	0	0	0	0	0	0	MCS	MCM0

MCS	CPU clock status
0	Operates with Ring-OSC clock
1	Operates with X1 input clock

MCM0	Selection of clock supplied to CPU
0	Ring-OSC clock
1	X1 input clock

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

Operation of the peripheral hardware with Ring-OSC clock cannot be guaranteed. Therefore, when Ring-OSC clock is selected as the clock supplied to the CPU, do not use peripheral hardware. In addition, stop the peripheral hardware before switching the clock supplied to the CPU from the X1 input clock to the Ring-OSC clock. Note, however, that the following peripheral hardware can be used when the CPU operates on the Ring-OSC clock.

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

2. Set MCS = 1 and MCM0 = 1 before switching subsystem clock operation to X1 input clock operation (bit 4 (CSS) of the processor clock control register (PCC) is changed from 1 to 0).

(4) Main OSC control register (MOC)

This register selects the operation mode of the X1 input clock.

This register is used to stop the X1 oscillator operation when the CPU is operating with the Ring-OSC clock. Therefore, this register is valid only when the CPU is operating with the Ring-OSC clock.

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

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

Figure 6-6. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	Control of X1 oscillator operation
0	X1 oscillator operating
1	X1 oscillator stopped

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

2. To stop X1 oscillation during operation with the subsystem clock, set bit 7 (MCC) of the processor clock control register (PCC) to 1 (setting by MSTOP is not possible).

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

This is the status register of the X1 input clock oscillation stabilization time counter. If the Ring-OSC clock is used as the CPU clock, the X1 input clock oscillation stabilization time can be checked.

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

★ $\overline{\text{RESET}}$ input, STOP instruction, MSTOP = 1, and MCC = 1 clear OSTC to 00H.

Figure 6-7. 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	$2^{11}/f_{XP}$ min. (204.8 μs min.)
1	1	0	0	0	$2^{13}/f_{XP}$ min. (819.2 μs min.)
1	1	1	0	0	$2^{14}/f_{XP}$ min. (1.64 ms min.)
1	1	1	1	0	$2^{15}/f_{XP}$ min. (3.27 ms min.)
1	1	1	1	1	$2^{16}/f_{XP}$ min. (6.55 ms min.)

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

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

2. f_{XP}: X1 input clock oscillation frequency

(6) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 oscillation stabilization wait time when STOP mode is released.

The wait time set by OSTS is valid only after STOP mode is released with the X1 input clock selected as CPU clock. After STOP mode is released with Ring-OSC selected as CPU clock, the oscillation stabilization time must be confirmed by OSTC.

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

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

Figure 6-8. 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

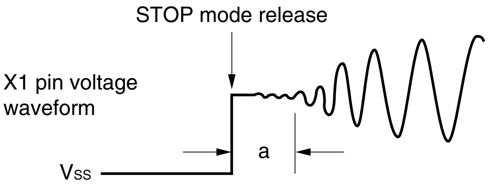
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	1	$2^{11}/f_{XP}$ (204.8 μs)
0	1	0	$2^{13}/f_{XP}$ (819.2 μs)
0	1	1	$2^{14}/f_{XP}$ (1.64 ms)
1	0	0	$2^{15}/f_{XP}$ (3.27 ms)
1	0	1	$2^{16}/f_{XP}$ (6.55 ms)
Other than above			Setting prohibited

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

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

The X1 oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

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



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

2. f_{XP}: X1 input clock oscillation frequency

6.4 System Clock Oscillator

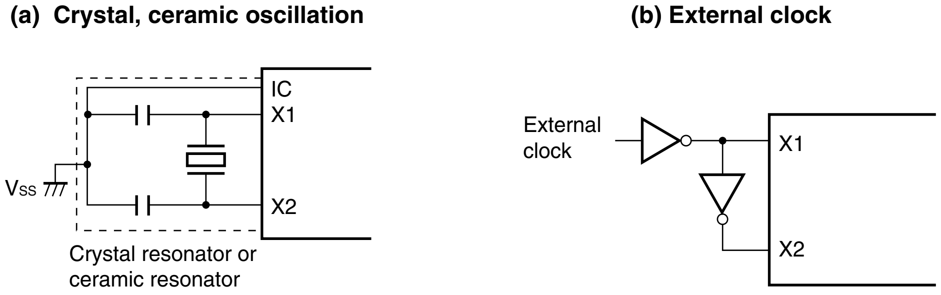
6.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (Standard: 8.38 MHz, 10 MHz when REGC pin is directly connected to V_{DD}) connected to the X1 and X2 pins.

An external clock can be input to the X1 oscillator when the REGC pin is directly connected to V_{DD}. In this case, input the clock signal to the X1 pin and input the inverse signal to the X2 pin.

Figure 6-9 shows the external circuit of the X1 oscillator.

Figure 6-9. External Circuit of X1 Oscillator



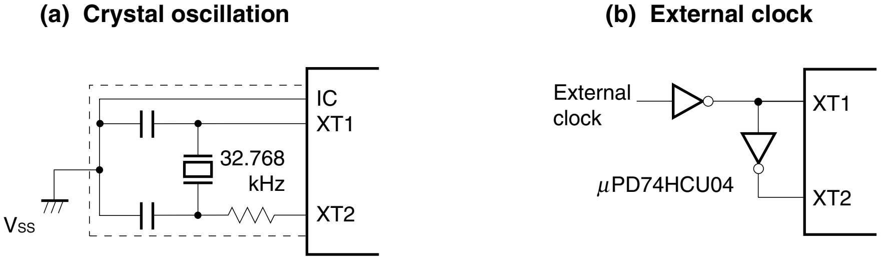
6.4.2 Subsystem clock oscillator

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

External clocks can be input to the subsystem clock oscillator when the REGC pin is directly connected to V_{DD}. In this case, input the clock signal to the XT1 pin and the inverse signal to the XT2 pin.

Figure 6-10 shows an external circuit of the subsystem clock oscillator.

Figure 6-10. External Circuit of Subsystem Clock Oscillator



Cautions are listed on the next page.

Cautions 1. When using the X1 oscillator and subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the Figure 6-11 to avoid an adverse effect from wiring capacitance.

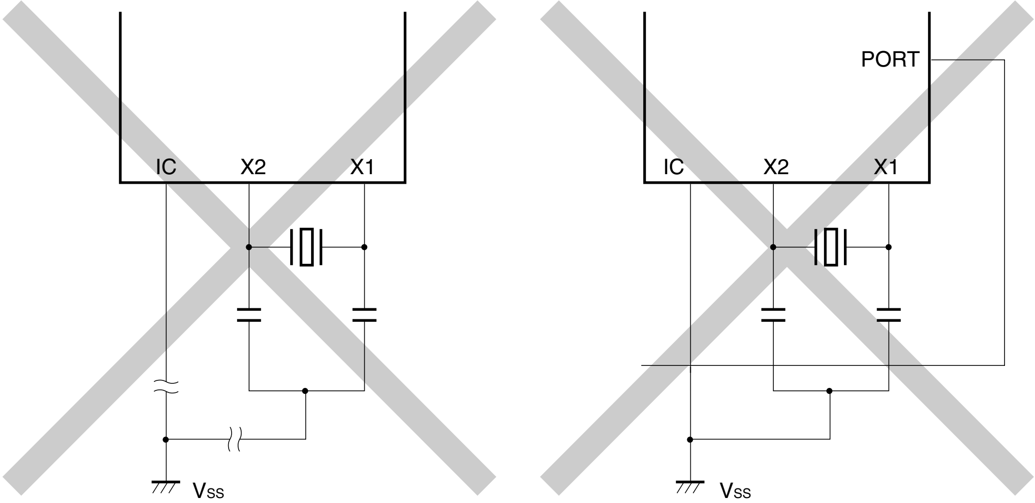
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do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

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

Figure 6-11 shows examples of incorrect resonator connection.

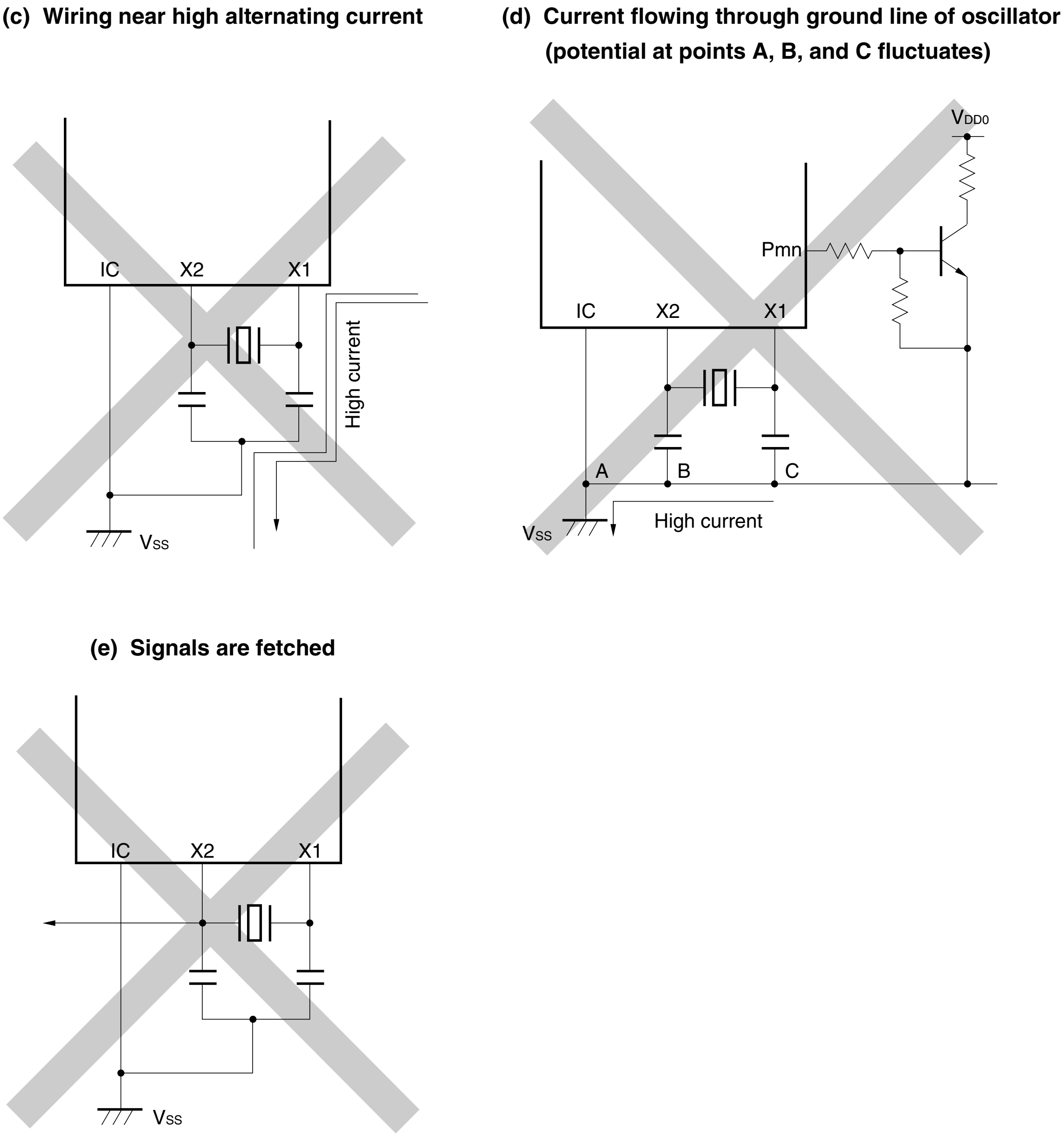
Figure 6-11. Examples of Incorrect Resonator Connection (1/2)

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



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

Figure 6-11. Examples of Incorrect Resonator Connection (2/2)



(c) Wiring near high alternating current

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

(e) Signals are fetched

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

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

To prevent that from occurring, it is recommended to wire X2 and XT1 so that they are not in parallel, and to connect the IC pin between X2 and XT1 directly to V_{SS}.

6.4.3 When subsystem clock is not used

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

XT1: Connect to EV_{DD} or V_{DD}
XT2: Leave open

In this state, however, some current may leak via the on-chip feedback resistor of the subsystem clock oscillator when the X1 input clock and Ring-OSC clock stop. To minimize leakage current, the above on-chip feedback resistor can be set not to be used via bit 6 (FRC) of the processor clock control register (PCC). In this case also, connect the XT1 and XT2 pins as described above.

6.4.4 Ring-OSC oscillator

Ring-OSC oscillator is incorporated in the μPD780143, 780144, 780146, 780148, and 78F0148.

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

6.4.5 Prescaler

The prescaler generates various clocks by dividing the X1 oscillator output (f_X) when the X1 input clock is selected as the clock to be supplied to the CPU.

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

6.5 Clock Generator Operation

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

- X1 input clock f_{XP}
- Ring-OSC clock f_R
- Subsystem clock f_{XT}
- CPU clock f_{CPU}
- Clock to peripheral hardware

The CPU starts operation when the on-chip Ring-OSC oscillator starts outputting after reset release in the 78K0/KF1 Series, thus enabling the following.

(1) Enhancement of security function

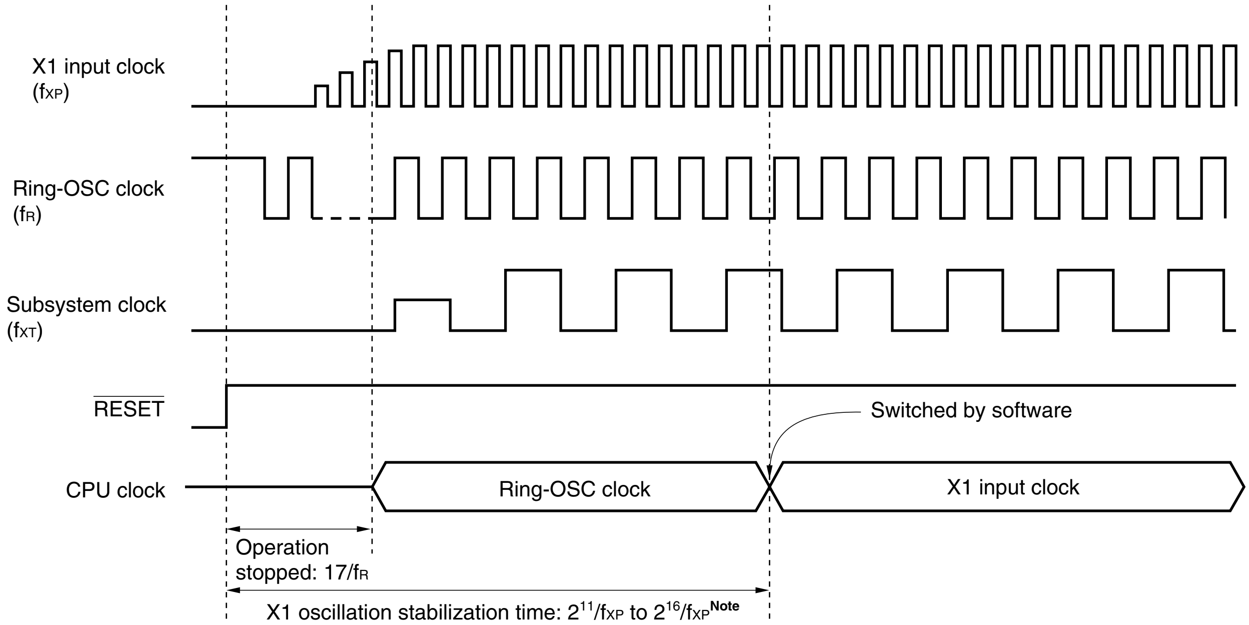
When the X1 input clock is set as the CPU clock by the default setting, the device cannot operate if the X1 input clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the on-chip Ring-OSC clock, so the device can be started by the Ring-OSC clock after reset release by the clock monitor (detection of X1 input 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 X1 input clock oscillation stabilization time, the total performance can be improved.

A timing diagram of the CPU default start using Ring-OSC is shown in Figure 6-12.

Figure 6-12. Timing Diagram of CPU Default Start Using Ring-OSC



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

(a) When the $\overline{\text{RESET}}$ signal is generated, bit 0 of the main clock mode register (MCM) is set to 0 and the Ring-OSC clock is set as the CPU clock. However, a clock is supplied to the CPU after 17 clocks of the Ring-OSC clock have elapsed after $\overline{\text{RESET}}$ release (or clock supply to the CPU stops for 17 clocks). During the $\overline{\text{RESET}}$ period, oscillation of the X1 input clock and Ring-OSC clock is stopped.

(b) After $\overline{\text{RESET}}$ release, the CPU clock can be switched from the Ring-OSC clock to the X1 input clock using bit 0 (MCM0) of the main clock mode register (MCM) after the X1 input clock oscillation stabilization time has elapsed. At this time, check the oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) before switching the CPU clock. The CPU clock status can be checked using bit 1 (MCS) of MCM.

(c) Ring-OSC can be set to stopped/oscillating using the Ring-OSC mode register (RCM) when “Can be stopped by software” is selected for the Ring-OSC by a mask option, if the X1 input or subsystem clock is used as the CPU clock. Make sure that MCS is 1 at this time.

(d) When Ring-OSC is used as the CPU clock, the X1 input clock can be set to stopped/oscillating using the main OSC control register (MOC). Make sure that MCS is 0 at this time.

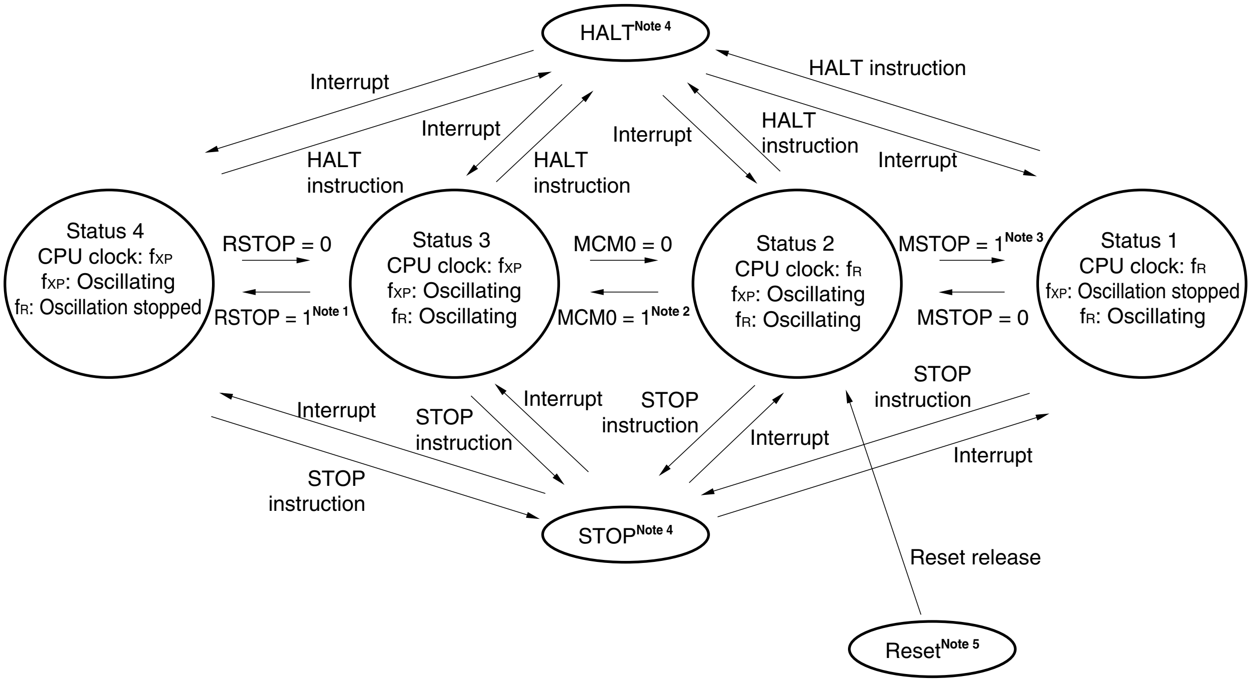
When the subsystem clock is used as the CPU clock, whether the X1 input clock stops or oscillates can be set by the processor clock control register (PCC). In addition, HALT mode can be used during operation with the subsystem clock, but STOP mode cannot be used (subsystem clock oscillation cannot be stopped by the STOP instruction).

(e) Select the X1 input clock oscillation stabilization time ($2^{11}/f_{XP}$, $2^{13}/f_{XP}$, $2^{14}/f_{XP}$, $2^{15}/f_{XP}$, $2^{16}/f_{XP}$) using the oscillation stabilization time select register (OSTS) when releasing STOP mode while X1 input clock is being used as the CPU clock. In addition, when releasing STOP mode while $\overline{\text{RESET}}$ is released and Ring-OSC clock is being used as the CPU clock, check the X1 input clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC).

A status transition diagram of this product is shown in Figure 6-13, 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 6-3 and 6-4, respectively.

Figure 6-13. Status Transition Diagram (1/4)

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



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

2. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

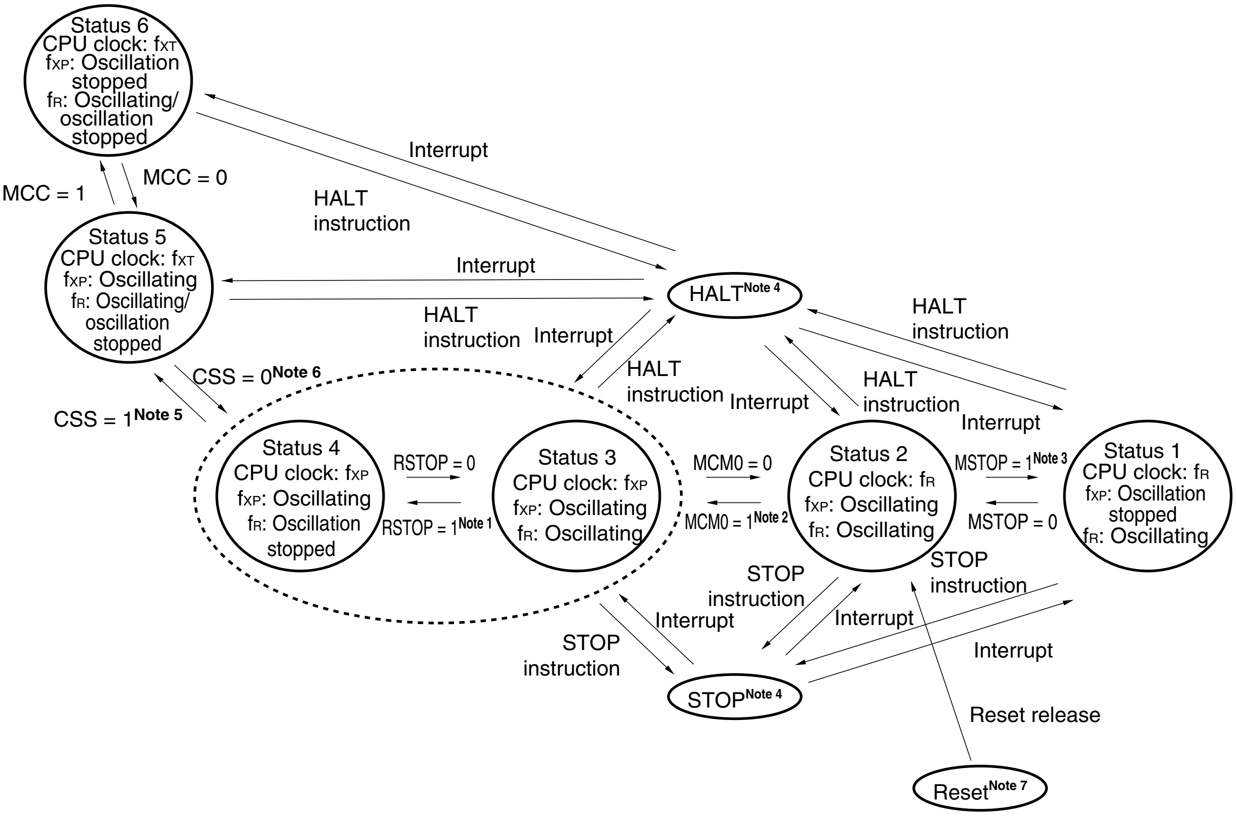
3. When shifting from status 2 to status 1, make sure that MCS is 0.

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

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

Figure 6-13. Status Transition Diagram (2/4)

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



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

2. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

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

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

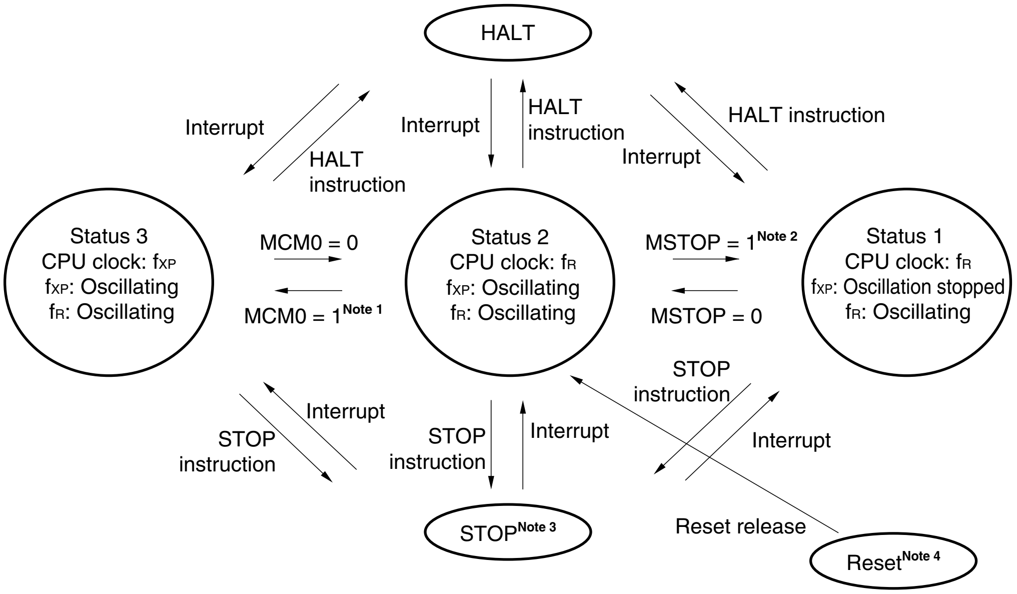
5. Shifting to status 5 (subsystem clock operation) can be performed only from status 3 or 4 (X1 input clock operation).

6. Shifting to status 1 or status 2 from status 5 is not possible.

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

Figure 6-13. Status Transition Diagram (3/4)

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



Notes 1. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

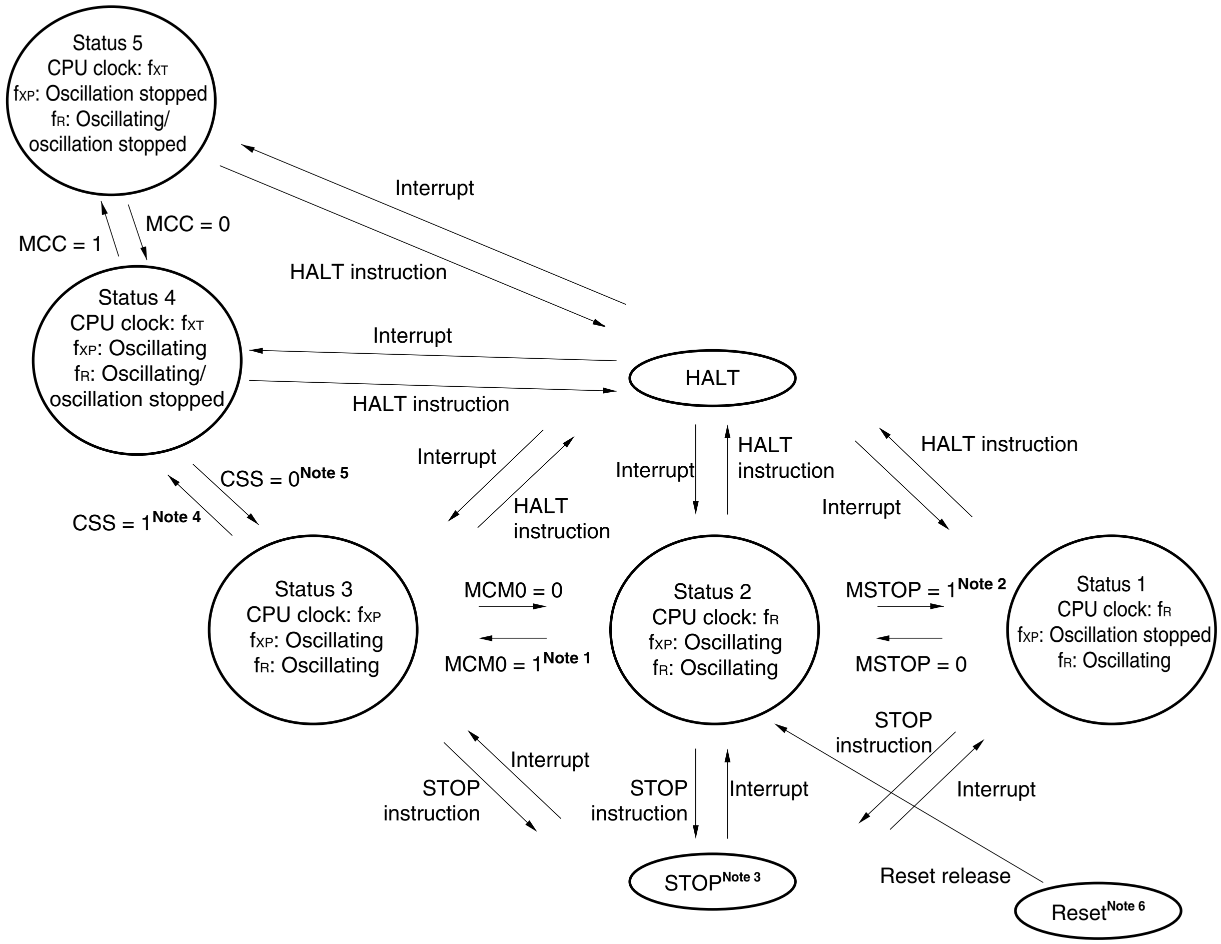
2. When shifting from status 2 to status 1, make sure that MCS is 0.

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

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

Figure 6-13. Status Transition Diagram (4/4)

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



Notes 1. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

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

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

4. Shifting to status 4 (subsystem clock operation) can be performed only from status 3 (X1 input clock operation).

5. Shifting to status 1 or status 2 from status 4 is not possible.

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

Table 6-3. Relationship Between Operation Clocks in Each Operation Status

Status / Operation Mode	X1 Oscillator	Ring-OSC Oscillator			Subsystem Clock 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			Stopped	Ring-OSC	Stopped	
STOP		Oscillating	Oscillating	Stopped	Oscillating	**Note 3**	Stopped	
HALT	Oscillating					**Note 4**	Ring-OSC	X1

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

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

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

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

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

Remark RSTOP: Bit 0 of the Ring-OSC mode register (RCM)

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

★

Table 6-4. Oscillation Control Flags and Clock Oscillation Status

		X1 Oscillator	Ring-OSC Oscillator
MSTOP = 1[Note]	RSTOP = 0	Stopped	Oscillating
	RSTOP = 1	Setting prohibited	
MSTOP = 0[Note]	RSTOP = 0	Oscillating	Oscillating
	RSTOP = 1		Stopped
MCC = 1[Note]	RSTOP = 0	Stopped	Oscillating
	RSTOP = 1		Stopped
MCC = 0[Note]	RSTOP = 0	Oscillating	Oscillating
	RSTOP = 1		Stopped

Note Setting X1 oscillator oscillating/stopped differs depending on the CPU clock used.

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

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

Remark MSTOP: Bit 7 of the main OSC control register (MOC)

MCC: Bit 7 of the processor clock control register (PCC)

RSTOP: Bit 0 of the Ring-OSC mode register (RCM)

6.6 Time Required to Switch Between Ring-OSC Clock and X1 Input Clock

Bit 0 (MCM0) of the main clock mode register (MCM) is used to switch between the Ring-OSC clock and X1 input 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 6-5**).

Bit 1 (MCS) of MCM is used to judge that operation is performed using either the Ring-OSC clock or X1 input clock.

To stop the clock, wait for the number of clocks shown in Table 6-5 before stopping.

Table 6-5. Time Required to Switch Between Ring-OSC Clock and X1 Input Clock

PCC			Time Required for Switching	
PCC2	PCC1	PCC0	X1→Ring-OSC	Ring-OSC→X1
0	0	0	f_{XP}/f_R + 1 clock	2 clocks
0	0	1	$f_{XP}/2f_R$ + 1 clock	
0	1	0	$f_{XP}/4f_R$ + 1 clock	
0	1	1	$f_{XP}/8f_R$ + 1 clock	
1	0	0	$f_{XP}/16f_R$ + 1 clock	

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

Remarks 1. PCC: Processor clock control register

2. f_{XP}: X1 input clock oscillation frequency

3. f_R: Ring-OSC clock oscillation frequency

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

6.7 Changing System Clock and CPU Clock Settings

6.7.1 Time required for switching between system clock and CPU clock

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

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

Whether the system is operating on the X1 input clock (or Ring-OSC clock) or the subsystem clock can be ascertained using bit 5 (CLS) of the PCC register.

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

<table>
<tr><th colspan="4">Set Value Before Switchover</th><th colspan="24">Set Value After Switchover</th></tr>
<tr><th rowspan="2">CSS</th><th rowspan="2">PCC2</th><th rowspan="2">PCC1</th><th rowspan="2">PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td rowspan="5">0</td><td>0</td><td>0</td><td>0</td><td colspan="4"></td><td colspan="4">16 clocks</td><td colspan="4">16 clocks</td><td colspan="4">16 clocks</td><td colspan="4">16 clocks</td><td colspan="4">f_{XP}/f_{XT} clocks
(306 clocks)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="4">8 clocks</td><td colspan="4"></td><td colspan="4">8 clocks</td><td colspan="4">8 clocks</td><td colspan="4">8 clocks</td><td colspan="4">f_{XP}/2f_{XT} clocks
(153 clocks)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td colspan="4">4 clocks</td><td colspan="4">4 clocks</td><td colspan="4"></td><td colspan="4">4 clocks</td><td colspan="4">4 clocks</td><td colspan="4">f_{XP}/4f_{XT} clocks
(77 clocks)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="4">2 clocks</td><td colspan="4">2 clocks</td><td colspan="4">2 clocks</td><td colspan="4"></td><td colspan="4">2 clocks</td><td colspan="4">f_{XP}/8f_{XT} clocks
(39 clocks)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="4">1 clock</td><td colspan="4">1 clock</td><td colspan="4">1 clock</td><td colspan="4">1 clock</td><td colspan="4"></td><td colspan="4">f_{XP}/16f_{XT} clocks
(20 clocks)</td></tr>
<tr><td>1</td><td>×</td><td>×</td><td>×</td><td colspan="4">1 clock</td><td colspan="4">1 clock</td><td colspan="4">1 clock</td><td colspan="4">1 clock</td><td colspan="4">1 clock</td><td colspan="4"></td></tr>
</table>

Remarks 1. The maximum time is the number of clocks of the CPU clock before switching.

2. Figures in parentheses apply to operation with f_{XP} = 10 MHz and f_{XT} = 32.768 kHz.

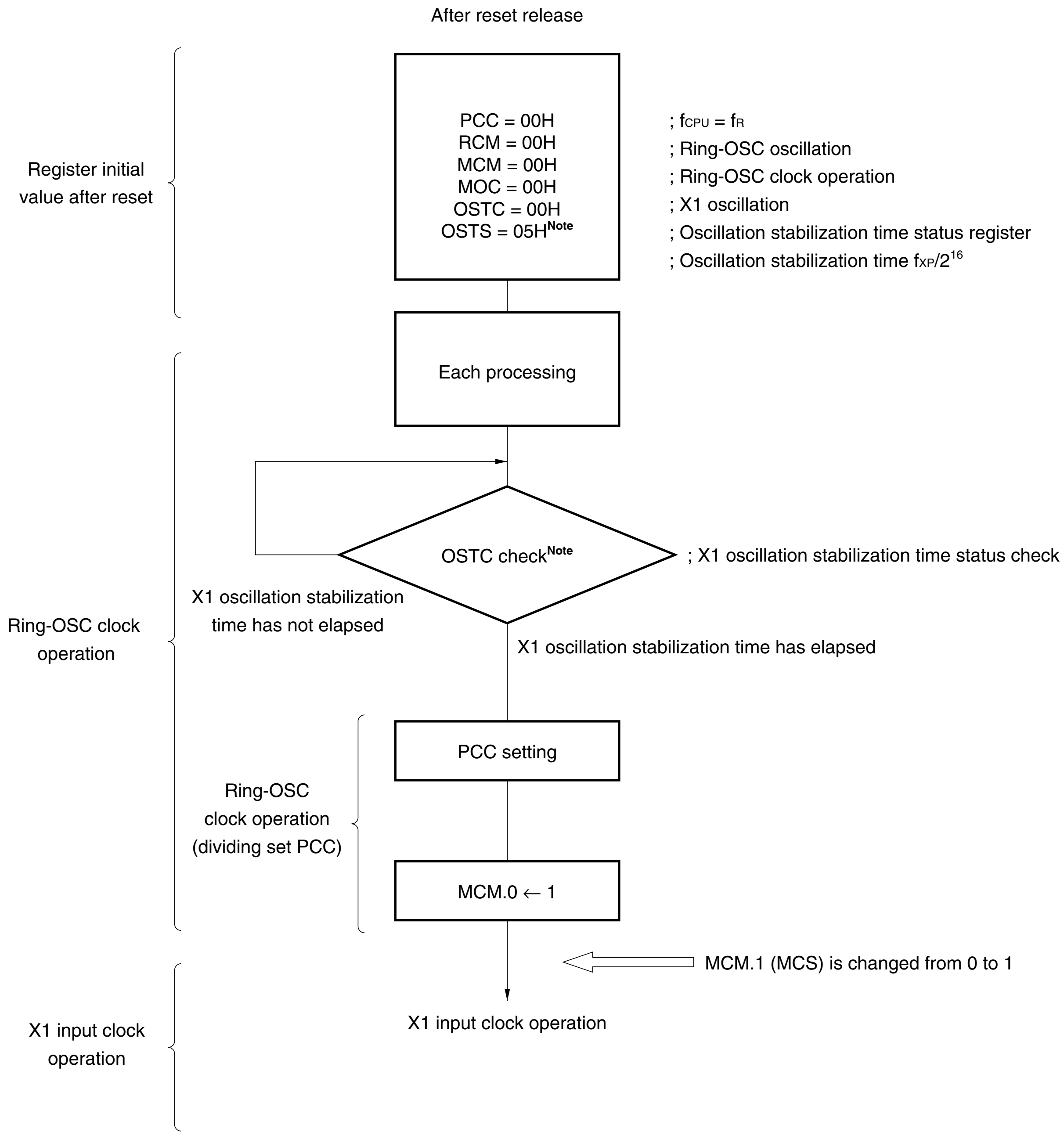
Caution Selection of the CPU clock cycle division factor (PCC0 to PCC2) and switchover from the X1 input clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously.

Simultaneous setting is possible, however, for selection of the CPU clock cycle division factor (PCC0 to PCC2) and switchover from the subsystem clock to the X1 input clock (changing CSS from 1 to 0).

6.8 Clock Switching Flowchart and Register Setting

6.8.1 Switching from Ring-OSC clock to X1 input clock

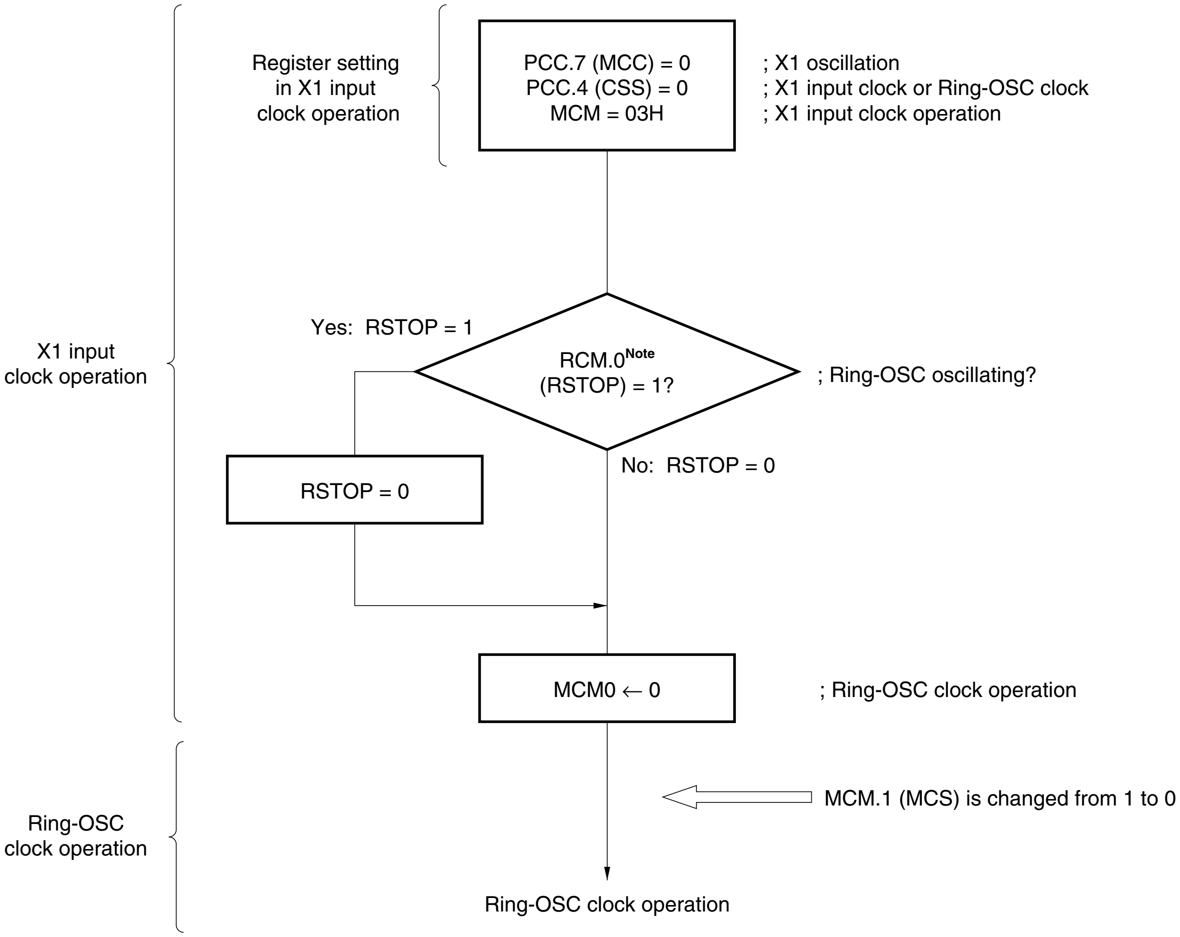
Figure 6-14. Switching from Ring-OSC Clock to X1 Input Clock


After reset release
Register initial value after reset
PCC = 00H
RCM = 00H
MCM = 00H
MOC = 00H
OSTC = 00H
OSTS = 05H Note
; $f_{CPU} = f_R$
; Ring-OSC oscillation
; Ring-OSC clock operation
; X1 oscillation
; Oscillation stabilization time status register
; Oscillation stabilization time $f_{XP}/2^{16}$
Each processing
OSTC check Note
; X1 oscillation stabilization time status check
X1 oscillation stabilization time has not elapsed
X1 oscillation stabilization time has elapsed
Ring-OSC clock operation
PCC setting
Ring-OSC clock operation (dividing set PCC)
MCM.0 ← 1
MCM.1 (MCS) is changed from 0 to 1
X1 input clock operation
X1 input clock operation

Note Check the oscillation stabilization wait time of the X1 oscillator after reset release using the OSTC register and then switch to the X1 input clock operation after the oscillation stabilization wait time has elapsed. The OSTS register setting is valid only after STOP mode is released during X1 input clock operation.

6.8.2 Switching from X1 input clock to Ring-OSC clock

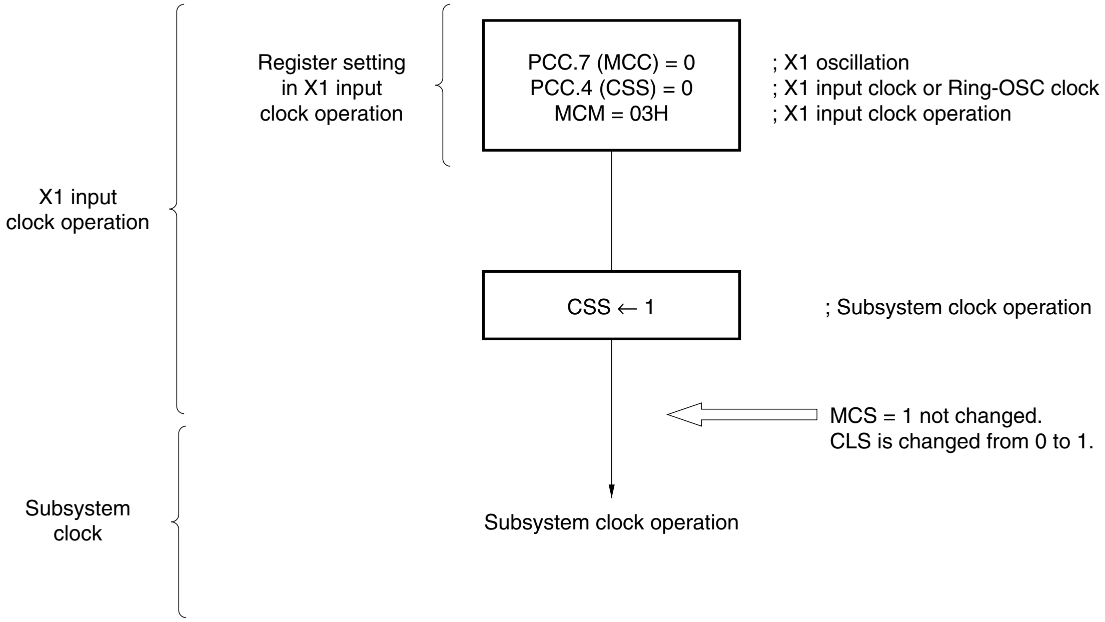
Figure 6-15. Switching from X1 Input Clock to Ring-OSC Clock (Flowchart)



Note Required only when “clock can be stopped by software” is selected for Ring-OSC by a mask option.

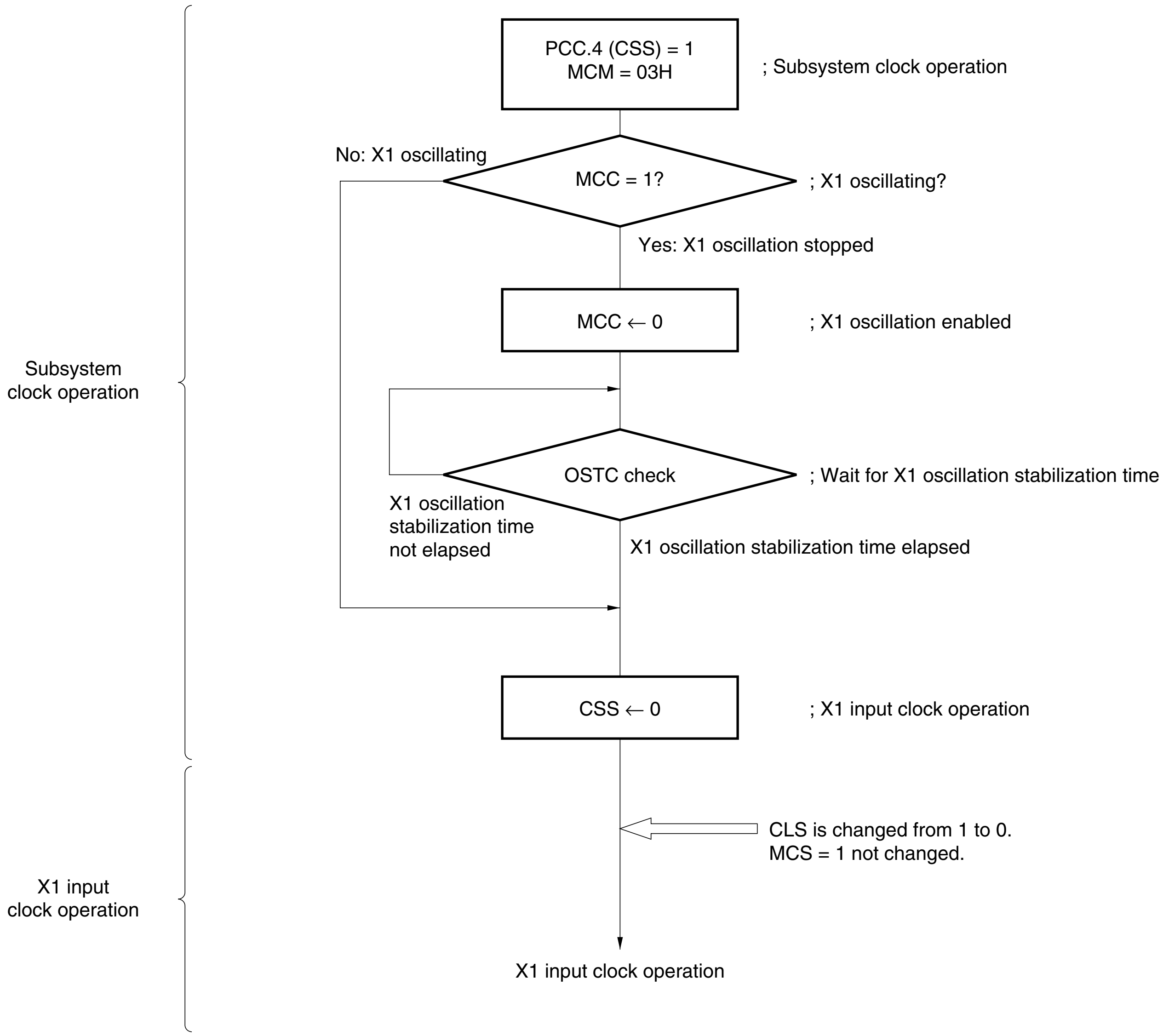
6.8.3 Switching from X1 input clock to subsystem clock

Figure 6-16. Switching from X1 Input Clock to Subsystem Clock (Flowchart)


X1 input clock operation
Register setting in X1 input clock operation
PCC.7 (MCC) = 0
PCC.4 (CSS) = 0
MCM = 03H
; X1 oscillation
; X1 input clock or Ring-OSC clock
; X1 input clock operation
CSS ← 1
; Subsystem clock operation
MCS = 1 not changed.
CLS is changed from 0 to 1.
Subsystem clock
Subsystem clock operation

6.8.4 Switching from subsystem clock to X1 input clock

Figure 6-17. Switching from Subsystem Clock to X1 Input Clock (Flowchart)



6.8.5 Register settings

Table 6-7. Clock and Register Setting

f_{CPU}	Mode	Setting Flag					Status Flag	
		PCC Register		MCM Register	MOC Register	RCM Register	PCC Register	MCM Register
		MCC	CSS	MCM0	MSTOP	RSTOP[Note 1]	CLS	MCS
X1 input clock[Note 2]	Ring-OSC oscillating	0	0	1	0	0	0	1
	Ring-OSC stopped	0	0	1	0	1	0	1
Ring-OSC clock	X1 oscillating	0	0	0	0	0	0	0
	X1 stopped	0[Note 3]	0	0	1	0	0	0
Subsystem clock[Note 4]	X1 oscillating, Ring-OSC oscillating	0	1	1[Note 5]	0[Note 6]	0	1	1
	X1 stopped, Ring-OSC oscillating	1	1	1[Note 5]	0[Note 6]	0	1	1
	X1 oscillating, Ring-OSC stopped	0	1	1[Note 5]	0[Note 6]	1	1	1
	X1 stopped, Ring-OSC stopped	1	1	1[Note 5]	0[Note 6]	1	1	1

Notes 1. Valid only when “clock can be stopped by software” is selected for Ring-OSC by a mask option.

2. Do not set MCC = 1 or MSTOP = 1 during X1 input clock operation (even if MCC = 1 or MSTOP = 1 is set, the X1 oscillation does not stop).

3. Do not set MCC = 1 during Ring-OSC operation (even if MCC = 1 is set, the X1 oscillation does not stop). To stop X1 oscillation during Ring-OSC operation, use MSTOP.

4. Shifting to subsystem clock operation mode must be performed from the X1 input clock operation mode. From subsystem clock operation mode, only X1 input clock operation mode can be shifted to.

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

6. Do not set MSTOP = 1 during subsystem clock operation (even if MSTOP = 1 is set, X1 oscillation does not stop). To stop X1 oscillation during subsystem clock operation, use MCC.

CHAPTER 7 16-BIT TIMER/EVENT COUNTERS 00 AND 01

The μPD780143 and 780144 incorporate 16-bit timer/event counter 00, and the μPD780146, 780148, and 78F0148 incorporate 16-bit timer/event counters 00 and 01.

7.1 Functions of 16-Bit Timer/Event Counters 00 and 01

16-bit timer/event counters 00 and 01[Note] have 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 counters 00 and 01 generate an interrupt request at the preset time interval.

(2) PPG output

16-bit timer/event counters 00 and 01 can output a rectangular wave whose frequency and output pulse width can be set freely.

(3) Pulse width measurement

16-bit timer/event counters 00 and 01 can measure the pulse width of an externally input signal.

(4) External event counter

16-bit timer/event counters 00 and 01 can measure the number of pulses of an externally input signal.

(5) Square-wave output

16-bit timer/event counters 00 and 01 can output a square wave with any selected frequency.

(6) One-shot pulse output

16-bit timer/event counters 00 and 01 can output a one-shot pulse whose output pulse width can be set freely.

Note Available only for the μPD780146, 780148, and 78F0148.

7.2 Configuration of 16-Bit Timer/Event Counters 00 and 01

16-bit timer/event counters 00 and 01 consist of the following hardware.

Table 7-1. Configuration of 16-Bit Timer/Event Counters 00 and 01

Item	Configuration
Timer counter	16 bits × 1 (TM0n)
Register	16-bit timer capture/compare register: 16 bits × 2 (CR00n, CR01n)
Timer output	1 (TO0n)
Control registers	16-bit timer mode control register 0n (TMC0n) 16-bit timer capture/compare control register 0n (CRC0n) 16-bit timer output control register 0n (TOC0n) Prescaler mode register 0n (PRM0n) Port mode register 0 (PM0)[Note]

Note See **Figure 4-2 Block Diagram of P00, P03, and P05** and **Figure 4-3 Block Diagram of P01 and P06**.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figures 7-1 and 7-2 show the block diagrams.

Figure 7-1. Block Diagram of 16-Bit Timer/Event Counter 00

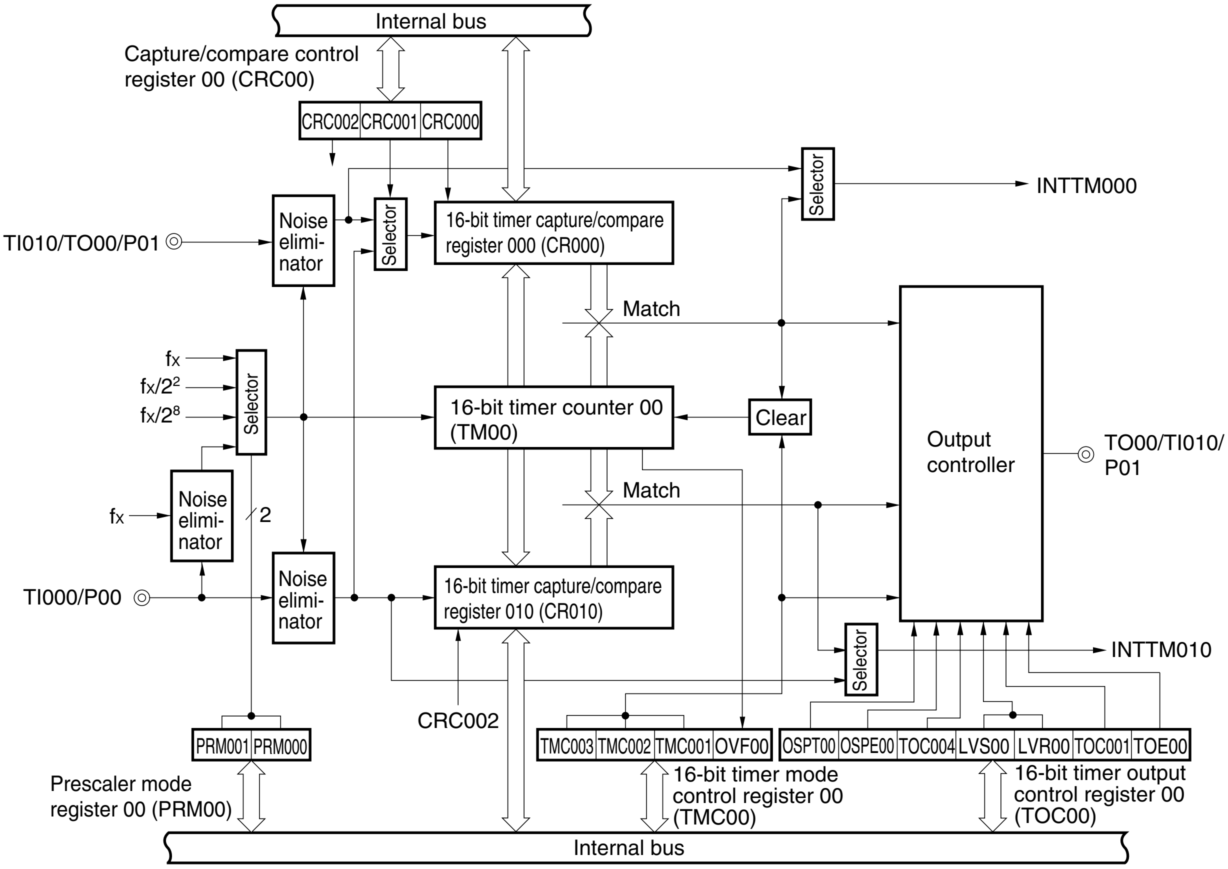
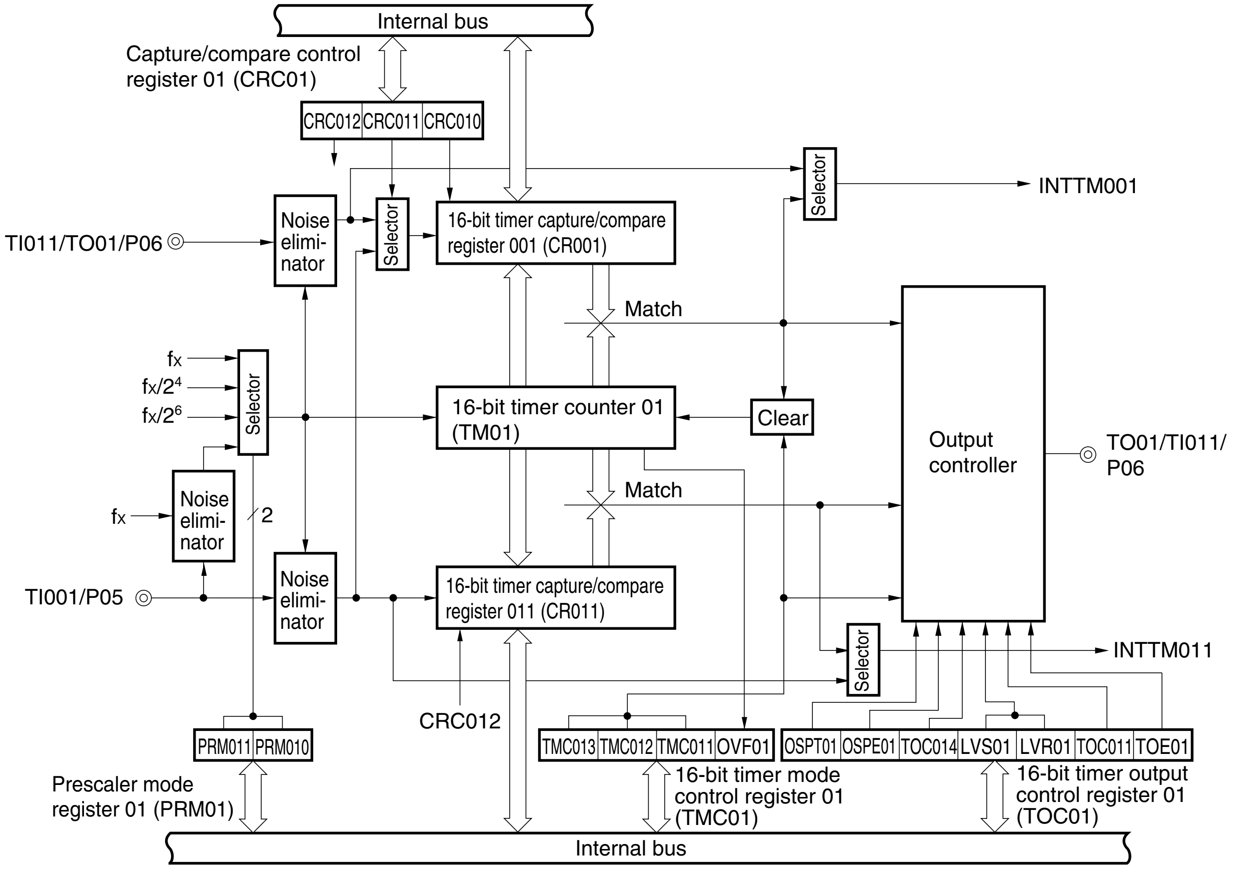


Figure 7-2. Block Diagram of 16-Bit Timer/Event Counter 01



(1) 16-bit timer counter 0n (TM0n)

TM0n 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. The count value is reset to 0000H in the following cases.

<1> At $\overline{\text{RESET}}$ input
<2> If TMC0n3 and TMC0n2 are cleared
<3> If the valid edge of TI00n is input in the mode in which clear & start occurs when inputting the valid edge of TI00n
<4> If TM0n and CR00n match in the mode in which clear & start occurs on a match of TM0n and CR00n
<5> OSPT0n is set in one-shot pulse output mode

(2) 16-bit timer capture/compare register 00n (CR00n)

CR00n 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 (CRC0n0) of capture/compare control register 0n (CRC0n).

- **When CR00n is used as a compare register**

  The value set in CR00n is constantly compared with the 16-bit timer counter 0n (TM0n) count value, and an interrupt request (INTTM00n) is generated if they match. It can also be used as the register that holds the interval time when TM0n is set to interval timer operation.

- **When CR00n is used as a capture register**

  It is possible to select the valid edge of the TI00n pin or the TI01n pin as the capture trigger. The TI00n or TI01n valid edge is set using prescaler mode register 0n (PRM0n).

  If the capture trigger is specified to be the valid edge of the TI00n pin, the situation is as shown in Table 7-2. On the other hand, when the capture trigger is specified to be the valid edge of the TI01n pin, the situation is as shown in Table 7-3.

Table 7-2. TI00n Pin Valid Edge and CR00n, CR01n Capture Trigger

ES0n1	ES0n0	TI00n Pin Valid Edge	CR00n Capture Trigger	CR01n Capture Trigger
0	0	Falling edge	Rising edge	Falling edge
0	1	Rising edge	Falling edge	Rising edge
1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation	Both rising and falling edges

Table 7-3. TI01n Pin Valid Edge and CR00n Capture Trigger

ES1n1	ES1n0	TI01n Pin Valid Edge	CR00n Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

CR00n can be set by a 16-bit memory manipulation instruction.

★ RESET input clears this register to 0000H.

Cautions 1. Set a value other than 0000H in CR00n in the mode in which clear & start occurs on a match of TM0n and CR00n. However, in the free-running mode and in the clear mode using the valid edge of TI00n, if CR00n is set to 0000H, an interrupt request (INTTM00n) is generated following overflow (FFFFH).

2. If the changed value of CR00n is smaller than the value of 16-bit timer counter 0n (TM0n), TM0n continues counting and starts counting again from 0 after overflow. Therefore, if the value of CR00n after the change is smaller than before the change, the timer should be restarted after CR00n is changed.

3. When P01 or P06 is used as the valid edge of TI01n, it cannot be used as the timer output (TO0n). Moreover, when P01 or P06 is used as TO0n, it cannot be used as the valid edge of TI01n.

4. When CR00n is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

5. Do not rewrite CR00n during TM0n operation.

(3) 16-bit timer capture/compare register 01n (CR01n)

CR01n 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 (CRC0n2) of capture/compare control register 0n (CRC0n).

- **When CR01n is used as a compare register**

  The value set in the CR01n is constantly compared with the 16-bit timer counter 0n (TM0n) count value, and an interrupt request (INTTM01n) is generated if they match.

- **When CR01n is used as a capture register**

  It is possible to select the valid edge of the TI00n pin as the capture trigger. The TI00n valid edge is set by prescaler mode register 0n (PRM0n).

CR01n can be set by a 16-bit memory manipulation instruction.

★ RESET input clears this register to 0000H.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Cautions 1. Set CR01n to other than 0000H. This means a 1-pulse count operation cannot be performed when CR01n is used as the event counter.
However, in the free-running mode and in the clear mode using the valid edge of TI00n, if CR01n is set to 0000H, an interrupt request (INTTM01n) is generated following overflow (FFFFH).

2. When CR01n is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

★ **3. CR01n can be rewritten during TM0n operation. For details, refer to Remark 2 in Figure 7-17.**

7.3 Registers Controlling 16-Bit Timer/Event Counters 00 and 01

The following five registers are used to control 16-bit timer/event counters 00 and 01.

- 16-bit timer mode control register 0n (TMC0n)
- Capture/compare control register 0n (CRC0n)
- 16-bit timer output control register 0n (TOC0n)
- Prescaler mode register 0n (PRM0n)
- Port mode register 0 (PM0)

(1) 16-bit timer mode control register 0n (TMC0n)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 0n (TM0n) clear mode, and output timing, and detects an overflow.

TMC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC0n to 00H.

Caution 16-bit timer counter 0n (TM0n) starts operation at the moment TMC0n2 and TMC0n3 are set to values other than 0, 0 (operation stop mode), respectively. Set TMC0n2 and TMC0n3 to 0, 0 to stop the operation.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-3. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address FFBAH After reset: 00H R/W

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

TMC003	TMC002	TMC001	Operating mode and clear mode selection	TO00 output timing selection	Interrupt request generation
0	0	0	Operation stop (TM00 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM00 and CR000 or match between TM00 and CR010	Generated on match between TM00 and CR000, or match between TM00 and CR010
0	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 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/P00 pin using prescaler mode register 00 (PRM00).

3. If any 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 TI00 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.

Remarks 1. TO00: 16-bit timer/event counter 00 output pin

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

3. TM00: 16-bit timer counter 00

4. CR000: 16-bit timer capture/compare register 000

5. CR010: 16-bit timer capture/compare register 010

Figure 7-4. Format of 16-Bit Timer Mode Control Register 01 (TMC01)

Address FFB6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	$\boxed{0}$
TMC01	0	0	0	0	TMC013	TMC012	TMC011	OVF01

TMC013	TMC012	TMC011	Operating mode and clear mode selection	TO01 output timing selection	Interrupt request generation
0	0	0	Operation stop (TM01 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM01 and CR001 or match between TM01 and CR011	Generated on match between TM01 and CR001, or match between TM01 and CR011
0	1	1		Match between TM01 and CR001, match between TM01 and CR011 or TI001 valid edge	
1	0	0	Clear & start occurs on TI001 valid edge	–	
1	0	1			
1	1	0	Clear & start occurs on match between TM01 and CR001	Match between TM01 and CR001 or match between TM01 and CR011	
1	1	1		Match between TM01 and CR001, match between TM01 and CR011 or TI001 valid edge	

OVF01	16-bit timer counter 01 (TM01) overflow detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Timer operation must be stopped before writing to bits other than the OVF01 flag.

2. Set the valid edge of the TI001/P05 pin using prescaler mode register 01 (PRM01).

3. If any the following modes: the mode in which clear & start occurs on match between TM01 and CR001, the mode in which clear & start occurs at the TI01 valid edge, or free-running mode is selected, when the set value of CR001 is FFFFH and the TM01 value changes from FFFFH to 0000H, the OVF01 flag is set to 1.

Remarks 1. TO01: 16-bit timer/event counter 01 output pin

2. TI001: 16-bit timer/event counter 01 input pin

3. TM01: 16-bit timer counter 01

4. CR001: 16-bit timer capture/compare register 001

5. CR011: 16-bit timer capture/compare register 011

(2) Capture/compare control register 0n (CRC0n)

This register controls the operation of the 16-bit timer capture/compare registers (CR00n, CR01n).

CRC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CRC0n to 00H.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-5. 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

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

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

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

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

Figure 7-6. Format of Capture/Compare Control Register 01 (CRC01)

Address: FFB8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC01	0	0	0	0	0	CRC012	CRC011	CRC010

CRC012	CR011 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC011	CR001 capture trigger selection
0	Captures on valid edge of TI011
1	Captures on valid edge of TI001 by reverse phase

CRC010	CR001 operating mode selection
0	Operates as compare register
1	Operates as capture register

Cautions 1. Timer operation must be stopped before setting CRC01.

2. When the mode in which clear & start occurs on a match between TM01 and CR001 is selected with 16-bit timer mode control register 01 (TMC01), CR001 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 times longer than the count clock selected by prescaler mode register 01 (PRM01).

(3) 16-bit timer output control register 0n (TOC0n)

This register controls the operation of the 16-bit timer/event counter 0n output controller. It sets/resets the R-S type flip-flop (LV0n), enables/disables output inversion and 16-bit timer/event counter 0n timer output, enables/disables the one-shot pulse output operation, and sets the one-shot pulse output trigger via software.

TOC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TOC0n to 00H.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

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

Address: FFBDH After reset: 00H R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
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	16-bit timer/event counter 00 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	16-bit timer/event counter 00 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. **If LVS00 and LVR00 are read after data is set, 0 is read.**
3. **OSPT00 is automatically cleared after data is set, so 0 is read.**
4. **Do not set OSPT00 to 1 other than in one-shot pulse output mode.**
5. **A write interval of two cycles or more of the operating clock is required to write to OSPT00 successively.**

Figure 7-8. Format of 16-Bit Timer Output Control Register 01 (TOC01)

Address: FFB9H After reset: 00H R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TOC01	0	OSPT01	OSPE01	TOC014	LVS01	LVR01	TOC011	TOE01

OSPT01	One-shot pulse output trigger control via software
0	No one-shot pulse trigger
1	One-shot pulse trigger

OSPE01	One-shot pulse output operation control
0	Successive pulse output mode
1	One-shot pulse output mode[Note]

TOC014	Timer output F/F control using match of CR011 and TM01
0	Disables inversion operation
1	Enables inversion operation

LVS01	LVR01	16-bit timer/event counter 01 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

TOC011	Timer output F/F control using match of CR001 and TM01
0	Disables inversion operation
1	Enables inversion operation

TOE01	16-bit timer/event counter 01 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 TI001 valid edge. In the mode in which clear & start occurs on a match between the TM01 register and CR001 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 TOC014.**

2. If LVS01 and LVR01 are read after data is set, 0 is read.

3. OSPT01 is automatically cleared after data is set, so 0 is read.

4. Do not set OSPT01 to 1 other than in one-shot pulse output mode.

5. A write interval of two cycles or more of the operating clock is required to write to OSPT01 successively.

(4) Prescaler mode register 0n (PRM0n)

This register is used to set the 16-bit timer counter 0n (TM0n) count clock and TI00n and TI01n input valid edges.

★ PRM0n can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM0n to 00H.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-9. 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	fx (10 MHz)
0	1	fx/2^2 (2.5 MHz)
1	0	fx/2^8 (39.06 kHz)
1	1	TI000 valid edge[Note]

Note The external clock requires a pulse two times longer than internal clock (fx).

Cautions 1. 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.

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

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

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

Remarks 1. fx: X1 input 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.

Figure 7-10. Format of Prescaler Mode Register 01 (PRM01)

Address: FFB7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM01	ES111	ES110	ES011	ES010	0	0	PRM011	PRM010

ES111	ES110	TI011 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES011	ES010	TI001 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM011	PRM010	Count clock selection
0	0	fx (10 MHz)
0	1	fx/2^4 (625 kHz)
1	0	fx/2^6 (156.25 kHz)
1	1	TI001 valid edge[Note]

Note The external clock requires a pulse two times longer than internal clock (fx).

Cautions 1. If the valid edge of TI001 is to be set for the count clock, do not set the clear & start mode using the valid edge of TI001 and the capture trigger.

2. Always set data to PRM01 after stopping the timer operation.

3. If the TI001 or TI011 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 TI001 pin or TI011 pin to enable the operation of 16-bit timer counter 01 (TM01). Care is therefore required when pulling up the TI001 or TI011 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

4. When P06 is used as the TI011 valid edge, it cannot be used as the timer output (TO01), and when used as TO01, it cannot be used as the TI011 valid edge.

Remarks 1. fx: X1 input clock oscillation frequency

2. TI001, TI011: 16-bit timer/event counter 01 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 and P06/TO01[Note]/TI011[Note] pins for timer output, set PM01 and PM06 and the output latch of P01 and P06 to 0.

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

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

Figure 7-11. 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	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Available only for the μPD780146, 780148, and 78F0148.

7.4 Operation of 16-Bit Timer/Event Counters 00 and 01

7.4.1 Interval timer operation

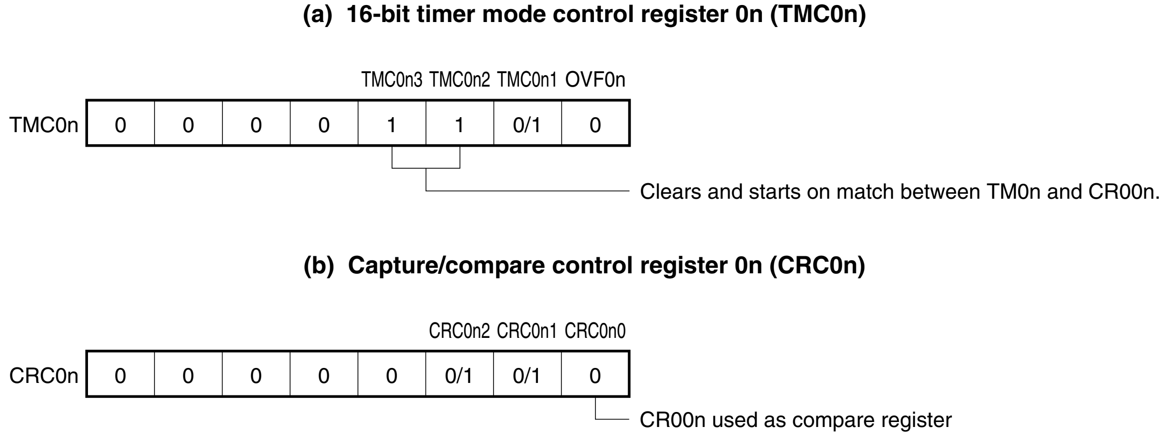
Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 7-12 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value preset in 16-bit timer capture/compare register 00n (CR00n) as the interval.

When the count value of 16-bit timer counter 0n (TM0n) matches the value set in CR00n, counting continues with the TM0n value cleared to 0 and the interrupt request signal (INTTM00n) is generated.

The count clock of the 16-bit timer/event counter 0n can be selected with bits 0 and 1 (PRM0n0, PRM0n1) of prescaler mode register 0n (PRM0n).

See **7.5 Cautions for 16-Bit Timer/Event Counters 00 and 01 (2) 16-bit timer capture/compare register setting** for details of the operation when the compare register value is changed during timer count operation.

Figure 7-12. Control Register Settings for Interval Timer Operation



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	1	0/1	0

Clears and starts on match between TM0n and CR00n.

(b) Capture/compare control register 0n (CRC0n)

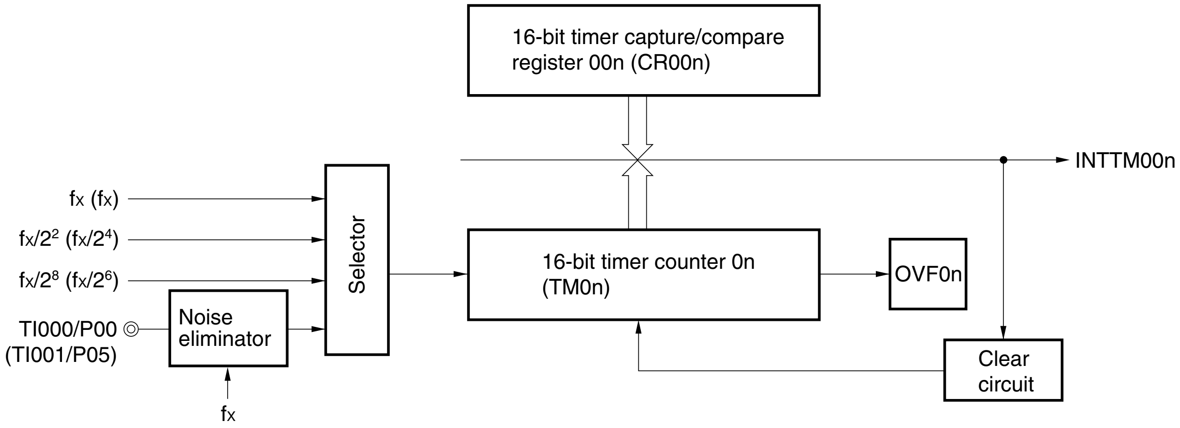
						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0/1	0/1	0

CR00n used as compare register

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. For details, see **Figures 7-3** to **7-6**.

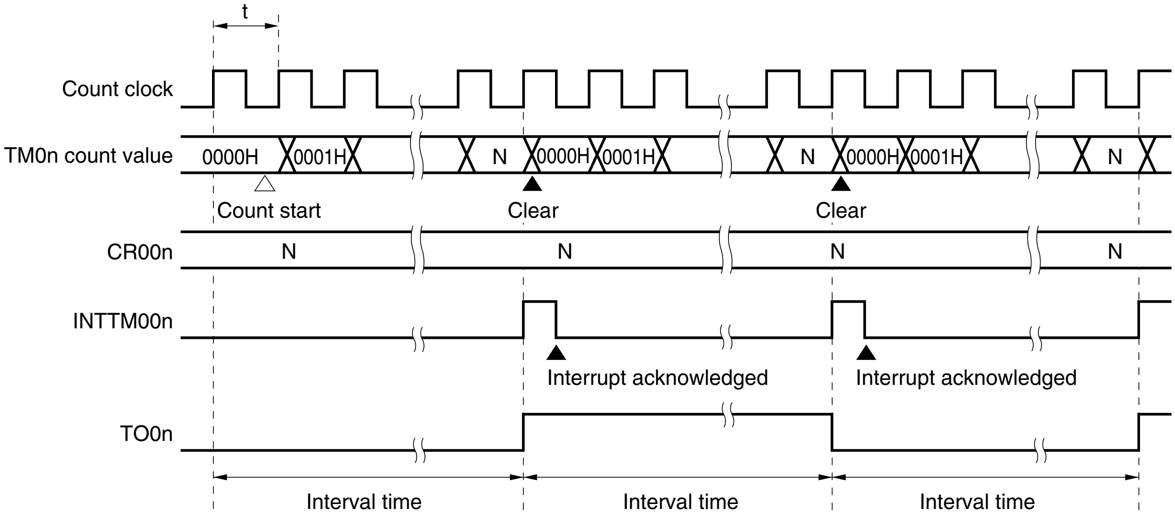
2. n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-13. Interval Timer Configuration Diagram



Remark Frequencies and pin names without parentheses are for 16-bit timer/event counter 00, and those in parentheses are for 16-bit timer/event counter 01.

Figure 7-14. Timing of Interval Timer Operation



Remark Interval time = (N + 1) × t

N = 0001H to FFFFH

n = 0: μPD780143, 780144

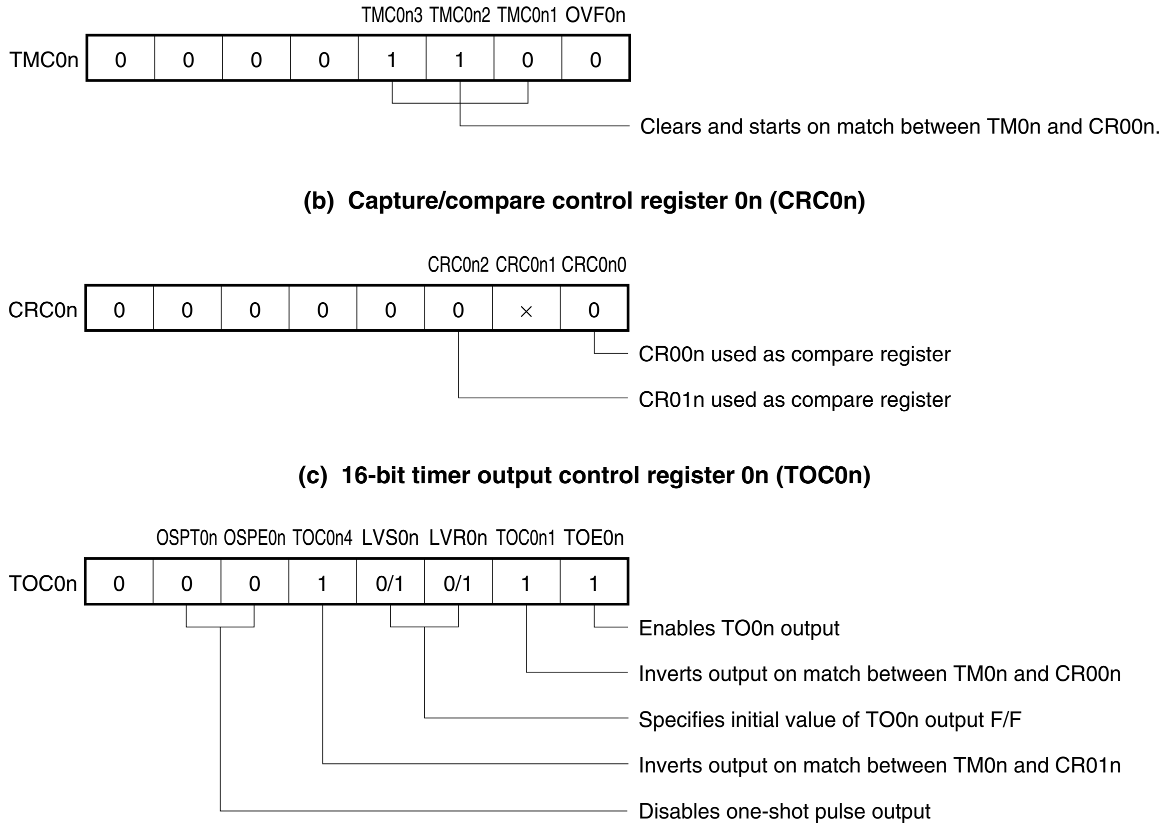
n = 0, 1: μPD780146, 780148, 78F0148

7.4.2 PPG output operations

Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 7-15 allows operation as PPG (Programmable Pulse Generator) output.

In the PPG output operation, rectangular waves are output from the TO0n pin with the pulse width and the cycle that correspond to the count values preset in 16-bit timer capture/compare register 01n (CR01n) and in 16-bit timer capture/compare register 00n (CR00n), respectively.

Figure 7-15. Control Register Settings for PPG Output Operation


(a) 16-bit timer mode control register 0n (TMC0n)

TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 1 1 0 0
Clears and starts on match between TM0n and CR00n.

(b) Capture/compare control register 0n (CRC0n)

CRC0n2 CRC0n1 CRC0n0
CRC0n
0 0 0 0 0 0 × 0
CR00n used as compare register
CR01n used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

OSPT0n OSPE0n TOC0n4 LVS0n LVR0n TOC0n1 TOE0n
TOC0n
0 0 0 1 0/1 0/1 1 1
Enables TO0n output
Inverts output on match between TM0n and CR00n
Specifies initial value of TO0n output F/F
Inverts output on match between TM0n and CR01n
Disables one-shot pulse output

Cautions 1. Values in the following range should be set in CR00n and CR01n:
0000H < CR01n < CR00n ≤ FFFFH

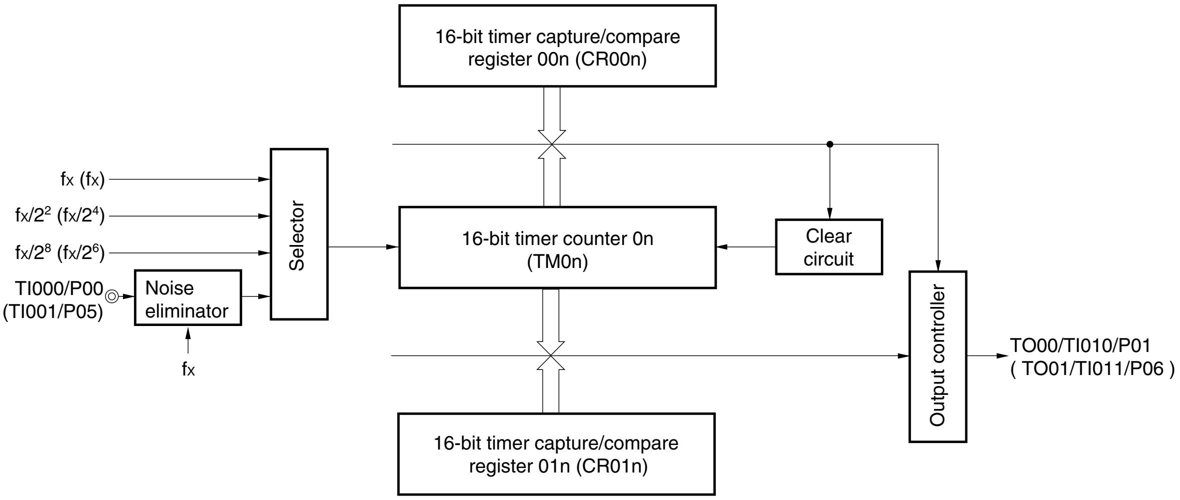
2. The cycle of the pulse generated through PPG output (CR00n setting value + 1) has a duty of (CR01n setting value + 1)/(CR00n setting value + 1).

Remark ×: Don’t care

n = 0: μPD780143, 780144

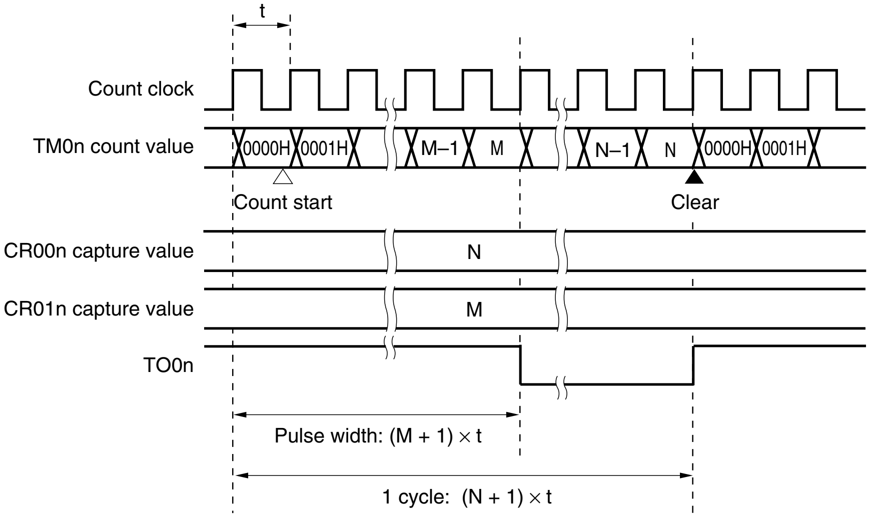
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-16. Configuration of PPG Output



Remark Frequencies and pin names without parentheses are for 16-bit timer/event counter 00, and those in parentheses are for 16-bit timer/event counter 01.

Figure 7-17. PPG Output Operation Timing



★ **Caution CR00n cannot be rewritten during TM0n operation.**

Remarks 1. 0000H < M < N ≤ FFFFH

2. In the PPG output operation, change the pulse width (rewrite CR01n) during TM0n operation using the following procedure.

<1> Disable the timer output inversion operation by match of TM0n and CR01n (TOC0n4 = 0)
<2> Disable the INTTM01n interrupt (TMMK01n = 1)
<3> Rewrite CR01n
<4> Wait for 1 cycle of the TM0n count clock
<5> Enable the timer output inversion operation by match of TM0n and CR01n (TOC0n4 = 1)
<6> Clear the interrupt request flag of INTTM01n (TMIF01n = 0)
<7> Enable the INTTM01n interrupt (TMMK01n = 0)

3. n = 0: μPD780143, 780144, n = 0, 1: μPD780146, 780148, 78F0148

7.4.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00n pin and TI01n pin using 16-bit timer counter 0n (TM0n).

There are two measurement methods: measuring with TM0n used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00n pin.

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

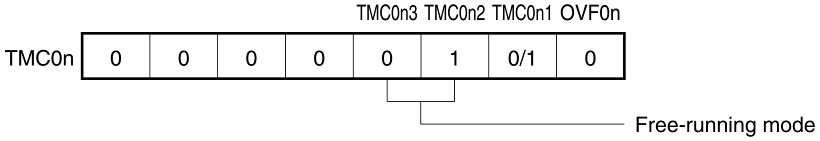
When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 7-18**), and the edge specified by prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

Any of three edges—rising, falling, or both edges—can be selected using bits 4 and 5 (ES0n0 and ES0n1) of PRM0n.

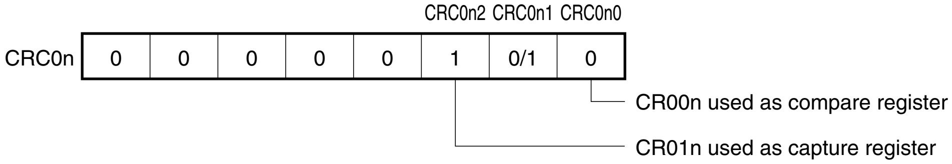
For valid edge detection, sampling is performed using the count clock selected by PRM0n, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 7-18. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)

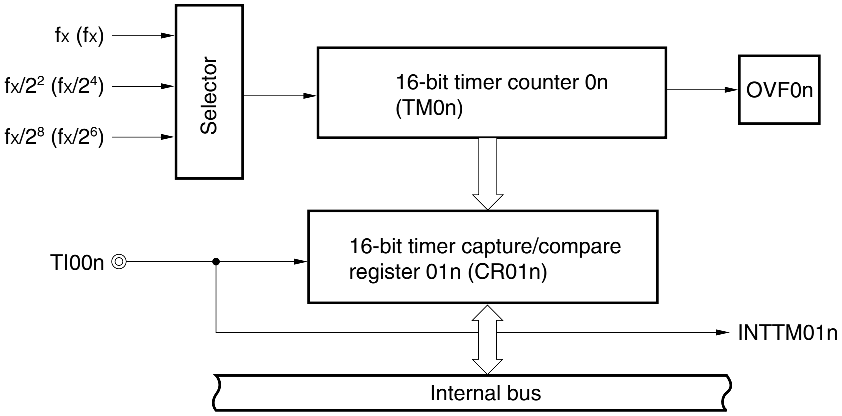

CRC0n2 CRC0n1 CRC0n0
CRC0n
0
0
0
0
0
1
0/1
0
CR00n used as compare register
CR01n used as capture register

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 7-3** to **7-6**.

n = 0: μPD780143, 780144

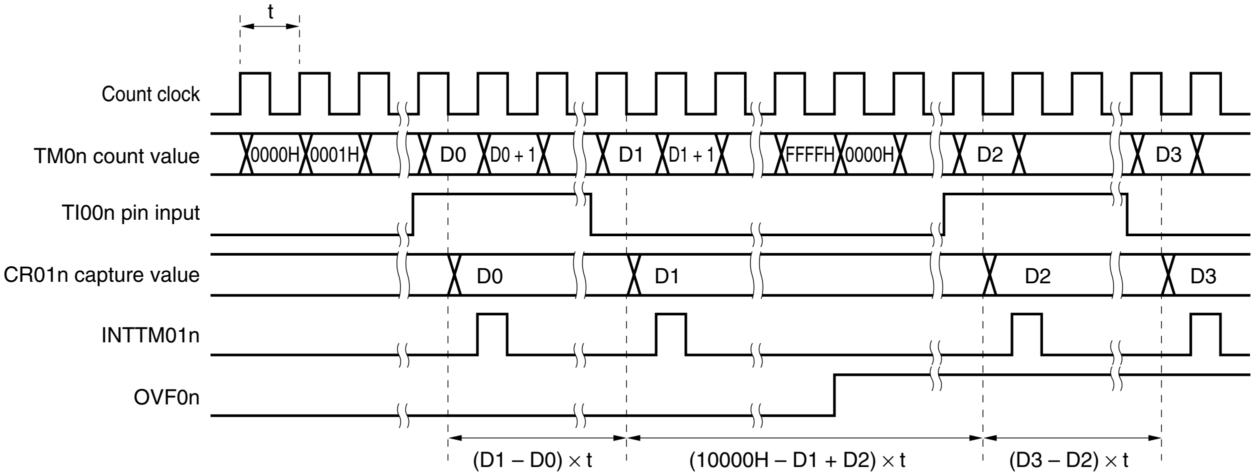
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-19. Configuration Diagram for Pulse Width Measurement with Free-Running Counter


fx (fx)
fx/2² (fx/2⁴)
fx/2⁸ (fx/2⁶)
Selector
16-bit timer counter 0n (TM0n)
OVF0n
TI00n
16-bit timer capture/compare register 01n (CR01n)
INTTM01n
Internal bus

Remark Frequencies without parentheses are for 16-bit timer/event counter 00, and those in parentheses are for 16-bit timer/event counter 01.

Figure 7-20. Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)



Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

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

When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see **Figure 7-21**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00n pin and the TI01n pin.

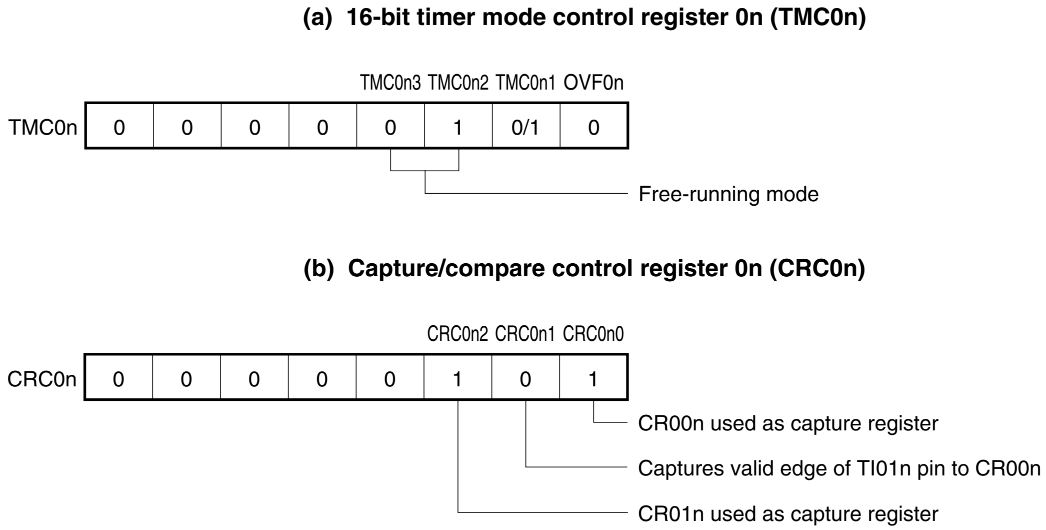
When the edge specified by bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an interrupt request signal (INTTM01n) is set.

Also, when the edge specified by bits 6 and 7 (ES1n0 and ES1n1) of PRM0n is input to the TI01n pin, the value of TM0n is taken into 16-bit timer capture/compare register 00n (CR00n) and an interrupt request signal (INTTM00n) is set.

Any of three edges—rising, falling, or both edges—can be selected as the valid edge of the TI00n pin and the TI01n pin, specified using bits 4 and 5 (ES0n0 and ES0n1) and bits 6 and 7 (ES1n0 and ES1n1) of PRM0n, respectively.

For valid edge detection of the TI00n and TI01n pins, sampling is performed at the interval selected by prescaler mode register 0n (PRM0n), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 7-21. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter


TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 0 1 0/1 0
Free-running mode
CRC0n2 CRC0n1 CRC0n0
CRC0n
0 0 0 0 0 1 0 1
CR00n used as capture register
Captures valid edge of TI01n pin to CR00n
CR01n used as capture register

(a) 16-bit timer mode control register 0n (TMC0n)

(b) Capture/compare control register 0n (CRC0n)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. For details, see **Figures 7-3** and **7-4**.

n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

- **Capture operation (free-running mode)**

  The capture register operation when capture trigger is input is shown below.

Figure 7-22. CR01n Capture Operation with Rising Edge Specified

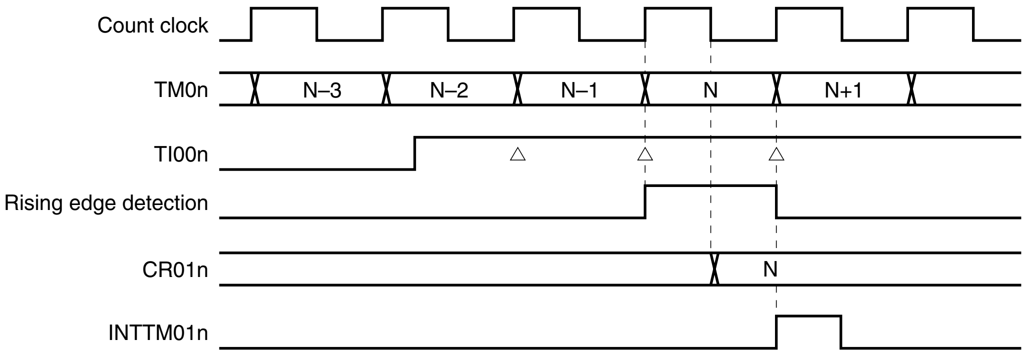
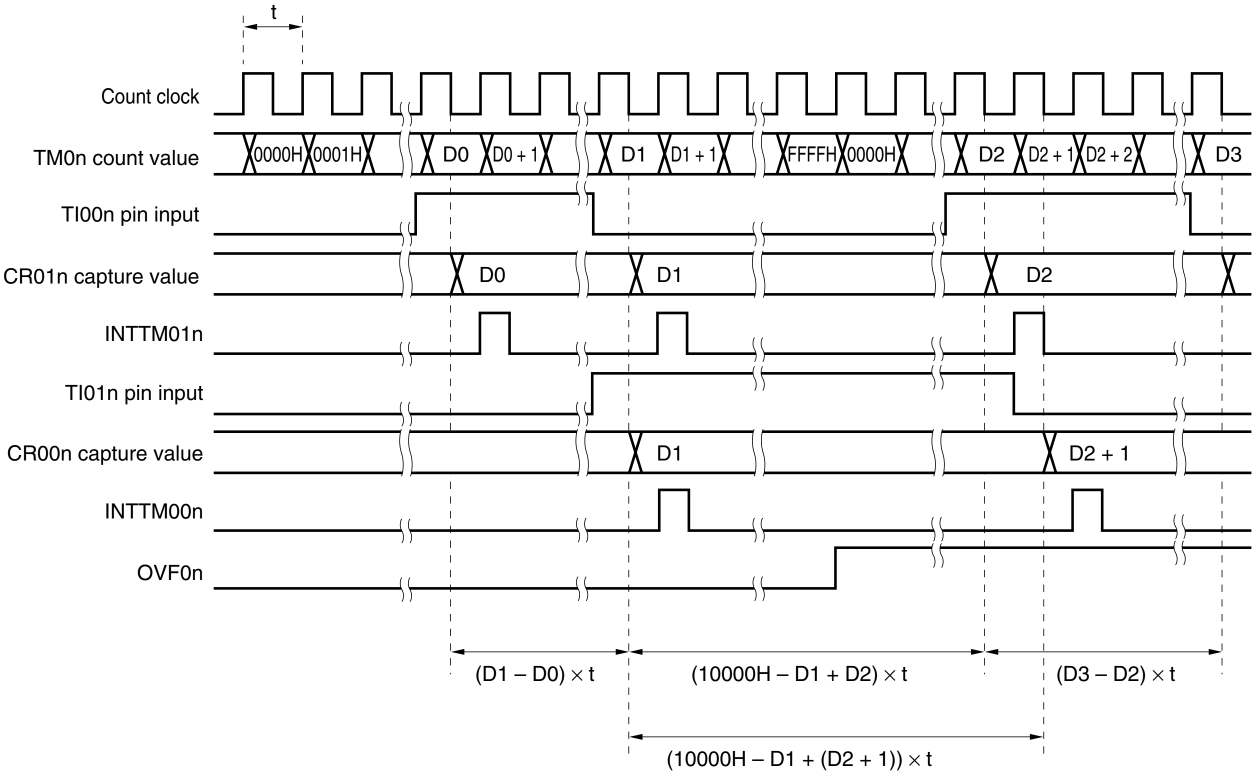


Figure 7-23. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



Remark n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

(3) Pulse width measurement with free-running counter and two capture registers

When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see **Figure 7-24**), it is possible to measure the pulse width of the signal input to the TI00n pin.

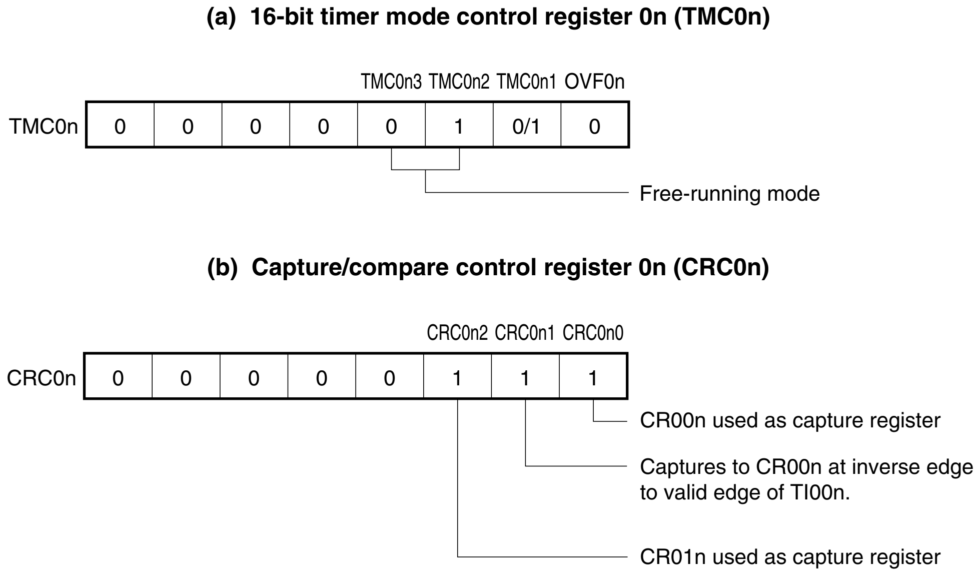
When the edge specified by bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an interrupt request signal (INTTM01n) is set.

Also, when the inverse edge to that of the capture operation is input into CR01n, the value of TM0n is taken into 16-bit timer capture/compare register 00n (CR00n).

Either of two edges—rising or falling—can be selected as the valid edge of the TI00n pin specified using bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

For TI00n pin valid edge detection, sampling is performed at the interval selected by prescaler mode register 0n (PRM0n), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 7-24. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers



(a) 16-bit timer mode control register 0n (TMC0n)

TMC0n3
TMC0n2
TMC0n1
OVF0n
TMC0n
0
0
0
0
0
1
0/1
0
Free-running mode

(b) Capture/compare control register 0n (CRC0n)

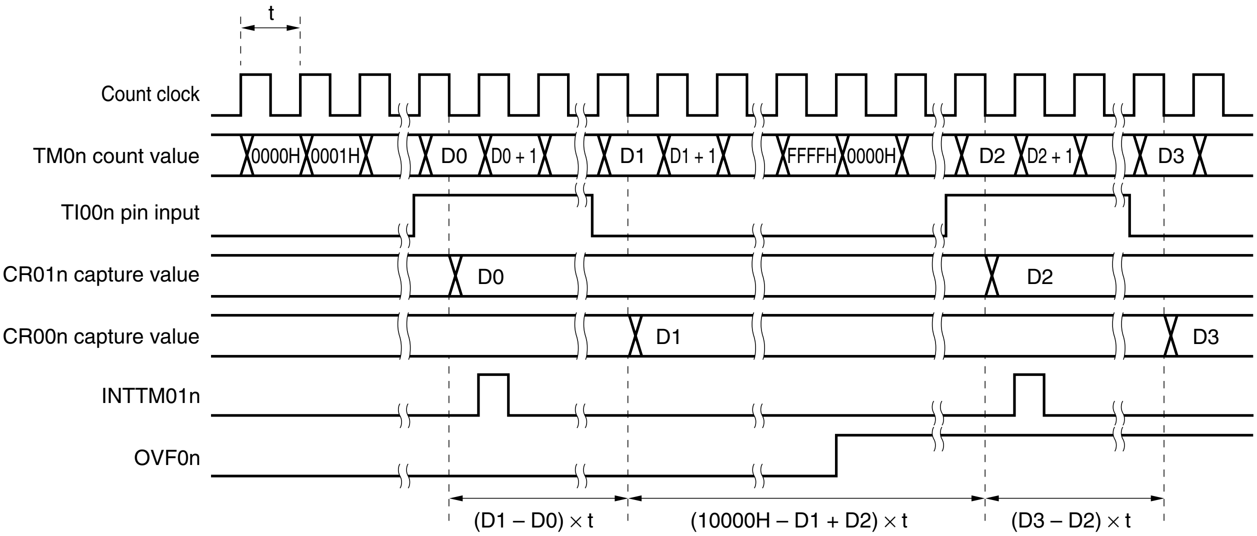
CRC0n2
CRC0n1
CRC0n0
CRC0n
0
0
0
0
0
1
1
1
CR00n used as capture register
Captures to CR00n at inverse edge to valid edge of TI00n.
CR01n used as capture register

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.

n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-25. Timing of Pulse Width Measurement Operation with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



(4) Pulse width measurement by means of restart

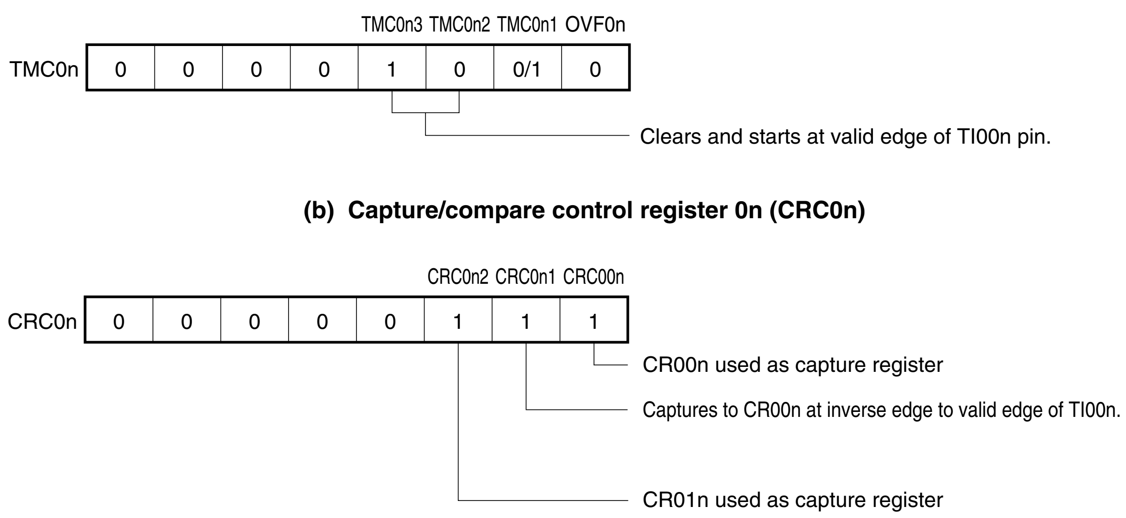
When input of a valid edge to the TI00n pin is detected, the count value of 16-bit timer counter 0n (TM0n) is taken into 16-bit timer capture/compare register 01n (CR01n), and then the pulse width of the signal input to the TI00n pin is measured by clearing TM0n and restarting the count operation (see **Figure 7-26**).

Either of two edges—rising or falling—can be selected using bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

In valid edge detection, sampling is performed using the count clock cycle selected by prescaler mode register 0n (PRM0n) and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-26. Control Register Settings for Pulse Width Measurement by Means of Restart


(a) 16-bit timer mode control register 0n (TMC0n)

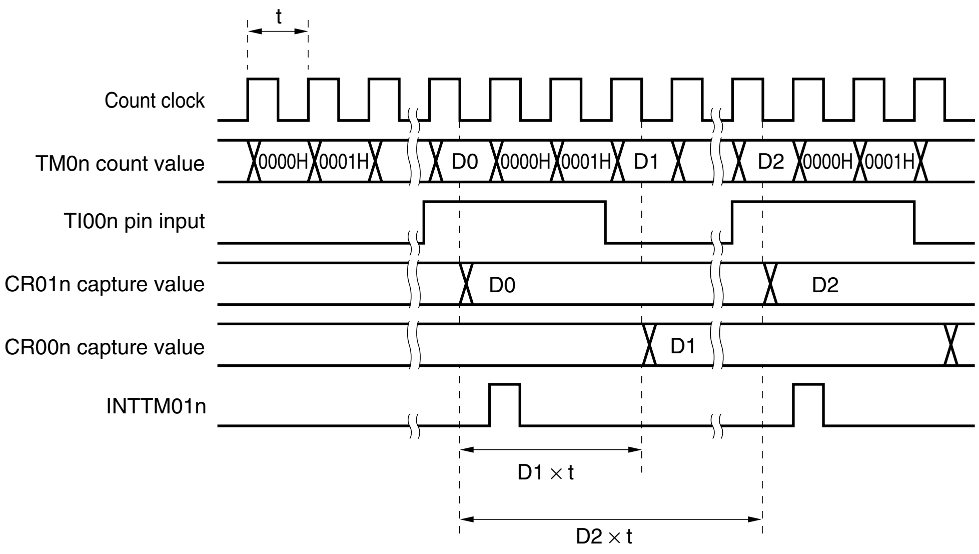
TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0
0
0
0
1
0
0/1
0
Clears and starts at valid edge of TI00n pin.

(b) Capture/compare control register 0n (CRC0n)

CRC0n2 CRC0n1 CRC00n
CRC0n
0
0
0
0
0
1
1
1
CR00n used as capture register
Captures to CR00n at inverse edge to valid edge of TI00n.
CR01n used as capture register

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. For details, see **Figures 7-3** and **7-4**.

Figure 7-27. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

7.4.4 External event counter operation

The external event counter counts the number of external clock pulses input to the TI00n pin using 16-bit timer counter 0n (TM0n).

TM0n is incremented each time the valid edge specified by prescaler mode register 0n (PRM0n) is input.

When the TM0n count value matches the 16-bit timer capture/compare register 00n (CR00n) value, TM0n is cleared to 0 and the interrupt request signal (INTTM00n) is generated.

Input a value other than 0000H to CR00n (a count operation with 1-bit pulse cannot be carried out).

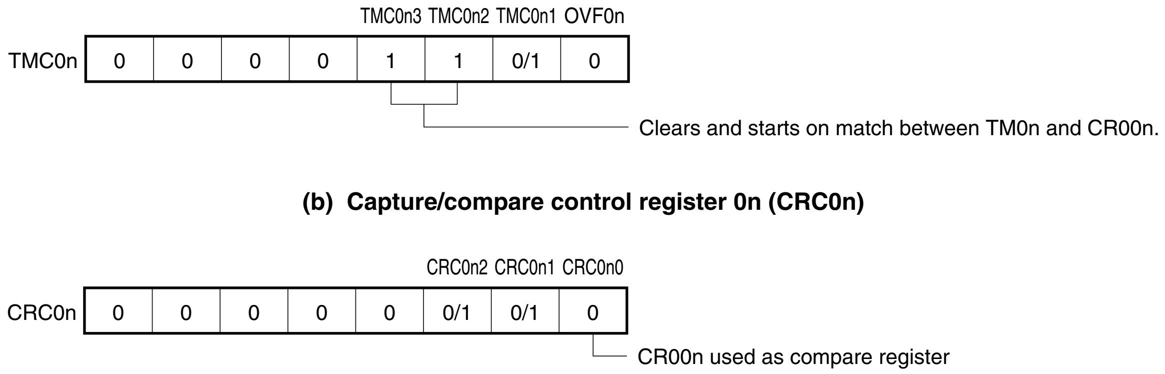
Any of three edges—rising, falling, or both edges—can be selected using bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

Because operation is carried out only after the valid edge is detected twice by sampling using the internal clock (fx), noise with short pulse widths can be eliminated.

Figure 7-28. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control register 0n (TMC0n)

(b) Capture/compare control register 0n (CRC0n)

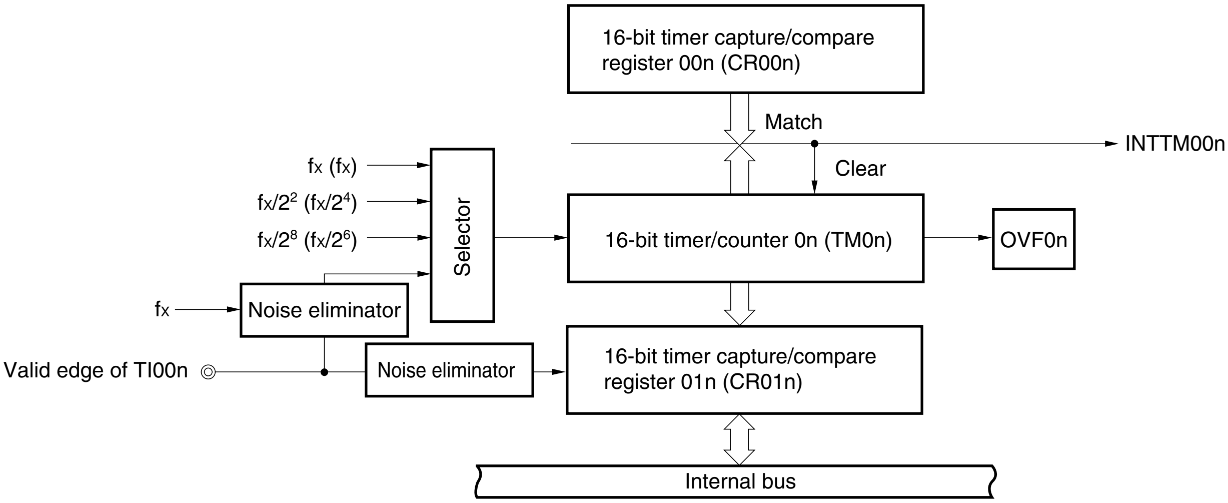

TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0
0
0
0
1
1
0/1
0
Clears and starts on match between TM0n and CR00n.
CRC0n2 CRC0n1 CRC0n0
CRC0n
0
0
0
0
0
0/1
0/1
0
CR00n used as compare register

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. For details, see **Figures 7-3** to **7-6**.

n = 0: μPD780143, 780144

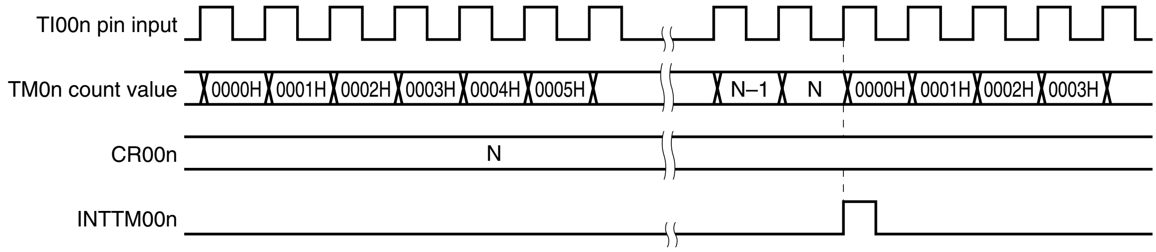
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-29. Configuration Diagram of External Event Counter



Remark Frequencies without parentheses are for 16-bit timer/event counter 00, and those in parentheses are for 16-bit timer/event counter 01.

Figure 7-30. External Event Counter Operation Timing (with Rising Edge Specified)



Caution When reading the external event counter count value, TM0n should be read.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

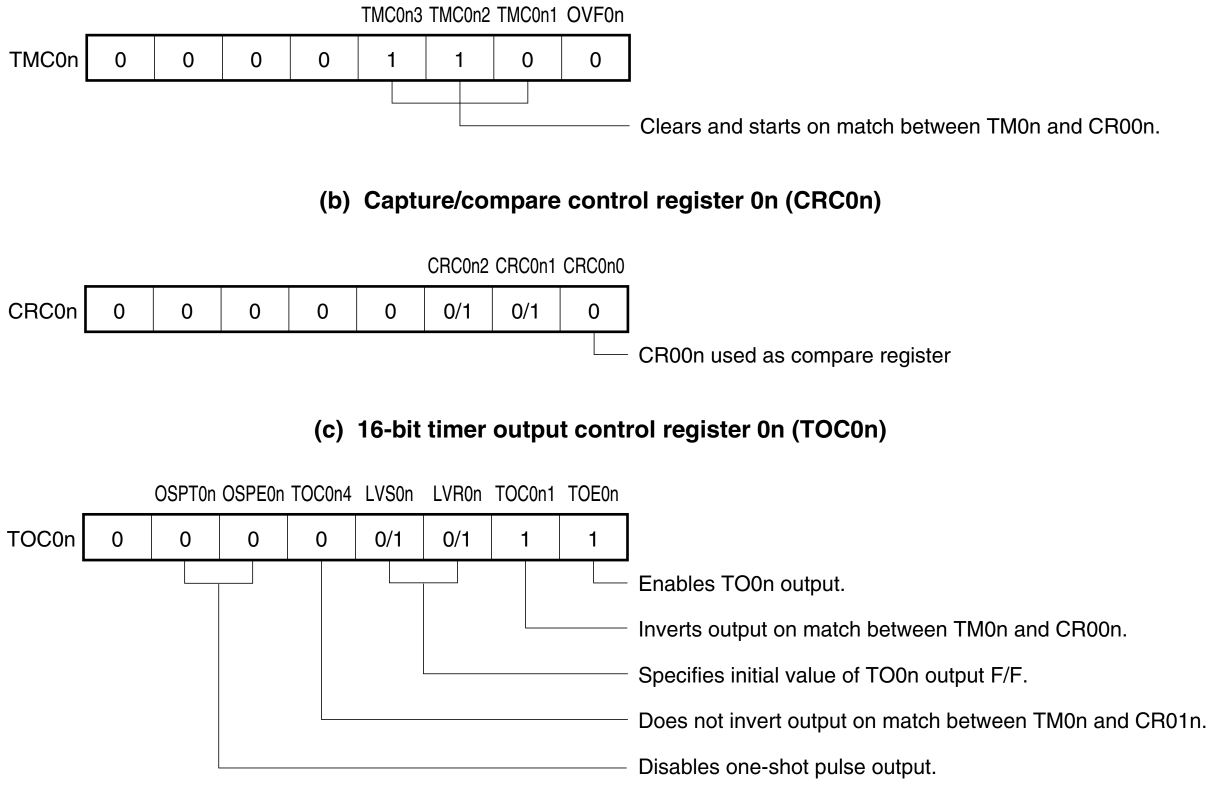
7.4.5 Square-wave output operation

A square wave with any selected frequency can be output at intervals of the count value preset to 16-bit timer capture/compare register 00n (CR00n).

The TO0n pin output status is reversed at intervals of the count value preset to CR00n by setting bit 0 (TOE0n) and bit 1 (TOC0n1) of 16-bit timer output control register 0n (TOC0n) to 1. This enables a square wave with any selected frequency to be output.

Figure 7-31. Control Register Settings in Square-Wave Output Mode

(a) 16-bit timer mode control register 0n (TMC0n)


TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 1 1 0 0
Clears and starts on match between TM0n and CR00n.
CRC0n2 CRC0n1 CRC0n0
CRC0n
0 0 0 0 0 0/1 0/1 0
CR00n used as compare register
OSPT0n OSPE0n TOC0n4 LVS0n LVR0n TOC0n1 TOE0n
TOC0n
0 0 0 0 0/1 0/1 1 1
Enables TO0n output.
Inverts output on match between TM0n and CR00n.
Specifies initial value of TO0n output F/F.
Does not invert output on match between TM0n and CR01n.
Disables one-shot pulse output.

(b) Capture/compare control register 0n (CRC0n)

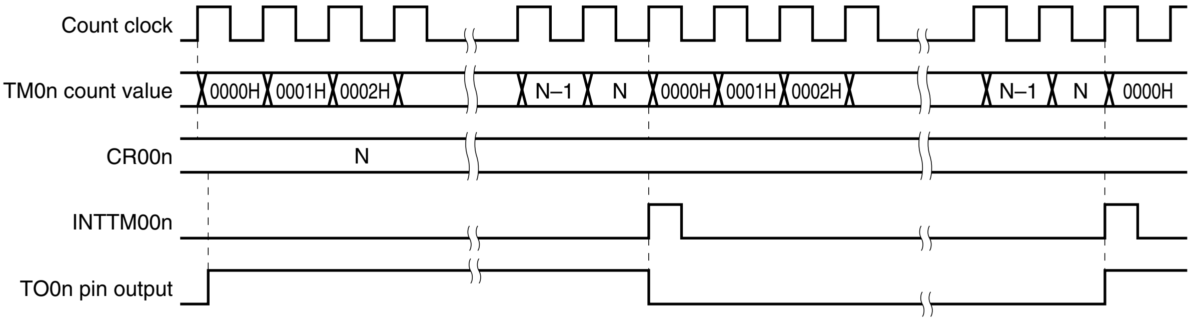
(c) 16-bit timer output control register 0n (TOC0n)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. For details, see **Figures 7-5** to **7-8**.

n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-32. Square-Wave Output Operation Timing



7.4.6 One-shot pulse output operation

16-bit timer/event counter 0n can output a one-shot pulse in synchronization with a software trigger or an external trigger (TI00n pin input).

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TO0n pin by setting 16-bit timer mode control register 0n (TMC0n), capture/compare control register 0n (CRC0n), and 16-bit timer output control register 0n (TOC0n) as shown in Figure 7-33, and by setting bit 6 (OSPT0n) of the TOC0n register to 1 by software.

By setting the OSPT0n bit to 1, 16-bit timer/event counter 0n is cleared and started, and its output becomes active at the count value (N) set in advance to 16-bit timer capture/compare register 01n (CR01n). After that, the output becomes inactive at the count value (M) set in advance to 16-bit timer capture/compare register 00n (CR00n)[Note].

Even after the one-shot pulse has been output, the TM0n register continues its operation. To stop the TM0n register, the TMC0n3 and TMC0n2 bits of the TMC0n register must be set to 00.

Note The case where N < M is described here. When N > M, the output becomes active with the CR00n register and inactive with the CR01n register.

Cautions 1. Do not set the OSPT0n bit 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.

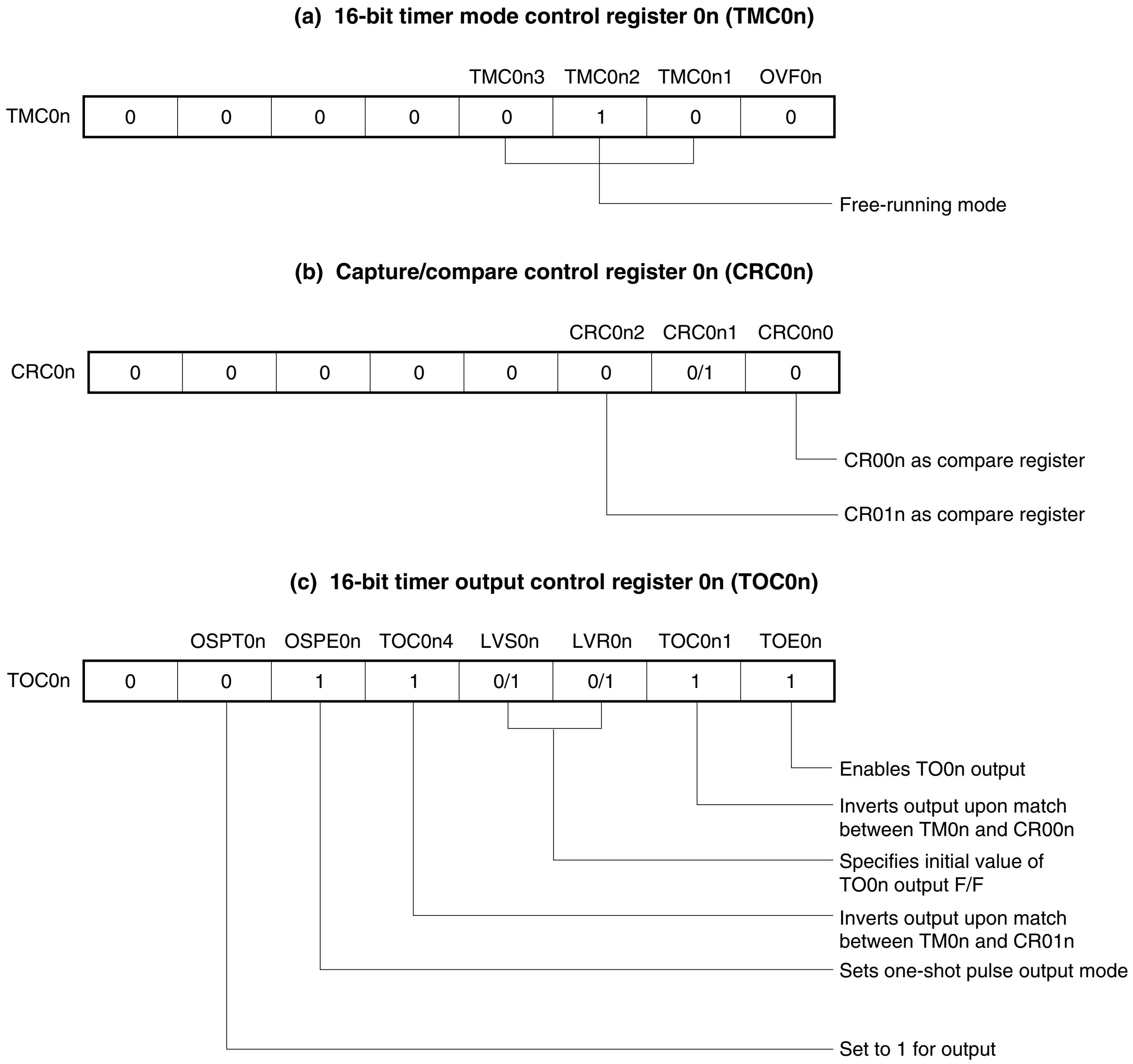
2. When using the one-shot pulse output of 16-bit timer/event counter 0n with a software trigger, do not change the level of the TI00n pin or its alternate-function port pin.

Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI00n pin or its alternate-function port pin, resulting in the output of a pulse at an undesired timing.

Remark n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-33. Control Register Settings for One-Shot Pulse Output with Software Trigger



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	0	1	0	0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0	0/1	0

CR00n as compare register
CR01n as compare register

(c) 16-bit timer output control register 0n (TOC0n)

		OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n
TOC0n	0	0	1	1	0/1	0/1	1	1

Enables TO0n output
Inverts output upon match between TM0n and CR00n
Specifies initial value of TO0n output F/F
Inverts output upon match between TM0n and CR01n
Sets one-shot pulse output mode
Set to 1 for output

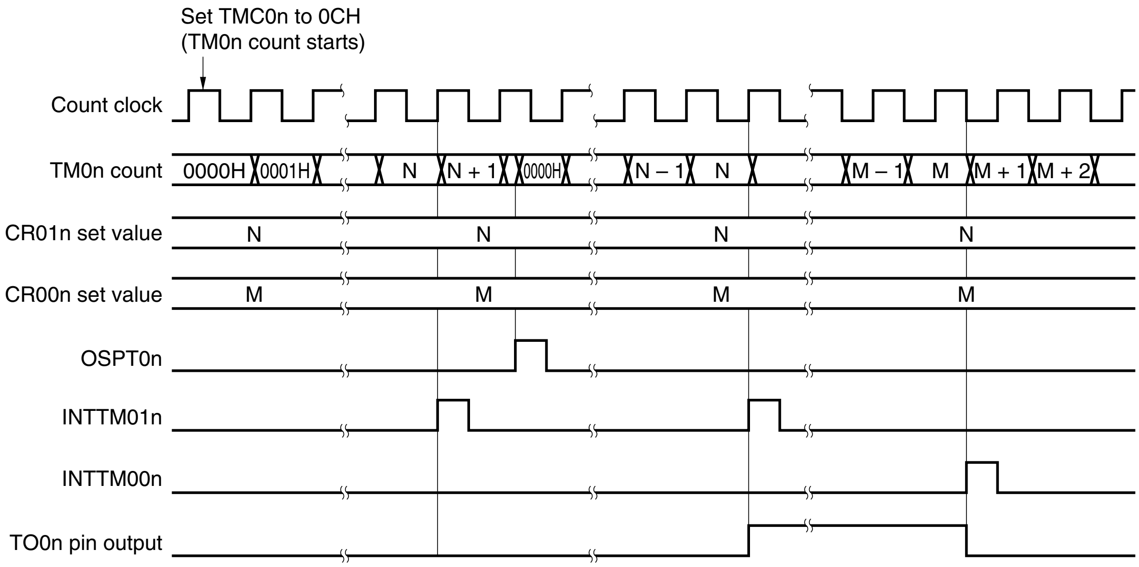
Caution Do not set 0000H to the CR00n and CR01n registers.

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. For details, see **Figures 7-5** to **7-8**.

n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-34. Timing of One-Shot Pulse Output Operation with Software Trigger



Caution **16-bit timer counter 0n starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC0n3 and TMC0n2 bits.**

Remark N < M

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TO0n pin by setting 16-bit timer mode control register 0n (TMC0n), capture/compare control register 0n (CRC0n), and 16-bit timer output control register 0n (TOC0n) as shown in Figure 7-35, and by using the valid edge of the TI00n pin as an external trigger.

The valid edge of the TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of prescaler mode register 0n (PRM0n). The rising, falling, or both the rising and falling edges can be specified.

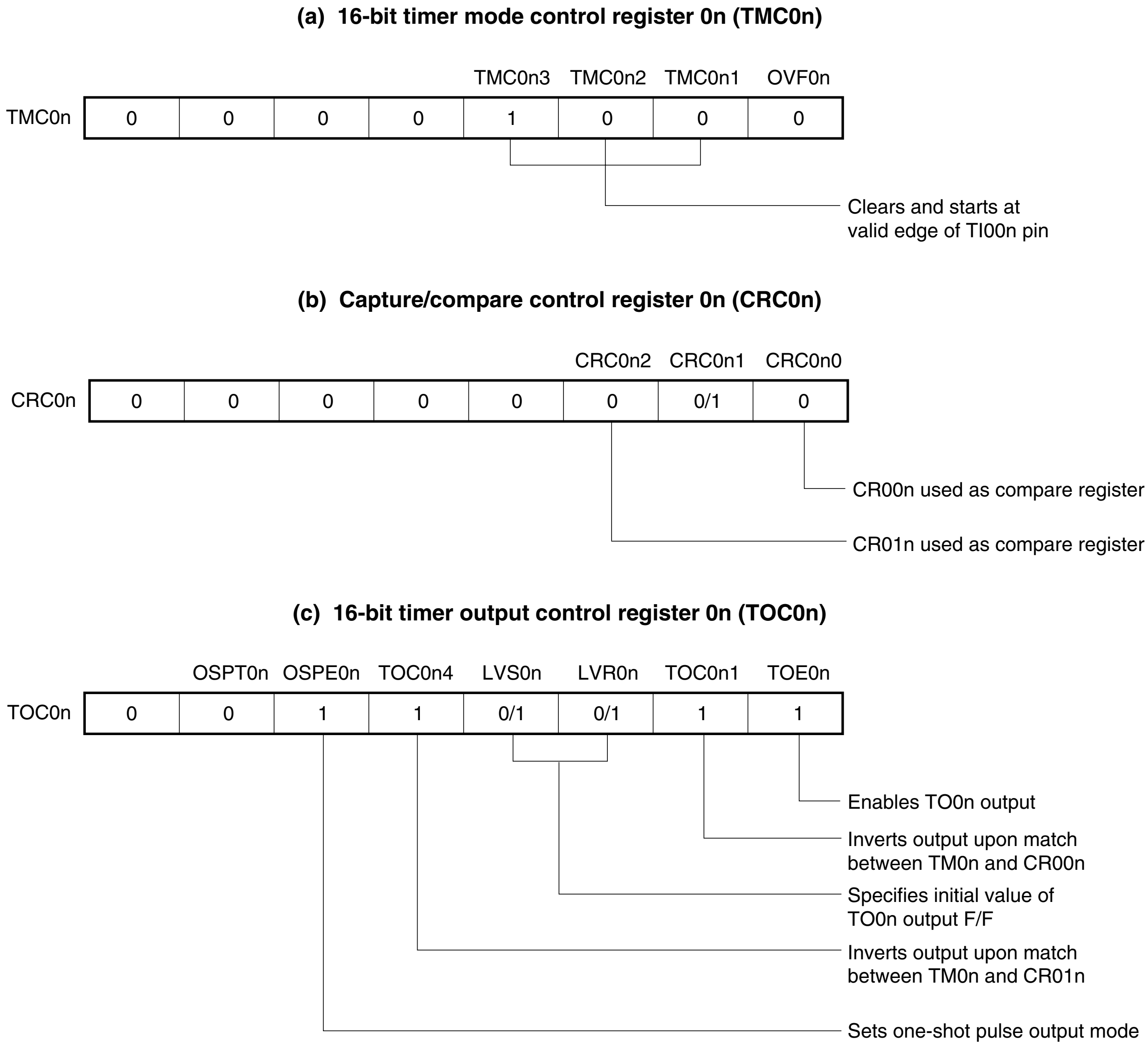
When the valid edge of the TI00n pin is detected, the 16-bit timer/event counter is cleared and started, and the output becomes active at the count value set in advance to 16-bit timer capture/compare register 01n (CR01n). After that, the output becomes inactive at the count value set in advance to 16-bit timer capture/compare register 00n (CR00n)[Note].

Note The case where N < M is described here. When N > M, the output becomes active with the CR00n register and inactive with the CR01n register.

Caution **Even if the external trigger is generated again while the one-shot pulse is output, it is ignored.**

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-35. Control Register Settings for One-Shot Pulse Output with External Trigger



(a) 16-bit timer mode control register 0n (TMC0n)

TMC0n3
TMC0n2
TMC0n1
OVF0n
TMC0n
0
0
0
0
1
0
0
0
Clears and starts at valid edge of TI00n pin

(b) Capture/compare control register 0n (CRC0n)

CRC0n2
CRC0n1
CRC0n0
CRC0n
0
0
0
0
0
0
0/1
0
CR00n used as compare register
CR01n used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

OSPT0n
OSPE0n
TOC0n4
LVS0n
LVR0n
TOC0n1
TOE0n
TOC0n
0
0
1
1
0/1
0/1
1
1
Enables TO0n output
Inverts output upon match between TM0n and CR00n
Specifies initial value of TO0n output F/F
Inverts output upon match between TM0n and CR01n
Sets one-shot pulse output mode

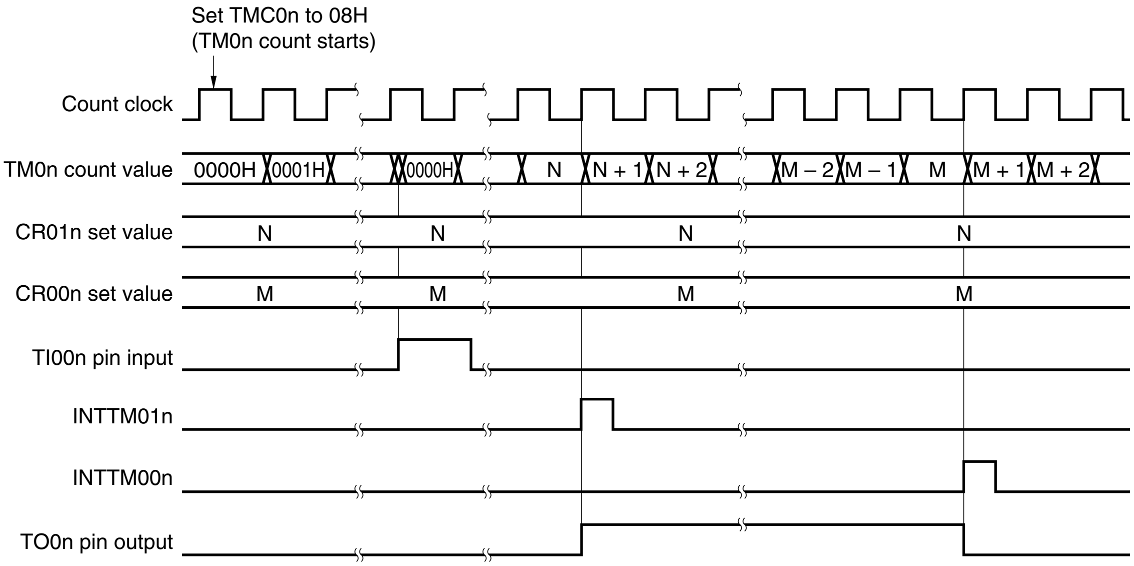
Caution Do not set 0000H to the CR00n and CR01n registers.

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. For details, see **Figures 7-5** to **7-8**.

n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

Figure 7-36. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution **16-bit timer counter 0n starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC0n2 and TMC0n3 bits.**

Remark N < M

n = 0: μPD780143, 780144

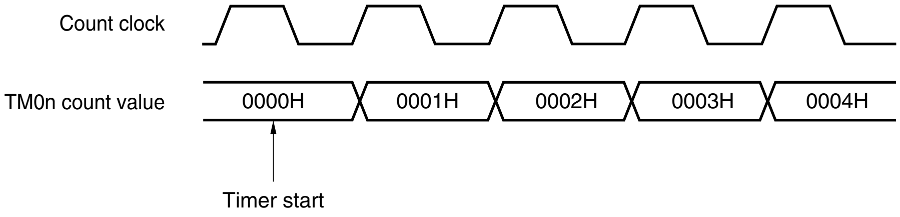
n = 0, 1: μPD780146, 780148, 78F0148

7.5 Cautions for 16-Bit Timer/Event Counters 00 and 01

(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 0n (TM0n) is started asynchronously to the count clock.

Figure 7-37. Start Timing of 16-Bit Timer Counter 0n (TM0n)



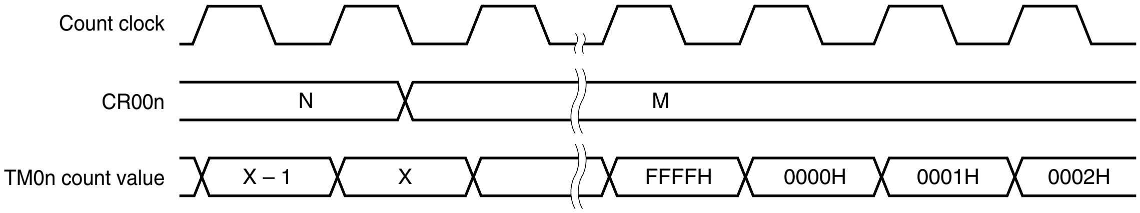
(2) 16-bit timer capture/compare register setting (in the mode in which clear & start occurs on match between TM0n and CR00n)

Set 16-bit timer capture/compare registers 00n, 01n (CR00n, CR01n) to other than 0000H. This means a 1-pulse count operation cannot be performed when 16-bit timer/event counter 0n is used as an event counter.

(3) Operation after compare register change during timer count operation

If the value after 16-bit timer capture/compare register 00n (CR00n) is changed is smaller than that of 16-bit timer counter 0n (TM0n), TM0n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR00n changes is smaller than that (N) before the change, it is necessary to restart the timer after changing CR00n.

Figure 7-38. Timings After Change of Compare Register During Timer Count Operation



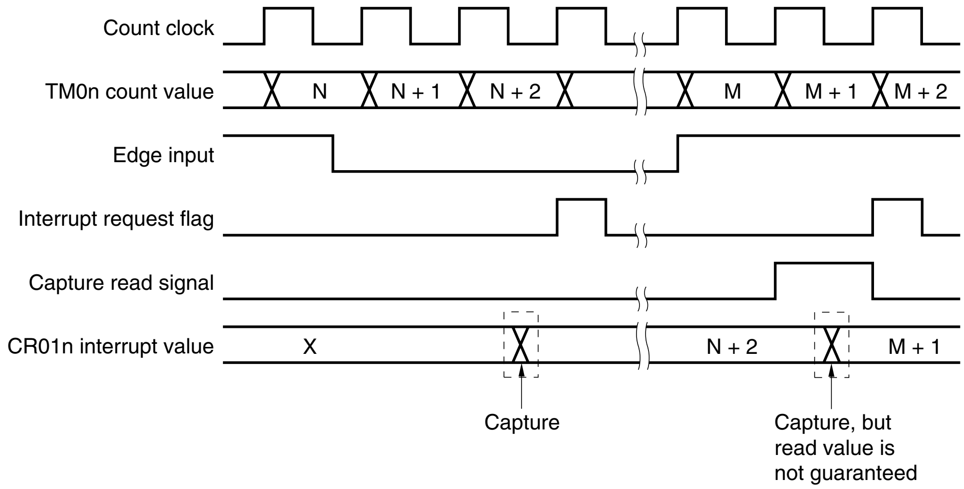
Remark N > X > M
n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

(4) Capture register data retention timing

If the valid edge of the TI00n pin is input during 16-bit timer capture/compare register 01n (CR01n) read, CR01n performs a capture operation. However, the value read at this time is not guaranteed.

The interrupt request flag (TMIF01n) is set upon detection of the valid edge.

Figure 7-39. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00n pin after setting bits 2 and 3 (TMC0n2 and TMC0n3) of 16-bit timer mode control register 0n (TMC0n) to 0, 0, respectively, and then stopping timer operation. The valid edge is set using bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

(6) Re-triggering one-shot pulse

(a) One-shot pulse output by software

When a one-shot pulse is output, do not set the OSPT0n bit to 1. Do not output the one-shot pulse again until INTTM00n, which occurs upon a match with the CR00n register, or INTTM01n, which occurs upon a match with the CR01n register, occurs.

(b) One-shot pulse output with external trigger

If the external trigger occurs again while a one-shot pulse is output, it is ignored.

(c) One-shot pulse output function

When using the one-shot pulse output of 16-bit timer/event counter 0n with a software trigger, do not change the level of the TI00n 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 TI00n pin or its alternate function port pin, resulting in the output of a pulse at an undesired timing.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

(7) Operation of OVF0n flag

<1> The OVF0n flag is set to 1 in the following case.

When of the following modes: the mode in which clear & start occurs on a match between TM0n and CR00n, the mode in which clear & start occurs on a TI0n valid edge, or the free-running mode, is selected

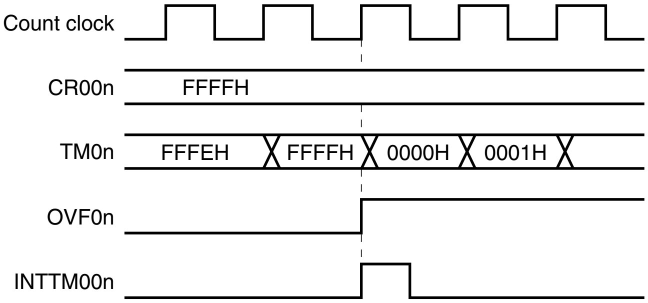
↓

CR00n is set to FFFFH

↓

TM0n is counted up from FFFFH to 0000H.

Figure 7-40. Operation Timing of OVF0n Flag



<2> Even if the OVF0n flag is cleared before the next count clock (before TM0n becomes 0001H) after the occurrence of TM0n overflow, the OVF0n flag is re-set newly and clear is disabled.

(8) Conflicting operations

Conflict between the read period of the 16-bit timer capture/compare register (CR00n/CR01n) and capture trigger input (CR00n/CR01n used as capture register)

Capture trigger input has priority. The data read from CR00n/CR01n is undefined.

(9) Timer operation

<1> Even if 16-bit timer counter 0n (TM0n) is read, the value is not captured by 16-bit timer capture/compare register 01n (CR01n).

<2> Regardless of the CPU's operation mode, when the timer stops, the input signals to the TI00n/TI01n 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 TI00n valid edge. In the mode in which clear & start occurs on a match between the TM0n register and CR00n register, one-shot pulse output is not possible because an overflow does not occur.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

(10) Capture operation

<1> If TI00n valid edge is specified as the count clock, a capture operation by the capture register specified as the trigger for TI00n is not possible.

<2> To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 0n (PRM0n).

<3> The capture operation is performed at the falling edge of the count clock. An interrupt request input (INTTM00n/INTTM01n), however, is generated at the rise of the next count clock.

(11) Compare operation

<1> When the 16-bit timer capture/compare register (CR00n/CR01n) is overwritten during timer operation, a match interrupt may be generated or a clear operation may not be performed normally if that value is close to or larger than the timer value.

<2> A capture operation may not be performed for CR00n/CR01n set in compare mode even if a capture trigger has been input.

(12) Edge detection

<1> If the TI00n or TI01n 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 TI00n or TI01n pin to enable the 16-bit timer counter 0n (TM0n) operation, a rising edge is detected immediately after the operation is enabled. Be careful therefore when pulling up the TI00n or TI01n pin. However, the rising edge is not detected at restart after the operation has been stopped once.

<2> The sampling clock used to remove noise differs when the TI00n 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 fx, and in the latter case the count clock is selected by prescaler mode register 0n (PRM0n). The capture operation is started only after a valid edge is detected twice by sampling, thus eliminating noise with a short pulse width.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

8.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

Figures 8-1 and 8-2 show the block diagrams of 8-bit timer/event counters 50 and 51.

Figure 8-1. Block Diagram of 8-Bit Timer/Event Counter 50

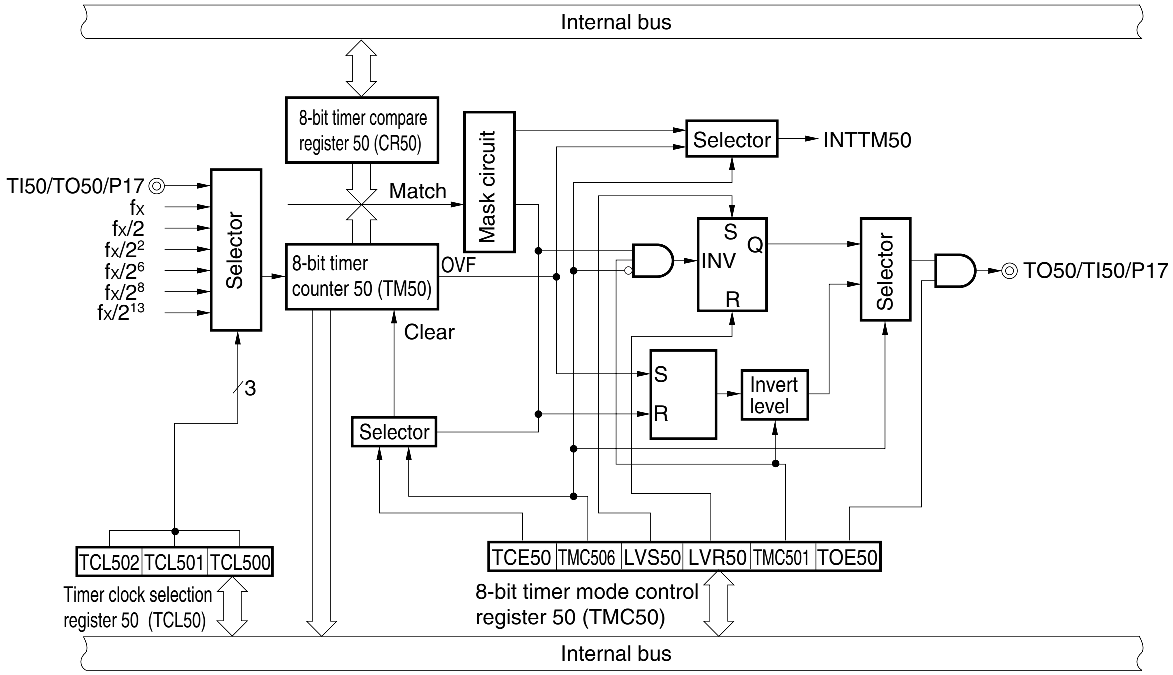
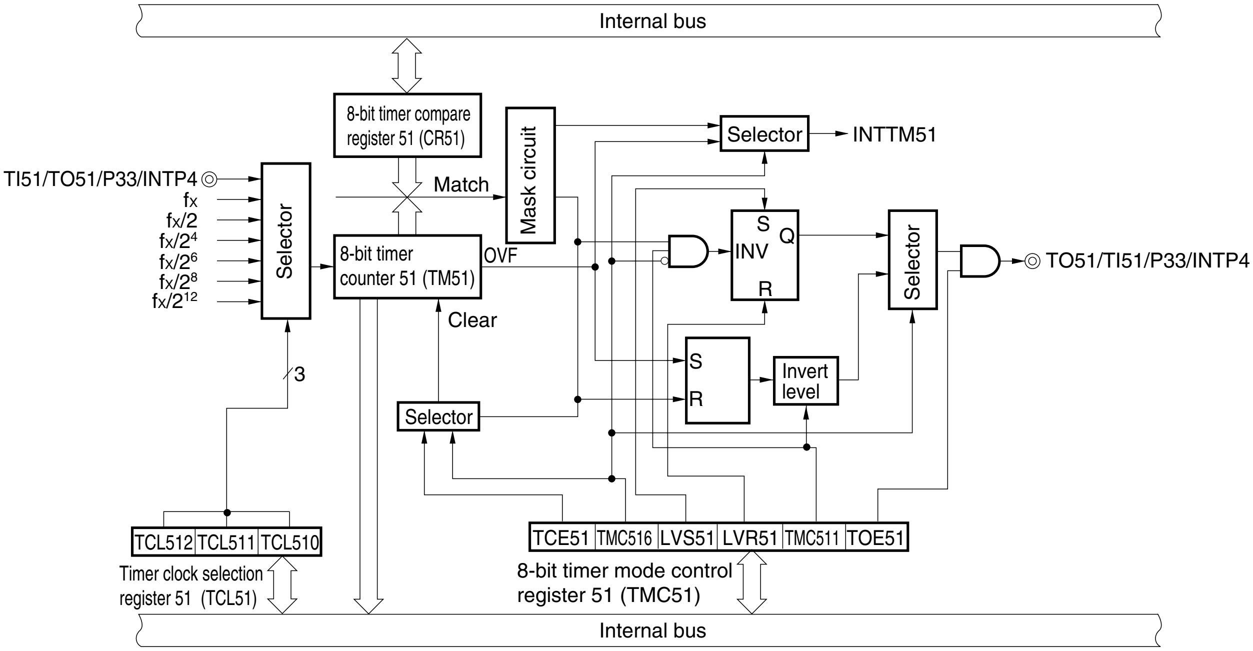


Figure 8-2. Block Diagram of 8-Bit Timer/Event Counter 51



8.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 consist of the following hardware.

Table 8-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer output	1 (TO5n)
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 1 (PM1)Note or port mode register 3 (PM3)Note

Note See **Figure 4-11 Block Diagram of P16 and P17** and **Figure 4-14 Block Diagram of P33**.

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit register that counts the count pulses and is read-only.

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

When the count value is read during operation, count clock input is temporary stopped, and then the count value is read. In the following situations, the count value is cleared to 00H.

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

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

(2) 8-bit timer compare register 5n (CR5n)

CR5n can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match.

In PWM mode, when the TO5n pin becomes active due to a TM5n overflow and the values of TM5n and CR5n match, the TO5n pin becomes inactive.

The value of CR5n can be set within 00H to FFH.

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

8.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51

The following three registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 1 (PM1) or port mode register 3 (PM3)

(1) Timer clock selection register 5n (TCL5n)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of TI5n input.
TCL5n can be set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL5n to 00H.

Remark n = 0, 1

Figure 8-3. 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	TI50 falling edge
0	0	1	TI50 rising edge
0	1	0	fx (10 MHz)
0	1	1	fx/2 (5 MHz)
1	0	0	fx/2^{2} (2.5 MHz)
1	0	1	fx/2^{6} (156.25 kHz)
1	1	0	fx/2^{8} (39.06 kHz)
1	1	1	fx/2^{13} (1.22 kHz)

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. fx: X1 input clock oscillation frequency
2. Figures in parentheses apply to operation at fx = 10 MHz.

Figure 8-4. Format of Timer Clock Selection Register 51 (TCL51)

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection
0	0	0	TI51 falling edge
0	0	1	TI51 rising edge
0	1	0	fx (10 MHz)
0	1	1	fx/2 (5 MHz)
1	0	0	fx/2^4 (625 kHz)
1	0	1	fx/2^6 (156.25 kHz)
1	1	0	fx/2^8 (39.06 kHz)
1	1	1	fx/2^{12} (2.44 kHz)

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

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

Remarks 1. fx: X1 input clock oscillation frequency

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

(2) 8-bit timer mode control register 5n (TMC5n)

TMC5n is a register that performs the following five types of settings.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Timer output F/F (flip-flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode
<5> Timer output control

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Remark n = 0, 1

Figure 8-5. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF6BH After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Mode in which clear & start occurs on a match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE50	Timer output control
0	Output disabled (TO50 pin outputs the low level)
1	Output enabled

Figure 8-6. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FF43H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TO51 pin outputs the low level)
1	Output enabled

Cautions 1. To clear TCE5n to 0, set the interrupt mask flag (TMMK5n) to 1 beforehand. Otherwise, an interrupt may occur when TCE5n is cleared.

TCE5n is cleared to 0 as follows.

TMMK5n = 1;	**Mask set**
TCE5n = 0;	**Timer clear**
TMIF5n = 0;	**Interrupt request flag clear**
TMMK5n = 0;	**Mask clear**
⋮	
TCE5n = 1;	**Timer start**
⋮	

2. The settings of LVS5n and LVR5n are valid in other than PWM mode.

3. Do not rewrite TMC5n1 and TOE5n simultaneously.

4. When switching to the PWM mode, do not rewrite TM5n6 and LVS5n or LVR5n simultaneously.

5. To rewrite TMC5n6, stop operation beforehand.

Remarks 1. In PWM mode, PWM output is made inactive by setting TCE5n to 0.

2. If LVS5n and LVR5n are read after data is set, 0 is read.

3. The values of the TMC5n6, LVS5n, LVR5n, TMC5n1, and TOE5n bits are reflected at the TO5n pin regardless of the value of TCE5n.

4. n = 0, 1

(3) Port mode register 1 (PM1) and 3 (PM3)

These registers set port 1 and 3 input/output in 1-bit units.

When using the P17/TO50/TI50 and P33/TO51/TI51 pins for timer output, set PM17 and PM33 and the output latches of P17 and P33 to 0.

PM1 and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 8-7. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 8-8. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	0	0	0	0	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operations of 8-Bit Timer/Event Counters 50 and 51

8.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

[Setting]

<1> Set the registers.

- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don’t care)

<2> After TCE5n = 1 is set, the count operation starts.

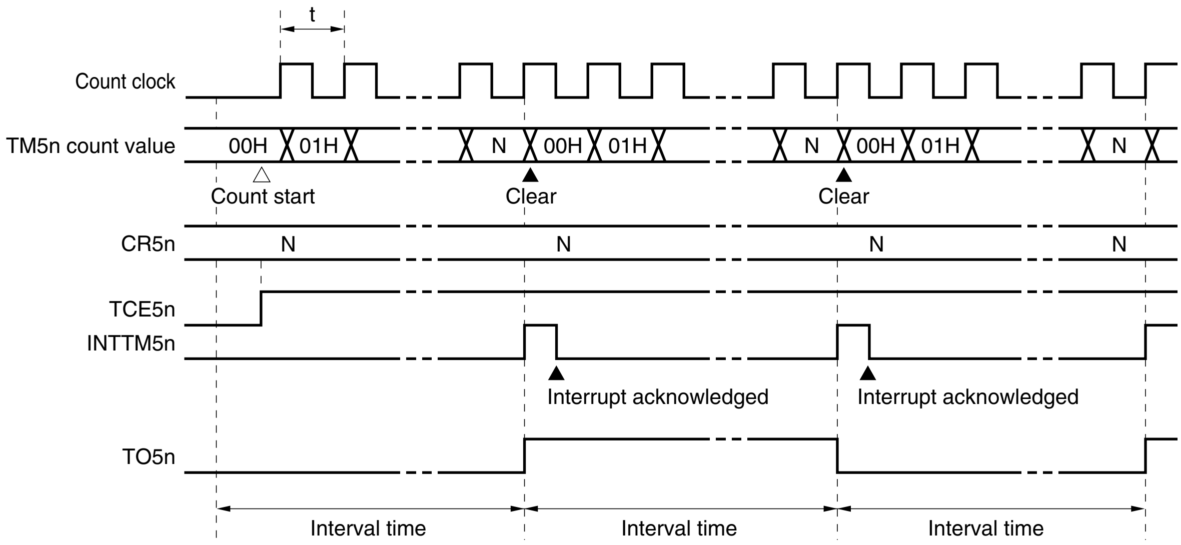
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n is generated repeatedly at the same interval.
Set TCE5n to 0 to stop the count operation.

Caution Do not write other values to CR5n during operation.

Figure 8-9. Interval Timer Operation Timing (1/2)

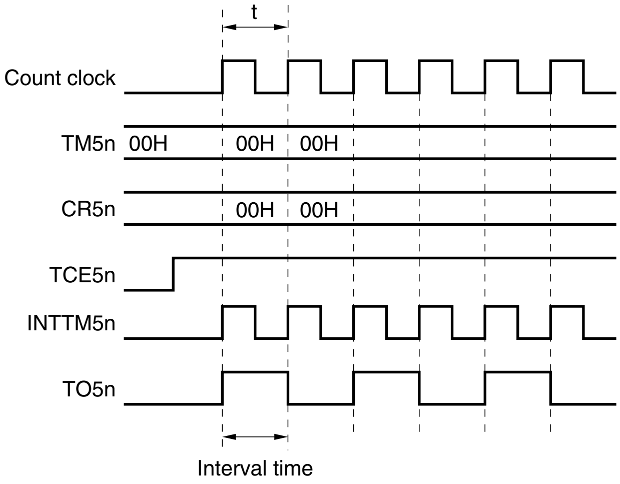
(a) Basic operation



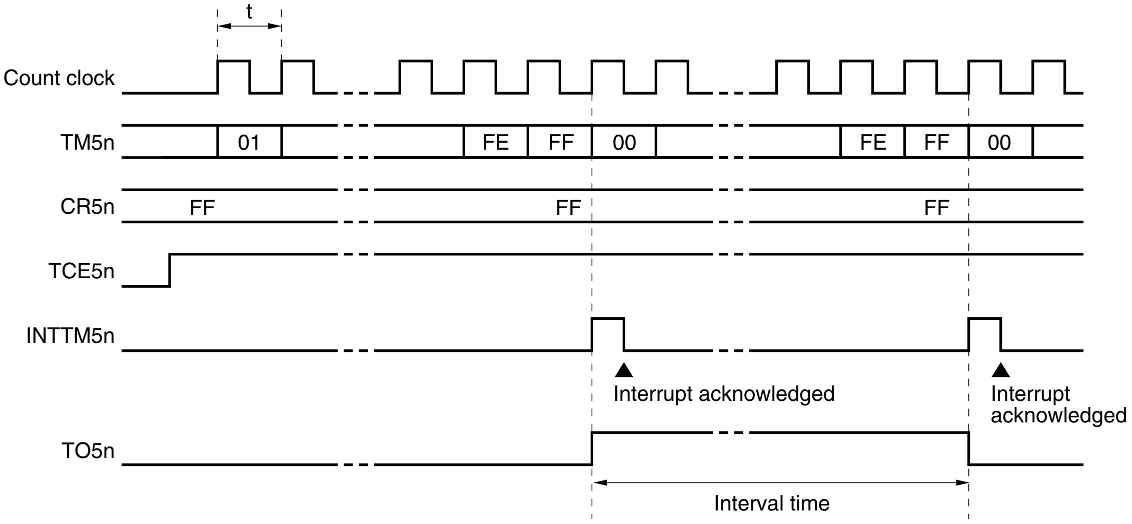
Remark Interval time = (N + 1) × t
N = 00H to FFH
n = 0, 1

Figure 8-9. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0, 1

8.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to TI5n by 8-bit timer counter 5n (TM5n).

TM5n is incremented each time the valid edge specified by timer clock selection register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

Whenever the TM5n value matches the value of CR5n, INTTM5n is generated.

[Setting]

<1> Set each register.

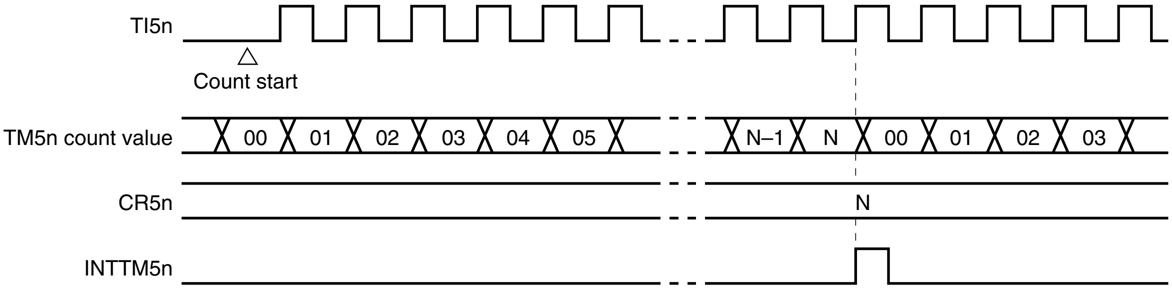
- TCL5n: Select TI5n input edge.
  TI5n falling edge → TCL5n = 00H
  TI5n rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 0000××00B × = Don’t care)

<2> When TCE5n = 1 is set, the number of pulses input from TI5n is counted.

<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

Figure 8-10. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0, 1

8.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals of the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is inverted at intervals of the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register.

- Set the port latches (P17 and P33)[Note] and port mode registers (PM17 and PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F inversion enabled
Timer output enabled
(TMC5n = 00001011B or 00000111B)

<2> After TCE5n = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n.

The frequency is as follows.

Frequency = $f_{CNT}/2 (N + 1)$

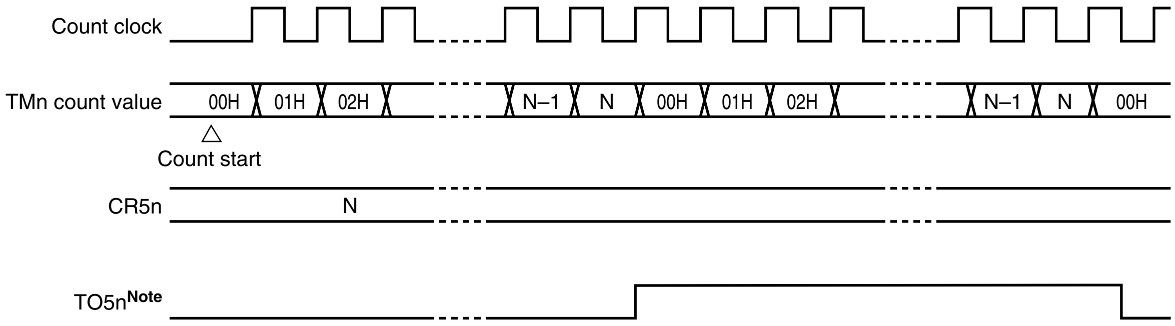
(N: 00H to FFH, f_{CNT}: Count clock)

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

Caution Do not write other values to CR5n during operation.

Remark n = 0, 1

Figure 8-11. Square-Wave Output Operation Timing



Note The initial value of TO5n output can be set by bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

8.4.4 PWM output operation

8-bit timer/event counter 5n operates as a PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty ratio pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

PWM output can be enabled/disabled with bit 0 (TOE5n) of TMC5n.

Caution In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

(1) PWM output basic operation

[Setting]

<1> Set each register.

- Set the port latches (P17, P33)[Note] and port mode registers (PM17, PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Set TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

[PWM output operation]

<1> PWM output (output from TO5n) outputs an inactive level after the count operation starts until an overflow occurs.

<2> When an overflow occurs, the active level set in <1> above is output.
The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).

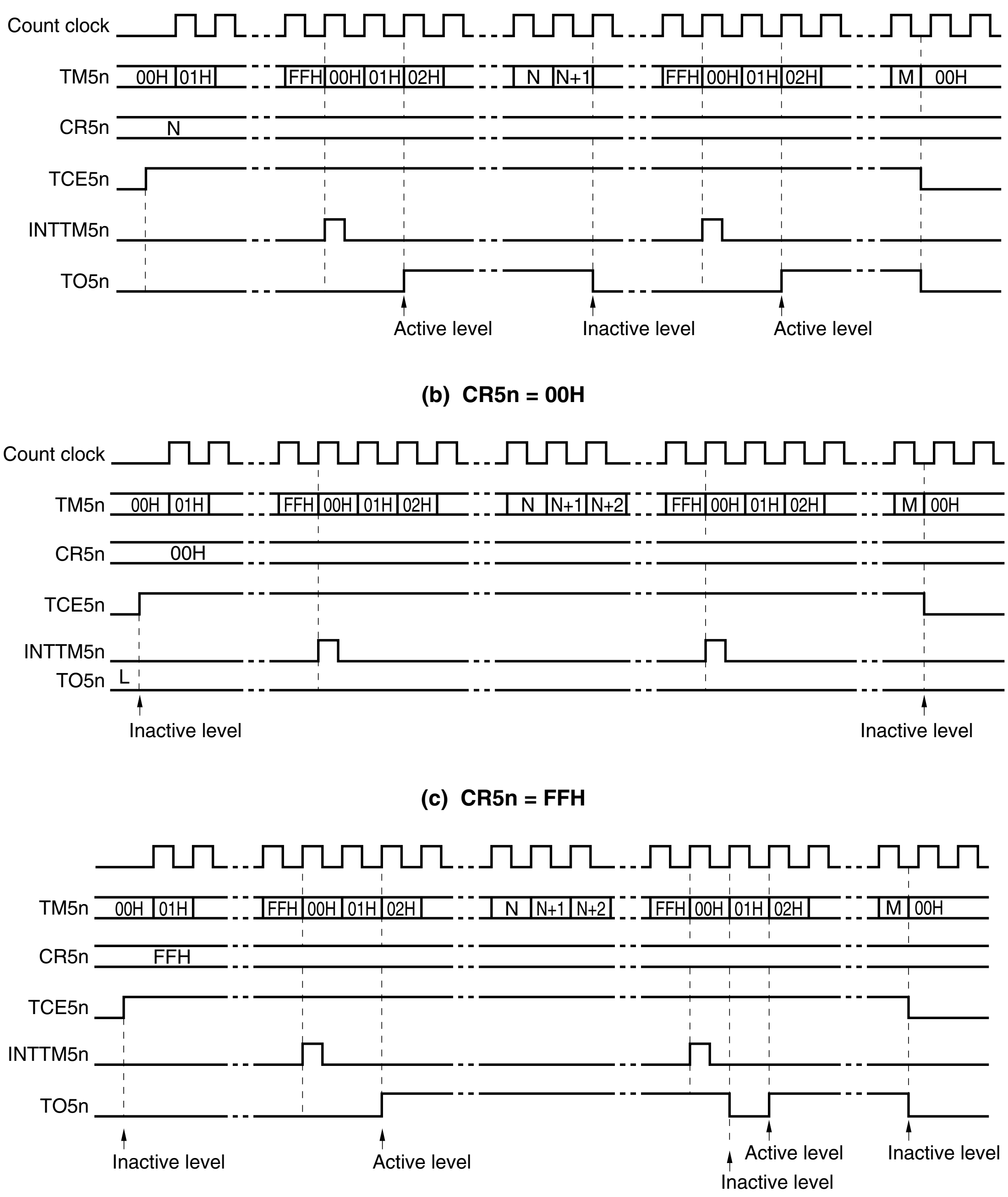
<3> After the CR5n matches the count value, the inactive level is output until an overflow occurs again.

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

<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.

Remark n = 0, 1

Figure 8-12. PWM Output Operation Timing



(a) Basic operation (active level = H)

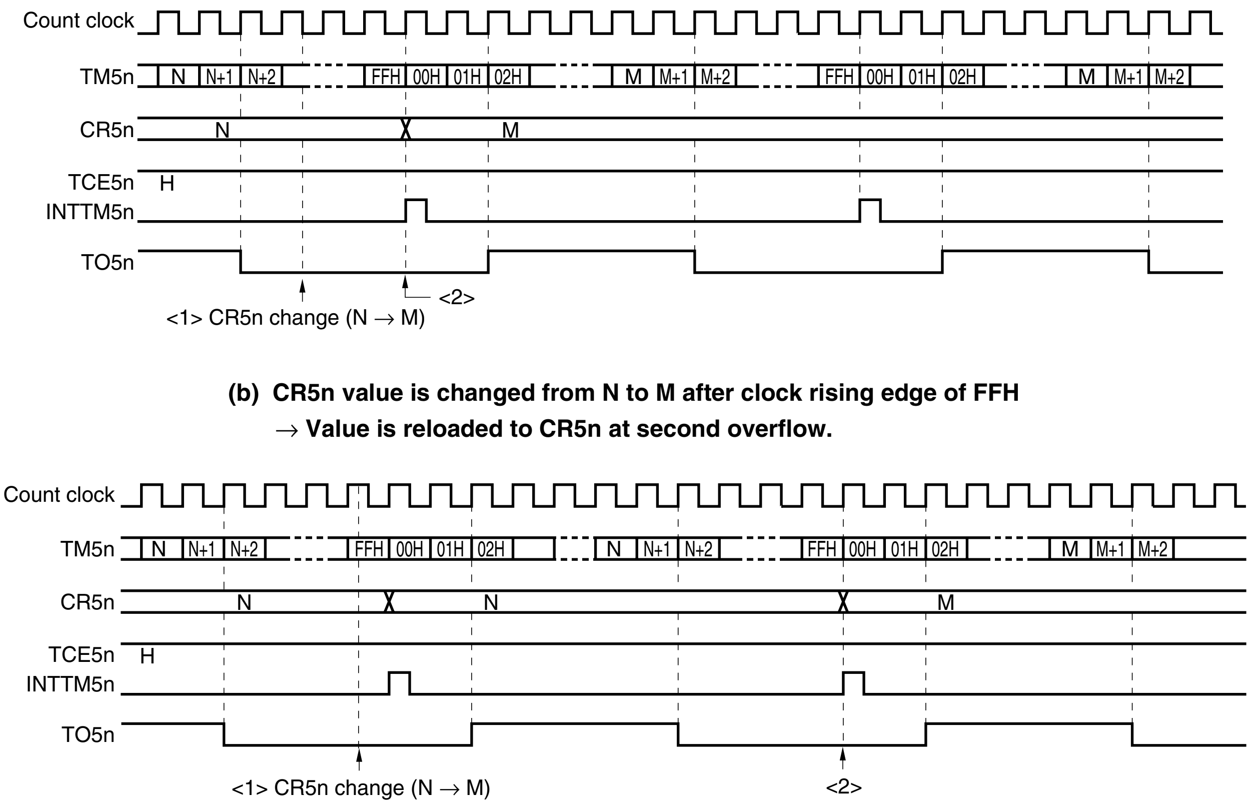
(b) CR5n = 00H

(c) CR5n = FFH

Remark n = 0, 1

(2) Operation with CR5n changed

Figure 8-13. Timing of Operation with CR5n Changed



(a) CR5n value is changed from N to M before clock rising edge of FFH
→ Value is reloaded to CR5n at overflow immediately after change.

(b) CR5n value is changed from N to M after clock rising edge of FFH
→ Value is reloaded to CR5n at second overflow.

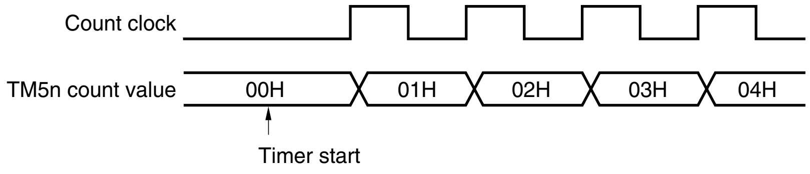
Caution When reading from CR5n between <1> and <2> in Figure 8-13, the value read differs from the actual value (read value: M, actual value of CR5n: N).

8.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counters 50 and 51 (TM50, TM51) are started asynchronously to the count clock.

Figure 8-14. 8-Bit Timer Counter 5n Start Timing



Remark n = 0, 1

CHAPTER 9 8-BIT TIMERS H0 AND H1

9.1 Functions of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 have the following functions.

- 8-bit-accuracy interval timer
- 8-bit-accuracy PWM pulse generator mode
- 8-bit-accuracy carrier generator mode (8-bit timer H1 only)

9.2 Configuration of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 consist of the following hardware.

Table 9-1. Configuration of 8-Bit Timers H0 and H1

Item	Configuration
Timer register	8-bit timer counter Hn (TMHn)
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	Two outputs (TOHn)
Control registers	8-bit timer H mode register n (TMHMDn) 8-bit timer H carrier control register 1 (TMCYC1)[Note]

Note 8-bit timer H1 only

Remark n = 0, 1

Figures 9-1 and 9-2 show the block diagrams.

Figure 9-1. Block Diagram of 8-Bit Timer H0

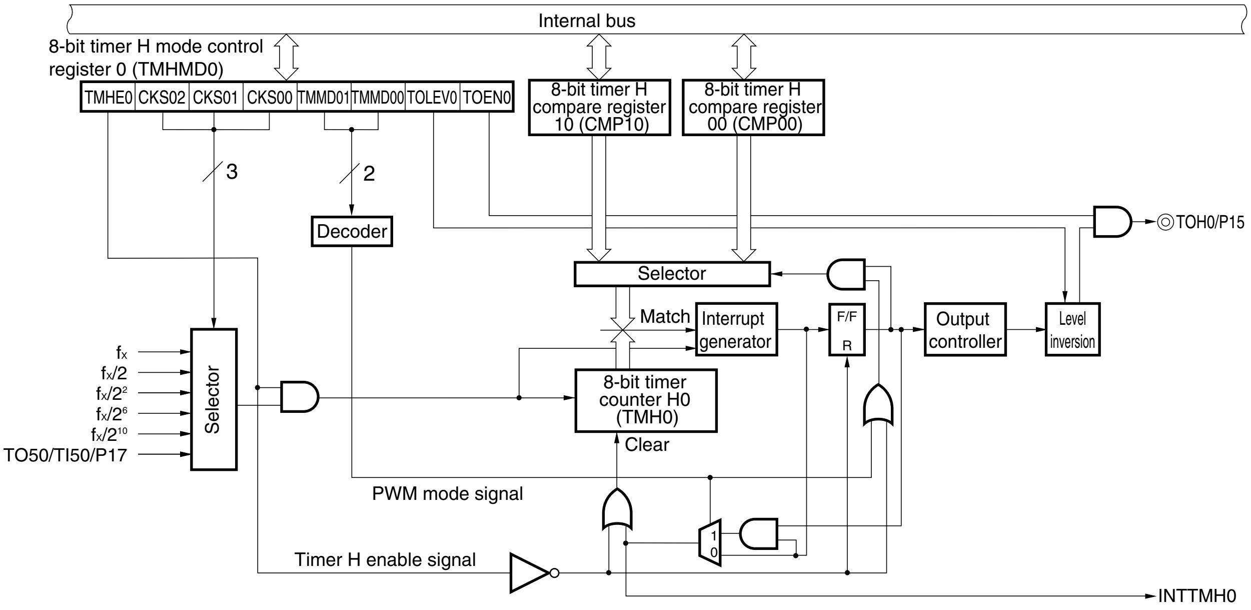
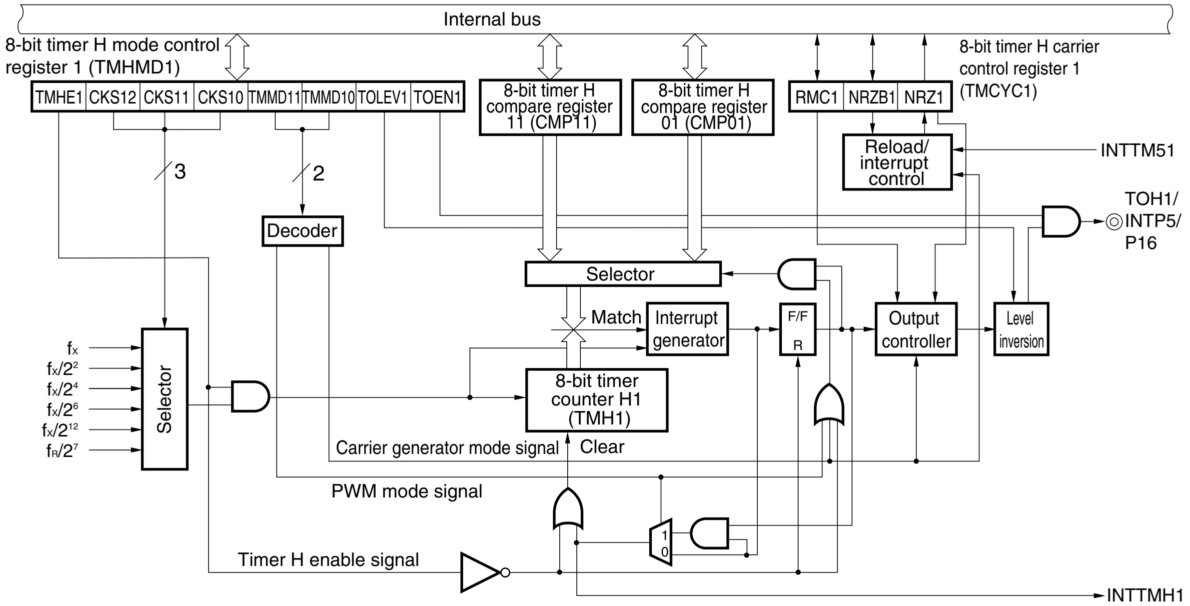


Figure 9-2. Block Diagram of 8-Bit Timer H1


Internal bus
8-bit timer H mode control register 1 (TMHMD1)
TMHE1
CKS12
CKS11
CKS10
TMMD11
TMMD10
TOLEV1
TOEN1
8-bit timer H compare register 11 (CMP11)
8-bit timer H compare register 01 (CMP01)
RMC1
NRZB1
NRZ1
8-bit timer H carrier control register 1 (TMCYC1)
Reload/ interrupt control
INTTM51
3
2
Decoder
TOH1/ INTP5/ P16
Selector
Match
Interrupt generator
F/F
R
Output controller
Level inversion
fx
fx/2²
fx/2⁴
fx/2⁶
fx/2¹²
fR/2⁷
Selector
8-bit timer counter H1 (TMH1)
Carrier generator mode signal
Clear
PWM mode signal
1
0
Timer H enable signal
INTTMH1

(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.

After reset: 00H R/W Address: FF18H, FF1AH

	7	6	5	4	3	2	1	0
CMP0n								

Caution This register 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.

After reset: 00H R/W Address: FF19H, FF1BH

	7	6	5	4	3	2	1	0
CMP1n								

The CMP1n register can be rewritten during timer count operation.

An interrupt request signal (INTTMHn) is generated if the values of the timer counter and CMP1n register match after setting the CMP1n register. The timer counter value is cleared at the same time. If the CMP1n register value is rewritten during timer operation, reloading is performed at the timing at which the counter value and CMP1n register value match. If the transfer timing and writing from CPU to CMP1n register conflict, transfer is not performed.

★ **Caution In the PWM pulse generator mode and carrier generator mode, be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).**

Remark n = 0, 1

9.3 Registers Controlling 8-Bit Timers H0 and H1

8-bit timers H0 and H1 are controlled by 8-bit timer H mode registers 0 and 1 (TMHMD0, TMHMD1) and 8-bit timer H carrier control register 1 (TMCYC1)[Note].

Note 8-bit timer H1 only

(1) 8-bit timer H mode registers 0 and 1 (TMHMD0, TMHMD1)

These registers control the mode of timer H.

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

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

Figure 9-3. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

	$\boxed{7}$	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation
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	TO50	
Other than above			Setting prohibited	

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM pulse generator 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

Cautions 1. When TMHE0 = 1, setting the other bits of the TMHMD0 register is prohibited.

★ **2. In the PWM pulse generator 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 the CMP10 register).**

Remarks 1. f_X: X1 input clock oscillation frequency

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

Figure 9-4. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	$\boxed{7}$	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stops timer count operation
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)
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 pulse generator mode
Other than above		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

Cautions 1. When TMHE1 = 1, setting the other bits of the TMHMD1 register is prohibited.

★ **2. In the PWM pulse generator 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).**

★ **3. When the carrier generator mode is used, set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.**

Remarks 1. f_X: X1 input clock oscillation frequency

2. f_R: Ring-OSC clock oscillation frequency

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

(2) 8-bit timer H carrier control register 1 (TMCYC1)

This register controls the remote control output and carrier pulse output status of 8-bit timer H1.

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

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

Figure 9-5. 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	$\boxed{0}$
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	Remote control output
0	0	Low-level output
0	1	High-level output
1	0	Low-level output
1	1	Carrier pulse output

NRZ1	Carrier pulse output status flag
0	Carrier output disabled status (low-level status)
1	Carrier output enabled status (RMC1 = 1: Carrier pulse output, RMC1 = 0: High-level status)

Note Bit 0 is read-only.

9.4 Operation of 8-Bit Timers H0 and H1

9.4.1 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.

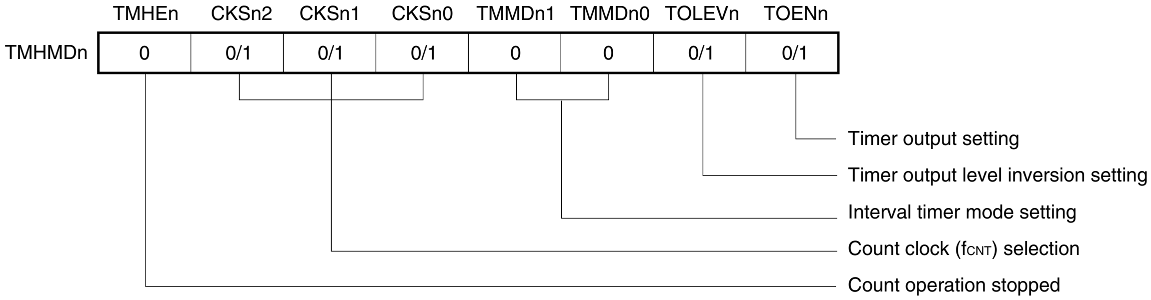
(1) Usage

Generates the INTTMHn signal repeatedly at the same interval.

<1> Set each register.

Figure 9-6. Register Setting in Interval Timer Mode

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



	TMHEn	CKSn2	CKSn1	CKSn0	TMMDn1	TMMDn0	TOLEVn	TOENn
TMHMDn	0	0/1	0/1	0/1	0	0	0/1	0/1

Timer output setting
Timer output level inversion setting
Interval timer mode setting
Count clock (f_{CNT}) selection
Count operation stopped

(ii) CMP0n register setting

- Compare value (N)

<2> Count operation starts when TMHEn = 1.

<3> When the values of 8-bit timer counter Hn and the CMP0n register match, the INTTMHn signal is generated and 8-bit timer counter Hn is cleared to 00H.

$$\boxed{\text{Interval timer} = (N+1)/f_{CNT}}$$

<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, set TMHEn to 0.

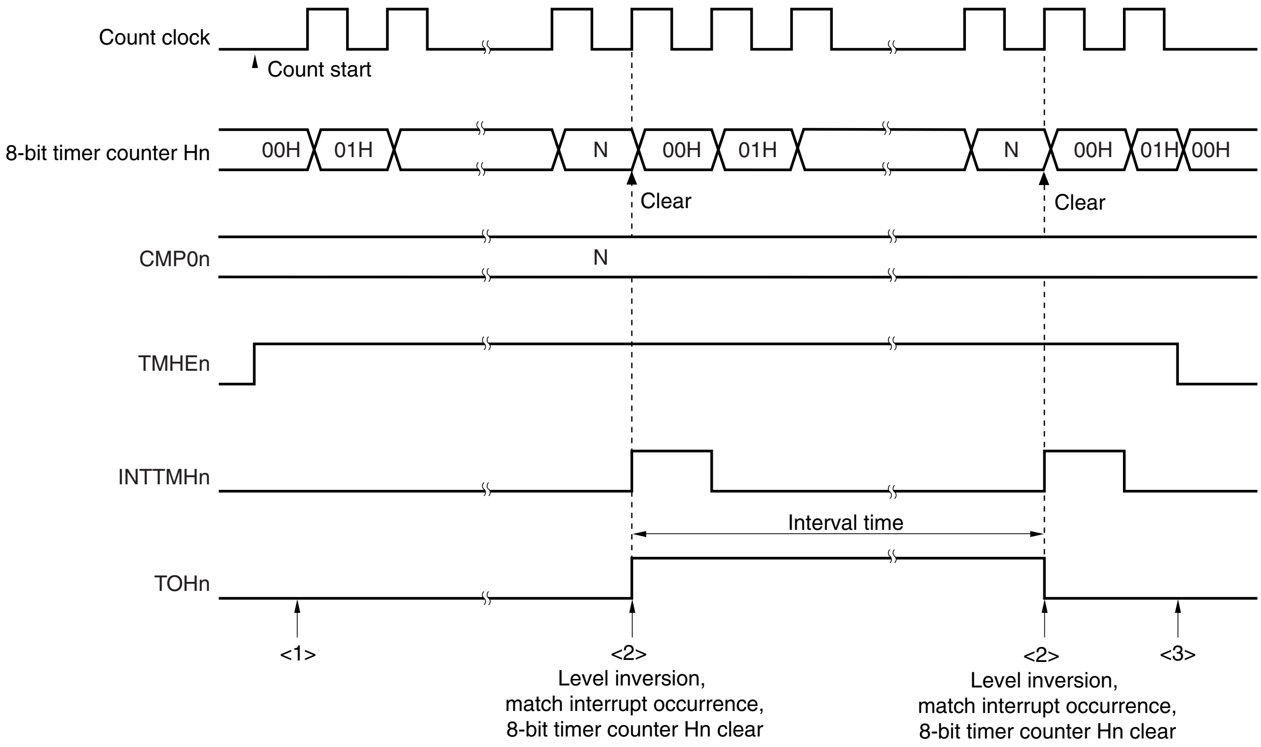
Remark n = 0, 1

(2) Timing chart

The timing in interval timer mode is shown below.

Figure 9-7. Timing of Interval Timer 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.

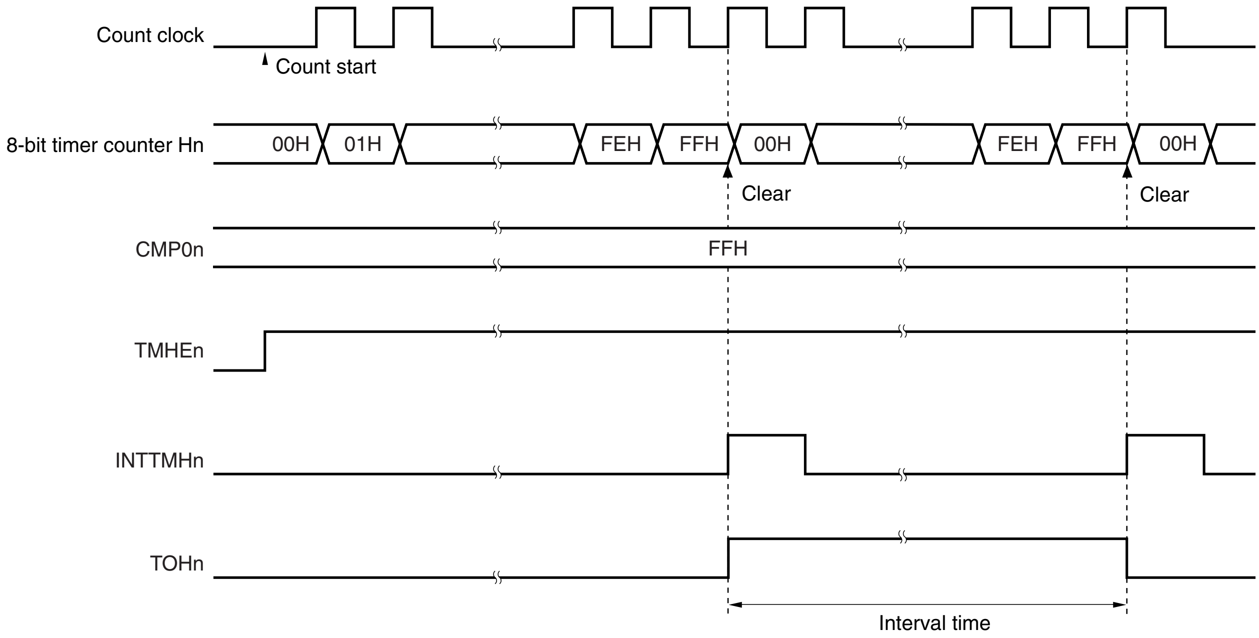
<2> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, the TOHn output level is inverted, and the INTTMHn signal is output.

<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.

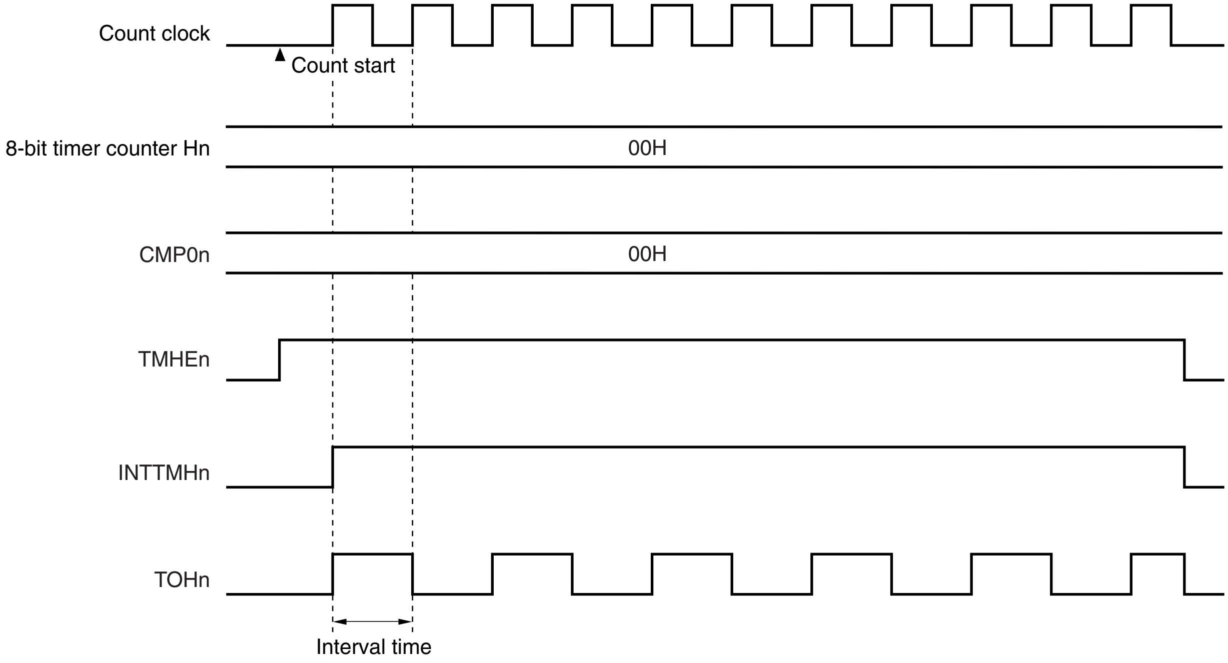
Remark n = 0, 1
N = 00H to FFH

Figure 9-7. Timing of Interval Timer Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



Remark n = 0, 1

9.4.2 Operation as PWM pulse generator

In PWM 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 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.

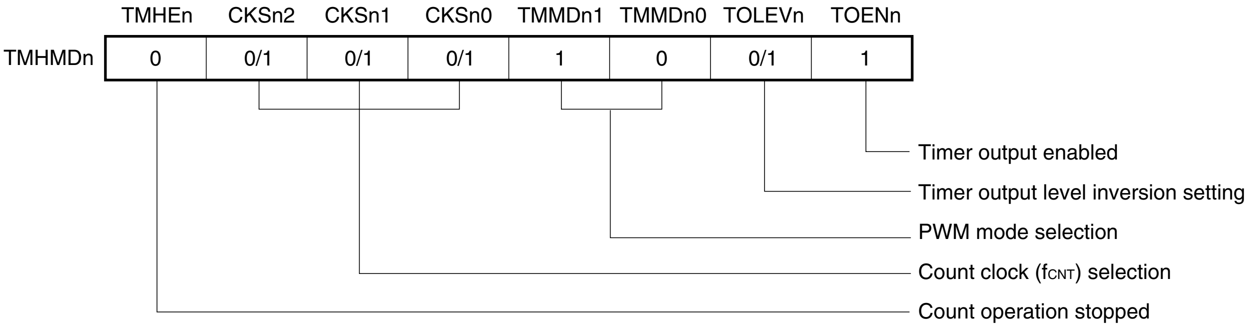
(1) Usage

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

<1> Set each register.

Figure 9-8. Register Setting in PWM Pulse Generator Mode

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



	TMHEn	CKSn2	CKSn1	CKSn0	TMMDn1	TMMDn0	TOLEVn	TOENn
TMHMDn	0	0/1	0/1	0/1	1	0	0/1	1

Timer output enabled
Timer output level inversion setting
PWM mode selection
Count clock (f_{CNT}) selection
Count operation stopped

(ii) Setting CMP0n register

- Compare value (N): Cycle setting

(iii) Setting CMP1n register

- Compare value (M): Duty setting

Remarks 1. n = 0, 1

2. 00H ≤ CMP1n (M) < CMP0n (N) < FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of 8-bit timer counter Hn and the CMP0n register match, 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, and TOHn output becomes active. At the same time, the compare register to be compared with 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

<4> When 8-bit timer counter Hn and the CMP1n register match, TOHn output becomes inactive and the compare register to be compared with 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty ratio 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 ratio are as follows.

PWM pulse output cycle = (N+1)/f_{CNT}
Duty ratio = Inactive width : Active width = (M + 1) : (N – M)

Cautions 1. In PWM 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

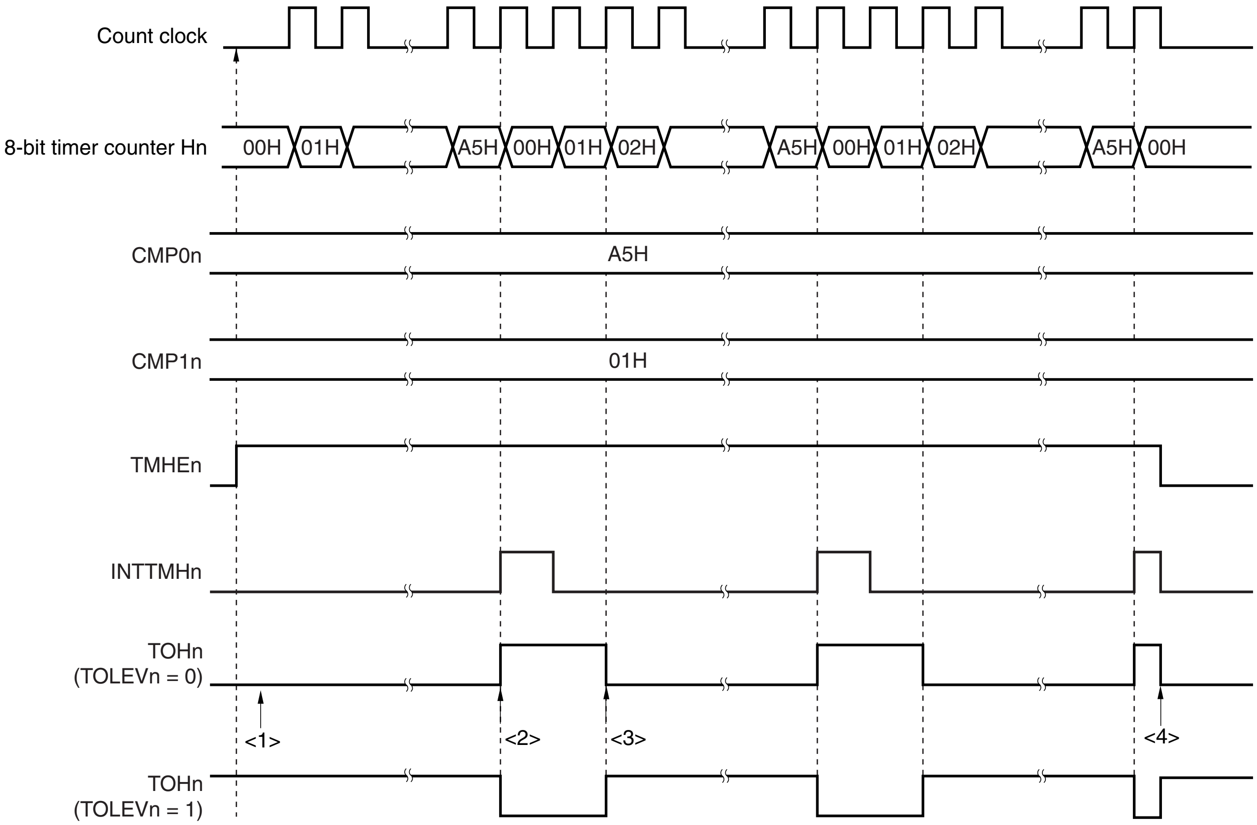
The operation timing in PWM mode is shown below.

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 9-9. Operation Timing in PWM Pulse Generator Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, TOHn output remains inactive (when TOLEVn = 0).

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

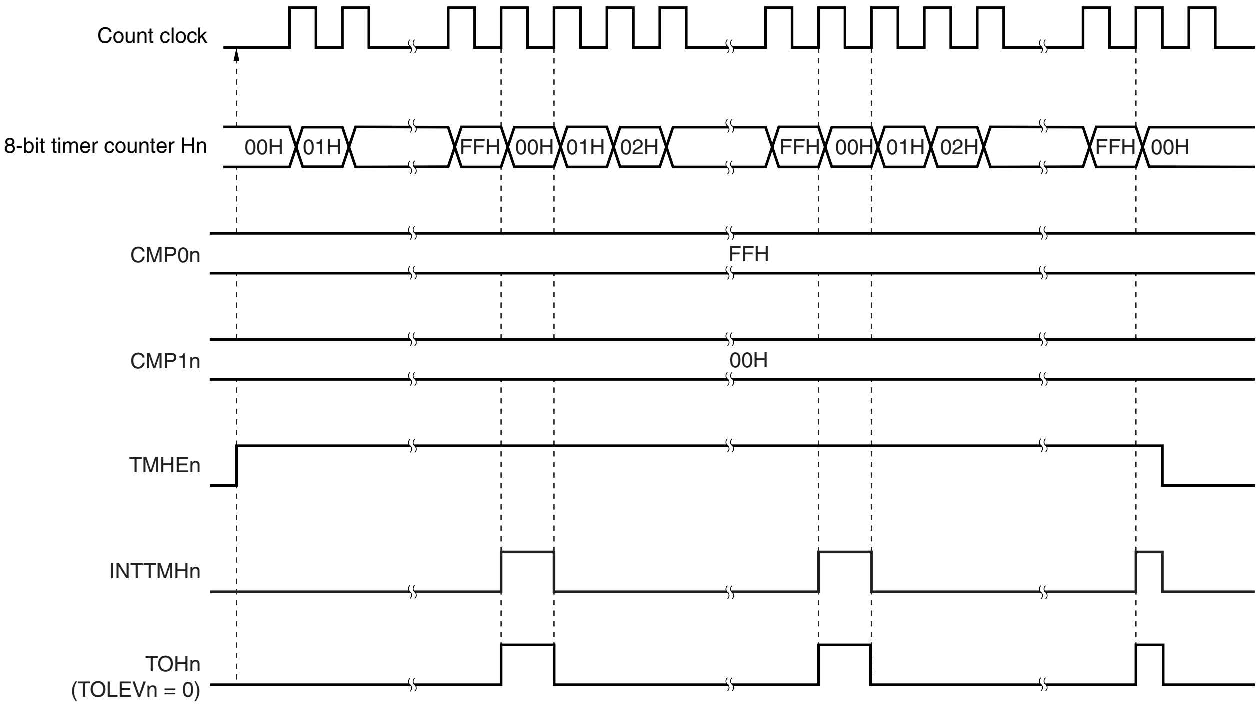
<3> When the values of 8-bit timer counter Hn and the CMP1n register match, the level of the TOHn output is returned. At this time, the 8-bit timer counter value is not cleared and the INTTMHn signal is not output.

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

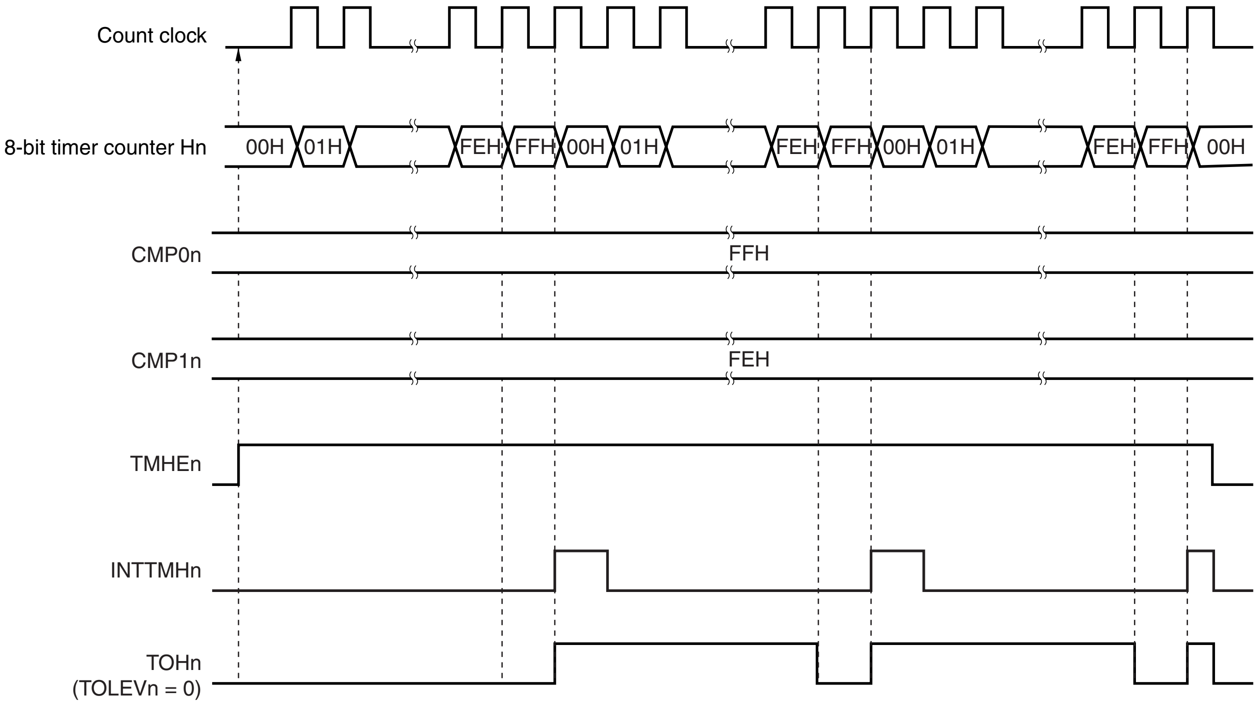
Remark n = 0, 1

Figure 9-9. Operation Timing in PWM Pulse Generator Mode (2/4)

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



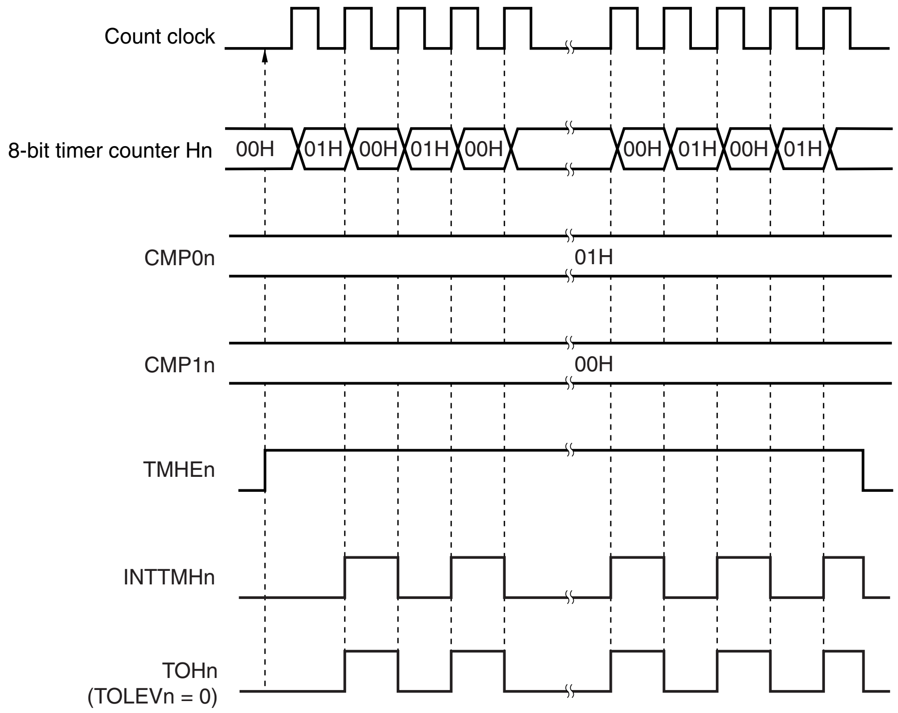
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0, 1

Figure 9-9. Operation Timing in PWM Pulse Generator Mode (3/4)

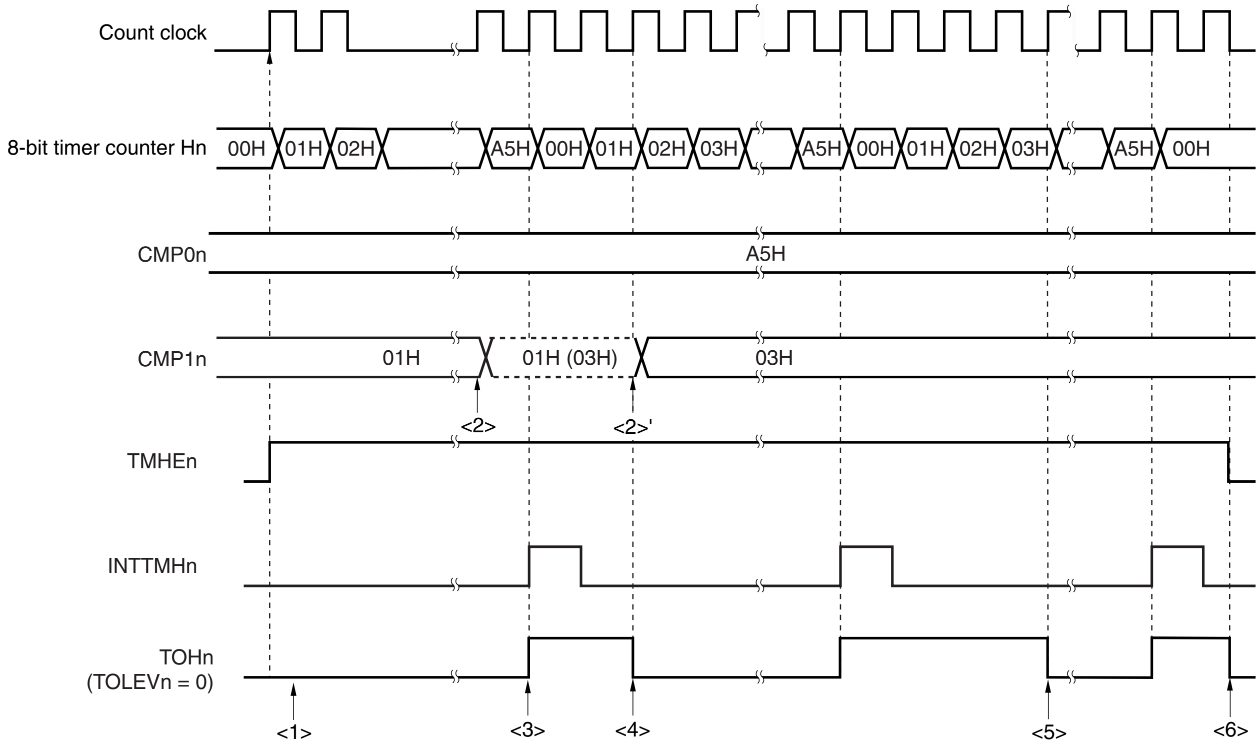
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0, 1

Figure 9-9. Operation Timing in PWM Pulse Generator 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

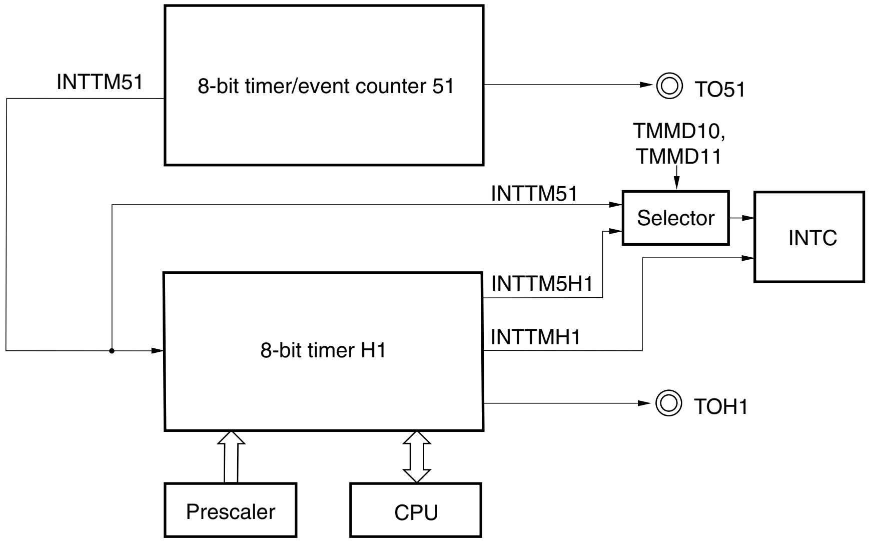
9.4.3 Carrier generator mode operation (8-bit timer H1 only)

The carrier clock generated by 8-bit timer H1 is output in the cycle set by 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

In carrier generator mode, the connection between 8-bit timer H1 and 8-bit timer/event counter 51 is as shown below.

Figure 9-10. Example of Connection Between 8-Bit Timer H1 and 8-Bit Timer/Event Counter 51



(1) Carrier generation

In carrier generator mode, 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

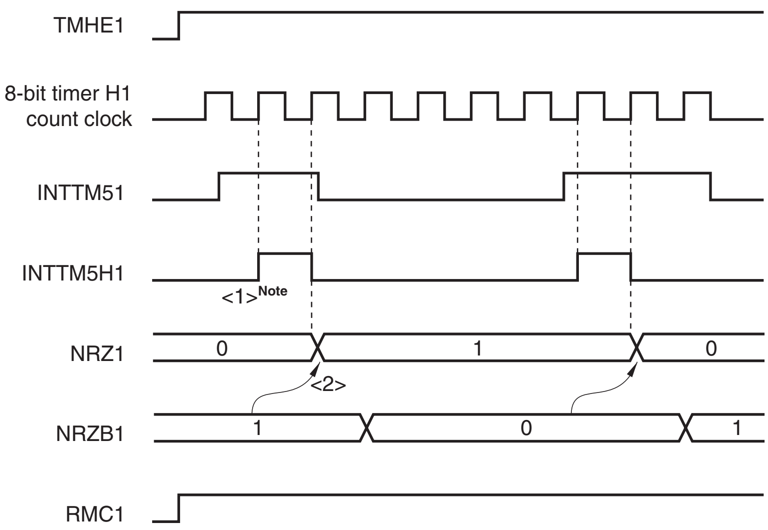
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of 8-bit timer/event counter 51 and the NRZ1 and RMC1 bits of the 8-bit timer H carrier control register (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZ1 Bit	Output
0	0	Low-level output
0	1	High-level output
1	0	Low-level output
1	1	Carrier pulse output

To control the carrier pulse output during a count operation, the NRZ1 and NRZB1 bits of the TMCYC1 register have a master and slave bit configuration. The NRZ1 bit is read-only but the NRZB1 bit can be read and written. The INTTM51 signal is synchronized with the 8-bit timer H1 count clock and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal of the NRZ1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit. The timing for transfer from the NRZB1 bit to the NRZ1 bit is as shown below.

Figure 9-11. Transfer Timing



<1> The INTTM51 signal is synchronized with the count clock of 8-bit timer H1 and is output as the INTTM5H1 signal.

<2> The value of the NRZB1 bit is transferred to the NRZ1 bit at the second clock from the rising edge of the INTTM5H1 signal.

Note When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When 8-bit timer/event counter 51 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

Caution 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.

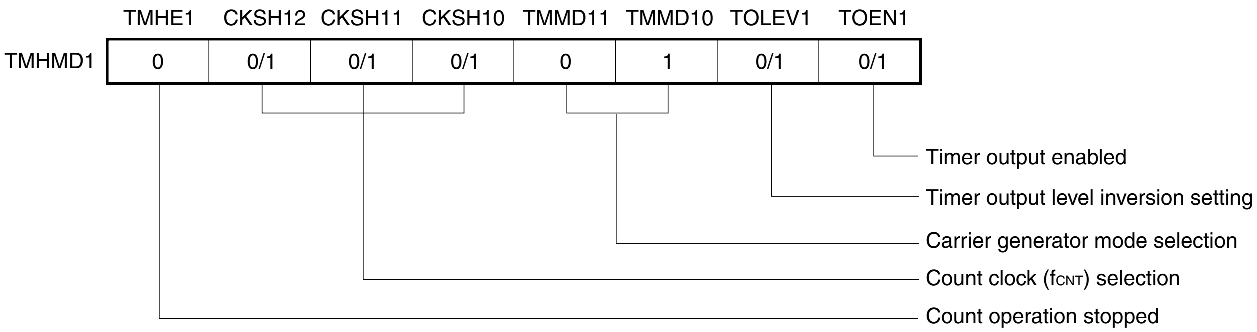
(3) Usage

Outputs an arbitrary carrier clock from the TOH1 pin.

<1> Set each register.

Figure 9-12. Register Setting in Carrier Generator Mode

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



	TMHE1	CKSH12	CKSH11	CKSH10	TMMD11	TMMD10	TOLEV1	TOEN1
TMHMD1	0	0/1	0/1	0/1	0	1	0/1	0/1

Timer output enabled
Timer output level inversion setting
Carrier generator mode selection
Count clock (f_{CNT}) selection
Count operation stopped

(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... carrier output enable bit

(v) TCL51 and TMC51 register setting

- Refer to **8.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51**.

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

<3> When TCE51 of 8-bit timer mode control register 51 (TMC51) is set to 1, 8-bit timer/event counter 51 starts counting.

<4> After the count operation is enabled, the first compare register to be compared is the CMP01 register. When the count value of 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared, and at the same time, the compare register to be compared with 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared, and at the same time, the compare register to be compared with 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register.

<6> By performing procedures <4> and <5> repeatedly, a carrier clock is generated.

<7> The INTTM51 signal is synchronized with 8-bit timer H1 and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<8> When the NRZ1 bit is high level, a carrier clock is output from the TOH1 pin.

<9> By performing the procedures above, an arbitrary carrier clock is obtained. To stop the count operation, set TMHE1 to 0.

If the setting value of the CMP01 register is 1, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the carrier clock output cycle and duty ratio are as follows.

Carrier clock output cycle = $(1 + M + 2)/f_{CNT}$
Duty ratio = High-level width : Low-level width = $(M + 1) : (1 + 1)$

★ **Cautions 1. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).**

★ **2. Set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.**

(4) Timing chart

The carrier output control timing is shown below.

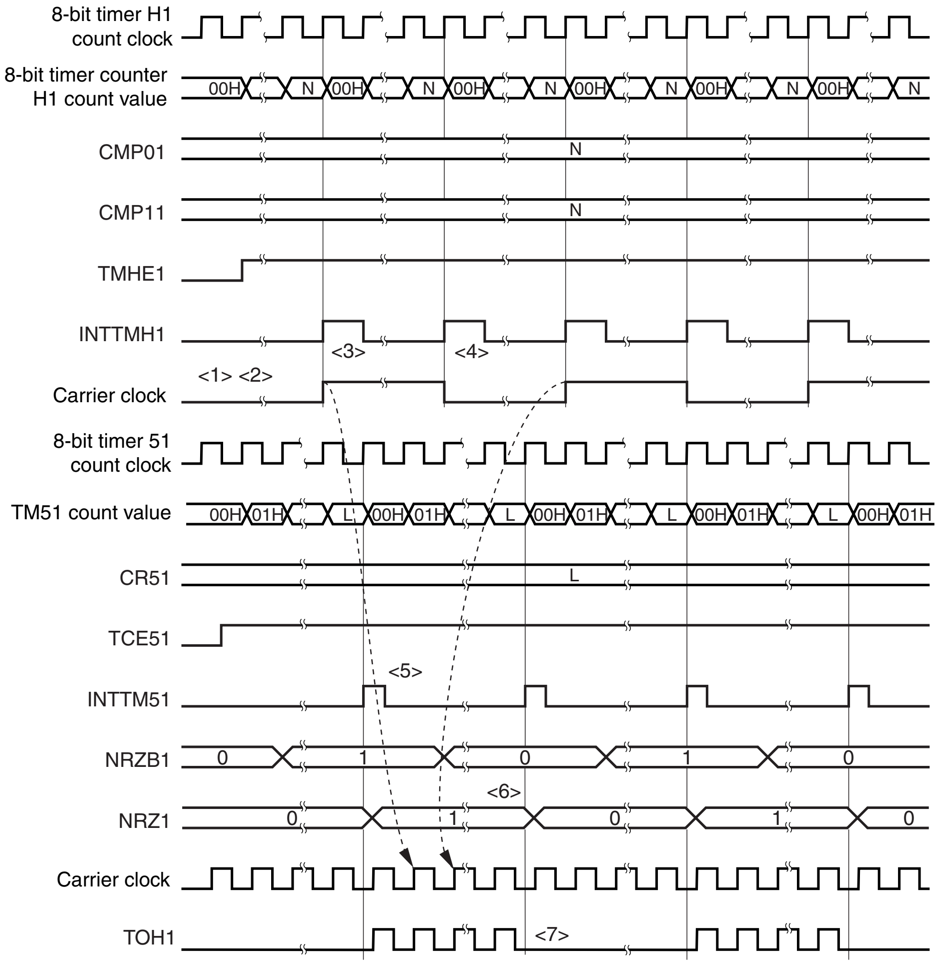
Cautions 1. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

2. In the carrier generator mode, three operating clocks (signal selected by CKSH12 to CKSH10 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 9-13. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = 1, CMP11 = 1



<1> When TMHE1 = 0 and TCE51 = 0, 8-bit timer H1 operation is stopped.

<2> When TMHE1 = 1 is set, 8-bit timer 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 ratio fixed to 50% is generated.

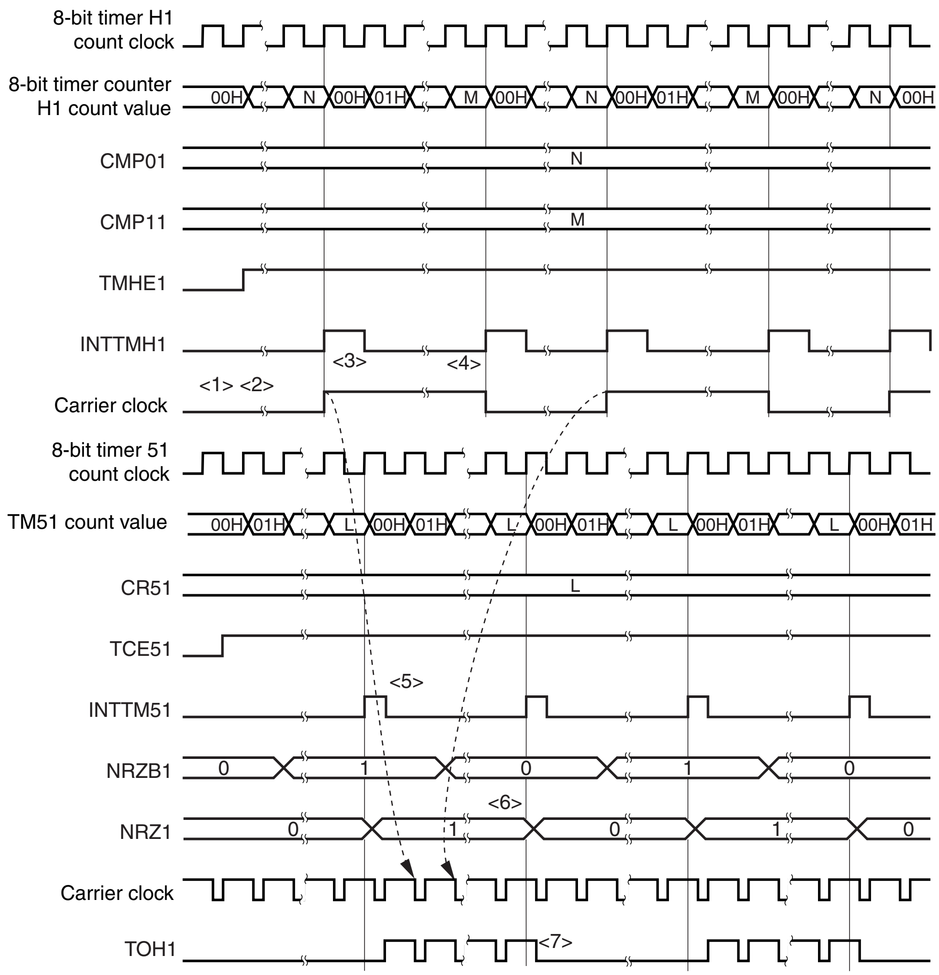
<5> When the INTTM51 signal is generated, it is synchronized with 8-bit timer H1 count clock and output as the INTTM5H1 signal.

<6> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<7> When NRZ1 = 0 is set, the TOH1 output becomes low level.

Figure 9-13. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = 1, CMP11 = M (operation when carrier clock phase is asynchronous to NRZ1 phase)



<1> When TMHE1 = 0 and TCE51 = 0, 8-bit timer H1 operation is stopped.

<2> When TMHE1 = 1 is set, 8-bit timer 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 ratio fixed to other than 50% is generated.

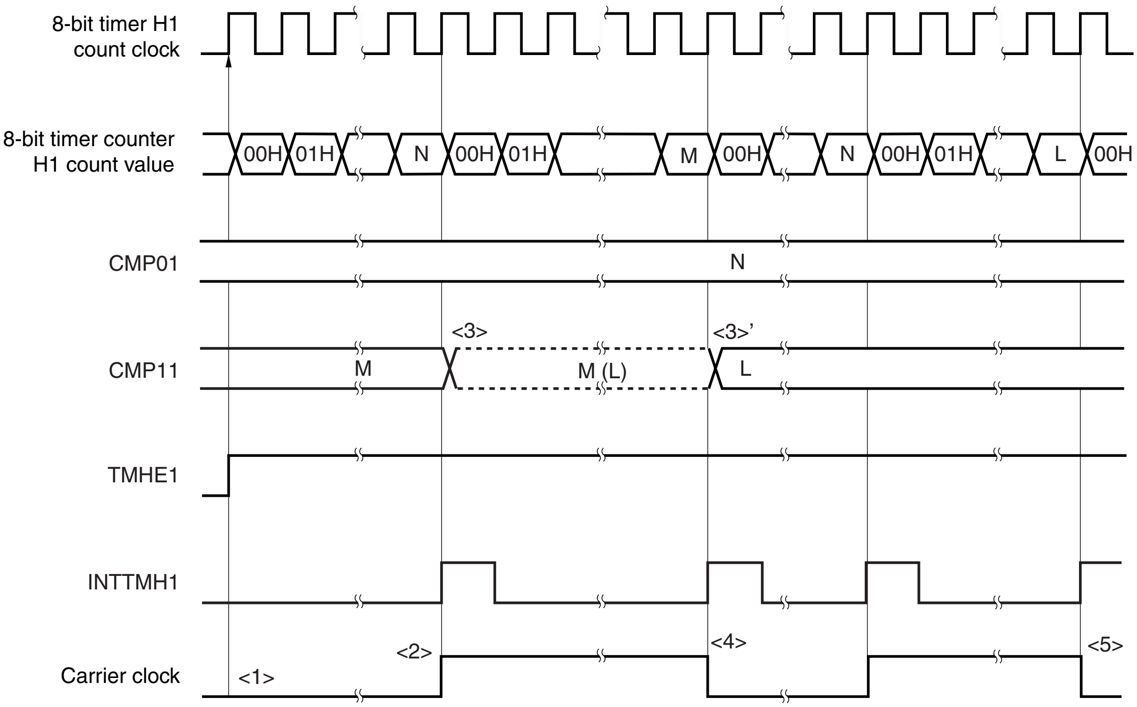
<5> When the INTTM51 signal is generated, it is synchronized with 8-bit timer H1 count clock and output as the INTTM5H1 signal.

<6> When the carrier clock phase becomes asynchronous to the NRZ1 bit phase, 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 9-13. Carrier Generator Mode Operation Timing (3/3)

(c) Operation when CMP11 is changed



<1> When TMHE1 = 1 is set, 8-bit timer 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).

10.1 Functions of Watch Timer

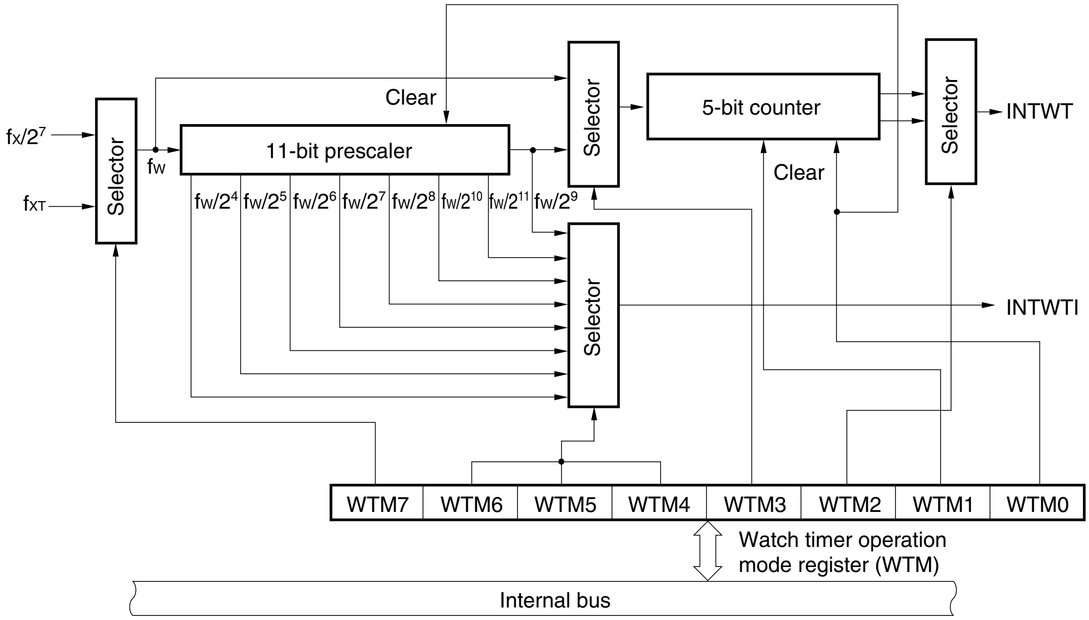
The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.

Figure 10-1 shows the watch timer block diagram.

Figure 10-1. Watch Timer Block Diagram



Remark fx: X1 input clock oscillation frequency
fxT: Subsystem clock oscillation frequency
fw: Watch timer clock frequency

(1) Watch timer

When the X1 input clock or subsystem clock is used, interrupt requests (INTWT) are generated at preset intervals.

Table 10-1. Watch Timer Interrupt Time

Interrupt Time	When Operated at f_{XT} = 32.768 kHz	When Operated at f_X = 10 MHz
$2^4/f_W$	488 μs	205 μs
$2^5/f_W$	977 μs	410 μs
$2^{13}/f_W$	0.25 s	0.105 s
$2^{14}/f_W$	0.5 s	0.210 s

Remark f_X: X1 input clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency
f_W: Watch timer clock frequency

(2) Interval timer

Interrupt requests (INTWTI) are generated at preset time intervals.

Table 10-2. Interval Timer Interval Time

Interval Time	When Operated at f_{XT} = 32.768 kHz	When Operated at f_X = 10 MHz
$2^4/f_W$	488 μs	205 μs
$2^5/f_W$	977 μs	410 μs
$2^6/f_W$	1.95 ms	820 μs
$2^7/f_W$	3.91 ms	1.64 ms
$2^8/f_W$	7.81 ms	3.28 ms
$2^9/f_W$	15.6 ms	6.55 ms
$2^{10}/f_W$	31.2 ms	13.1 ms
$2^{11}/f_W$	62.4 ms	26.2 ms

Remark f_X: X1 input clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency
f_W: Watch timer clock frequency

10.2 Configuration of Watch Timer

The watch timer consists of the following hardware.

Table 10-3. Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control register	Watch timer operation mode register (WTM)

10.3 Register Controlling Watch Timer

The watch timer is controlled by the watch timer operation mode register (WTM).

- **Watch timer operation mode register (WTM)**

  This register sets the watch timer count clock, enables/disables operation, prescaler interval time, and 5-bit counter operation control.
  WTM is set by a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input clears WTM to 00H.

Figure 10-2. Format of Watch Timer Operation Mode Register (WTM)

Address: FF6FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	Watch timer count clock selection
0	$f_X/2^7$ (78.125 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler interval time selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
1	1	0	$2^{10}/f_W$
1	1	1	$2^{11}/f_W$

WTM3	WTM2	Interrupt time selection
0	0	$2^{14}/f_W$
0	1	$2^{13}/f_W$
1	0	$2^5/f_W$
1	1	$2^4/f_W$

WTM1	5-bit counter operation control
0	Clear after operation stop
1	Start

WTM0	Watch timer operation enable
0	Operation stop (clear both prescaler and timer)
1	Operation enable

Caution Do not change the count clock and interval time (by setting bits 4 to 7 (WTM4 to WTM7) of WTM) during watch timer operation.

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$ or f_{XT})

2. f_X: X1 input clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

4. Figures in parentheses apply to operation with f_X = 10 MHz, f_{XT} = 32.768 kHz.

10.4 Watch Timer Operations

10.4.1 Watch timer operation

The watch timer generates an interrupt request (INTWT) at a specific time interval by using the X1 input clock or subsystem clock.

When bit 0 (WTM0) and bit 1 (WTM1) of the watch timer operation mode register (WTM) are set to 1, the count operation starts. When these bits are set to 0, the 5-bit counter is cleared and the count operation stops.

When the interval timer is simultaneously operated, zero-second start can be achieved only for the watch timer by setting WTM1 to 0. In this case, however, the 11-bit prescaler is not cleared. Therefore, an error up to $2^{11} \times 1/f_W$ seconds occurs in the first overflow (INTWT) after zero-second start.

The interrupt request is generated at the following time intervals.

Table 10-4. Watch Timer Interrupt Time

WTM3	WTM2	Interrupt Time Selection	When Operated at f_{XT} = 32.768 kHz (WTM7 = 1)	When Operated at f_X = 10 MHz (WTM7 = 0)
0	0	$2^{14}/f_W$	0.5 s	0.210 s
0	1	$2^{13}/f_W$	0.25 s	0.105 s
1	0	$2^{5}/f_W$	977 μs	410 μs
1	1	$2^{4}/f_W$	488 μs	205 μs

Remark f_X: X1 input clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

f_W: Watch timer clock frequency

10.4.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt requests (INTWTI) repeatedly at an interval of the preset count value.

The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer operation mode register (WTM).

When bit 0 (WTM0) of the WTM is set to 1, the count operation starts. When this bit is set to 0, the count operation stops.

Table 10-5. Interval Timer Interval Time

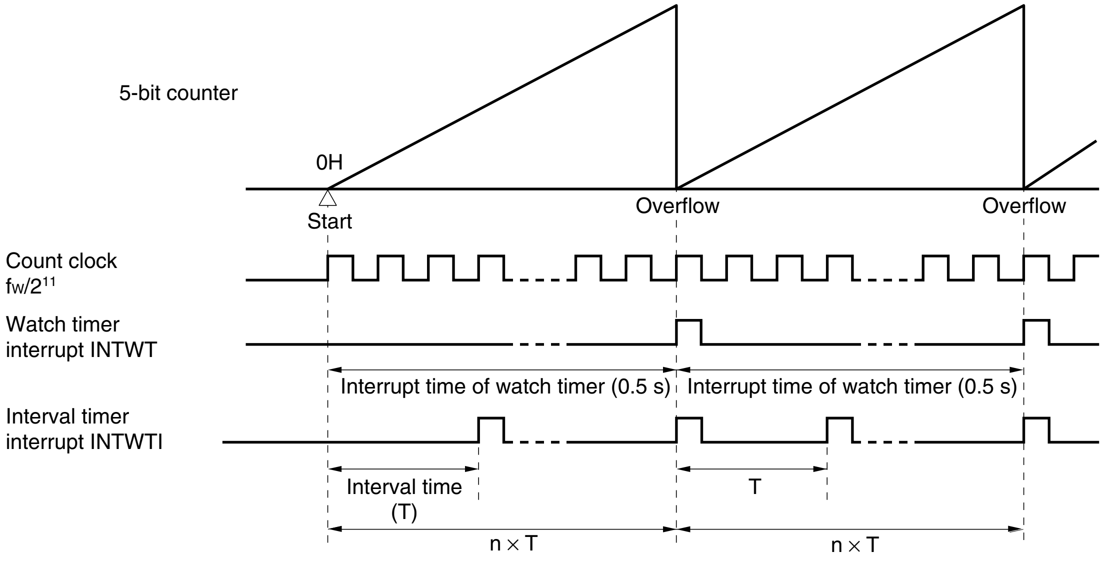
WTM6	WTM5	WTM4	Interval Time	When Operated at f_{XT} = 32.768 kHz (WTM7 = 1)	When Operated at f_X = 10 MHz (WTM7 = 0)
0	0	0	$2^4/f_W$	488 μs	205 μs
0	0	1	$2^5/f_W$	977 μs	410 μs
0	1	0	$2^6/f_W$	1.95 ms	820 μs
0	1	1	$2^7/f_W$	3.91 ms	1.64 ms
1	0	0	$2^8/f_W$	7.81 ms	3.28 ms
1	0	1	$2^9/f_W$	15.6 ms	6.55 ms
1	1	0	$2^{10}/f_W$	31.2 ms	13.1 ms
1	1	1	$2^{11}/f_W$	62.4 ms	26.2 ms

Remark f_X: X1 input clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

f_W: Watch timer clock frequency

Figure 10-3. Operation Timing of Watch Timer/Interval Timer



Caution **When operation of the watch timer and 5-bit counter is enabled by the watch timer mode control register (WTM) (by setting bits 0 (WTM0) and 1 (WTM1) of WTM to 1), the interval until the first interrupt request (INTWT) is generated after the register is set does not exactly match the specification made with bit 3 (WTM3) of WTM. This is because there is a delay of one 11-bit prescaler output cycle until the 5-bit counter starts counting. Subsequently, however, the INTWT signal is generated at the specified intervals.**

Remark fw: Watch timer clock frequency

n: The number of times of interval timer operations

Figures in parentheses are for operation with fw = 32.768 kHz (WTM7 = 1, WTM3, WTM2 = 0, 0)

CHAPTER 11 WATCHDOG TIMER

11.1 Functions of Watchdog Timer

The watchdog timer detects an inadvertent program loop. If a program loop is detected, an internal reset signal (WDTRES) 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 22 RESET FUNCTION**.

Table 11-1. Loop Detection Time of Watchdog Timer

Loop Detection Time	
During Ring-OSC Clock Operation	During X1 Input Clock Operation
$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 μs)
$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

Remarks 1. f_R: Ring-OSC clock oscillation frequency

2. f_{XP}: X1 input clock oscillation frequency

3. Figures in parentheses apply to operation at f_R = 240 kHz (TYP.), f_{XP} = 10 MHz

The operation mode of the watchdog timer (WDT) is switched according to the mask option setting of the on-chip Ring-OSC as shown in Table 11-2.

Table 11-2. Mask Option Setting and Watchdog Timer Operation Mode

	Mask Option	
	Ring-OSC Cannot Be Stopped	Ring-OSC Can Be Stopped by Software
Watchdog timer clock source	Fixed to f_R[Note 1].	• Selectable by software (f_{XP}, f_R or stopped) • When reset is released: f_R
Operation after reset	Operation starts with the maximum interval ($f_R/2^{18}$).	Operation starts with maximum interval ($f_R/2^{18}$).
Operation mode selection	The interval can be changed only once.	The clock selection/interval can be changed only once.
Features	• The watchdog timer cannot be stopped. • Current in STOP mode ≒ 10 μA	The watchdog timer can be stopped in standby mode[Note 2].

Notes 1. As long as power is being supplied, Ring-OSC oscillation cannot be stopped (except in the reset period).

2. Clock supply to the watchdog timer is stopped in accordance with the watchdog timer clock source as follows:

<1> When the clock source is f_{XP}

Clock supply to the watchdog timer is stopped while f_{XP} is stopped, during HALT/STOP instruction execution, and during the oscillation stabilization time.

<2> When the clock source is f_R

Clock supply to the watchdog timer is stopped if f_R is stopped by software before STOP instruction execution when the CPU clock is f_{XP} and during HALT/STOP instruction execution.

Remarks 1. f_R: Ring-OSC clock oscillation frequency

2. f_{XP}: X1 input clock oscillation frequency

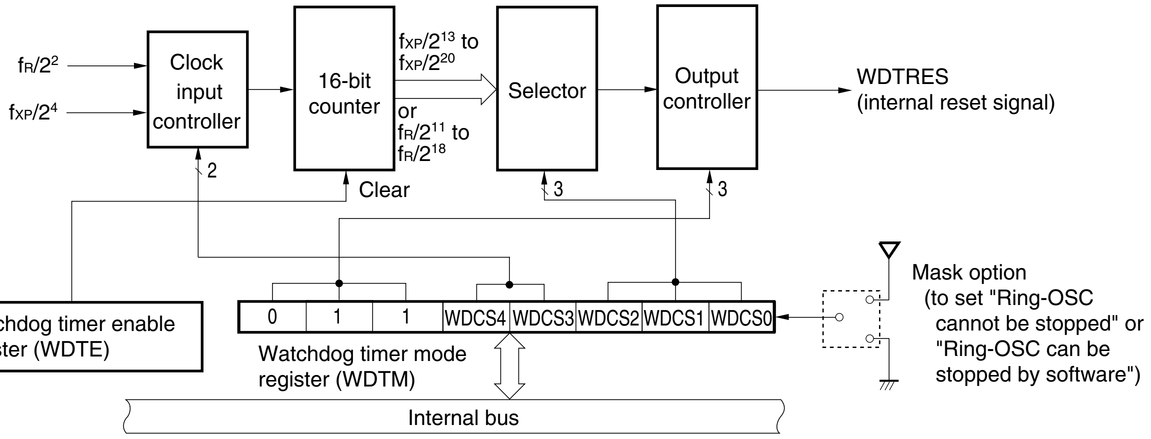
11.2 Configuration of Watchdog Timer

The watchdog timer consists of following hardware.

Table 11-3. Configuration of Watchdog Timer

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

Figure 11-1. Block Diagram of Watchdog Timer



11.3 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 11-2. Format of Watchdog Timer Mode Register (WDTM)

Address: FF98H After reset: 67H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	0	1	1	WDCS4	WDCS3	WDCS2	WDCS1	WDCS0

WDCS4[Note 1]	WDCS3[Note 1]	Operation clock selection
0	0	Ring-OSC clock (f_R)
0	1	X1 input clock (f_{XP})
1	×	Watchdog timer operation stopped

WDCS2[Note 2]	WDCS1[Note 2]	WDCS0[Note 2]	Overflow time setting	
			During Ring-OSC clock operation	During X1 input clock operation
0	0	0	$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 μs)
0	0	1	$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
0	1	0	$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
0	1	1	$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
1	0	0	$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
1	0	1	$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
1	1	0	$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
1	1	1	$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

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

2. Reset is released at the maximum cycle (WDCS2, 1, 0 = 1, 1, 1).

Cautions 1. If data is written to WDTM, a wait cycle is generated. Do not write data to WDTM when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

2. Set bits 7, 6, and 5 to 0, 1, and 1, respectively (when “Ring-OSC cannot be stopped” is selected by 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.

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

Remarks 1. f_R: Ring-OSC clock oscillation frequency

2. f_{XP}: X1 input clock oscillation frequency

3. ×: Don’t care

4. Figures in parentheses apply to operation at f_R = 240 kHz (TYP.), f_{XP} = 10 MHz

(2) Watchdog timer enable register (WDTE)

Writing ACH to WDTE clears the watchdog timer counter and starts counting again.

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

$\overline{\text{RESET}}$ input sets this register to 9AH.

Figure 11-3. 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.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated (an error occurs in the assembler).

3. The value read from WDTE is 9AH (this differs from the written value (ACH)).

11.4 Operation of Watchdog Timer

11.4.1 Watchdog timer operation when "Ring-OSC cannot be stopped" is selected by mask option

The operation clock of watchdog timer is fixed to the Ring-OSC.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1). The watchdog timer operation cannot be stopped.

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Ring-OSC clock
   - Cycle: $f_R/2^{18}$ (1.09 seconds: At operation with f_R = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2].
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

Notes 1. The operation clock (Ring-OSC clock) cannot be changed. If any value is written to bits 3 and 4 (WDCS3, WDCS4) of WDTM, it is ignored.

2. As soon as WDTM is written, the counter of the watchdog timer is cleared.

Caution In this mode, operation of the watchdog timer absolutely cannot be stopped even during STOP instruction execution. For 8-bit timer H1 (TMH1), a division of the Ring-OSC can be selected as the count source, so clear the watchdog timer using the interrupt request of TMH1 before the watchdog timer overflows. If this processing is not performed, an internal reset signal is generated when the watchdog timer overflows after STOP instruction execution.

11.4.2 Watchdog timer operation when "Ring-OSC can be stopped by software" is selected by mask option

The operation clock of the watchdog timer can be selected as either the Ring-OSC clock or the X1 input clock.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1).

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Ring-OSC clock oscillation frequency (f_R)
   - Cycle: $f_R/2^{18}$ (1.09 seconds: At operation with f_R = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2, 3].
   - Operation clock: Any of the following can be selected using bits 3 and 4 (WDCS3 and WDCS4).
     Ring-OSC clock (f_R)
     X1 input clock (f_{XP})
     Watchdog timer operation stopped
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

Notes 1. As soon as WDTM is written, the counter of the watchdog timer is cleared.

2. Set bits 7, 6, and 5 to 0, 1, 1, respectively. If other values are set, the watchdog timer cannot be operated (an error occurs in the assembler).

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.

For the watchdog timer operation during STOP mode and HALT mode in each status, refer to 11.4.3 Watchdog timer operation in STOP mode and 11.4.4 Watchdog timer operation in HALT mode.

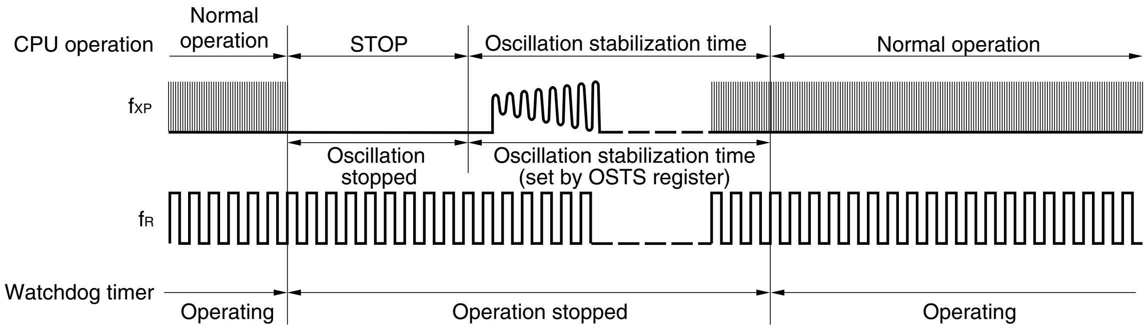
11.4.3 Watchdog timer operation in STOP mode (when “Ring-OSC can be stopped by software” is selected by mask option)

The watchdog timer stops counting during STOP instruction execution regardless of whether the X1 input clock or Ring-OSC clock is being used.

(1) When the CPU clock and the watchdog timer operation clock are the X1 input clock (f_{XP}) when the STOP instruction is executed

When 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.

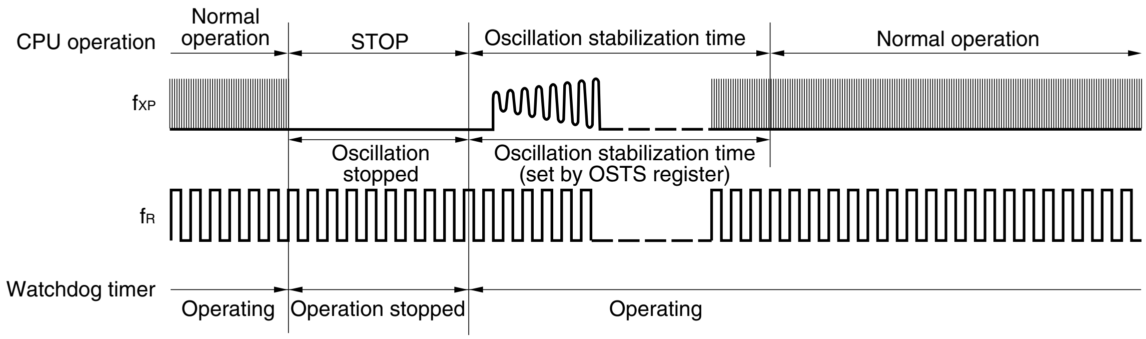
Figure 11-4. Operation in STOP Mode (CPU Clock and WDT Operation Clock: X1 Input Clock)



(2) When the CPU clock is the X1 input clock (f_{XP}) and the watchdog timer operation clock is the Ring-OSC clock (f_R) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

Figure 11-5. Operation in STOP Mode (CPU Clock: X1 Input Clock, WDT Operation Clock: Ring-OSC Clock)



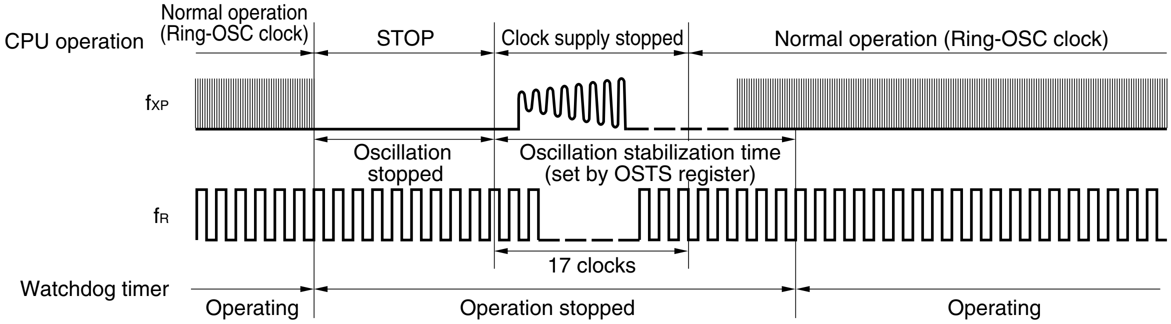
(3) When the CPU clock is the Ring-OSC clock (f_R) and the watchdog timer operation clock is the X1 input clock (f_{XP}) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is stopped until the timing of <1> or <2>, whichever is earlier, and then counting is started using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

<1> The oscillation stabilization time set by the oscillation stabilization time select register (OSTS) elapses.
<2> The CPU clock is switched to the X1 input clock (f_{XP}).

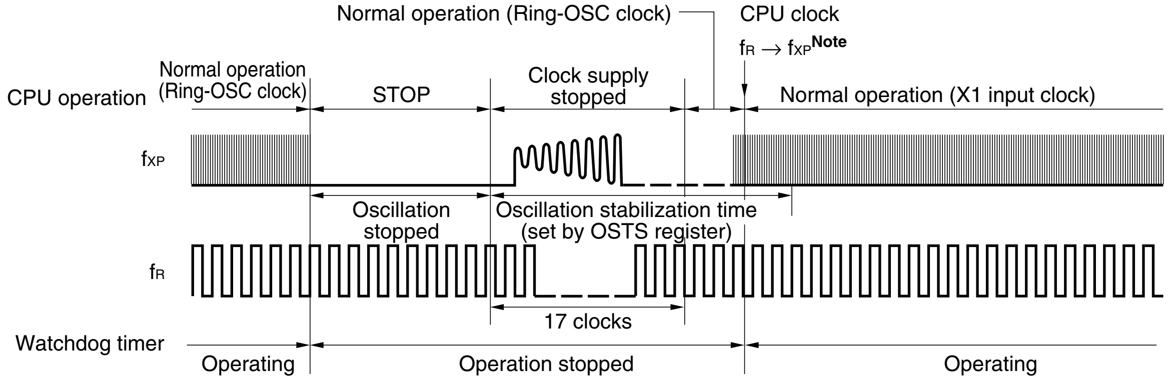
Figure 11-6. Operation in STOP Mode (CPU Clock: Ring-OSC Clock, WDT Operation Clock: X1 Input Clock)

<1> Timing when counting is started after the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) has elapsed



Normal operation (Ring-OSC clock)
CPU operation
STOP
Clock supply stopped
Normal operation (Ring-OSC clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

<2> Timing when counting is started after the CPU clock is switched to the X1 input clock (f_{XP})



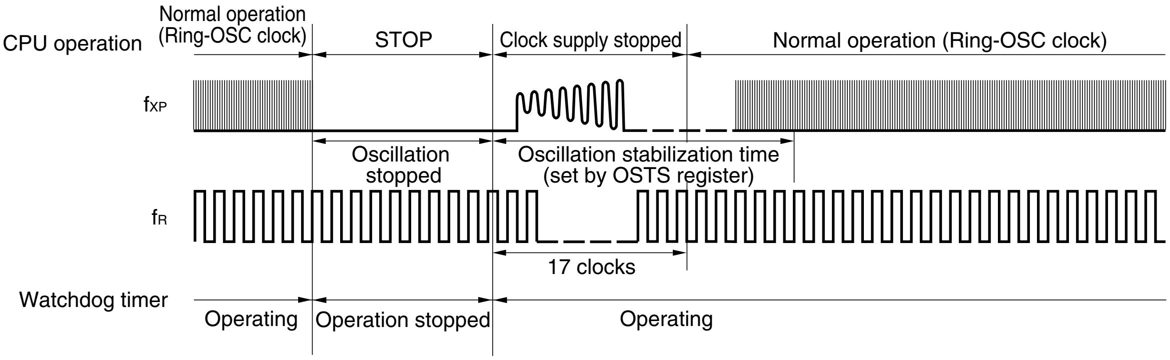
Normal operation (Ring-OSC clock)
CPU clock
fR → fXP Note
Normal operation (Ring-OSC clock)
CPU operation
STOP
Clock supply stopped
Normal operation (X1 input clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

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

(4) When CPU clock and watchdog timer operation clock are the Ring-OSC clocks (f_R) during STOP instruction execution

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.

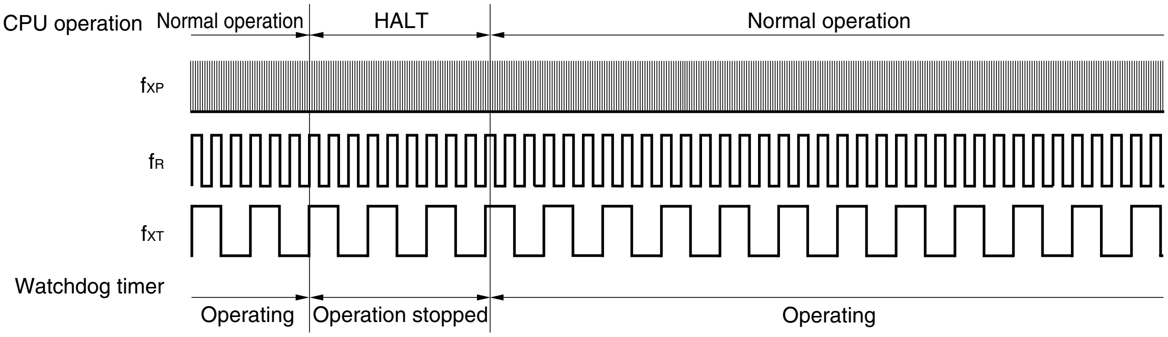
Figure 11-7. Operation in STOP Mode (CPU Clock and WDT Operation Clock: Ring-OSC Clock)


CPU operation
Normal operation (Ring-OSC clock)
STOP
Clock supply stopped
Normal operation (Ring-OSC clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

11.4.4 Watchdog timer operation in HALT mode (when “Ring-OSC 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 X1 input clock (f_{XP}), Ring-OSC clock (f_R), or subsystem clock (f_{XT}), or whether the operation clock of the watchdog timer is the X1 input clock (f_{XP}) or Ring-OSC clock (f_R). After HALT mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

Figure 11-8. Operation in HALT Mode



CHAPTER 12 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

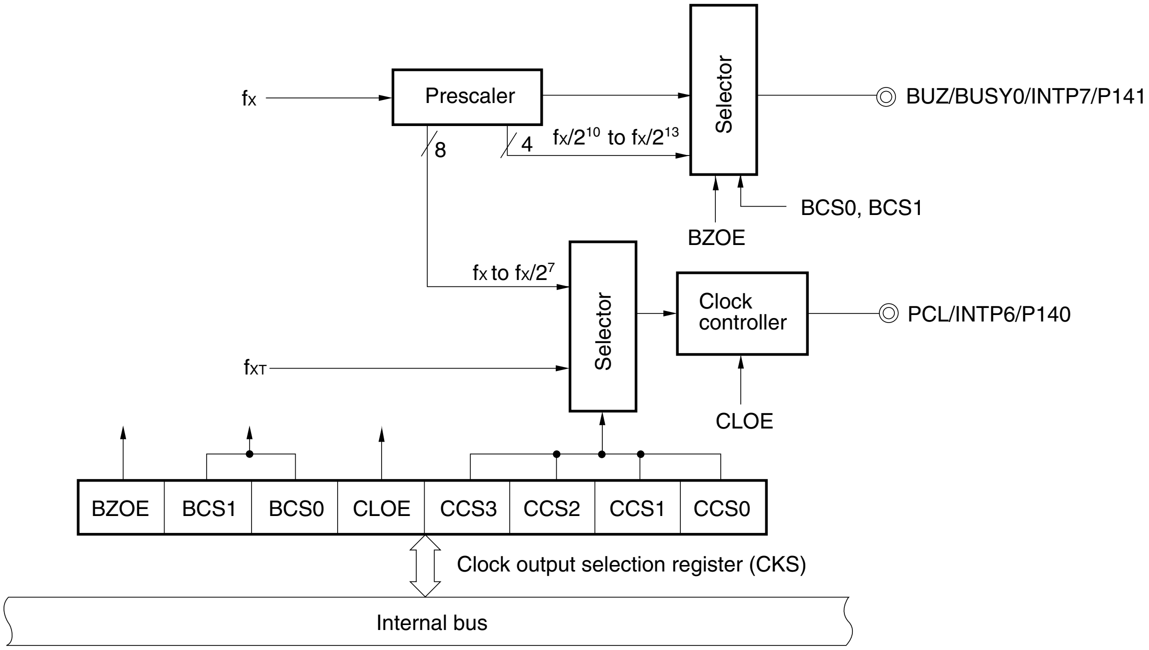
12.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

In addition, the buzzer output is intended for square-wave output of buzzer frequency selected with CKS.

Figure 12-1 shows the block diagram of clock output/buzzer output controller.

Figure 12-1. Block Diagram of Clock Output/Buzzer Output Controller



12.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller consists of the following hardware.

Table 12-1. Clock Output/Buzzer Output Controller Configuration

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 14 (PM14)[Note]

Note See **Figure 4-23 Block Diagram of P140 and P141**.

12.3 Register Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output selection register (CKS)
- Port mode register 14 (PM14)

(1) Clock output selection register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

CKS is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CKS to 00H.

Figure 12-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	$\boxed{4}$	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Clock division circuit operation stopped. BUZ fixed to low level.
1	Clock division circuit operation enabled. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection
0	0	$f_X/2^{10}$ (9.77 kHz)
0	1	$f_X/2^{11}$ (4.88 kHz)
1	0	$f_X/2^{12}$ (2.44 kHz)
1	1	$f_X/2^{13}$ (1.22 kHz)

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	0	f_X (10 MHz)
0	0	0	1	$f_X/2$ (5 MHz)
0	0	1	0	$f_X/2^2$ (2.5 MHz)
0	0	1	1	$f_X/2^3$ (1.25 MHz)
0	1	0	0	$f_X/2^4$ (625 kHz)
0	1	0	1	$f_X/2^5$ (312.5 kHz)
0	1	1	0	$f_X/2^6$ (156.25 kHz)
0	1	1	1	$f_X/2^7$ (78.125 kHz)
1	0	0	0	f_{XT} (32.768 kHz)
Other than above				Setting prohibited

Remarks 1. f_X: X1 input clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 10 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register 14 (PM14)

This register sets port 14 input/output in 1-bit units.

When using the P140/INTP6/PCL pin for clock output and the P141/BUSY0/INTP7/BUZ pin for buzzer output, set PM140, PM141 and the output latch of P140, P141 to 0.

PM14 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM14 to FFH.

Figure 12-3. Format of Port Mode Register 14 (PM14)

Address: FF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PM14n	P14n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

12.4 Clock Output/Buzzer Output Controller Operations

12.4.1 Clock output operation

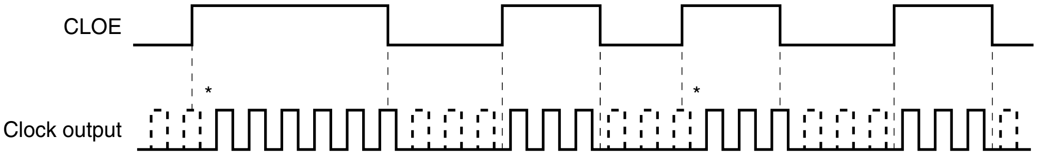
The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).

<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 12-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after securing high level of the clock.

Figure 12-4. Remote Control Output Application Example



12.4.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output selection register (CKS) (buzzer output in disabled status).

<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

CHAPTER 13 A/D CONVERTER

13.1 Functions of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of up to eight channels (ANI0 to ANI7) with a resolution of 10 bits.

The A/D converter has the following two functions.

(1) 10-bit resolution A/D conversion

10-bit resolution A/D conversion is carried out repeatedly for one channel selected from analog inputs ANI0 to ANI7. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

(2) Power-fail detection function

This function is used to detect a voltage drop in a battery. The A/D conversion result (ADCR register value) and power-fail comparison threshold register (PFT) value are compared. INTAD is generated only when a comparative condition has been matched.

Figure 13-1. Block Diagram of A/D Converter

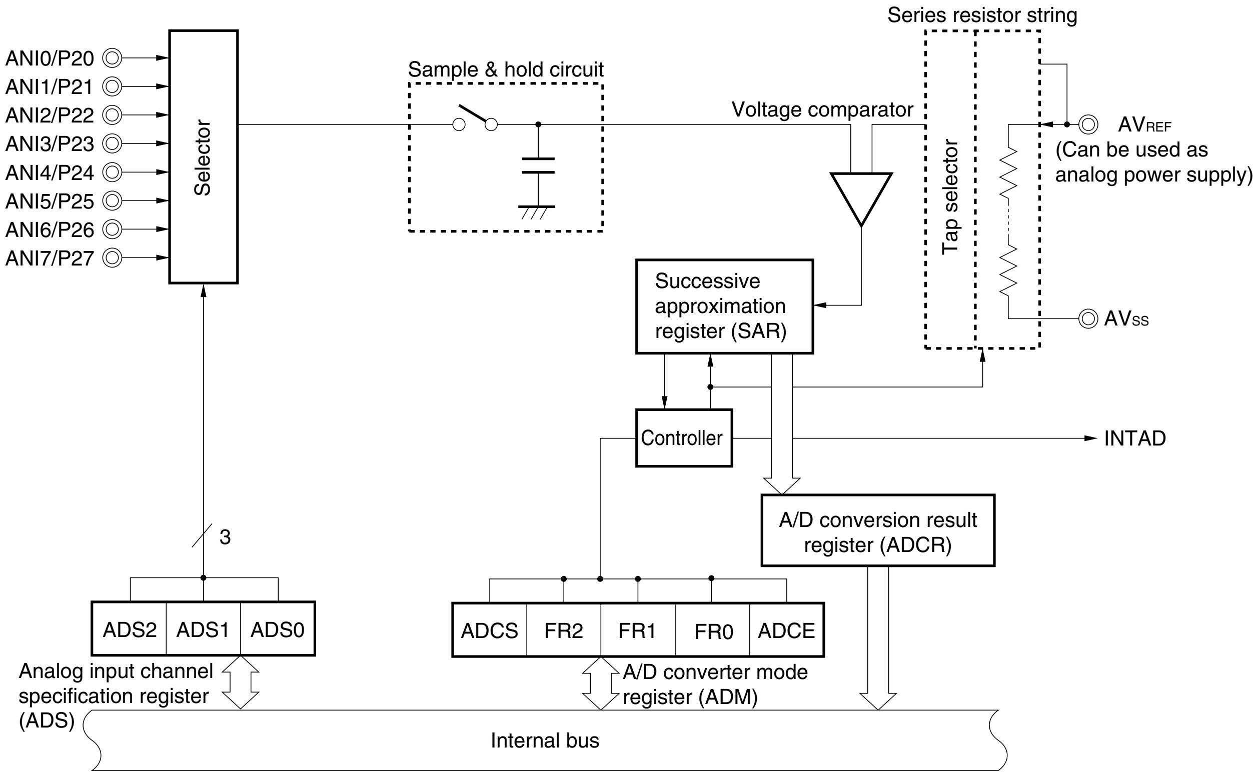
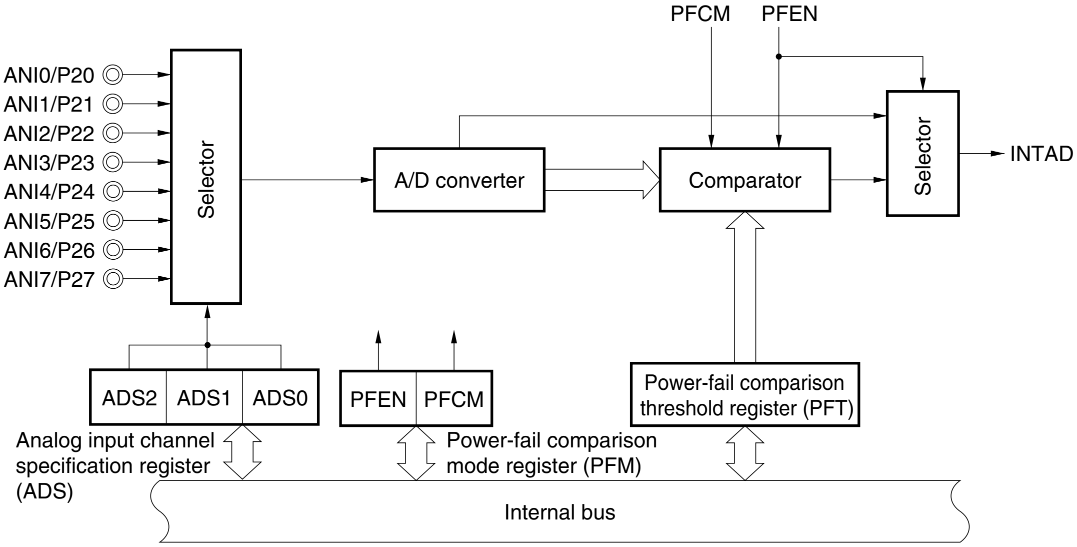


Figure 13-2. Block Diagram of Power-Fail Detection Function



13.2 Configuration of A/D Converter

The A/D converter consists of the following hardware.

Table 13-1. Configuration of A/D Converter

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR)
Control registers	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) Successive approximation register (SAR)

This register compares the analog input voltage value with the voltage tap (compare voltage) value applied from the series resistor string, and holds the result starting from the most significant bit (MSB).

When the result up to the least significant bit (LSB) is held (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

(2) A/D conversion result register (ADCR)

The ADCR is 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).

ADCR can be read by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR undefined.

Figure 13-3. Format of A/D Conversion Register (ADCR)

Address: FF08H, FF09H After reset: Undefined R

Symbol	FF09H								FF08H							
ADCR											0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCR, a wait cycle is generated. Do not read data from ADCR when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input with the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV$_{REF}$ and AV$_{SS}$, and generates a voltage to be compared with the analog input.

(6) ANI0 to ANI7 pins

These eight-channel analog input pins input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 are alternate-function pins that can also be used for digital input.

Cautions 1. Observe the rated range of the ANI0 to ANI7 input voltage. If a voltage of AV$_{REF}$ or higher or a voltage of AV$_{SS}$ or lower (even if within 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.

2. The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27). When A/D conversion is performed with any of ANI0 to ANI7 selected, do not execute the input instruction to port 2 while conversion is in progress; otherwise the conversion resolution may be degraded. 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.

(7) AV$_{REF}$ pin

The AV$_{REF}$ pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI7 into digital signals based on a voltage between AV$_{REF}$ and AV$_{SS}$.

In a standby mode, the current flowing into series resistor strings can be reduced by changing the input voltage of the AV$_{REF}$ pin to AV$_{SS}$ level.

It can also be used as the analog power supply. When the A/D converter is used, be sure to use the AV$_{REF}$ pin for the power supply.

Caution A series resistor string of several tens of kΩ is connected between the AV$_{REF}$ and AV$_{SS}$ pins. Therefore, if the output impedance of the reference voltage source is high, this will result in series connection to the series resistor string between the AV$_{REF}$ and AV$_{SS}$ pins, resulting in a large reference voltage error.

(8) AV$_{SS}$ pin

The AV$_{SS}$ pin is the GND potential pin for the A/D converter. Always use the AV$_{SS}$ pin at the same potential as the V$_{SS0}$ pin, even when the A/D converter is not used.

13.3 Registers Controlling A/D Converter

The following four registers are used to control the A/D converter.

- 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) 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 13-4. 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[Note 1]	34.3 μs	28.8 μs
0	0	1	240/f_X	120 μs[Note 1]	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[Note 1]
1	1	0	96/f_X	48 μs	11.5 μs[Note 1]	9.6 μs[Note 1]
Other			Setting prohibited			

ADCE	Boost reference voltage generator operation control[Note 2]
0	Stops operation of reference voltage generator
1	Enables operation of reference voltage generator

Notes 1. Set so that the A/D conversion time is 14 μs or longer but less than 100 μs.

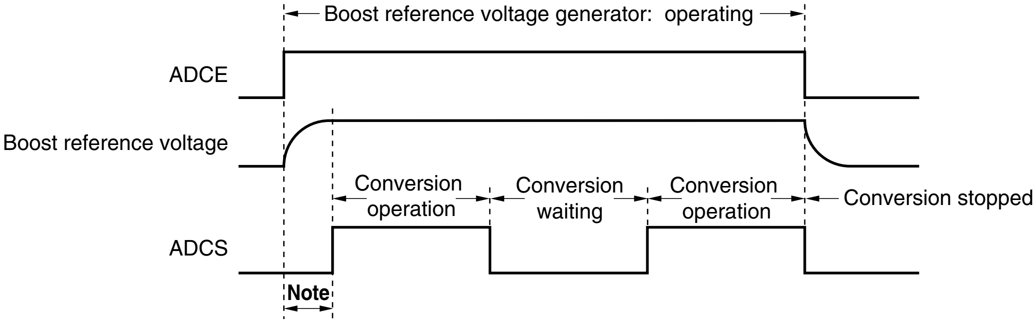
2. A booster circuit is incorporated to realize low-voltage operation. The operation of the circuit that generates the reference voltage for boosting is controlled by ADCE, and it takes 14 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 14 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result.

Table 13-2. 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 13-5. Timing Chart When Boost Reference Voltage Generator Is Used



Note 14 µs or more is required for reference voltage stabilization.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR2 to values other than the identical data.

2. For the sampling time of the A/D converter and the A/D conversion start delay time, refer to (11) in 13.6 Cautions for A/D Converter.

3. If data is written to ADM, a wait cycle is generated. Do not write data to ADM when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

Remark fx: X1 input clock oscillation frequency

(2) Analog input channel specification register (ADS)

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

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

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

Figure 13-6. Format of Analog Input Channel Specification Register (ADS)

Address: FF29H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

Cautions 1. Be sure to set bits 3 to 7 of ADS to 0.

2. If data is written to ADS, a wait cycle is generated. Do not write data to ADS when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

(3) Power-fail comparison mode register (PFM)

The power-fail comparison mode register (PFM) is a register that controls the comparison operation.

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

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

Figure 13-7. Format of Power-Fail Comparison Mode Register (PFM)

Address: FF2AH After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	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	ADCR3 ≥ PFT3	Interrupt request signal (INTAD) generation
	ADCR3 < PFT3	No INTAD generation
1	ADCR3 ≥ PFT3	No INTAD generation
	ADCR3 < PFT3	INTAD generation

Caution If data is written to PFM, a wait cycle is generated. Do not write data to PFM when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

(4) 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 13-8. Format of Power-Fail Comparison Threshold Register (PFT)

Address: FF2BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFT	PFT7	PFT6	PFT5	PFT4	PFT3	PFT2	PFT1	PFT0

Caution If data is written to PFT, a wait cycle is generated. Do not write data to PFT when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

13.4 A/D Converter Operations

13.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion with analog input channel specification register (ADS).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> 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 is ended.

<4> 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.

<5> 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.

<6> 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) V_{DD}
- Bit 9 = 0: (1/4) V_{DD}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

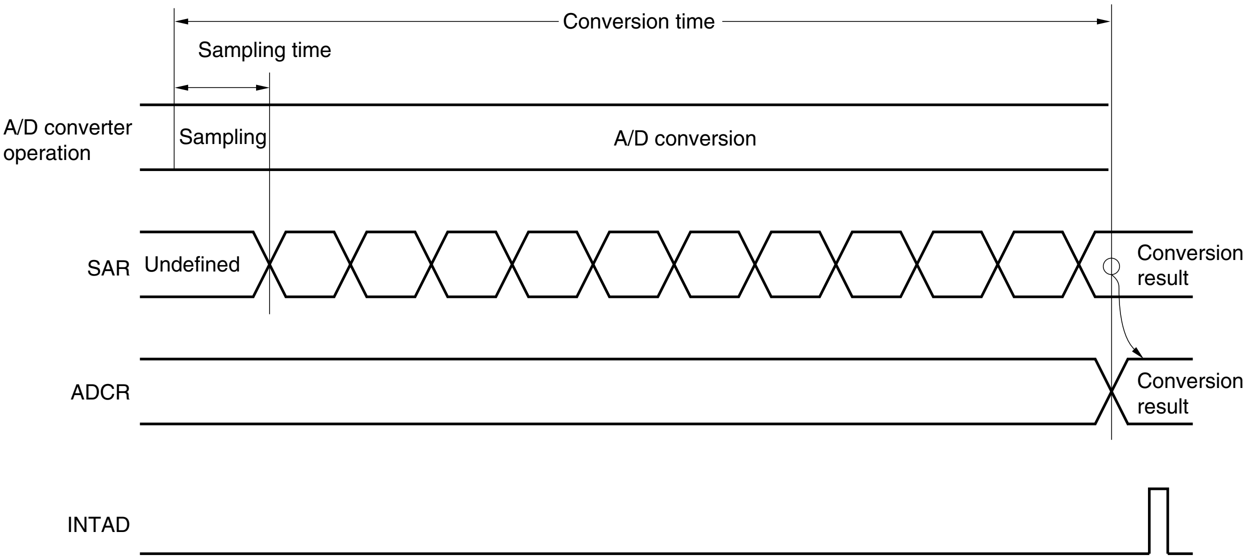
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<7> Comparison is continued in this way up to bit 0 of SAR.

<8> 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.

Caution The first A/D conversion value immediately after A/D conversion operations start may not fall within the rating.

Figure 13-9. 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.

13.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (stored in the A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{ADCR} = \mathrm{INT}\ (\frac{V_{\mathrm{IN}}}{\mathrm{AV}_{\mathrm{REF}}} \times 1024 + 0.5)$$

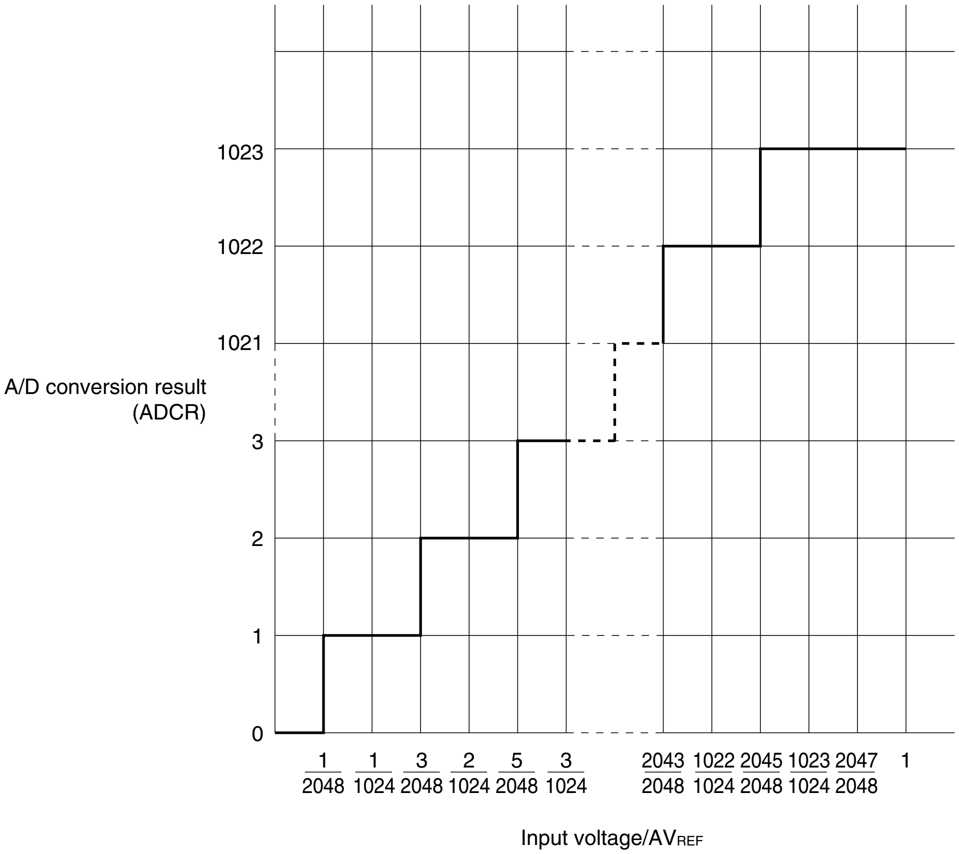
or

$$(\mathrm{ADCR} - 0.5) \times \frac{\mathrm{AV}_{\mathrm{REF}}}{1024} - V_{\mathrm{IN}} < (\mathrm{ADCR} + 0.5) \times \frac{\mathrm{AV}_{\mathrm{REF}}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{IN}: Analog input voltage
$\mathrm{AV}_{\mathrm{REF}}$: $\mathrm{AV}_{\mathrm{REF}}$ pin voltage
ADCR: A/D conversion result register (ADCR) value

Figure 13-10 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 13-10. Relationship Between Analog Input Voltage and A/D Conversion Result



13.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI7 by the analog input channel specification register (ADS) and A/D conversion is executed.

In addition, the following two functions can be selected by setting 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)

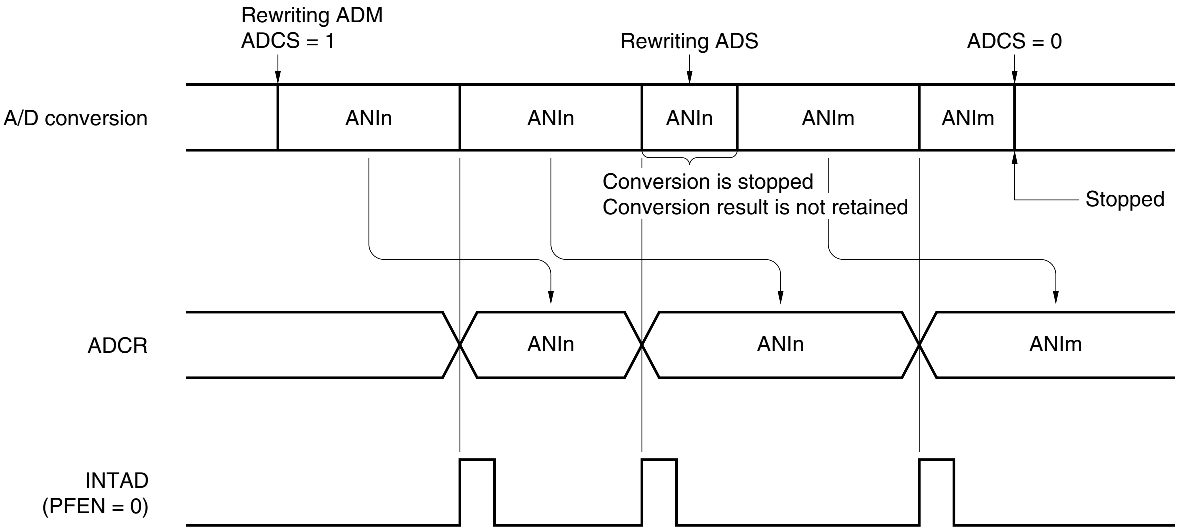
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 0, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. Once the A/D conversion has started and when one A/D conversion has been completed, the next A/D conversion operation is immediately started. The A/D conversion operations are repeated until new data is written to ADS.

If ADS is rewritten during A/D conversion, the A/D conversion under execution is suspended, and the A/D conversion of the newly selected analog input channel is started.

If 0 is written to ADCS of ADM during A/D conversion, the conversion operation is immediately stopped.

Figure 13-11. A/D Conversion Operation



Remarks 1. n = 0 to 7

2. m = 0 to 7

(2) Power-fail detection function (when PFEN = 1)

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

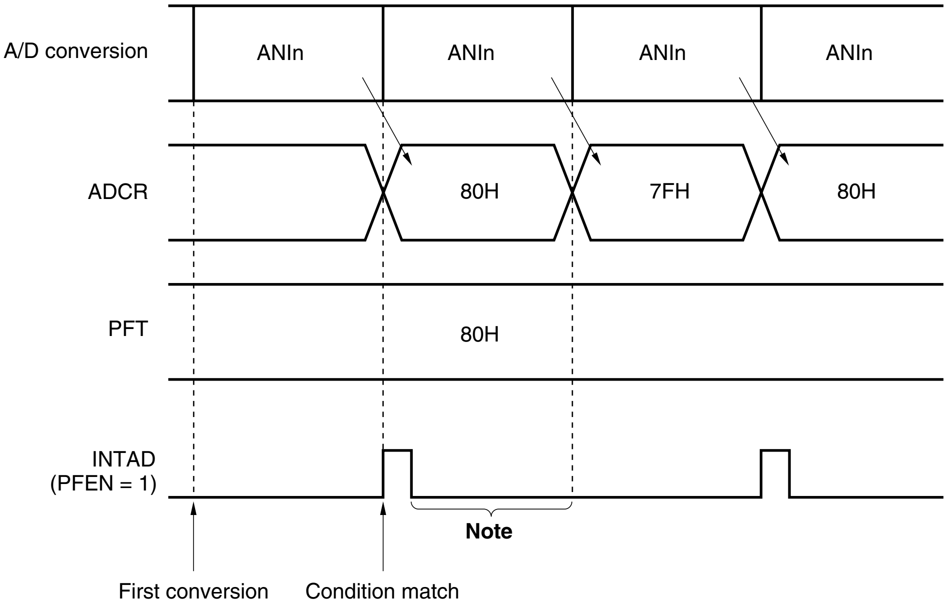
When the A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), the values are compared with power-fail comparison threshold register (PFT), and an interrupt request signal (INTAD) is generated under the condition specified by bit 6 (PFCM) of PFM.

<1> When PFEN = 0
INTAD is generated at the end of each A/D conversion.

<2> When PFEN = 1 and PFCM = 0
The ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when ADCR ≥ PFT.

<3> When PFEN = 1 and PFCM = 1
The ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when ADCR < PFT.

Figure 13-12. Power-Fail Detection (When PFEN = 1 and PFCM = 0)



Note If the conversion result is not read before the end of the next conversion after INTAD is output, the result is replaced by the next conversion result.

Remark n = 0 to 7

The setting methods are described below.

- When used as A/D conversion operation

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
<2> Select the channel and conversion time using bits 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS) and bits 5 to 3 (FR2 to FR0) of ADM.
<3> Set bit 7 (ADCS) of ADM to 1.
<4> An interrupt request signal (INTAD) is generated.
<5> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<Change the channel>

<6> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS.
<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 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS) and bits 5 to 3 (FR2 to FR0) of ADM.
<5> Set a threshold value to the power-fail comparison threshold register (PFT).
<6> Set bit 7 (ADCS) of ADM to 1.
<7> Transfer the A/D conversion data to the A/D conversion result register (ADCR).
<8> ADCR and PFT are compared and an interrupt request signal (INTAD) is generated if the conditions match.

<Change the channel>

<9> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS.
<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR).
<11> 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> can be omitted. However, do not use the first conversion result after <6> in this case.
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.

13.5 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.

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

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

(2) Overall error

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

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

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

(3) Quantization error

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

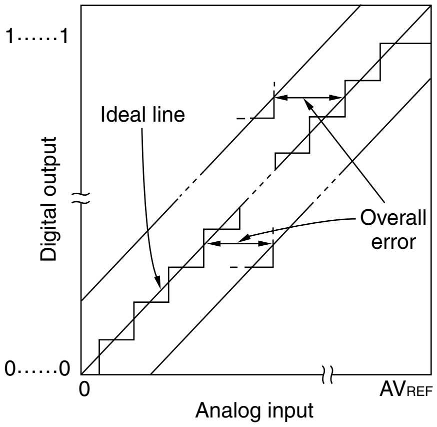
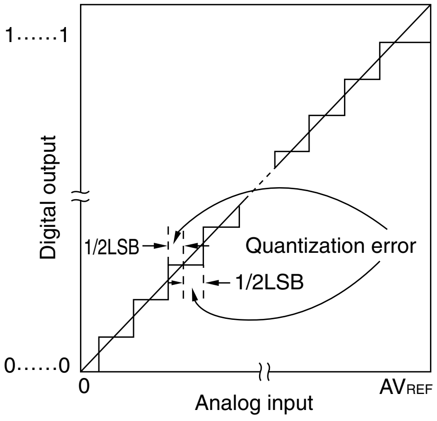


Figure 13-14. Quantization Error



(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 13-15. Zero-Scale Error

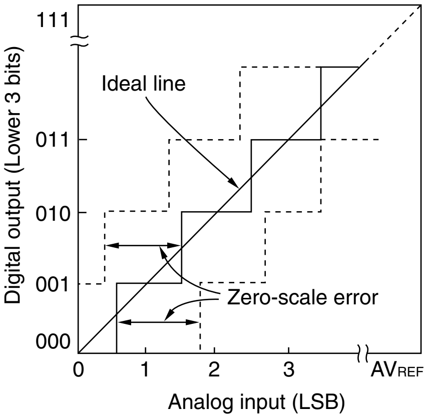


Figure 13-16. Full-Scale Error

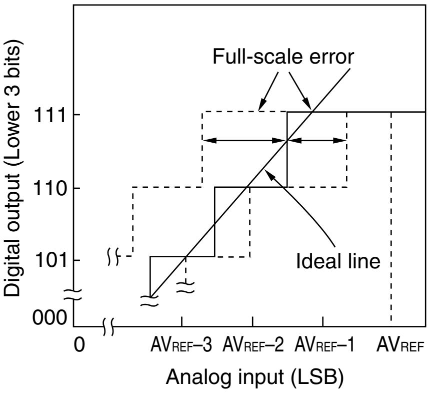


Figure 13-17. Integral Linearity Error

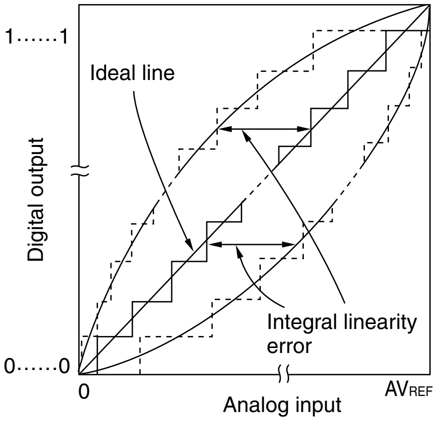
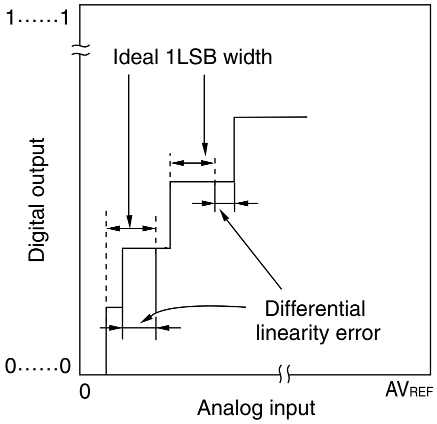


Figure 13-18. Differential Linearity Error



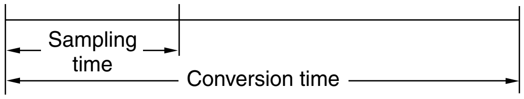
(8) Conversion time

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.



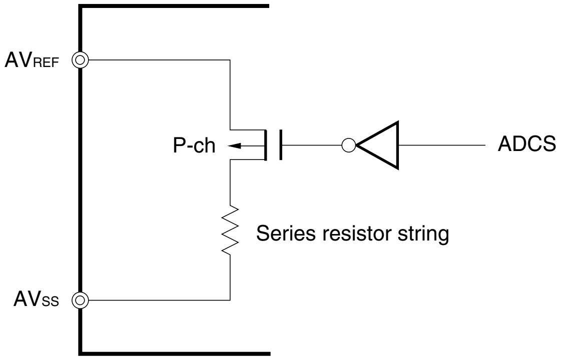
13.6 Cautions for A/D Converter

(1) Current consumption in standby mode

The A/D converter stops operating in the standby mode. At this time, the current consumption can be reduced by stopping the conversion operation (by setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 0).

Figure 13-19 shows how to reduce the current consumption in the standby mode.

Figure 13-19. Example of Method of Reducing Current Consumption in Standby Mode



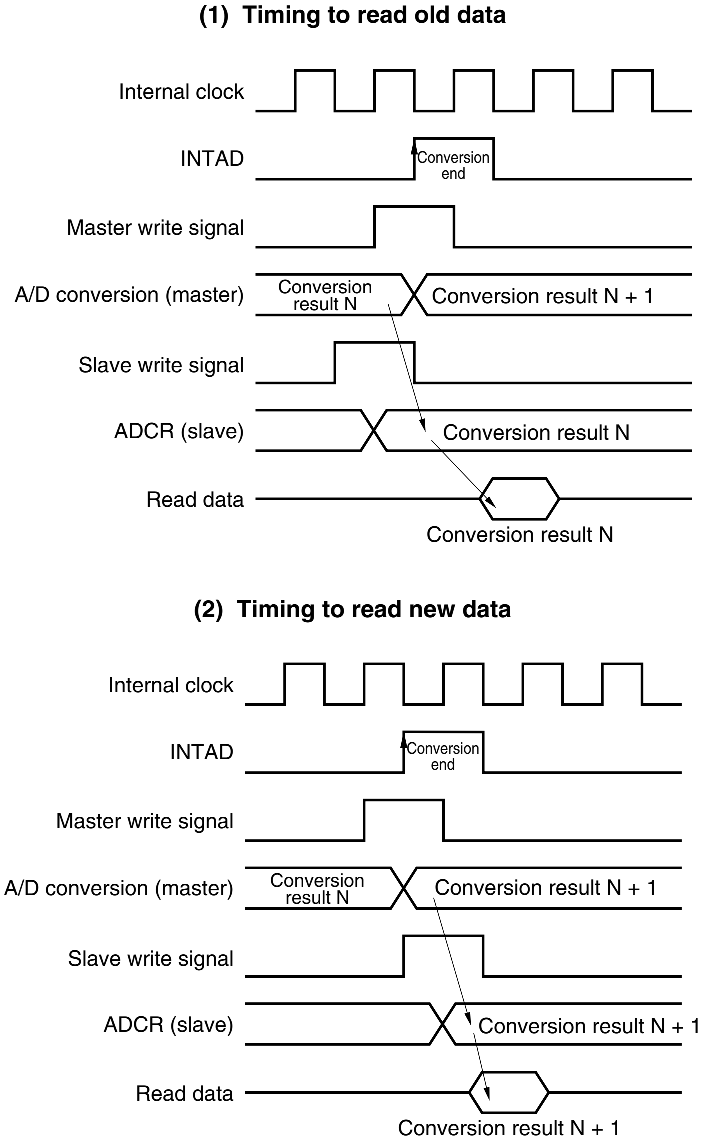
(2) Input range of ANI0 to ANI7

Observe the rated range of the ANI0 to ANI7 input voltage. If a voltage of AV_{REF} or higher and AV_{SS} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR) write and ADCR read by instruction upon the end of conversion
ADCR read has priority. After the read operation, the new conversion result is written to ADCR.
Old data can be read from ADCR at the timing of (1) and new data can be read from ADCR at the timing of (2) as shown in Figure 13-20. A master-slave configuration is employed for transferring the A/D conversion result to ADCR.

Figure 13-20. Storing Conversion Result in ADCR and Timing of Data Read from ADCR


(1) Timing to read old data

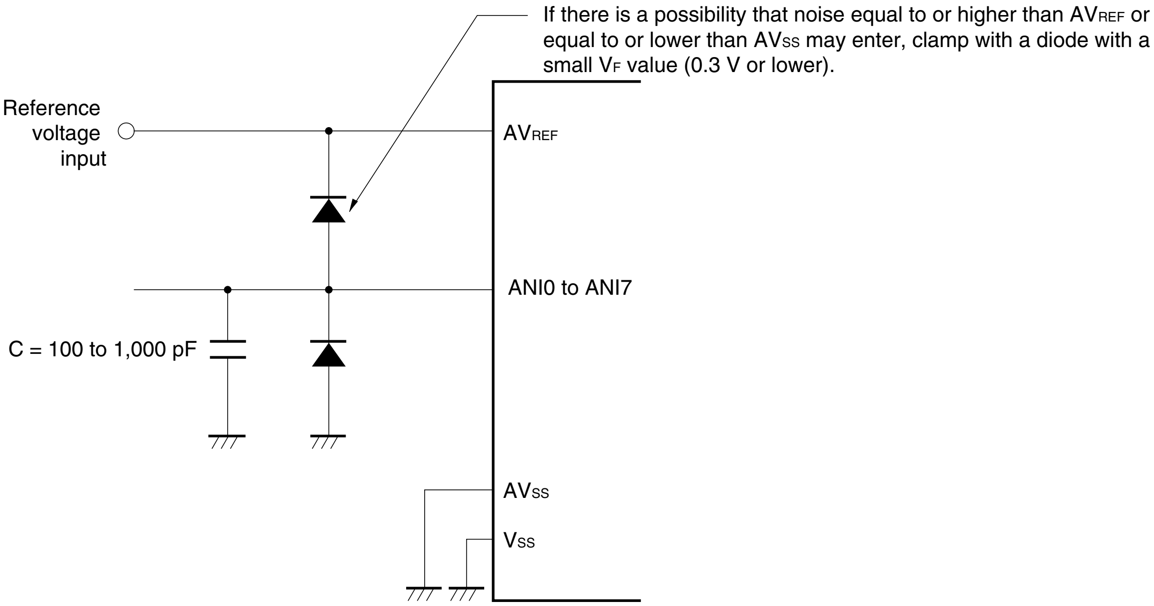
(2) Timing to read new data

<2> Conflict between ADCR write and A/D converter mode register (ADM) write or analog input channel specification register (ADS) write
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 ANI7. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally, as shown in Figure 13-21, to reduce noise.

Figure 13-21. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF or equal to or lower than AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREF
ANI0 to ANI7
C = 100 to 1,000 pF
AVSS
VSS

(5) ANI0/P20 to ANI7/P27

The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27).

When A/D conversion is performed with any of ANI0 to ANI7 selected, do not execute the input instruction to port 2 while conversion is in progress; otherwise the conversion resolution may be degraded.

If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI0 to ANI7 pins

In this A/D converter, the internal sampling capacitor is charged and sampling is performed for approx. one tenth of the conversion time.

Since only the leakage current flows other than during sampling and the current for charging the capacitor also flows during sampling, the input impedance fluctuates and has no meaning.

To perform sufficient sampling, however, it is recommended to make the output impedance of the analog input source 10 kΩ or lower, or attach a capacitor of around 100 pF to the ANI0 to ANI7 pins (see **Figure 13-21**).

(7) AV_{REF} pin input impedance

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

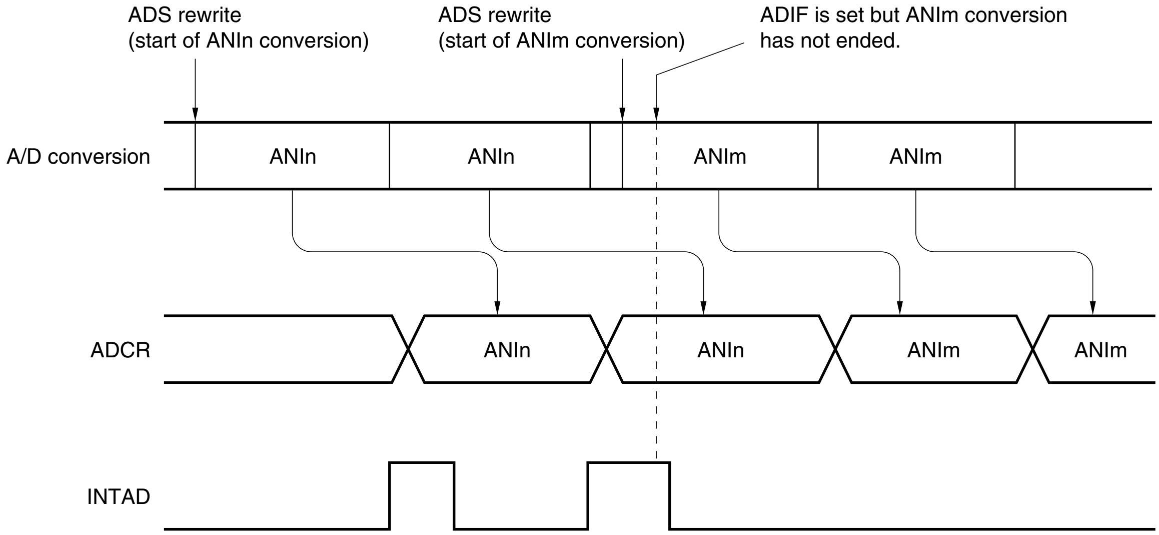
(8) Interrupt request flag (ADIF)

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 13-22. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 7

2. m = 0 to 7

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

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating. 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.

★ Do not read ADCR when the CPU is operating on the subsystem clock and oscillation of the X1 input clock is stopped.

(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 13-23 and Table 13-3.

Figure 13-23. Timing of A/D Converter Sampling and A/D Conversion Start Delay

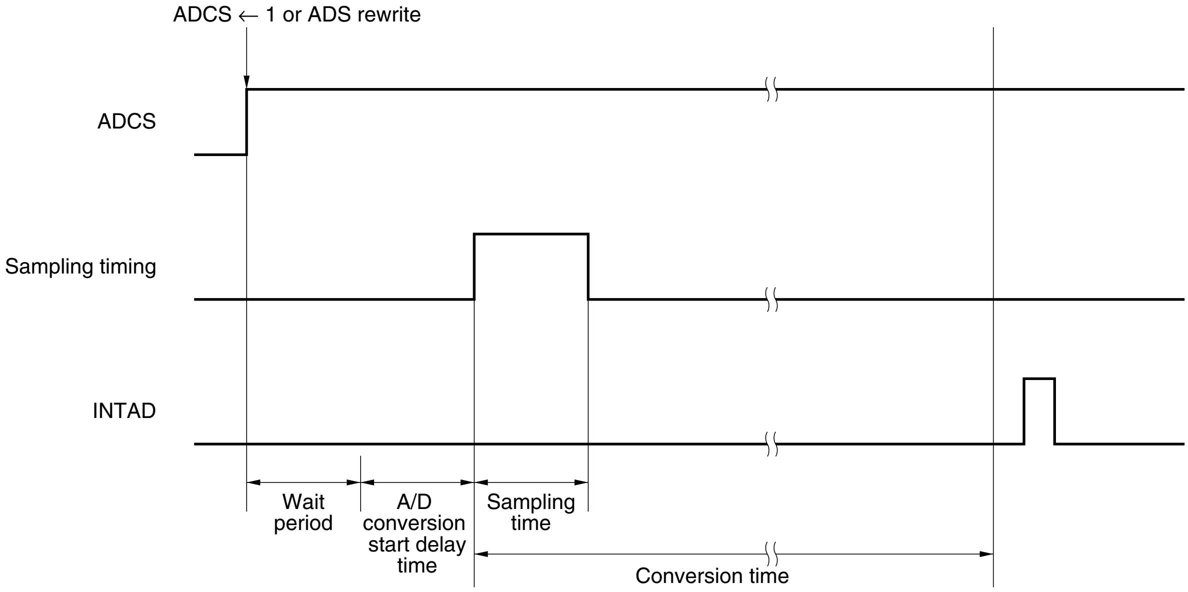


Table 13-3. A/D Converter Sampling Time and A/D Conversion Start Delay Time (ADM Set Value)

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 32 CAUTIONS FOR WAIT**.

Remark fx: X1 clock oscillation frequency

CHAPTER 14 SERIAL INTERFACE UART0

14.1 Functions of Serial Interface UART0

Serial interface UART0 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 **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

The functions of this mode are outlined below.

- Two-pin configuration TxD0: Transmit data output pin
  RxB0: Receive data input pin
- Length of transfer data can be selected from 7 or 8 bits.
- Dedicated on-chip 5-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently.
- Four operating clock inputs selectable
- Fixed to LSB-first transfer

Cautions 1. The default value of the TxD0 pin is high level. Exercise care when using the TxD0 pin as a port pin.

2. If clock supply to serial interface UART0 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART0 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD0 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER0 = 0, RXE0 = 0, and TXE0 = 0.

3. Set POWER0 = 1 and then set TXE0 = 1 (transmission) or RXE0 = 1 (reception) to start communication.

4. TXE0 and RXE0 are synchronized with the base clock (f_{XCLK}) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.

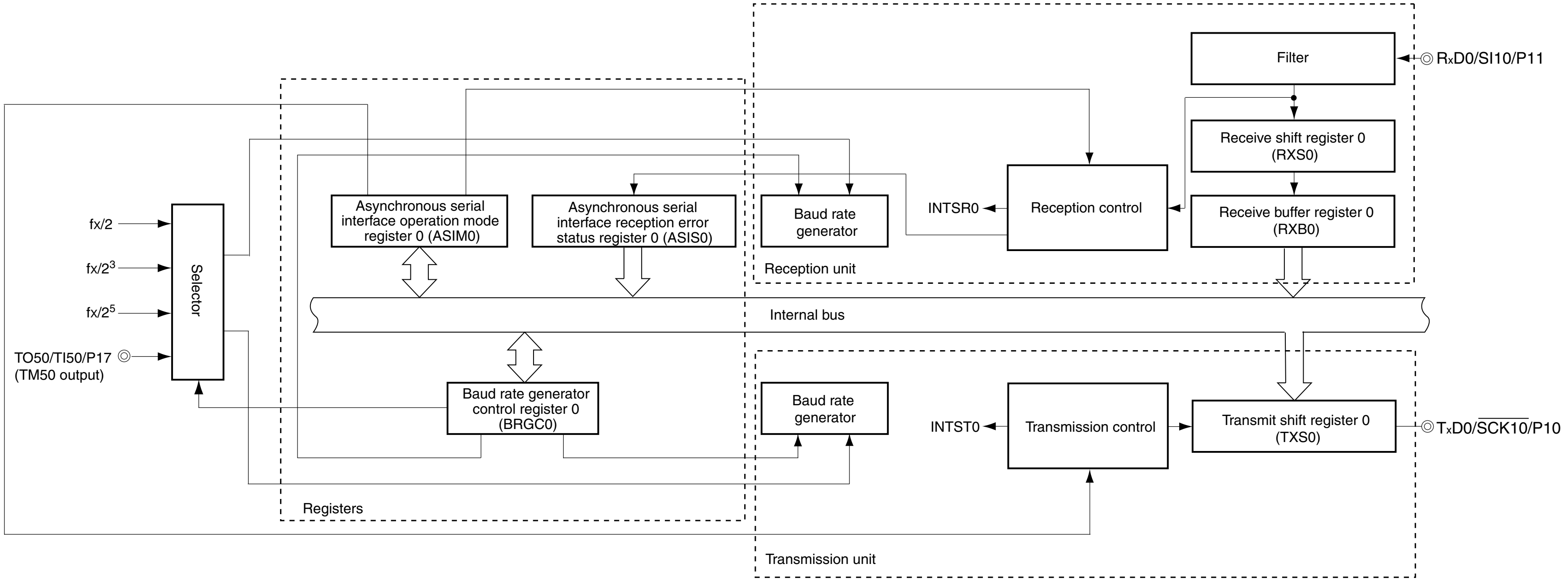
14.2 Configuration of Serial Interface UART0

Serial interface UART0 consists of the following hardware.

Table 14-1. Configuration of Serial Interface UART0

Item	Configuration
Registers	Receive buffer register 0 (RXB0) Receive shift register 0 (RXS0) Transmit shift register 0 (TXS0)
Control registers	Asynchronous serial interface operation mode register 0 (ASIM0) Asynchronous serial interface reception error status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0)

Figure 14-1. Block Diagram of Serial Interface UART0


Filter
RxD0/SI10/P11
Receive shift register 0 (RXS0)
Receive buffer register 0 (RXB0)
Reception control
INTSR0
Baud rate generator
Reception unit
Asynchronous serial interface operation mode register 0 (ASIM0)
Asynchronous serial interface reception error status register 0 (ASIS0)
Internal bus
fx/2
fx/2^3
fx/2^5
TO50/TI50/P17
(TM50 output)
Selector
Baud rate generator control register 0 (BRGC0)
Registers
Baud rate generator
INTST0
Transmission control
Transmit shift register 0 (TXS0)
TxD0/SCK10/P10
Transmission unit

(1) Receive buffer register 0 (RXB0)

This 8-bit register stores parallel data converted by receive shift register 0 (RXS0).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 0 (RXS0).

If the data length is set to 7 bits the receive data is transferred to bits 0 to 6 of RXB0 and the MSB of RXB0 is always 0.

If an overrun error (OVE0) occurs, the receive data is not transferred to RXB0.

$\overline{\text{RESET}}$ input or POWER0 = 0 sets this register to FFH.

RXB0 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

(2) Receive shift register 0 (RXS0)

This register converts the serial data input to the RxD0 pin into parallel data.

RXS0 cannot be directly manipulated by a program.

(3) Transmit shift register 0 (TXS0)

This register is used to set transmit data. Transmission is started when data is written to TXS0, and serial data is transmitted from the TxD0 pins.

$\overline{\text{RESET}}$ input, POWER0 = 0, or TXE0 = 0 sets this register to FFH.

TXS0 can be written by an 8-bit memory manipulation instruction. This register cannot be read.

Caution Do not write the next transmit data to TXS0 before the transmission completion interrupt signal (INTST0) is generated.

14.3 Registers Controlling Serial Interface UART0

Serial interface UART0 is controlled by the following three registers.

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)

(1) Asynchronous serial interface operation mode register 0 (ASIM0)

This 8-bit register controls the serial transfer operations of serial interface UART0.

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

$\overline{\text{RESET}}$ input sets this register to 01H.

Figure 14-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (1/2)

Address: FF70H After reset: 01H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

Note The input from the RxD0 pin is fixed to high level when POWER0 = 0.

Cautions
1. **At startup, set POWER0 to 1 and then set TXE0 to 1. Clear TXE0 to 0 first, and then clear POWER0 to 0.**
2. **At startup, set POWER0 to 1 and then set RXE0 to 1. Clear RXE0 to 0 first, and then clear POWER0 to 0.**
3. **TXE0 and RXE0 are synchronized with the base clock (f_{XCLK}) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.**
4. **Be sure to set bit 0 to 1.**

Figure 14-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (2/2)

PS01	PS00	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL0	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL0	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE0) of asynchronous serial interface status register 0 (ASIS0) is not set and the error interrupt does not occur.

Cautions 1. Clear the TXE0 and RXE0 bits to 0 before rewriting the PS01, PS00, and CL0 bits.

2. Make sure that TXE0 = 0 when rewriting the SL0 bit. Reception is always performed with “number of stop bits = 1”, and therefore, is not affected by the set value of the SL0 bit.

(2) Asynchronous serial interface reception error status register 0 (ASIS0)

This register indicates an error status on completion of reception by serial interface UART0. It includes three error flag bits (PE0, FE0, OVE0).

This register can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$ input clears this register to 00H if bit 7 (POWER0) and bit 5 (RXE0) of ASIM0 = 0. 00H is read when this register is read.

Figure 14-3. Format of Asynchronous Serial Interface Reception Error Status Register 0 (ASIS0)

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Status flag indicating parity error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE0	Status flag indicating framing error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the stop bit is not detected on completion of reception.

OVE0	Status flag indicating overrun error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If receive data is set to the RXB register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE0 bit differs depending on the set values of the PS01 and PS00 bits of asynchronous serial interface mode register 0 (ASIM0).

2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 0 (RXB0) but discarded.

4. If data is read from ASIS0, a wait cycle is generated. Do not read data from ASIS0 when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

(3) Baud rate generator control register 0 (BRGC0)

This register selects the base clock of serial interface UART0 and controls the baud rate.

BRGC0 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 1FH.

Figure 14-4. Format of Baud Rate Generator Control Register 0 (BRGC0)

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	Base clock (f_{XCLK}) selection
0	0	TM50 output (TO50)
0	1	$f_X/2$ (5 MHz)
1	0	$f_X/2^3$ (1.25 MHz)
1	1	$f_X/2^5$ (312.5 kHz)

MDL04	MDL03	MDL02	MDL01	MDL00	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	$f_{XCLK}/8$
0	1	0	0	1	9	$f_{XCLK}/9$
0	1	0	1	0	10	$f_{XCLK}/10$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	0	1	0	26	$f_{XCLK}/26$
1	1	0	1	1	27	$f_{XCLK}/27$
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Make sure that bit 6 (TXE0) and bit 5 (RXE0) of the ASIM0 register = 0 when rewriting the MDL04 to MDL00 bits.

2. The baud rate is the output clock of the 5-bit counter divided by 2.

Remarks 1. f_{XCLK}: Frequency of base clock (Clock) selected by the TPS01 and TPS00 bits

2. f_X: X1 input clock oscillation frequency

3. k: Value set by the MDL04 to MDL00 bits (k = 8, 9, 10, ..., 31)

4. ×: Don’t care

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

14.4 Operation of Serial Interface UART0

This section explains the two modes of serial interface UART0.

14.4.1 Operation stop mode

In this mode, serial transfer cannot be executed, thus reducing the power consumption. In addition, the pins can be used as ordinary port pins in this mode.

(1) Register setting

The operation stop mode is set by asynchronous serial interface operation mode register 0 (ASIM0).
ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets this register to 01H.

Address: FF70H After reset: 01H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

Note The input from the RxD0 pin is fixed to high level when POWER0 = 0.

Cautions 1. At startup, set POWER0 to 1 and then set TXE0 to 1. Clear TXE0 to 0 first, and then clear POWER0 to 0.

2. At startup, set POWER0 to 1 and then set RXE0 to 1. Clear RXE0 to 0 first, and then clear POWER0 to 0.

3. TXE0 and RXE0 are synchronized with the base clock (f_{XCLK}) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.

14.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Register setting

The UART mode is set by asynchronous serial interface operation mode register 0 (ASIM0), asynchronous serial interface reception error status register 0 (ASIS0), and baud rate generator control register 0 (BRGC0).

(a) Asynchronous serial interface operation mode register 0 (ASIM0)

This 8-bit register controls the serial transfer operations of serial interface UART0.

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 01H.

Address: FF70H After reset: 01H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Note The input from the RxD0 pin is fixed to high level when POWER0 = 0.

Cautions 1. At startup, set POWER0 to 1 and then set TXE0 to 1. Clear TXE0 to 0 first, and then clear POWER0 to 0.

2. At startup, set POWER0 to 1 and then set RXE0 to 1. Clear RXE0 to 0 first, and then clear POWER0 to 0.

3. TXE0 and RXE0 are synchronized with the base clock (f_{XCLK}) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.

4. Be sure to set bit 0 to 1.

PS01	PS00	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL0	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL0	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE0) of asynchronous serial interface status register 0 (ASIS0) is not set and the error interrupt does not occur.

Cautions 1. Clear the TXE0 and RXE0 bits to 0 before rewriting the PS01, PS00, and CL0 bits.

2. Make sure that TXE0 = 0 when rewriting the SL0 bit. Reception is always performed with “number of stop bits = 1”, and therefore, is not affected by the set value of the SL0 bit.

(b) Asynchronous serial interface reception error status register 0 (ASIS0)

This register indicates an error status on completion of reception by serial interface UART0. It includes three error flag bits (PE0, FE0, OVE0).

This register can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$ input clears this register to 00H if bit 7 (POWER0) and bit 5 (RXE0) of ASIM0 = 0. 00H is read when this register is read.

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Status flag indicating parity error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE0	Status flag indicating framing error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the stop bit is not detected on completion of reception.

OVE0	Status flag indicating overrun error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If receive data is set to the RXB register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE0 bit differs depending on the set values of the PS01 and PS00 bits of asynchronous serial interface mode register 0 (ASIM0).

2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 0 (RXB0) but discarded.

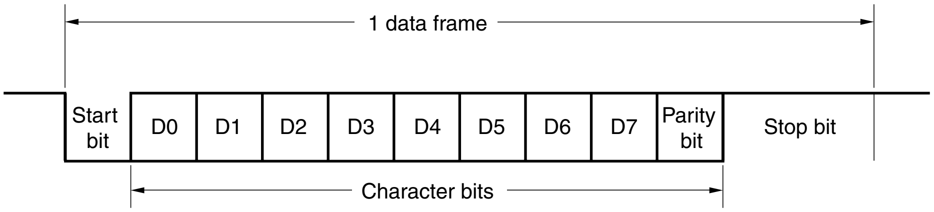
4. If data is read from ASIS0, a wait cycle is generated. Do not read data from ASIS0 when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

(2) Communication operation

(a) Normal transmit/receive data format

Figure 14-5 shows the format of the transmit/receive data.

Figure 14-5. Format of Normal UART Transmit/Receive Data



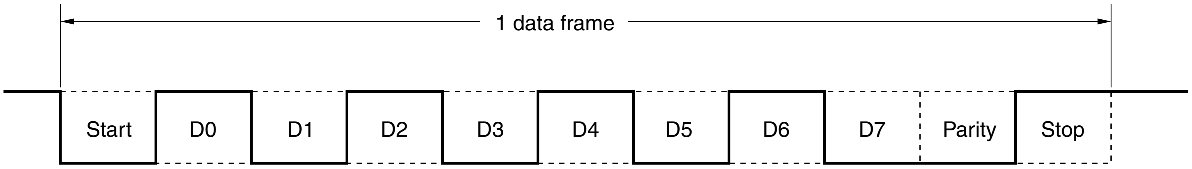
One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits (LSB first)
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

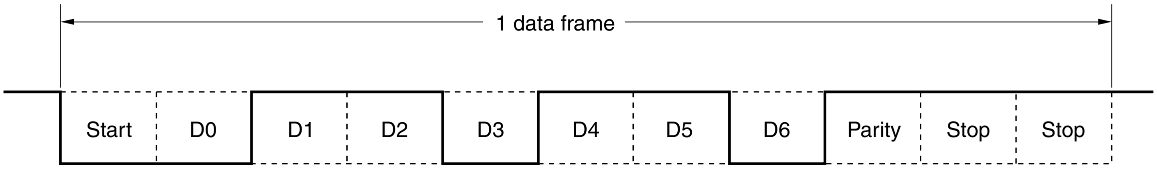
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface mode register 0 (ASIM0).

Figure 14-6. Example of Normal UART Transmit/Receive Data Format

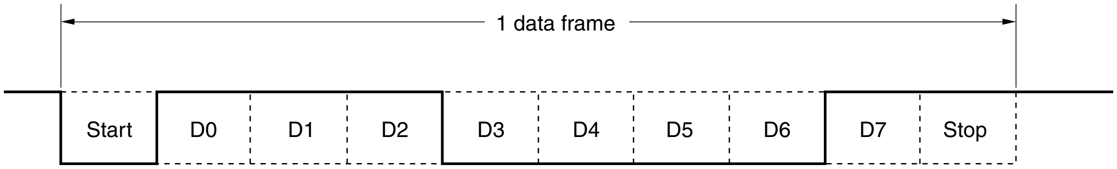
1. Data length: 8 bits, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H



2. Data length: 7 bits, Parity: Odd parity, Stop bit: 2 bits, Transfer data: 36H



3. Data length: 8 bits, Parity: None, Stop bit: 1 bit, Transfer data: 87H



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Transmission

The TxD0 pin outputs a high level when bit 7 (POWER0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1. If bit 6 (TXE0) of ASIM0 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit shift register 0 (TXS0). The start bit, parity bit, and stop bit are automatically appended to the data.

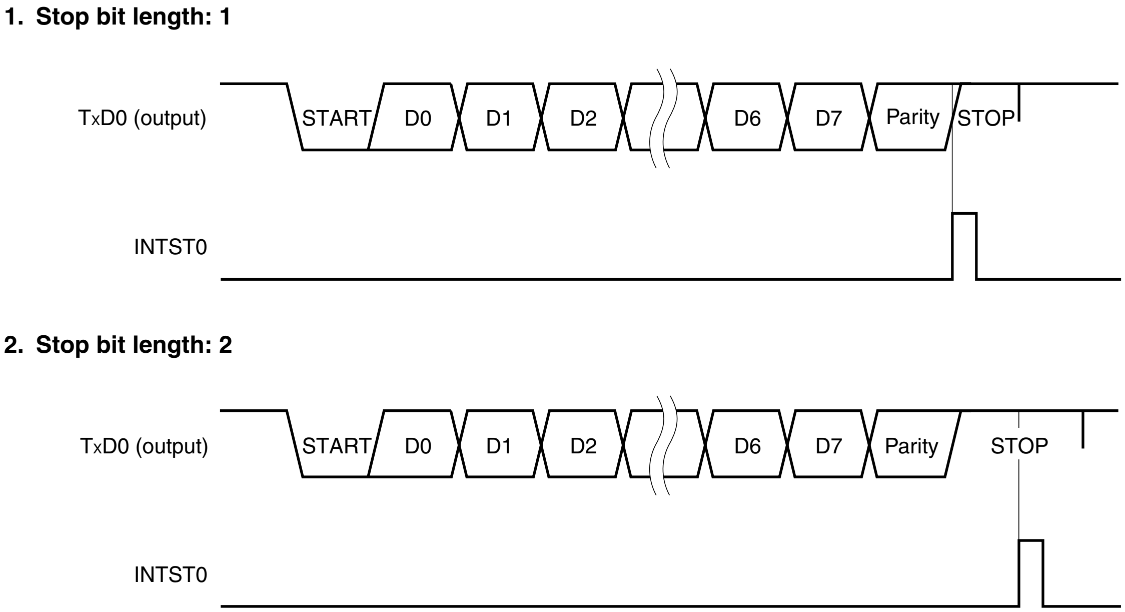
When transmission is started, the start bit is output from the TxD0 pin, followed by the rest of the data in order starting from the LSB. When transmission is completed, the parity and stop bits set by ASIM0 are appended and a transmission completion interrupt request (INTST0) is generated.

Transmission is stopped until the data to be transmitted next is written to TXS0.

Figure 14-7 shows the timing of the transmission completion interrupt request (INTST0). This interrupt occurs as soon as the last stop bit has been output.

Caution After transmit data is written to TXS0, do not write the next transmit data before the transmission completion interrupt signal (INTST0) is generated.

Figure 14-7. Normal Transmission Completion Interrupt Request Timing



1. Stop bit length: 1

2. Stop bit length: 2

(d) Reception

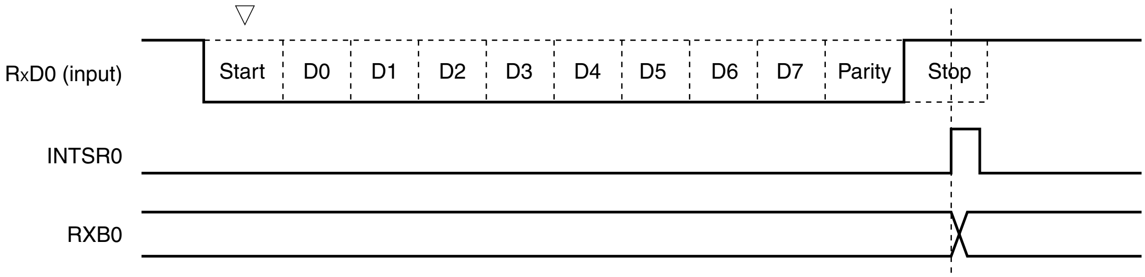
Reception is enabled and the RxD0 pin input is sampled when bit 7 (POWER0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1 and then bit 5 (RXE0) of ASIM0 is set to 1.

The 5-bit counter of the baud rate generator starts counting when the falling edge of the RxD0 pin input is detected. When the set value of baud rate generator control register 0 (BRGC0) has been counted, the RxD0 pin input is sampled again (▽ in Figure 14-8). If the RxD0 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in receive shift register 0 (RXS0) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR0) is generated and the data of RXS0 is written to receive buffer register 0 (RXB0). If an overrun error (OVE0) occurs, however, the receive data is not written to RXB0.

Even if a parity error (PE0) or a framing error (FE0) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSR0) is generated after completion of reception.

Figure 14-8. Reception Completion Interrupt Request Timing



Cautions 1. Be sure to read receive buffer register 0 (RXB0) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 0 (ASIS0) before reading RXB0.

(e) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 0 (ASIS0) is set as a result of data reception, a reception error interrupt request (INTSR0) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS0 in the reception error interrupt servicing (INTSR0) (refer to **Table 14-2**).

The contents of ASIS0 are reset to 0 when ASIS0 is read.

Table 14-2. Cause of Reception Error

Reception Error	Cause	Value of ASIS0
Parity error	The parity specified for transmission does not match the parity of the receive data.	04H
Framing error	Stop bit is not detected.	02H
Overrun error	Reception of the next data is completed before data is read from receive buffer register 0 (RXB0).	01H

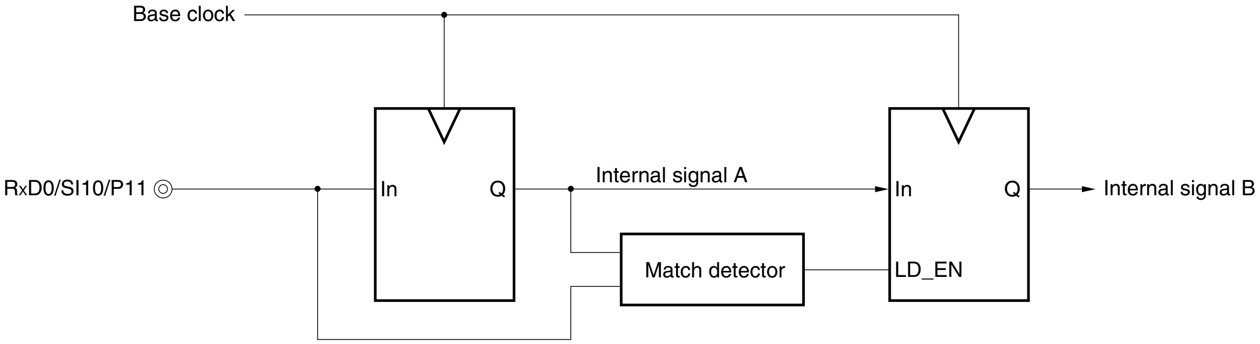
(f) Noise filter of receive data

The RxD0 signal is sampled using the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 14-9, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 14-9. Noise Filter Circuit



14.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and a 5-bit programmable counter, and generates a serial clock for transmission/reception of UART0.

Separate 5-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

- Base clock (Clock)

  The clock selected by bits 7 and 6 (TPS01 and TPS00) of baud rate generator control register 0 (BRGC0) is supplied to each module when bit 7 (POWER0) of asynchronous serial interface mode register 0 (ASIM0) is 1. This clock is called the base clock “Clock” and its frequency is called f_{XCLK}. “Clock” is fixed to low level when POWER0 = 0.

- Transmission counter

  This counter stops, cleared to 0, when bit 7 (POWER0) or bit 6 (TXE0) of asynchronous serial interface mode register 0 (ASIM0) is 0.

  It starts counting when POWER0 = 1 and TXE0 = 1.

  The counter is cleared to 0 when the first data transmitted is written to transmit shift register 0 (TXS0).

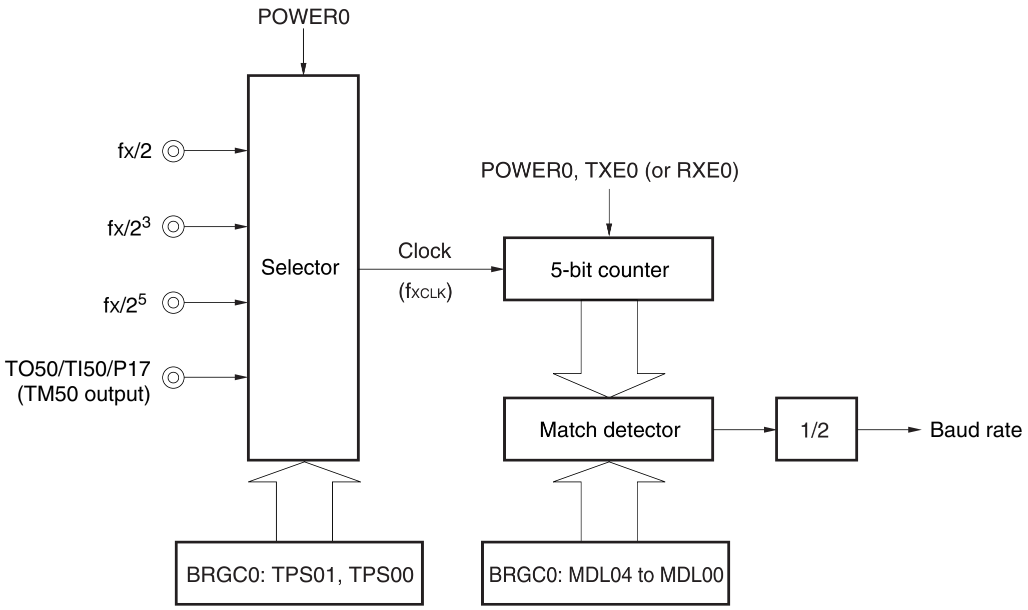
- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 5 (RXE0) of asynchronous serial interface mode register 0 (ASIM0) is 0.

  It starts counting when the start bit has been detected.

  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 14-10. Configuration of Baud Rate Generator



Remark POWER0: Bit 7 of asynchronous serial interface mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
BRGC0: Baud rate generator control register 0

(2) Generation of serial clock

A serial clock can be generated by using baud rate generator control register 0 (BRGC0).
Select the clock to be input to the 5-bit counter by using bits 7 and 6 (TPS01 and TPS00) of BRGC0.
Bits 4 to 0 (MDL04 to MDL00) of BRGC0 can be used to select the division value of the 5-bit counter.

(a) Baud rate generator control register 0 (BRGC0)

This register selects the base clock of serial interface UART0 and controls the baud rate.
BRGC0 can be set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets this register to 1FH.

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	Base clock (f_{XCLK}) selection
0	0	TM50 output (TO50)
0	1	$f_X/2$ (5 MHz)
1	0	$f_X/2^3$ (1.25 MHz)
1	1	$f_X/2^5$ (312.5 kHz)

MDL04	MDL03	MDL02	MDL01	MDL00	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	$f_{XCLK}/8$
0	1	0	0	1	9	$f_{XCLK}/9$
0	1	0	1	0	10	$f_{XCLK}/10$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	0	1	0	26	$f_{XCLK}/26$
1	1	0	1	1	27	$f_{XCLK}/27$
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Make sure that bit 6 (TXE0) and bit 5 (RXE0) of the ASIM0 register = 0 when rewriting the MDL04 to MDL00 bits.

2. The baud rate value is the output clock of the 5-bit counter divided by 2.

Remarks 1. f_{XCLK}: Frequency of base clock (Clock) selected by the TPS01 and TPS00 bits

2. f_X: X1 input clock oscillation frequency

3. k: Value set by the MDL04 to MDL00 bits (k = 8, 9, 10, ..., 31)

4. ×: Don’t care

5. Figures in parentheses apply to operation with f_X = 10 MHz

(b) Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK}}{2 \times k}$ [bps]

f_{XCLK}: Frequency of base clock (Clock) selected by the TPS01 and TPS00 bits of the BRGC0 register
k: Value set by the MDL04 to MDL00 bits of the BRGC0 register (k = 8, 9, 10, ..., 31)

(c) 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 (Clock) = 2.5 MHz = 2,500,000 Hz
Set value of MDL04 to MDL00 bits of BRGC0 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)
= 2,500,000/(2 × 16) = 78,125 [bps]

Error = (78,125/76,800 − 1) × 100
= 1.725 [%]

(3) Example of setting baud rate

Table 14-3. Set Data of Baud Rate Generator

Baud Rate [bps]	f_X = 10.0 MHz				f_X = 8.38 MHz				f_X = 4.19 MHz			
	TPS01, TPS00	k	Calculated value	ERR[%]	TPS01, TPS00	k	Calculated Value	ERR[%]	TPS01, TPS00	k	Calculated Value	ERR[%]
2400	–	–	–	–	–	–	–	–	3	27	2425	1.03
4800	–	–	–	–	3	27	4850	1.03	3	14	4676	–2.58
9600	3	16	9766	1.73	3	14	9353	–2.58	2	27	9699	1.03
10400	3	15	10417	0.16	3	13	10072	–3.15	2	25	10475	0.72
19200	3	8	19531	1.73	2	27	19398	1.03	2	14	18705	–2.58
31250	2	20	31250	0	2	17	30809	–1.41	–	–	–	–
38400	2	16	39063	1.73	2	14	38796	–2.58	2	27	38796	1.03
76800	2	8	78125	1.73	1	27	77593	1.03	1	14	74821	–2.58
115200	1	22	113636	–1.36	1	18	116389	1.03	1	9	116389	1.03
153600	1	16	156250	1.73	1	14	149643	–2.58	–	–	–	–
230400	1	11	227273	–1.36	1	9	232778	1.03	–	–	–	–

Remark TPS01, TPS00: Bits 7 and 6 of baud rate generator control register 0 (BRGC0) (setting of base clock (f_{XCLK}))

k: Value set by the MDL04 to MDL00 bits of BRGC0 (k = 8, 9, 10, ..., 31)

f_X: X1 input clock oscillation frequency

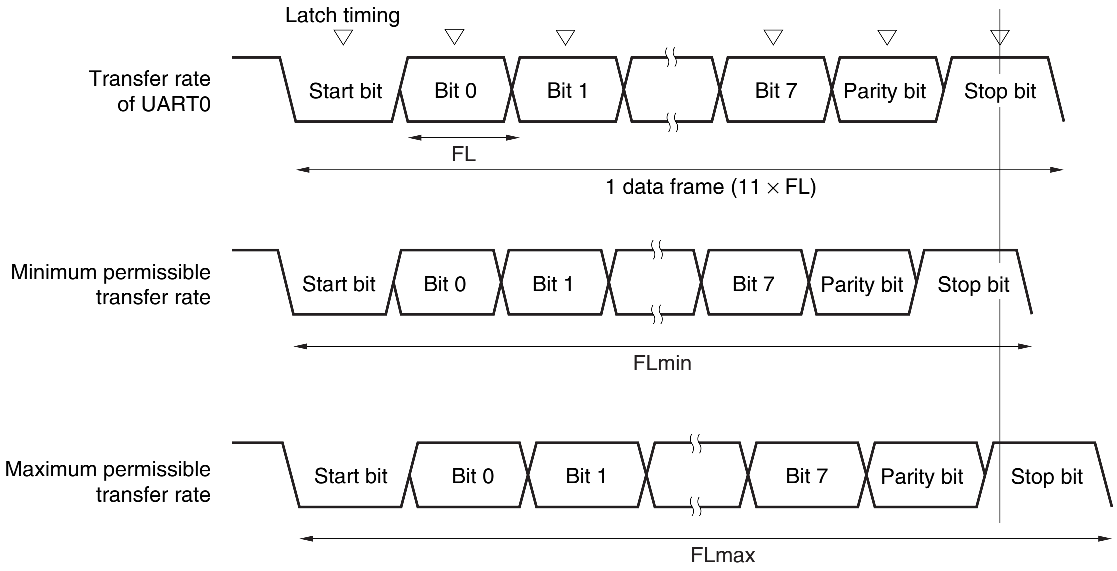
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 14-11. Permissible Baud Rate Range During Reception



As shown in Figure 14-11, the latch timing of the receive data is determined by the counter set by baud rate generator control register 0 (BRGC0) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

FL = $(\text{Brate})^{-1}$

Brate: Baud rate of UART0
k: Set value of BRGC0
FL: 1-bit data length
Margin of latch timing: 2 clocks

Minimum permissible transfer rate: $\text{FLmin} = 11 \times \text{FL} - \frac{k-2}{2k} \times \text{FL} = \frac{21k+2}{2k}\ \text{FL}$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k+2}\ \text{Brate}$$

Similarly, the maximum permissible transfer rate can be calculated as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2 \times k} \times \text{FL} = \frac{21k-2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k-2}{20k}\ \text{FL} \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2}\ \text{Brate}$$

The permissible baud rate error between UART0 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 14-4. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	–3.61%
16	+4.14%	–4.19%
24	+4.34%	–4.38%
31	+4.44%	–4.47%

Remarks 1. The accuracy 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 accuracy.

2. k: Set value of BRGC0

CHAPTER 15 SERIAL INTERFACE UART6

15.1 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 **15.4.1 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.

- Two-pin configuration TxD6: Transmit data output pin
  RxB6: Receive data input pin
- Data length of transfer 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 transfer selectable
- Inverted transmission operation
- Tuning break field transmission from 13 to 20 bits
- More than 11 bits can be identified for tuning break field reception (SBF reception flag provided).

Cautions 1. The default value of the TxD6 pin is the high level. Exercise care when using the TxD6 pin as a port pin.

2. 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 (it must be able to recognize a low-level start bit).

3. 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.

4. If data is continuously transmitted, the transfer rate from the stop bit to the next start bit is extended two clocks. However, this does not affect the result of transfer because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is incorporated in LIN.

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

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

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

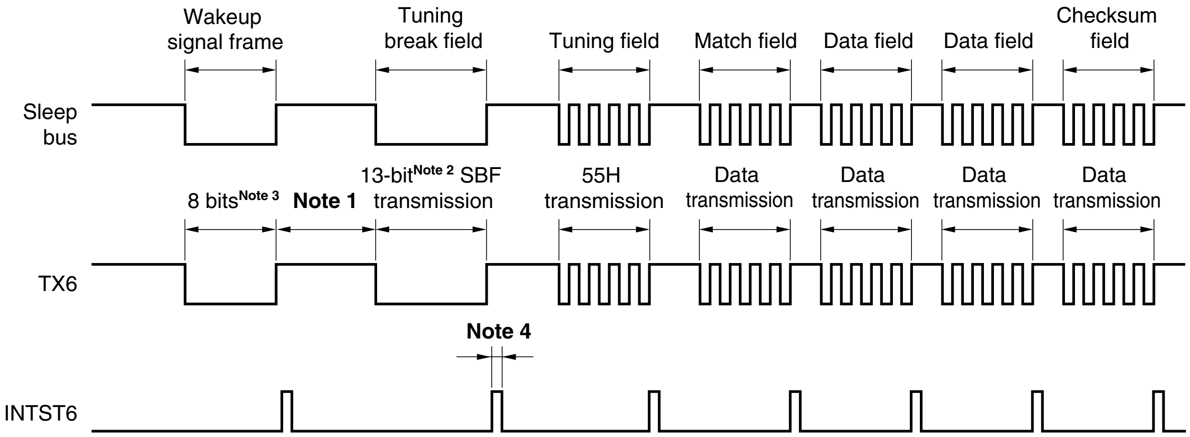
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 15-1 and 15-2 outline the transmission and reception operations of LIN.

Figure 15-1. LIN Transmission Operation



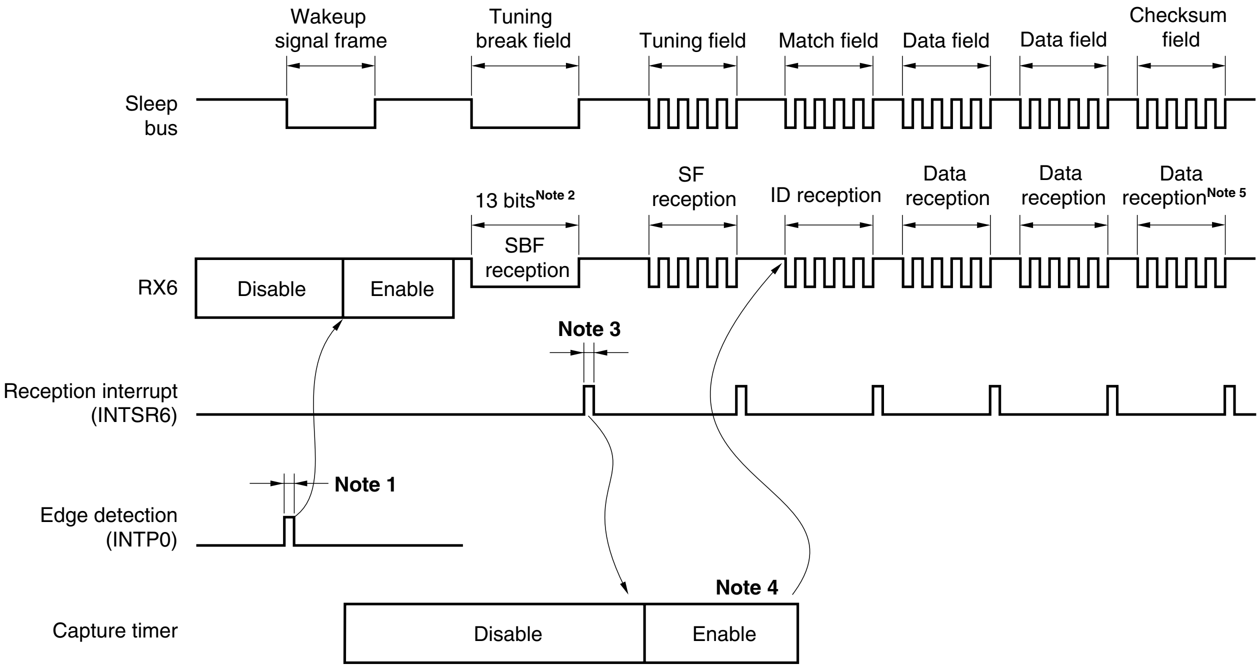
Notes 1. The interval between each field is controlled by software.

2. The tuning break field is output by hardware. The output width is equal to the bit length set by bits 5 to 3 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6). If the output width needs to be adjusted more accurately, use baud rate generator control register 6 (BRGC6).

3. The wakeup signal frame is substituted by 80H transfer in the 8-bit mode.

4. INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

Figure 15-2. 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 11 bits or more of SBF have been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If less than 11 bits of SBF have 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 value obtained from the capture timer, disable UART6 after SF reception, and then re-set baud rate generator control register 6 (BRGC6).

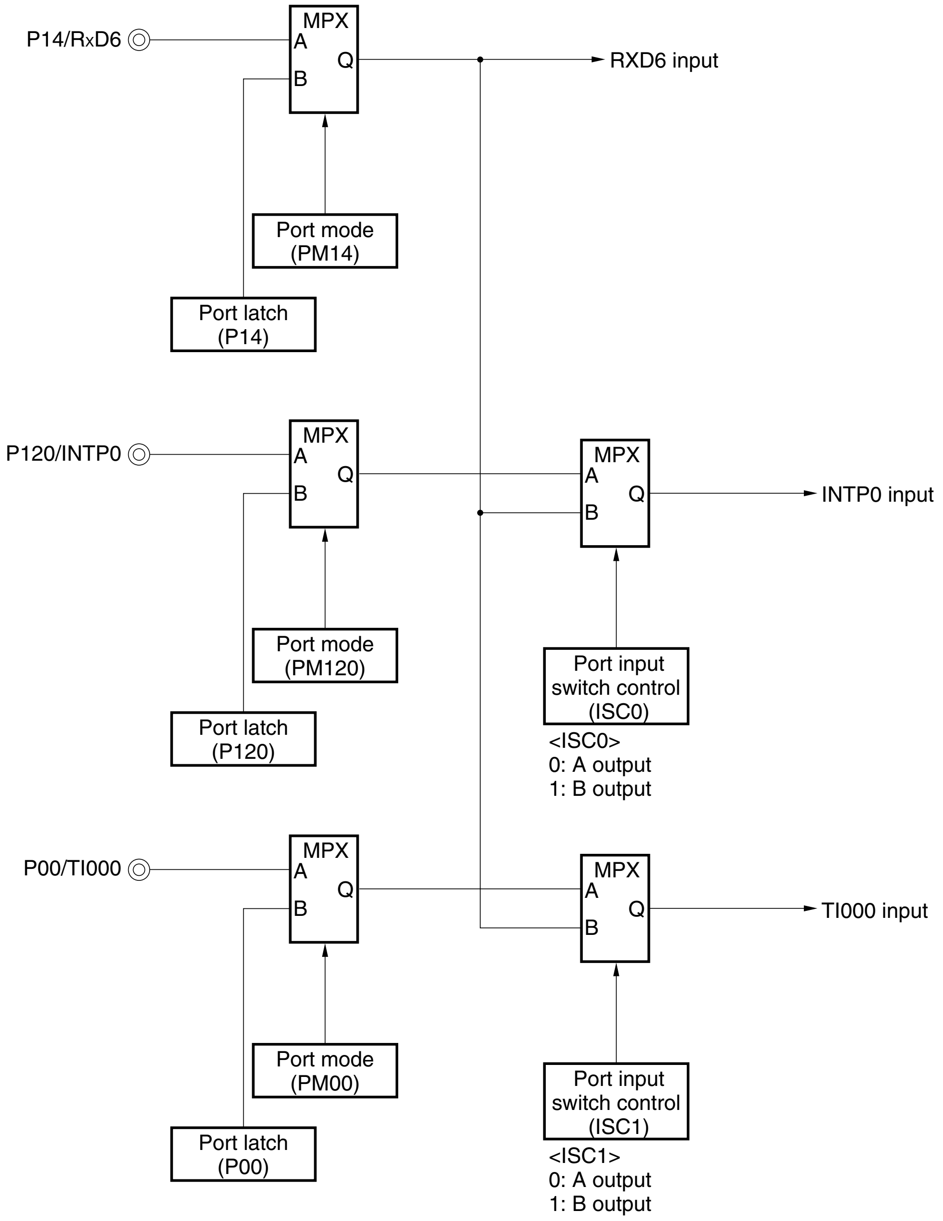
5. Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

To perform a LIN receive operation, use a configuration like the one shown in Figure 15-3.

The wakeup signal transmitted from the LIN master is received by detecting the edge of the external interrupt (INTP0). The length of the tuning break 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 using the time and number of bits of the tuning break field.

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 15-3. Port Configuration for LIN Reception Operation



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (see **Figure 4-29**)

The resources used in the LIN communication operation are shown below.

<Resources 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 tuning break field (SBF) length and divides it by the number of bits.
- Serial interface UART6

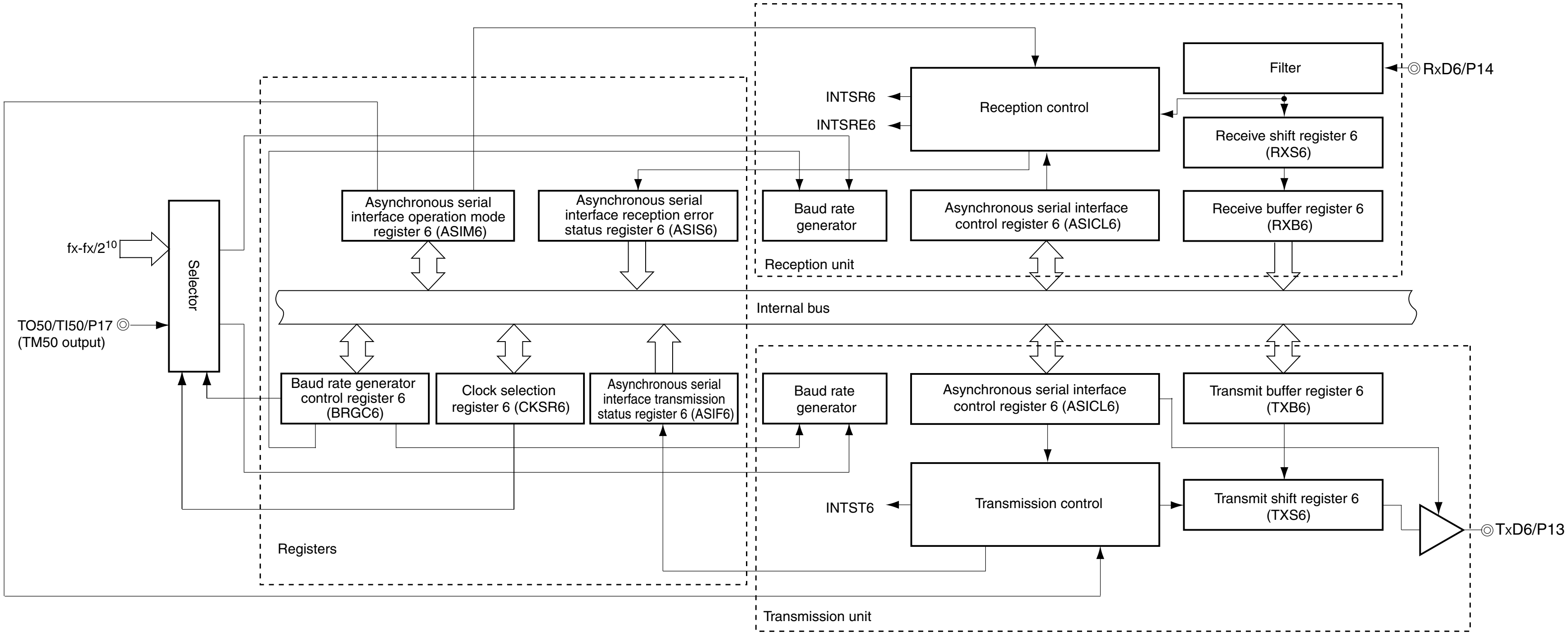
15.2 Configuration of Serial Interface UART6

Serial interface UART6 consists of the following hardware.

Table 15-1. 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)

Figure 15-4. Block Diagram of Serial Interface UART6


Filter
RxD6/P14
INTSR6
INTSRE6
Reception control
Receive shift register 6 (RXS6)
Asynchronous serial interface operation mode register 6 (ASIM6)
Asynchronous serial interface reception error status register 6 (ASIS6)
Baud rate generator
Asynchronous serial interface control register 6 (ASICL6)
Receive buffer register 6 (RXB6)
fx-fx/2^10
Selector
Reception unit
Internal bus
TO50/TI50/P17
(TM50 output)
Baud rate generator control register 6 (BRGC6)
Clock selection register 6 (CKSR6)
Asynchronous serial interface transmission status register 6 (ASIF6)
Baud rate generator
Asynchronous serial interface control register 6 (ASICL6)
Transmit buffer register 6 (TXB6)
INTST6
Transmission control
Transmit shift register 6 (TXS6)
TxD6/P13
Registers
Transmission unit

(1) Receive buffer register 6 (RXB6)

This 8-bit register stores parallel data converted by the receive shift register.

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). However, if the same value is continuously transmitted in the transmission mode (POWER6 = 1 and TXE6 = 1), the same value can be written.

(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 internal clock.

TXS6 cannot be directly manipulated by a program.

15.3 Registers Controlling Serial Interface UART6

Serial interface UART6 is controlled by the following six 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)

(1) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial transfer 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 15-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)

Address: FF50H After reset: 01H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	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.
1[Note 2]	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. Operation of the internal operation clock is enabled at the second input clock after 1 is written to the POWER6 bit.

Caution At startup, set POWER6 to 1 and then set TXE6 to 1. Clear TXE6 to 0 first, and then clear POWER6 to 0.

Figure 15-5. 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 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 RXE6 to 1. Clear RXE6 to 0 first, and then clear POWER6 to 0.

2. Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.

3. Fix the PS61 and PS60 bits to 0 when mounting the device on LIN.

4. 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.

5. 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 can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$ input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0. 00H is read when this register is read.

Figure 15-6. 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 RXB 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. Do not read data from ASIS6 when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 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 can be set by an 8-bit memory manipulation instruction, and is read-only.

$\overline{\text{RESET}}$ input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0.

Figure 15-7. 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 continuously transmit data, write the data of the first byte to TXB6, check that the value of the TXBF6 flag is 0, and then write the data of the second byte to TXB6. The operation is not guaranteed if data is written to TXB6 while the TXBF6 flag is 1.

2. While continuous transmission is being executed, check the value of the TXSF6 flag after the transmission completion interrupt to determine the subsequent write processing to TXB6.

- **If TXSF6 is 1: Continuous transmission is in progress. Data of 1 byte can be written.**
- **If TXSF6 is 0: Continuous transmission is complete. Data of 2 bytes can be written. When doing so, observe Caution 1 above.**

3. While continuous transmission is in progress, check that TXSF6 is 0 after the transmission completion interrupt, and then execute clearing (POWER6 = 0 or TXE6 = 0). If clearing 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 15-8. 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_{XCLK})
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 (TO50)
Other				Setting prohibited

Caution 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: X1 input clock oscillation frequency

(5) Baud rate generator control register 6 (BRGC6)

This register selects the base clock 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 15-9. 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_{XCLK}/8
0	0	0	0	1	0	0	1	9	f_{XCLK}/9
0	0	0	0	1	0	1	0	10	f_{XCLK}/10
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	f_{XCLK}/252
1	1	1	1	1	1	0	1	253	f_{XCLK}/253
1	1	1	1	1	1	1	0	254	f_{XCLK}/254
1	1	1	1	1	1	1	1	255	f_{XCLK}/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_{XCLK}: Frequency of base clock (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 transfer operations of serial interface UART6.

ASICL6 can be set by a 1-bit transfer instruction or an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to16H.

Remark 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, transfer is started by refresh because bit 6 (SBRT6) and bit 5 (SBTT6) of ASICL6 are cleared to 0 when communication is complete (when an interrupt signal is generated).

Figure 15-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (1/2)

Address: FF58H After reset: 16H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

Cautions 1. In the case of an SBF reception error, return the mode to the SBF reception mode and hold the status of the SBRF6 flag.

2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1.

3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1.

5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

Figure 15-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (2/2)

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bits length.
1	1	0	SBF is output with 14-bits length.
1	1	1	SBF is output with 15-bits length.
0	0	0	SBF is output with 16-bits length.
0	0	1	SBF is output with 17-bits length.
0	1	0	SBF is output with 18-bits length.
0	1	1	SBF is output with 19-bits length.
1	0	0	SBF is output with 20-bits length.

DIR6	MSB/LSB-first transfer
0	MSB-first transfer
1	LSB-first transfer

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

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

15.4 Operation of Serial Interface UART6

This section explains the two modes of serial interface UART6.

15.4.1 Operation stop mode

In this mode, serial transfer cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode.

(1) Register setting

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.

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).

Address: FF50H After reset: 01H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL	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.
1[Note 2]	Enables operation of the internal operation clock.

TXE6	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).
1	Enables transmission

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

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. Operation of the internal operation clock is enabled at the second input clock after 1 is written to the POWER6 bit.

Cautions 1. At startup, set POWER6 to 1 and then set TXE6 to 1. Clear TXE6 to 0 first, and then clear POWER6 to 0.

2. At startup, set POWER6 to 1 and then set RXE6 to 1. Clear RXE6 to 0 first, and then clear POWER6 to 0.

15.4.2 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) Register setting

The UART mode is set by 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), and asynchronous serial interface control register 6 (ASICL6).

(a) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial transfer operations of serial interface UART6.

ASIM6 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).

Address: FF50H After reset: 01H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	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.
1[Note 2]	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. Operation of the internal operation clock is enabled at the second input clock after 1 is written to the POWER6 bit.

Caution At startup, set POWER6 to 1 and then set TXE6 to 1. Clear TXE6 to 0 first, and then clear POWER6 to 0.

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 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 RXE6 to 1. Clear RXE6 to 0 first, and then clear POWER6 to 0.

2. Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.

3. Fix the PS61 and PS60 bits to 0 when mounting the device on LIN.

4. 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.

5. Make sure that RXE6 = 0 when rewriting the ISRM6 bit.

(b) 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 can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$ input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0. 00H is read when this register is read.

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 RXB 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. Do not read data from ASIS6 when the CPU is operating on the subsystem clock and the X1 input clock is stopped. For details, refer to CHAPTER 32 CAUTIONS FOR WAIT.

(c) 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 can be set by an 8-bit memory manipulation instruction, and is read-only.

$\overline{\text{RESET}}$ input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0.

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 continuously transmit data, write the data of the first byte to TXB6, check that the value of the TXBF6 flag is 0, and then write the data of the second byte to TXB6. The operation is not guaranteed if data is written to TXB6 while the TXBF6 flag is 1.

2. While continuous transmission is being executed, check the value of the TXSF6 flag after the transmission completion interrupt to determine the subsequent write processing to TXB6.

- **If TXSF6 is 1: Continuous transmission is in progress. Data of 1 byte can be written.**
- **If TXSF6 is 0: Continuous transmission is complete. Data of 2 bytes can be written. When doing so, observe Caution 1 above.**

3. While continuous transmission is in progress, check that TXSF6 is 0 after the transmission completion interrupt, and then execute clearing (POWER6 = 0 or TXE6 = 0). If clearing is executed while the TXSF6 flag is 1, the transmit data cannot be guaranteed.

(d) Asynchronous serial interface control register 6 (ASICL6)

This register controls the serial transfer operations of serial interface UART6.

ASICL6 can be set by a 1-bit transfer instruction or an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to16H.

Remark 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, transfer is started by refresh because bit 6 (SBRT6) and bit 5 (SBTT6) of ASICL6 are cleared to 0 when communication is complete (when an interrupt signal is generated).

Address: FF58H After reset: 16H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

Cautions 1. In the case of an SBF reception error, return the mode to the SBF reception mode and hold the status of the SBRF6 flag.

2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1.

3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1.

5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6	MSB/LSB-first transfer
0	MSB-first transfer
1	LSB-first transfer

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

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

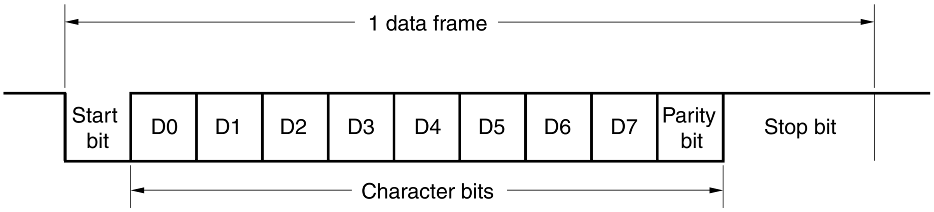
(2) Communication operation

(a) Normal transmit/receive data format

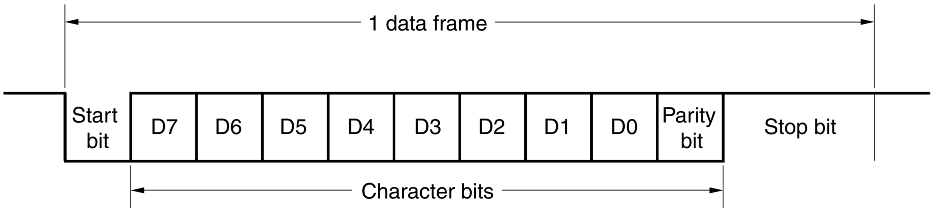
Figure 15-11 shows the format of the transmit/receive data.

Figure 15-11. 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

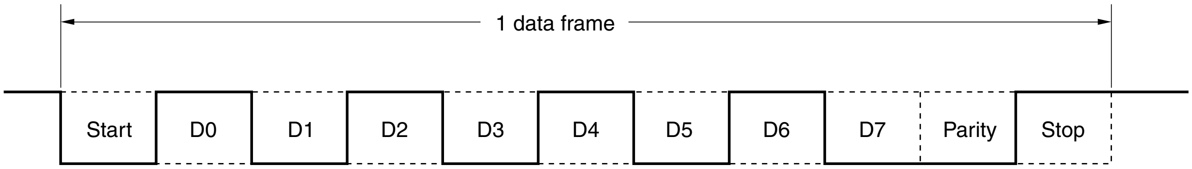
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface mode register 6 (ASIM6).

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

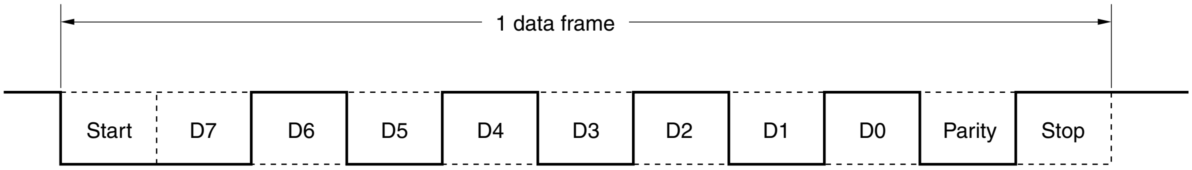
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 15-12. Example of Normal UART Transmit/Receive Data Format

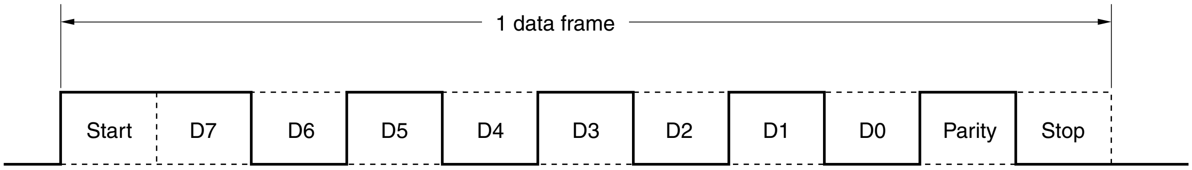
1. **Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H**



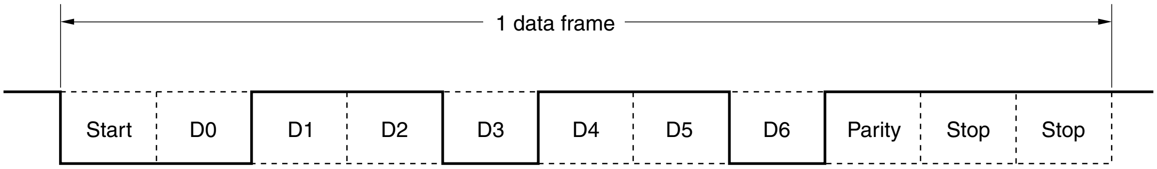
2. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H**



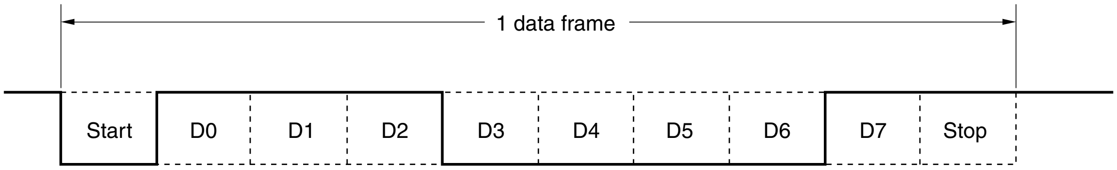
3. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H, TxD6 pin inverted output**



4. **Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Transfer data: 36H**



5. **Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Transfer data: 87H**



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

Caution Fix the PS61 and PS60 bits to 0 when the device is incorporated in LIN.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are "1" is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Normal transmission

The TxD6 pin outputs a high level when bit 7 (POWER6) of asynchronous serial interface 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.

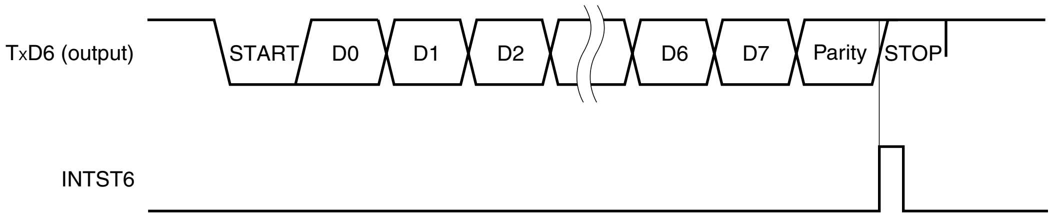
When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the data is sequentially output from TXS6 to the TxD6 pin, starting from the LSB. When transmission is completed, a transmission completion interrupt request (INTST6) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6.

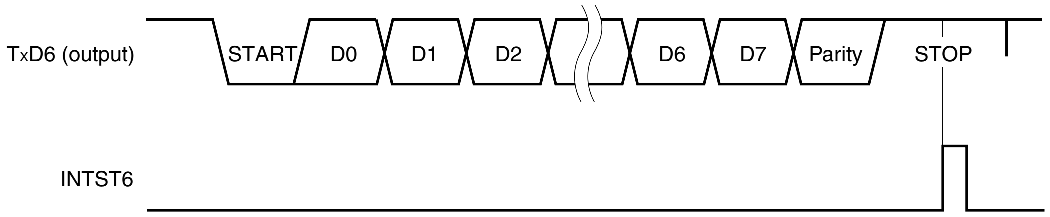
Figure 15-13 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

Figure 15-13. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Continuous transmission

When transmit shift register 6 (TXS6) has started the shift operation, the next transmit data can be written to transmit buffer register 6 (TXB6). As a result, data can be transmitted without intermission even while an interrupt that has occurred after transmission of one data frame is being serviced, thus realizing an efficient communication rate. To transmit data continuously, however, transmission processing must be executed while referencing bits 1 (TXBF6) and 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6).

Caution When the device is incorporated in LIN, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

Table 15-2. Write Processing and Writing to TXB6 During Execution of Continuous Transmission

TXBF6	TXSF6	Write Processing During Execution of Continuous Transmission	Writing to TXB6 During Execution of Continuous Transmission
0	0	Enables writing 2 bytes or transmission completion processing	Enables writing
0	1	Enables writing 1 byte	Enables writing
1	0	Enables writing 2 bytes or transmission completion processing	Disables writing
1	1	Enables writing 1 byte	Disables writing

Cautions 1. To continuously transmit data, write the data of the first byte to TXB6, check that the value of the TXBF6 flag is 0, and then write the data of the second byte to TXB6. The operation is not guaranteed if data is written to TXB6 while the TXBF6 flag is 1.

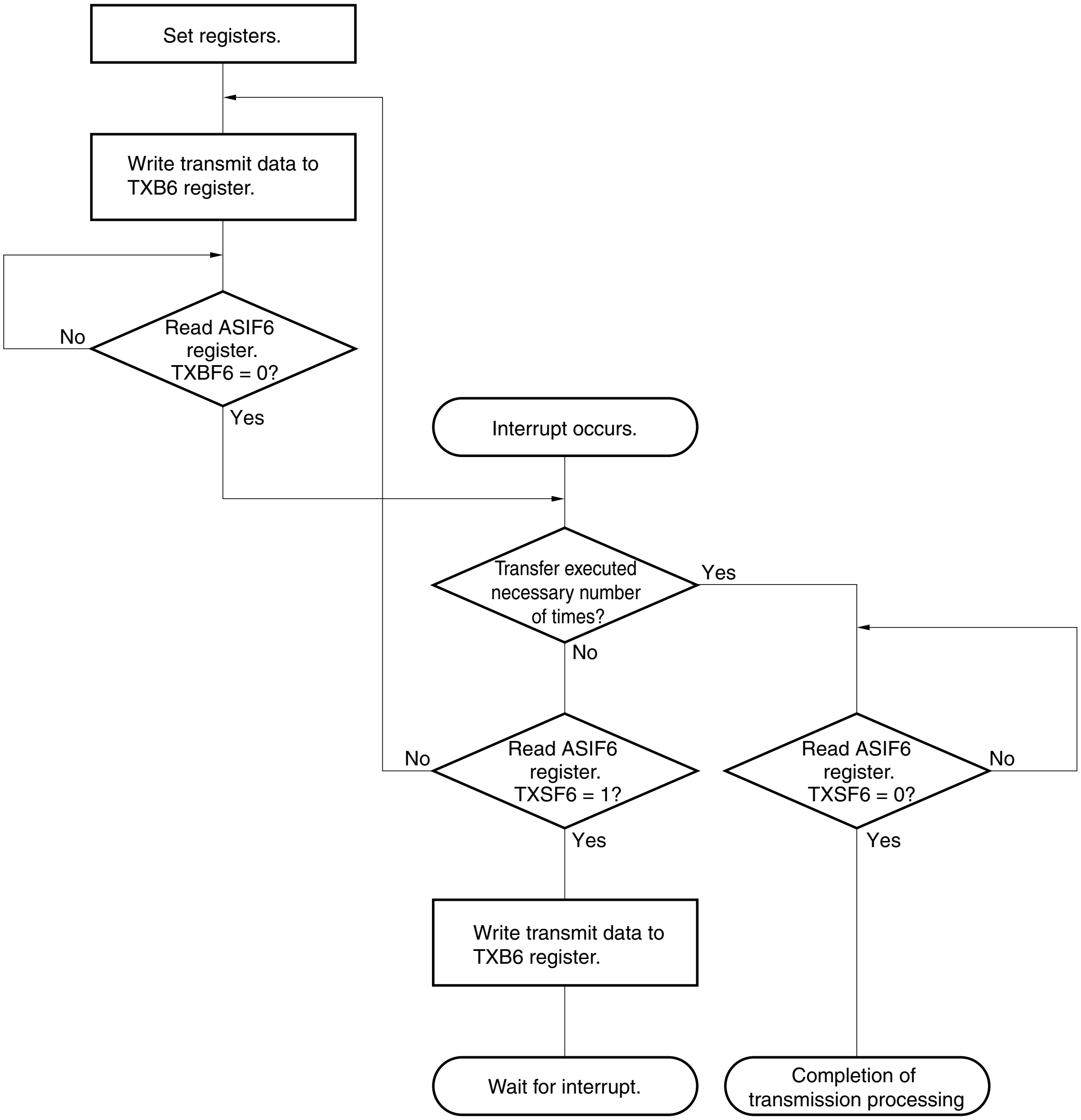
2. While continuous transmission is being executed, check the value of the TXSF6 flag after the transmission completion interrupt to determine the subsequent write processing to TXB6.

- **If TXSF6 is 1: Continuous transmission is in progress. Data of 1 byte can be written.**
- **If TXSF6 is 0: Continuous transmission is completed. Data of 2 bytes can be written. To do so, observe Caution 1 above.**

3. While continuous transmission is in progress, check that TXSF6 is 0 after the transmission completion interrupt, and then execute clearing (POWER6 = 0 or TXE6 = 0). If clearing is executed while the TXSF6 flag is 1, the transmit data cannot be guaranteed.

Figure 15-14 shows the processing flow of continuous transmission.

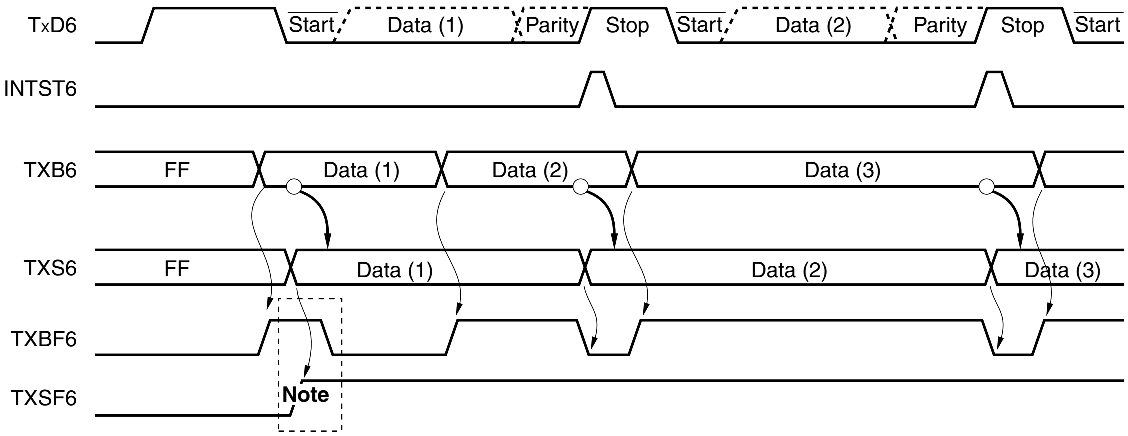
Figure 15-14. Processing Flow of Continuous Transmission


Set registers.
Write transmit data to TXB6 register.
No
Read ASIF6 register. TXBF6 = 0?
Yes
Interrupt occurs.
Transfer executed necessary number of times?
Yes
No
No
Read ASIF6 register. TXSF6 = 1?
Read ASIF6 register. TXSF6 = 0?
No
Yes
Yes
Write transmit data to TXB6 register.
Wait for interrupt.
Completion of transmission processing

Remark	TXB6:	Transmit buffer register 6
	ASIF6:	Asynchronous serial interface transmission status register 6
	TXBF6:	Bit 1 of ASIF6 (transmit buffer data flag)
	TXSF6:	Bit 0 of ASIF6 (transmit shift register data flag)

Figure 15-15 shows the timing of starting continuous transmission, and Figure 15-16 shows the timing of ending continuous transmission.

Figure 15-15. Timing of Starting Continuous Transmission

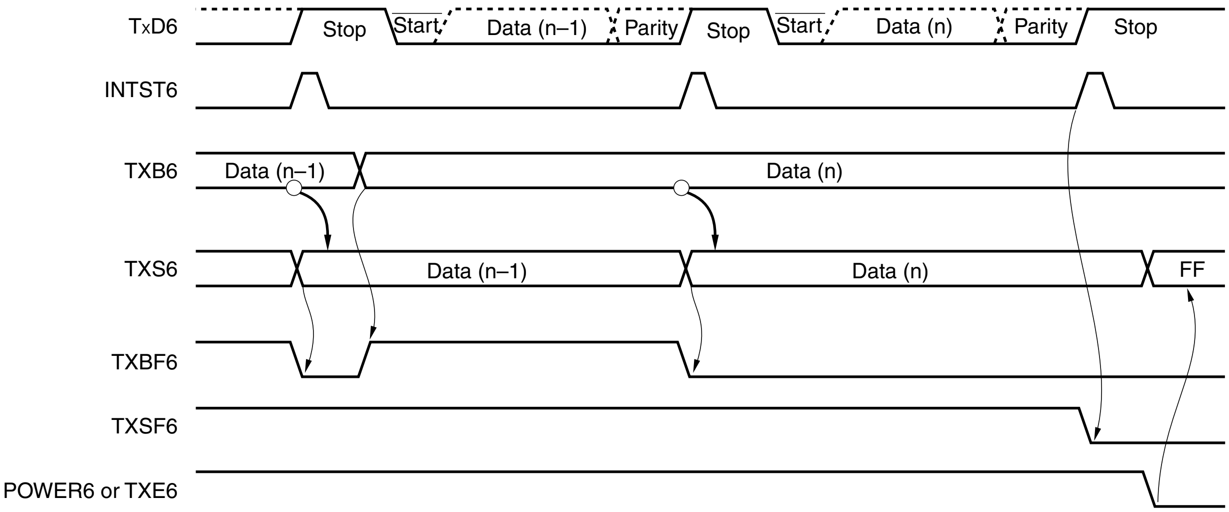


Note When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

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 15-16. 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 mode register (ASIM6)

TXE6: Bit 6 of asynchronous serial interface mode register (ASIM6)

(e) Normal reception

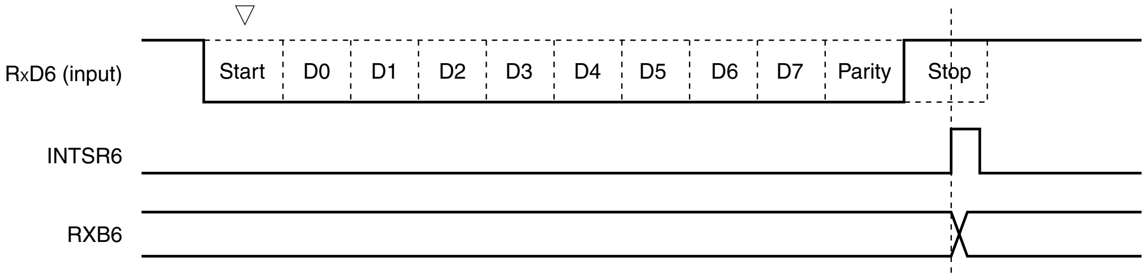
Reception is enabled and the RxD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

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 15-17). 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) or a framing error (FE6) 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 15-17. Reception Completion Interrupt Request Timing



Cautions 1. Be sure to read receive buffer register 6 (RXB6) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the “number of stop bits = 1”. The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 6 (ASIS6) before reading RXB6.

(f) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6 (ASIS6) is set as a result of data reception, a reception error interrupt request (INTSR6/INTSRE6) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6 in the reception error interrupt servicing (INTSR6/INTSRE6) (refer to **Table 15-3**).

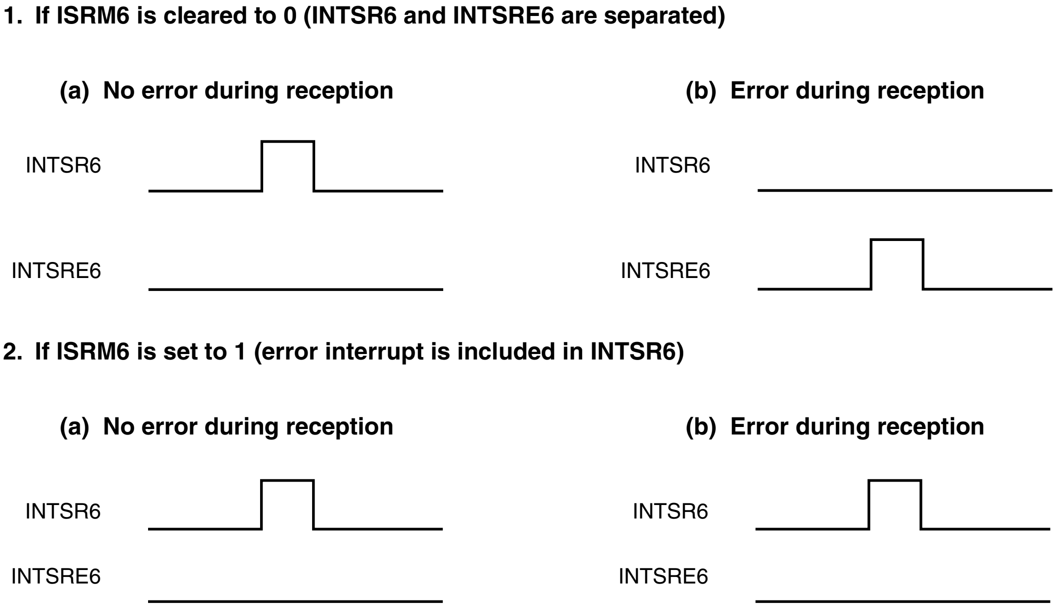
The contents of ASIS6 are reset to 0 when ASIS6 is read.

Table 15-3. Cause of Reception Error

Reception Error	Cause	Value of ASIS6
Parity error	The parity specified for transmission does not match the parity of the receive data.	04H
Framing error	Stop bit is not detected.	02H
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).	01H

The error interrupt can be separated into INTSR6 and INTSRE6 by clearing bit 0 (ISRM6) of asynchronous serial interface mode register 6 (ASIM6) to 0.

Figure 15-18. Reception Error Interrupt



1. If ISRM6 is cleared to 0 (INTSR6 and 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

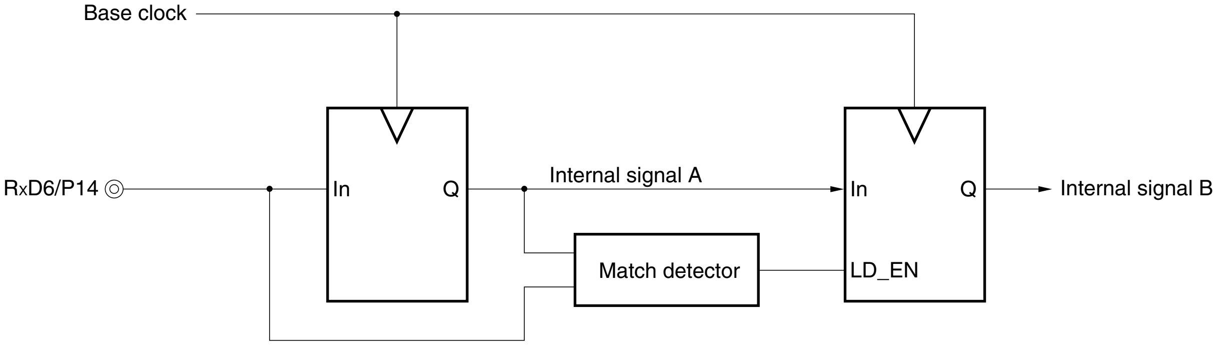
(g) Noise filter of receive data

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 15-19, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 15-19. Noise Filter Circuit



(h) SBF transmission

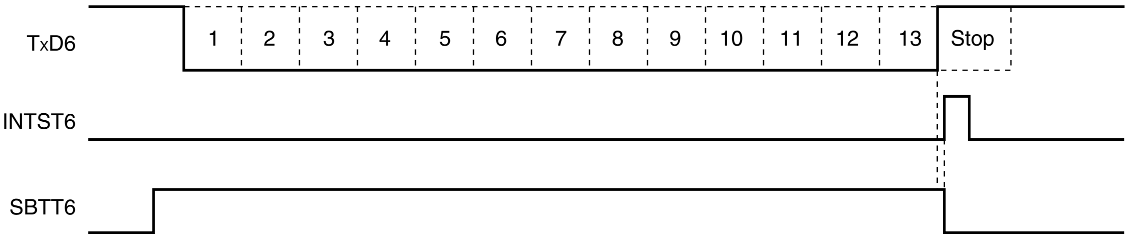
When the device is incorporated in LIN, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, refer to **Figure 15-1 LIN Transmission Operation**.

The TxD6 pin outputs a high level when bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1. Transmission is enabled when bit 6 (TXE6) of ASIM6 is set to 1 next time, and SBF transmission operation is started when bit 5 (SBTT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1.

After transmission has been started, the low levels of bits 13 to 20 (set by bits 4 to 2 (SBL62 to SBL60) of ASICL6) are output. When SBF transmission has been completed, a transmission completion interrupt request (INTST6) is generated, and SBTT6 is automatically cleared. After SBF transmission has been completed, the normal transmission mode is restored.

Transmission is stopped until the data to be transmitted next is written to transmit buffer register 6 (TXB6) or SBTT6 is set to 1.

Figure 15-20. SBF Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Transmission completion interrupt request
SBTT6: Bit 5 of asynchronous serial interface control register 6 (ASICL6)

(i) SBF reception

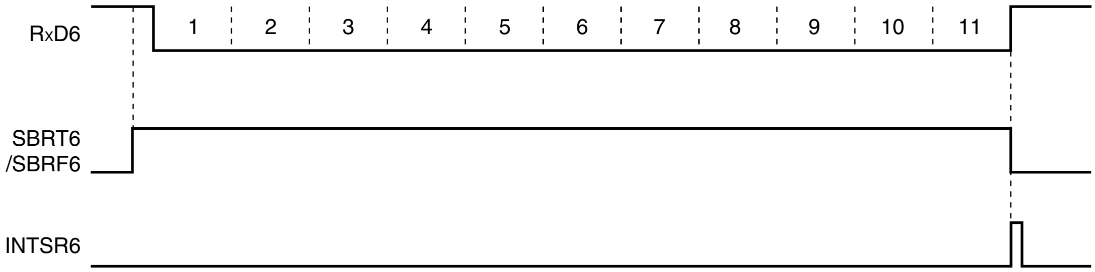
When the device is incorporated in LIN, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, refer to **Figure 15-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6) of asynchronous serial interface 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.

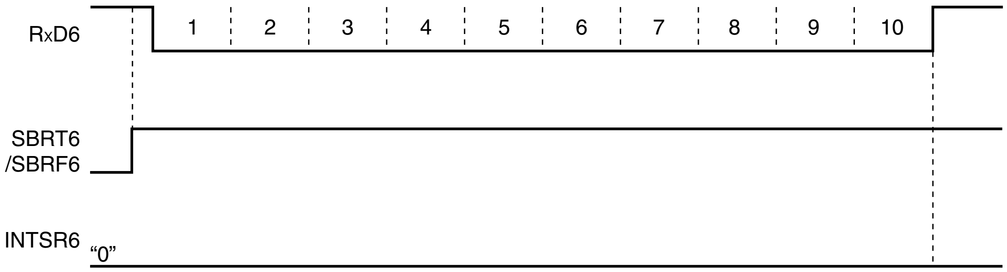
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal 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 15-21. 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

15.4.3 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 (Clock)

  The clock selected by bits 3 to 0 (TPS63 to TPS60) of clock selection register 6 (CKSR6) is supplied to each module when bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is 1. This clock is called the base clock (Clock) and its frequency is called f_{XCLK}. 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 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.

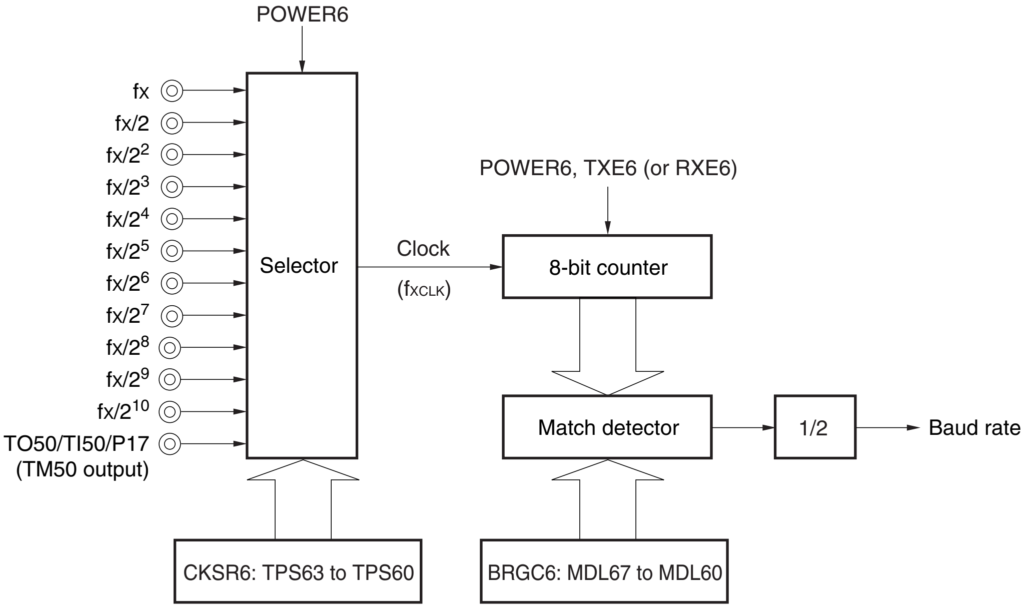
- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 5 (RXE6) of asynchronous serial interface 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 15-22. Configuration of Baud Rate Generator



Remark	POWER6:	Bit 7 of asynchronous serial interface 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) 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).

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_{XCLK})
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
Other				Setting prohibited

Caution 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: X1 input clock oscillation frequency

(b) Baud rate generator control register 6 (BRGC6)

This register selects the base clock 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).

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	$f_{XCLK}/8$
0	0	0	0	1	0	0	1	9	$f_{XCLK}/9$
0	0	0	0	1	0	1	0	10	$f_{XCLK}/10$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	1	1	1	1	0	0	252	$f_{XCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK}/255$

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_{XCLK}: Frequency of base clock (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

(c) Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK}}{2 \times k}$ [bps]

f_{XCLK}: Frequency of base clock (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)

(d) 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 (Clock) = 20 MHz = 20,000,000 Hz
Set value of MDL67 to MDL60 bits of BRGC6 register = 01000001B (k = 65)
Target baud rate = 153600 bps

Baud rate = 20 M/(2 × 65)
= 20000000/(2 × 65) = 153,846 [bps]

Error = (153846/153600 − 1) × 100
= 0.160 [%]

(3) Example of setting baud rate

Table 15-4. 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	19200	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	116388	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	232777	1.03	0H	9	232778	1.03

Caution The maximum permissible frequency (f_{XCLK}) of the base clock is 25 MHz.

Remark TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock (f_{XCLK}))

k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) (k = 8, 9, 10, ..., 255)

f_X: X1 input clock oscillation frequency

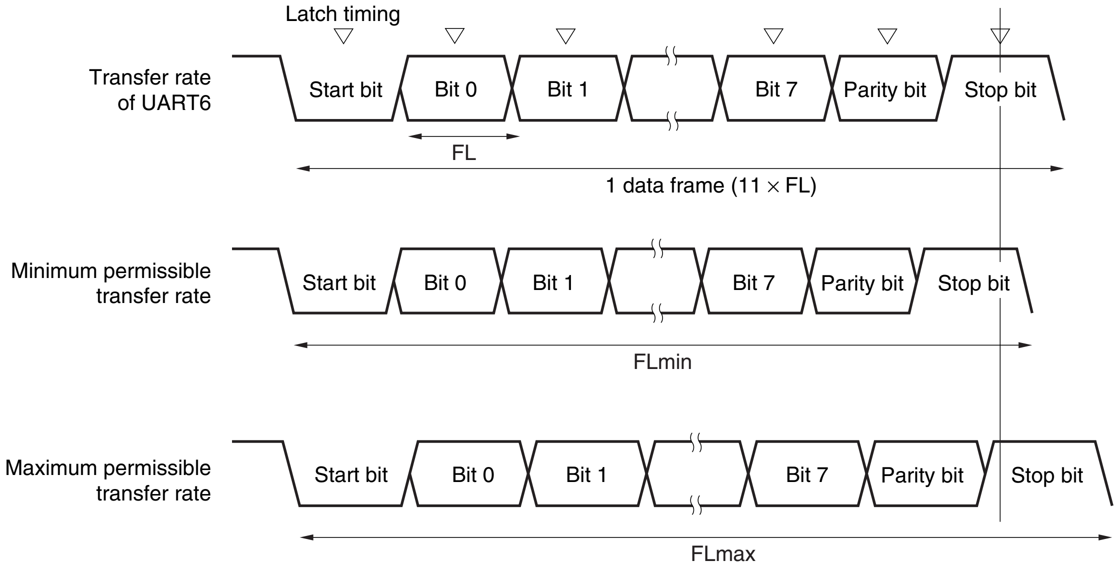
ERR: Baud rate error

(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 15-23. Permissible Baud Rate Range During Reception



As shown in Figure 15-23, 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 transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k+2}\ \text{Brate}$$

Similarly, the maximum permissible transfer rate can be calculated as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2 \times k} \times \text{FL} = \frac{21k-2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k-2}{20k}\ \text{FL} \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2}\ \text{Brate}$$

The permissible baud rate error between UART6 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 15-5. 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%

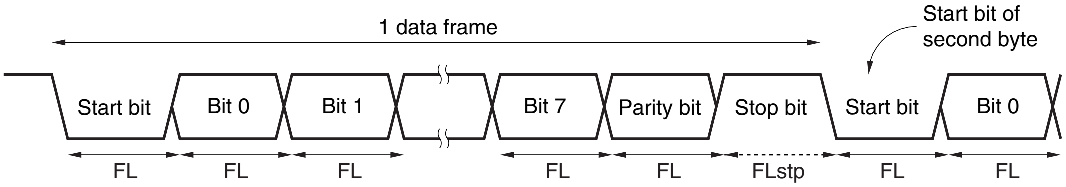
Remarks 1. The accuracy of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the accuracy.

2. k: Set value of BRGC6

(5) Transfer rate during continuous transmission

When data is continuously transmitted, the transfer rate from a stop bit to the next start bit is extended by two clocks from the normal value. However, the result of transfer is not affected because the timing is initialized on the reception side when the start bit is detected.

Figure 15-24. Transfer Rate During Continuous Transmission



Where the 1-bit data length is FL, the stop bit length is FLstp, and base clock frequency is f_{XCLK}, the following expression is satisfied.

$$FLstp = FL + 2/f_{XCLK}$$

Therefore, the transfer rate during continuous transmission is:

$$\text{Transfer rate} = 11 \times FL + 2/f_{XCLK}$$

CHAPTER 16 SERIAL INTERFACES CSI10 AND CSI11

The µPD780143 and 780144 incorporate serial interface CSI10, and the µPD780146, 780148, and 78F0148 incorporate serial interfaces CSI10 and CSI11.

16.1 Functions of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11[Note] have the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not performed and can enable a reduction in the power consumption.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to transfer 8-bit data using three lines: a serial clock line ($\overline{\text{SCK1n}}$) and two serial data lines (SI1n and SO1n).

The processing time of data transfer 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 transferred 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 is useful for connecting peripheral I/Os and display controllers with a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

Note µPD780146, 780148, and 78F0148 only

16.2 Configuration of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 consist of the following hardware.

Table 16-1. Configuration of Serial Interfaces CSI10 and CSI11

Item	Configuration
Registers	Transmit buffer register 1n (SOTB1n) Serial I/O shift register 1n (SIO1n)
Control registers	Serial operation mode register 1n (CSIM1n) Serial clock selection register 1n (CSIC1n)

Remark n = 0: µPD780143, 780144
n = 0, 1: µPD780146, 780148, 78F0148

Figure 16-1. Block Diagram of Serial Interface CSI10

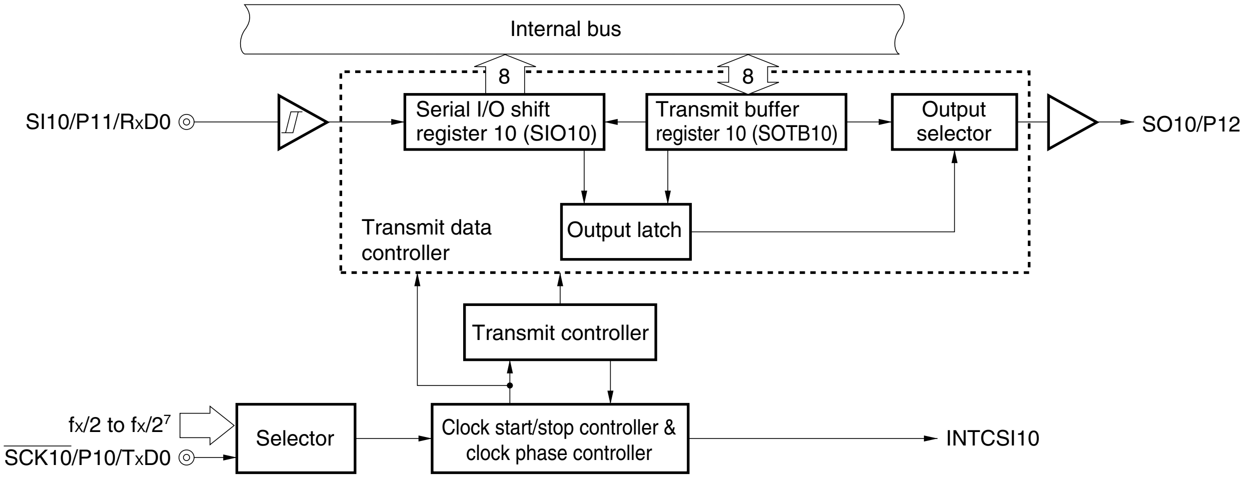
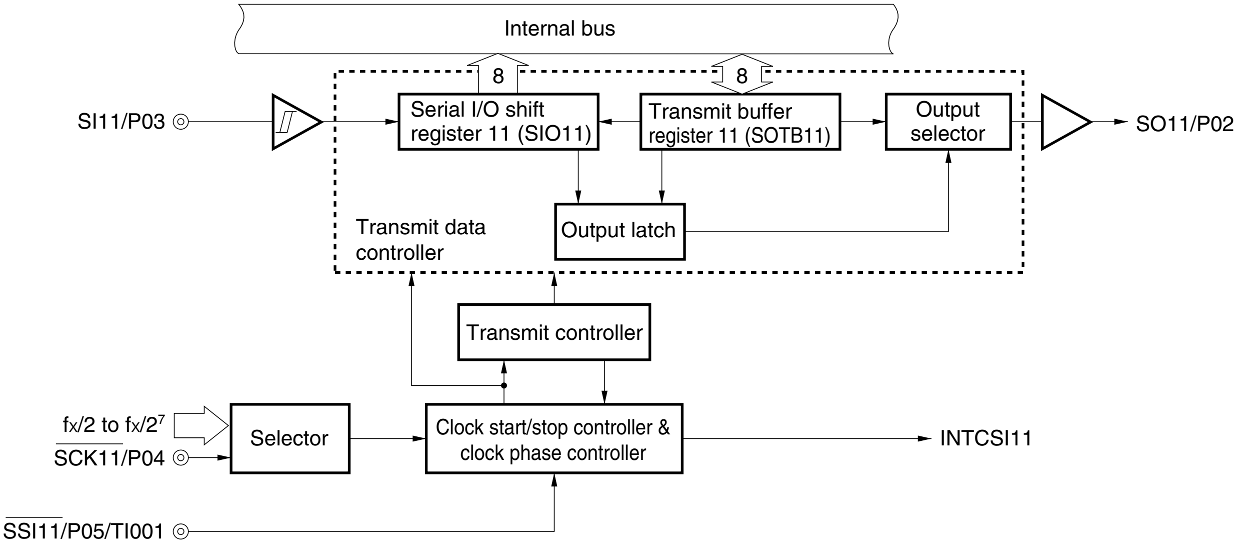


Figure 16-2. Block Diagram of Serial Interface CSI11 (μPD780146, 780148, and 78F0148 Only)



(1) Transmit buffer register 1n (SOTB1n)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB1n when bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1.

The data written to SOTB1n is converted from parallel data into serial data by serial I/O shift register 1n, and output to the serial output pin (SO1n).

SOTB1n can be written or read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

Caution Do not access SOTB1n when CSOT1n = 1 (during serial communication).

Remarks 1. When using serial interface CSI11 in the slave mode, the operation is as follows if bit 5 (SSE11) of serial operation mode register 11 (CSIM11) is 1.

(1) If $\overline{\text{SSI11}}$ is low
... This chip is selected and transmission is started by writing data to SOTB11.

(2) If $\overline{\text{SSI11}}$ is high
... Transmission is not started even if data is written to SOTB11 because this chip is not selected (transmission held pending).

(3) If data is written to SOTB11 when transmission is held pending because $\overline{\text{SSI11}}$ is high and then $\overline{\text{SSI11}}$ goes low
... Transmission is started.

(4) If $\overline{\text{SSI11}}$ goes high after transmission has been started by writing data to SOTB11 when $\overline{\text{SSI11}}$ is low
... Transmission is aborted.

2. n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

(2) Serial I/O shift register 1n (SIO1n)

This is an 8-bit register that converts data from parallel data into serial data and vice versa.

This register can be read by an 8-bit memory manipulation instruction.

Reception is started by reading data from SIO1n if bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

During reception, the data is read from the serial input pin (SI1n) to SIO1n.

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

Caution Do not access SIO1n when CSOT1n = 1 (during serial communication).

Remark When using serial interface CSI11 in the slave mode, the operation is as follows if bit 5 (SSE11) of serial operation mode register 11 (CSIM11) is 1.

(1) If $\overline{\text{SSI11}}$ is low
... This chip is selected and reception is started by reading data from SIO11.

(2) If $\overline{\text{SSI11}}$ is high
... Reception is not started even if data is read from SIO11 because this chip is not selected (reception held pending).

(3) If data is read from SIO11 when reception is held pending because $\overline{\text{SSI11}}$ is high and then $\overline{\text{SSI11}}$ goes low
... Reception is started.

(4) If $\overline{\text{SSI11}}$ goes high after reception has been started by reading data from SIO11 when $\overline{\text{SSI11}}$ is low
... Reception is aborted.

16.3 Registers Controlling Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 are controlled by the following two registers.

- Serial operation mode register 1n (CSIM1n)
- Serial clock selection register 1n (CSIC1n)

(1) Serial operation mode register 1n (CSIM1n)

CSIM1n is used to select the operation mode and enable or disable operation.

CSIM1n 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: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 16-3. Format of Serial Operation Mode Register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	$\boxed{7}$	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	Stops operation (SI10/P11/RxD0, SO10/P12, and $\overline{\text{SCK10}}$/P10/TxD0 pins can be used as general-purpose port pins).
1	Enables operation (SI10/P11/RxD0, SO10/P12, and $\overline{\text{SCK10}}$/P10/TxD0 pins are at active level).

TRMD10[Note 2]	Transmit/receive mode control
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10[Note 4]	First bit specification
0	MSB
1	LSB

CSOT10[Note 5]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

3. The SO10 pin is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

4. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

5. CSOT10 is cleared if CSIE10 is set to 0 (operation stopped).

Caution Be sure to set bit 5 to 0.

Figure 16-4. Format of Serial Operation Mode Register 11 (CSIM11)

Address: FF88H After reset: 00H R/W[Note 1]

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11	DIR11	0	0	0	CSOT11

CSIE11	Operation control in 3-wire serial I/O mode
0	Stops operation (SI11/P03, SO11/P02, and $\overline{\text{SCK11}}$/P04 pins can be used as general-purpose port pins).
1	Enables operation (SI11/P03, SO11/P02, and $\overline{\text{SCK11}}$/P04 pins are at active level).

TRMD11[Note 2]	Transmit/receive mode control
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

SSE11[Notes 4, 5]	$\overline{\text{SSI11}}$ pin use selection
0	$\overline{\text{SSI11}}$ pin is not used
1	$\overline{\text{SSI11}}$ pin is used

DIR11[Note 6]	First bit specification
0	MSB
1	LSB

CSOT11[Note 7]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD11 when CSOT11 = 1 (during serial communication).

3. The SO11 pin is fixed to the low level when TRMD11 is 0. Reception is started when data is read from SIO11.

4. Do not rewrite SSE11 when CSOT11 = 1 (during serial communication).

5. Before setting this bit to 1, fix the $\overline{\text{SSI11}}$ pin input level to 0 or 1.

6. Do not rewrite DIR11 when CSOT11 = 1 (during serial communication).

7. CSOT11 is cleared if CSIE11 is set to 0 (operation stopped).

(2) Serial clock selection register 1n (CSIC1n)

CSIC1n is used to select the phase of the data clock and set the count clock.

CSIC1n can be set by a 1-bit or 8-bit memory manipulation instruction.

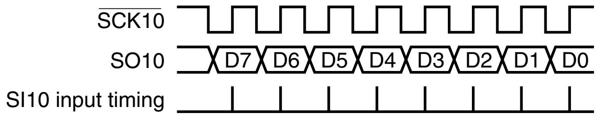
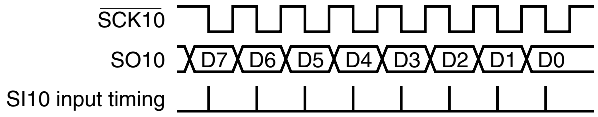
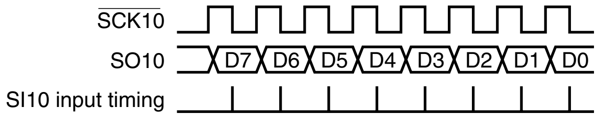
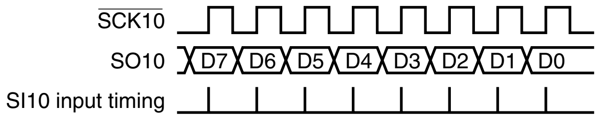
$\overline{\text{RESET}}$ input sets this register to 00H.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 16-5. 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	Data clock phase selection	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 count clock selection
0	0	0	fx/2 (5 MHz)
0	0	1	fx/2^2 (2.5 MHz)
0	1	0	fx/2^3 (1.25 MHz)
0	1	1	fx/2^4 (625 kHz)
1	0	0	fx/2^5 (312.5 kHz)
1	0	1	fx/2^6 (156.25 kHz)
1	1	0	fx/2^7 (78.13 kHz)
1	1	1	External clock input to $\overline{\text{SCK10}}$

Cautions 1. Do not write CSIC10 during a communication operation or when using P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, and P12/SO10 as general-purpose port pins.

2. The phase type of the data clock is type 1 after reset.

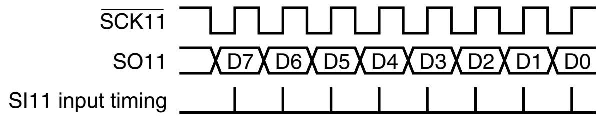
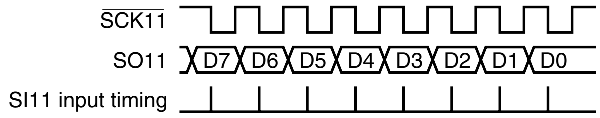
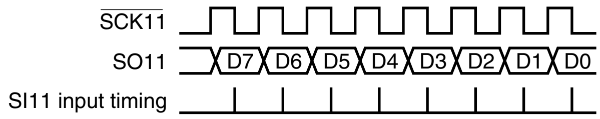
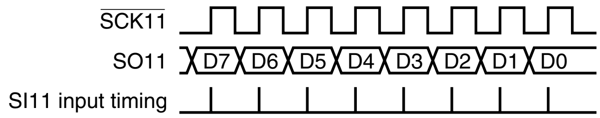
Remarks 1. Figures in parentheses are for operation with fx = 10 MHz

2. fx: X1 input clock oscillation frequency

Figure 16-6. Format of Serial Clock Selection Register 11 (CSIC11)

Address: FF89H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC11	0	0	0	CKP11	DAP11	CKS112	CKS111	CKS110

CKP11	DAP11	Data clock phase selection	Type
0	0	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	1
0	1	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	2
1	0	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	3
1	1	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	4

CKS112	CKS111	CKS110	CSI11 count clock selection
0	0	0	fx/2 (5 MHz)
0	0	1	fx/2^2 (2.5 MHz)
0	1	0	fx/2^3 (1.25 MHz)
0	1	1	fx/2^4 (625 kHz)
1	0	0	fx/2^5 (312.5 kHz)
1	0	1	fx/2^6 (156.25 kHz)
1	1	0	fx/2^7 (78.13 kHz)
1	1	1	External clock input to $\overline{\text{SCK11}}$

Cautions 1. Do not write CSIC11 during a communication operation or when using P02/SO11, P03/SI11, and P04/$\overline{\text{SCK11}}$ as general-purpose port pins.

2. 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: X1 input clock oscillation frequency

16.4 Operation of Serial Interfaces CSI10 and CSI11

Serial interface CSI10 and CSI11 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

16.4.1 Operation stop mode

Serial transfer is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, P12/SO10, P02/SO11[Note], P03/SI11[Note], and P04/$\overline{\text{SCK11}}$[Note] pins can be used as ordinary I/O port pins in this mode.

Note μPD780146, 780148, and 78F0148 only

(1) Register setting

The operation stop mode is set by serial operation mode register 1n (CSIM1n).

(a) Serial operation mode register 1n (CSIM1n)

CSIM1n can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM1n to 00H.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

- Serial operation mode register 10 (CSIM10)

Address: FF80H After reset: 00H R/W

Symbol	$\boxed{7}$	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	Stops operation (SI10/P11/RxD0, SO10/P12, and $\overline{\text{SCK10}}$/P10/TxD0 pins can be used as general-purpose port pins).
1	Enables operation (SI10/P11/RxD0, SO10/P12, and $\overline{\text{SCK10}}$/P10/TxD0 pins are at active level).

- Serial operation mode register 11 (CSIM11)

Address: FF88H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11	DIR11	0	0	0	CSOT11

CSIE11	Operation control in 3-wire serial I/O mode
0	Stops operation (SI11/P03, SO11/P02, and $\overline{\text{SCK11}}$/P04 pins can be used as general-purpose port pins).
1	Enables operation (SI11/P03, SO11/P02, and $\overline{\text{SCK11}}$/P04 pins are at active level).

16.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connecting peripheral I/Os and display controllers that have a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

In this mode, communication is executed by using three lines: the serial clock ($\overline{\text{SCK1n}}$), serial output (SO1n), and serial input (SI1n) lines.

(1) Register setting

The 3-wire serial I/O mode is set by serial operation mode register 1n (CSIM1n) and serial clock selection register 1n (CSIC1n).

(a) Serial operation mode register 1n (CSIM1n)

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

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

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

- Serial operation mode register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	$\boxed{7}$	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	Stops operation (SI10/P11/RxD0, SO10/P12, and $\overline{\text{SCK10}}$/P10/TxD0 pins can be used as general-purpose port pins).
1	Enables operation (SI10/P11/RxD0, SO10/P12, and $\overline{\text{SCK10}}$/P10/TxD0 pins are at active level).

TRMD10[Note 2]	Transmit/receive mode control
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10[Note 4]	First bit specification
0	MSB
1	LSB

CSOT10[Note 5]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

3. The SO10 pin is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

4. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

5. CSOT10 is cleared if CSIE10 is set to 0 (operation stopped).

Caution Be sure to set bit 5 to 0.

- Serial operation mode register 11 (CSIM11)

Address: FF88H After reset: 00H R/W[Note 1]

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11	DIR11	0	0	0	CSOT11

CSIE11	Operation control in 3-wire serial I/O mode
0	Stops operation (SI11/P03, SO11/P02, and $\overline{\text{SCK11}}$/P04 pins can be used as general-purpose port pins).
1	Enables operation (SI11/P03, SO11/P02, and $\overline{\text{SCK11}}$/P04 pins are at active level).

TRMD11[Note 2]	Transmit/receive mode control
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

SSE11[Notes 4, 5]	$\overline{\text{SSI11}}$ pin use selection
0	$\overline{\text{SSI11}}$ pin is not used
1	$\overline{\text{SSI11}}$ pin is used

DIR11[Note 6]	First bit specification
0	MSB
1	LSB

CSOT11[Note 7]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD11 when CSOT11 = 1 (during serial communication).

3. The SO11 pin is fixed to the low level when TRMD11 is 0. Reception is started when data is read from SIO11.

4. Do not rewrite SSE11 when CSOT11 = 1 (during serial communication).

5. Before setting this bit to 1, fix the $\overline{\text{SSI11}}$ pin input level to 0 or 1.

6. Do not rewrite DIR11 when CSOT11 = 1 (during serial communication).

7. CSOT11 is cleared if CSIE11 is set to 0 (operation stopped).

(b) Serial clock selection register 1n (CSIC1n)

CSIC1n can be set by a 1-bit or 8-bit memory manipulation instruction.

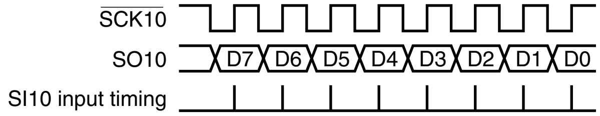
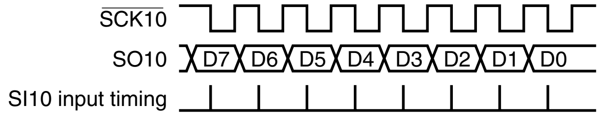
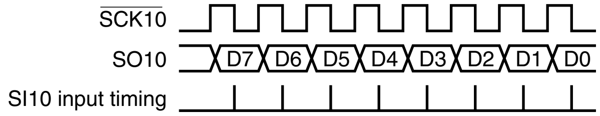
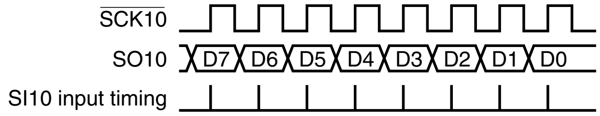
$\overline{\text{RESET}}$ input sets this register to 00H.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

- 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	Data clock phase selection	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 count clock selection
0	0	0	fx/2 (5 MHz)
0	0	1	fx/2^2 (2.5 MHz)
0	1	0	fx/2^3 (1.25 MHz)
0	1	1	fx/2^4 (625 kHz)
1	0	0	fx/2^5 (312.5 kHz)
1	0	1	fx/2^6 (156.25 kHz)
1	1	0	fx/2^7 (78.13 kHz)
1	1	1	External clock input to $\overline{\text{SCK10}}$

Cautions 1. Do not write CSIC10 during a communication operation or when using P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, and P12/SO10 as general-purpose port pins.

2. The phase type of the data clock is type 1 after reset.

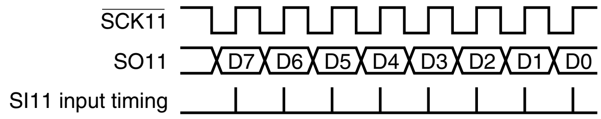
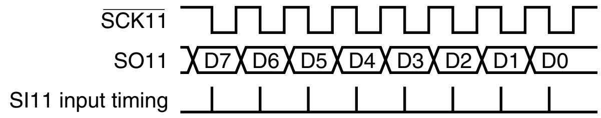
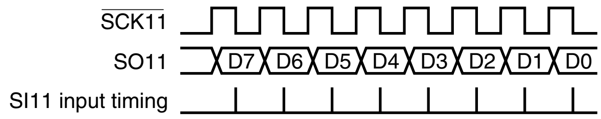
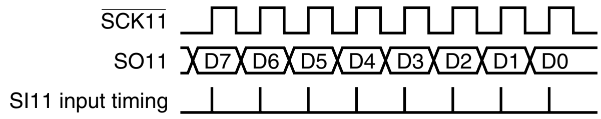
Remarks 1. Figures in parentheses are for operation with fx = 10 MHz

2. fx: X1 input clock oscillation frequency

- Serial clock selection register 11 (CSIC11)

Address: FF89H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC11	0	0	0	CKP11	DAP11	CKS112	CKS111	CKS110

CKP11	DAP11	Data clock phase selection	Type
0	0	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	1
0	1	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	2
1	0	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	3
1	1	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	4

CKS112	CKS111	CKS110	CSI11 count clock selection
0	0	0	fx/2 (5 MHz)
0	0	1	fx/2^2 (2.5 MHz)
0	1	0	fx/2^3 (1.25 MHz)
0	1	1	fx/2^4 (625 kHz)
1	0	0	fx/2^5 (312.5 kHz)
1	0	1	fx/2^6 (156.25 kHz)
1	1	0	fx/2^7 (78.13 kHz)
1	1	1	External clock input to $\overline{\text{SCK11}}$

Cautions 1. Do not write CSIC11 during a communication operation or when using P02/SO11, P03/SI11, and P04/$\overline{\text{SCK11}}$ as general-purpose port pins.

2. 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: X1 input clock oscillation frequency

(2) Setting of ports

- **Serial interface CSI10**

<1> Transmit/receive mode

(a) To use externally input clock as system clock ($\overline{\text{SCK10}}$)

Bit 1 (PM11) of port mode register 1: Set to 1
Bit 2 (PM12) of port mode register 1: Cleared to 0
Bit 0 (PM10) of port mode register 1: Set to 1
Bit 2 (P12) of port 1: Cleared to 0

(b) To use internal clock as system clock ($\overline{\text{SCK10}}$)

Bit 1 (PM11) of port mode register 1: Set to 1
Bit 2 (PM12) of port mode register 1: Cleared to 0
Bit 0 (PM10) of port mode register 1: Cleared to 0
Bit 2 (P12) of port 1: Cleared to 0
Bit 0 (P10) of port 1: Set to 1

<2> Receive mode (with transmission disabled)

(a) To use externally input clock as system clock ($\overline{\text{SCK10}}$)

Bit 1 (PM11) of port mode register 1: Set to 1
Bit 0 (PM10) of port mode register 1: Set to 1

(b) To use internal clock as system clock ($\overline{\text{SCK10}}$)

Bit 1 (PM11) of port mode register 1: Set to 1
Bit 0 (PM10) of port mode register 1: Cleared to 0
Bit 0 (P10) of port 1: Set to 1

Remark The transmit/receive mode or receive mode is selected by using bit 6 (TRMD10) of serial operation mode register 10 (CSIM10).

- **Serial interface CSI11**

<1> Transmit/receive mode

(a) To use externally input clock as system clock ($\overline{\text{SCK11}}$)

Bit 3 (PM03) of port mode register 0: Set to 1
Bit 2 (PM02) of port mode register 0: Cleared to 0
Bit 4 (PM04) of port mode register 0: Set to 1
Bit 2 (P02) of port 0: Cleared to 0

(b) To use internal clock as system clock ($\overline{\text{SCK11}}$)

Bit 3 (PM03) of port mode register 0: Set to 1
Bit 2 (PM02) of port mode register 0: Cleared to 0
Bit 4 (PM04) of port mode register 0: Cleared to 0
Bit 2 (P02) of port 0: Cleared to 0
Bit 4 (P04) of port 0: Set to 1

<2> Receive mode (with transmission disabled)

(a) To use externally input clock as system clock ($\overline{\text{SCK11}}$)

Bit 3 (PM03) of port mode register 0: Set to 1
Bit 4 (PM04) of port mode register 0: Set to 1

(b) To use internal clock as system clock ($\overline{\text{SCK11}}$)

Bit 3 (PM03) of port mode register 0: Set to 1
Bit 4 (PM04) of port mode register 0: Cleared to 0
Bit 4 (P04) of port 0: Set to 1

Remark The transmit/receive mode or receive mode is selected by using bit 6 (TRMD11) of serial operation mode register 11 (CSIM11).

(3) 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 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1. Transmission/reception is started when a value is written to transmit buffer register 1n (SOTB1n). In addition, data can be received when bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

Reception is started when data is read from serial I/O shift register 1n (SIO1n).

When using serial interface CSI11, however, the communication operation is not started if bit 5 (SSE11) of CSIM11 is 1 in the slave mode, and the $\overline{\text{SSI11}}$ pin is at the high level.

After communication has been started, bit 0 (CSOT1n) of CSIM1n is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF1n) is set, and CSOT1n is cleared to 0. Then the next communication is enabled.

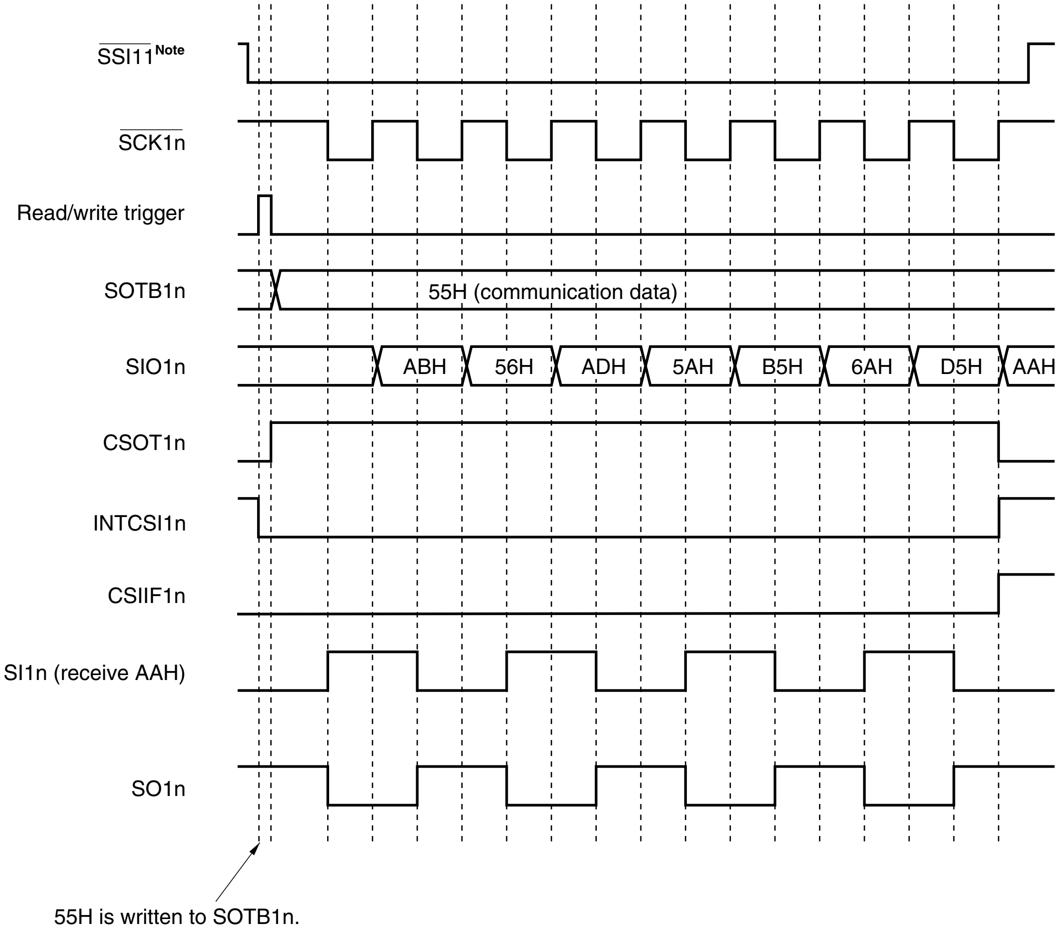
Cautions 1. Do not access the control register and data register when CSOT1n = 1 (during serial communication).

2. When using serial interface CSI11, wait for the duration of at least one clock before the clock operation is started to change the level of the $\overline{\text{SSI11}}$ pin in the slave mode; otherwise, malfunctioning may occur.

Remark n = 0, 1

Figure 16-7. Timing in 3-Wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 0, SSE11 = 1[Note]**)**

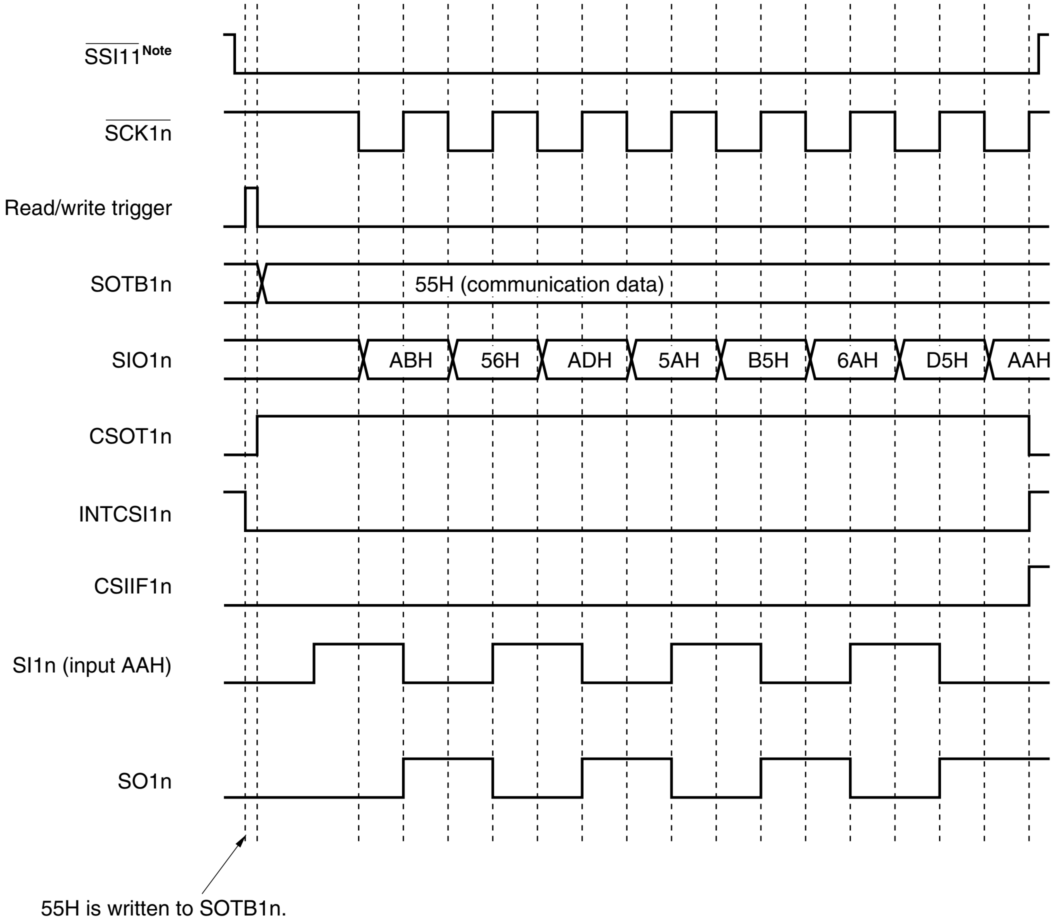


Note The SSE11 flag and $\overline{\text{SSI11}}$ pin are available only for serial interface CSI11, and are used in the slave mode.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 16-7. Timing in 3-Wire Serial I/O Mode (2/2)

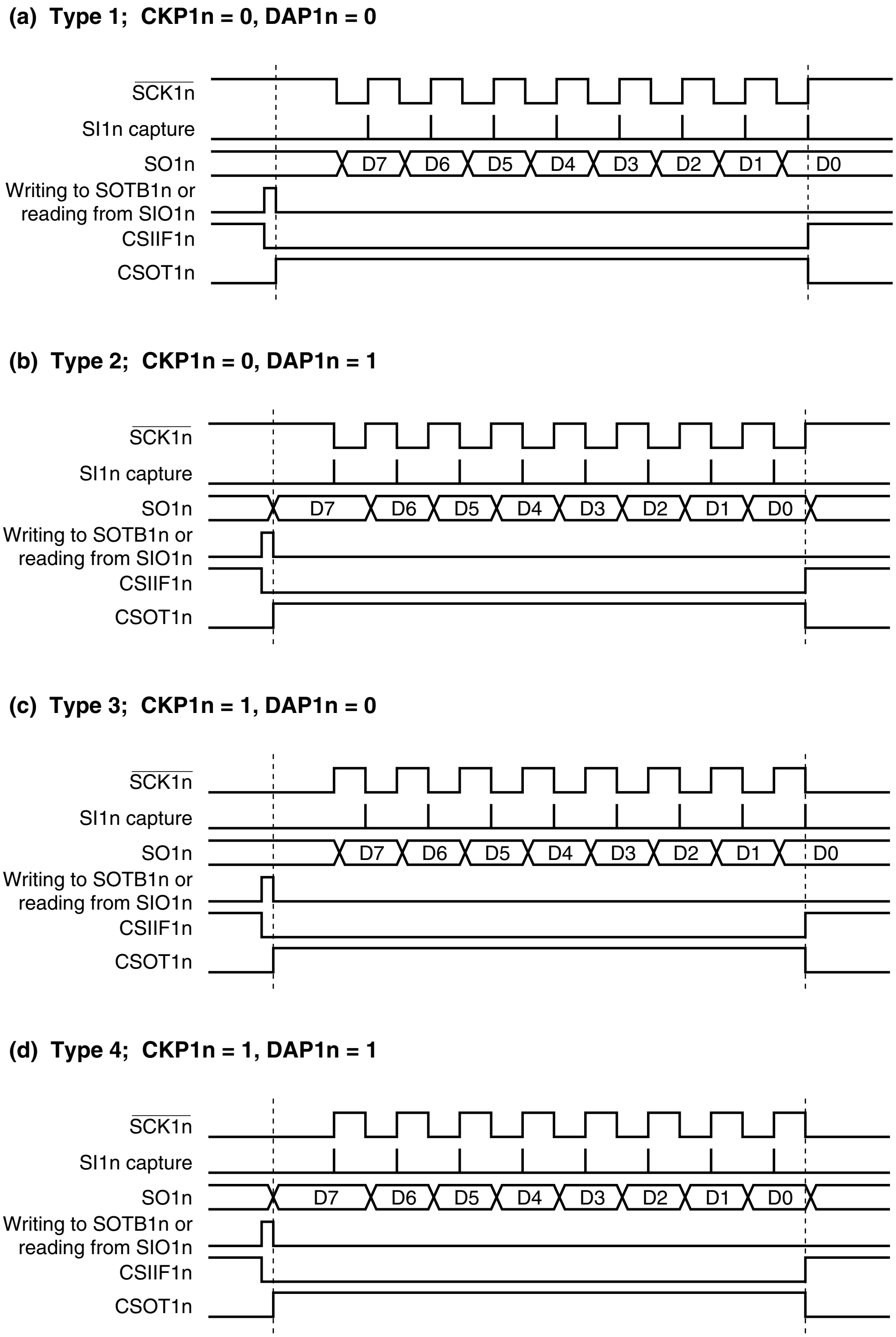
(2) Transmission/reception timing (Type 2; TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 1, SSE11 = 1^{Note})



Note The SSE11 flag and $\overline{SSI11}$ pin are available only for serial interface CSI11, and are used in the slave mode.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

Figure 16-8. Timing of Clock/Data Phase



(a) Type 1; CKP1n = 0, DAP1n = 0

(b) Type 2; CKP1n = 0, DAP1n = 1

(c) Type 3; CKP1n = 1, DAP1n = 0

(d) Type 4; CKP1n = 1, DAP1n = 1

$\overline{\text{SCK1n}}$
SI1n capture
SO1n
Writing to SOTB1n or reading from SIO1n
CSIIF1n
CSOT1n
D7
D6
D5
D4
D3
D2
D1
D0

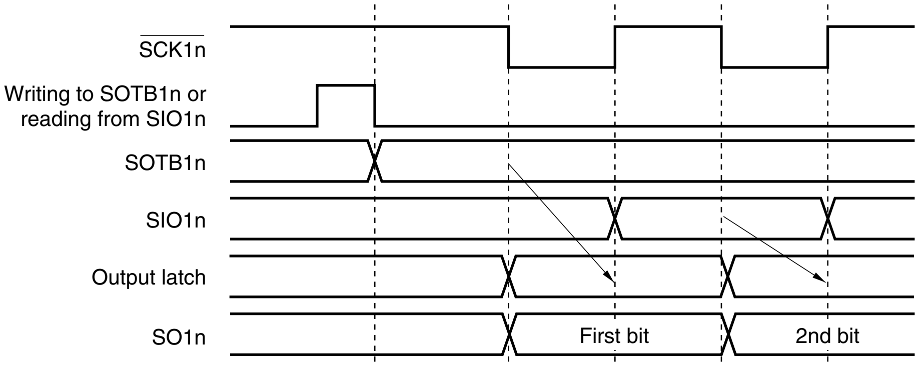
Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

(4) Timing of output to SO1n pin (first bit)

When communication is started, the value of transmit buffer register 1n (SOTB1n) is output from the SO1n pin. The output operation of the first bit at this time is described below.

Figure 16-9. Output Operation of First Bit

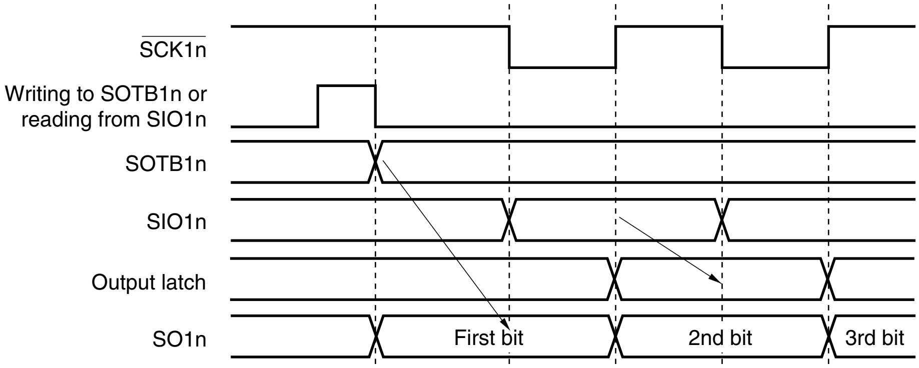
(1) When CKP1n = 0, DAP1n = 0 (or CKP1n = 1, DAP1n = 0)


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
SOTB1n
SIO1n
Output latch
SO1n
First bit
2nd bit

The first bit is directly latched by the SOTB1n register to the output latch at the falling (or rising) edge of $\overline{\text{SCK1n}}$, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next rising (or falling) edge of $\overline{\text{SCK1n}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin.

The second and subsequent bits are latched by the SIO1n register to the output latch at the next falling (or rising) edge of $\overline{\text{SCK1n}}$, and the data is output from the SO1n pin.

(2) When CKP1n = 0, DAP1n = 1 (or CKP1n = 1, DAP1n = 1)


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
SOTB1n
SIO1n
Output latch
SO1n
First bit
2nd bit
3rd bit

The first bit is directly latched by the SOTB1n register at the falling edge of the write signal of the SOTB1n register or the read signal of the SIO1n register, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next falling (or rising) edge of $\overline{\text{SCK1n}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin.

The second and subsequent bits are latched by the SIO1n register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK1n}}$, and the data is output from the SO1n pin.

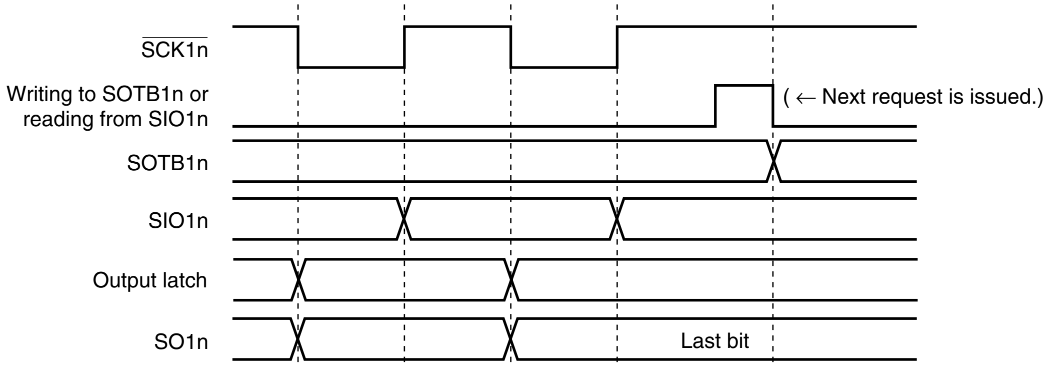
Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

(5) Output value of SO1n pin (last bit)

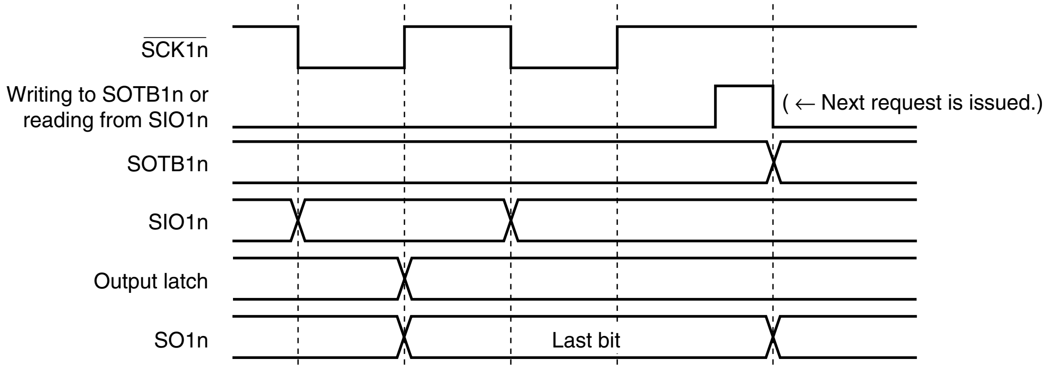
After communication has been completed, the SO1n pin holds the output value of the last bit.

Figure 16-10. Output Value of SO1n Pin (Last Bit)

(1) Type 1; when CKP1n = 0 and DAP1n = 0 (or CKP1n = 1, DAP1n = 0)


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
(← Next request is issued.)
SOTB1n
SIO1n
Output latch
SO1n
Last bit

(2) Type 2; when CKP1n = 0 and DAP1n = 1 (or CKP1n = 1, DAP1n = 1)


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
(← Next request is issued.)
SOTB1n
SIO1n
Output latch
SO1n
Last bit

Remark n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

(6) SO1n pin

The status of the SO1n pin is as follows if bit 7 (CSIE1n) of serial operation mode register 1n (CSIM1n) is cleared to 0.

Table 16-2. SO1n Pin Status

TRMD1n	DAP1n	DIR1n	SO1n Pin
TRMD1n = 0[Note]	–	–	Outputs low level[Note].
TRMD1n = 1	DAP1n = 0	–	Value of SO1n latch (low-level output)
	DAP1n = 1	DIR1n = 0	Value of bit 7 of SOTB1n
		DIR1n = 1	Value of bit 0 of SOTB1n

Note Status after reset

Caution If a value is written to TRMD1n, DAP1n, and DIR1n, the output value of the SO1n pin changes.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

CHAPTER 17 SERIAL INTERFACE CSIA0

17.1 Functions of Serial Interface CSIA0

Serial interface CSIA0 has the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

(1) Operation stop mode

This mode is used when serial transfer is not performed and can enable a reduction in the power consumption.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to transfer 8-bit data using three lines: a serial clock line ($\overline{\text{SCKA0}}$) and two serial data lines (SIA0 and SOA0).

The processing time of data transfer 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 transferred MSB or LSB first can be specified, so this interface can be connected to any device.

(3) 3-wire serial I/O mode with automatic transmit/receive function (MSB/LSB-first selectable)

This mode is used to transfer 8-bit data using three lines: a serial clock line ($\overline{\text{SCKA0}}$) and two serial data lines (SIA0 and SOA0).

The processing time of data transfer 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 transferred MSB or LSB first can be specified, so this interface can be connected to any device.

Data can be transferred to/from a display driver etc. without using software since a 32-byte transfer buffer RAM is incorporated. Also, the incorporation of handshake pins (STB0, BUSY0) has made connection to peripheral LSIs easy.

- Master mode/slave mode selectable
- Transfer data length: 8 bits
- MSB/LSB-first selectable for transfer data
- Automatic transmit/receive function:
  Number of transfer bytes can be specified between 1 and 32
  Transfer interval can be specified (0 to 63 clocks)
  Single transfer/repeat transfer selectable
- On-chip dedicated baud rate generator (6/8/16/32 divisions)
- 3-wire SOA0: Serial data output
  SIA0: Serial data input
  $\overline{\text{SCKA0}}$: Serial clock I/O
- Handshake function incorporated STB0: Strobe output
  BUSY0: Busy input
- Transmission/reception completion interrupt: INTACSI
- Internal 32-byte buffer RAM

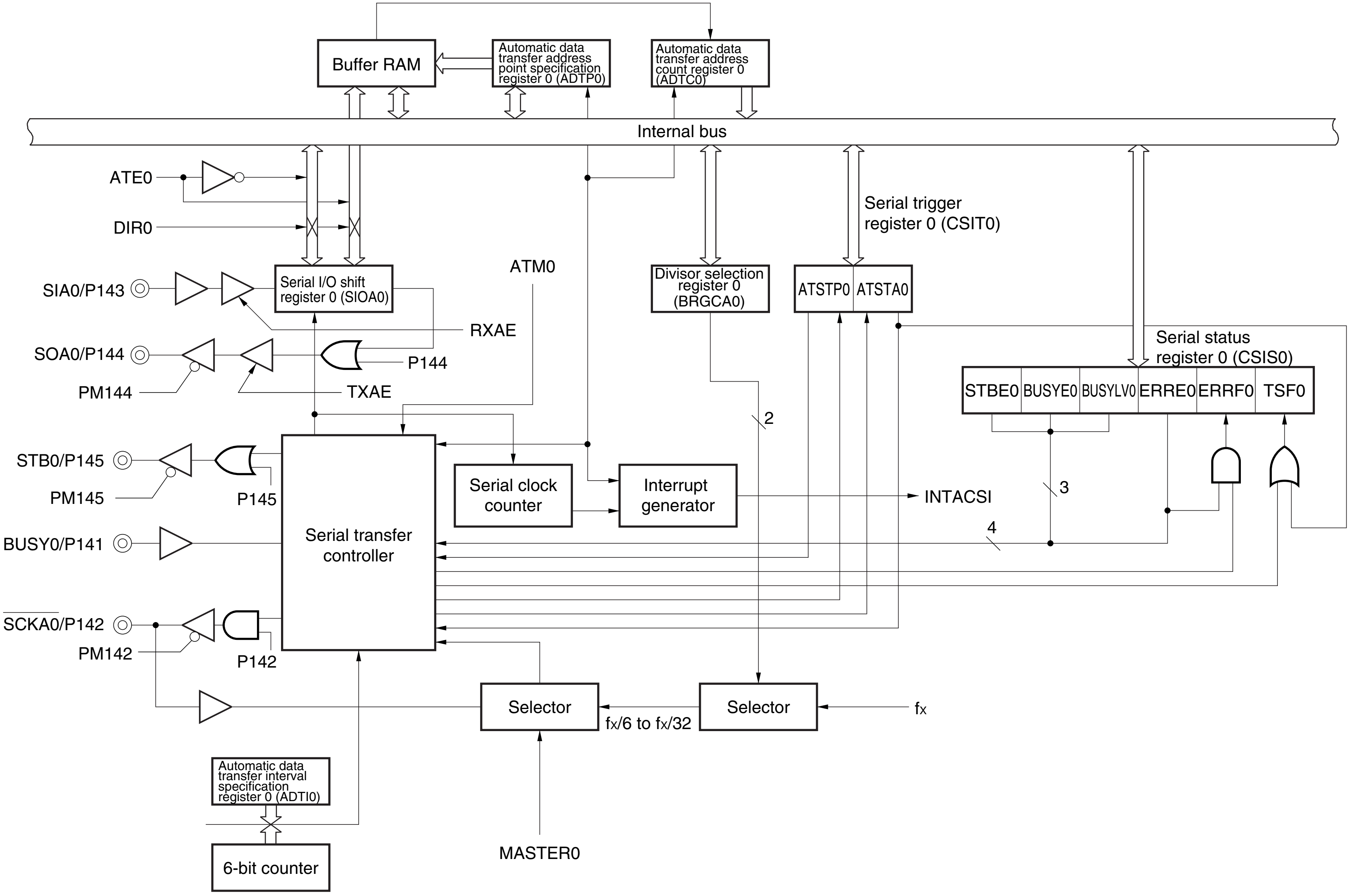
17.2 Configuration of Serial Interface CSIA0

Serial interface CSIA0 consists of the following hardware.

Table 17-1. Configuration of Serial Interface CSIA0

Item	Configuration
Register	Serial I/O shift register 0 (SIOA0) Automatic data transfer address count register 0 (ADTC0)
Control registers	Serial operation mode specification register 0 (CSIMA0) Serial status register 0 (CSIS0) Serial trigger register 0 (CSIT0) Divisor selection register 0 (BRGCA0) Automatic data transfer address point specification register 0 (ADTP0) Automatic data transfer interval specification register 0 (ADTI0)

Figure 17-1. Block Diagram of Serial Interface CSIA0


Buffer RAM
Automatic data transfer address point specification register 0 (ADTP0)
Automatic data transfer address count register 0 (ADTC0)
Internal bus
ATE0
DIR0
SIA0/P143
Serial I/O shift register 0 (SIOA0)
ATM0
RXAE
SOA0/P144
P144
PM144
TXAE
STB0/P145
PM145
P145
BUSY0/P141
SCKA0/P142
PM142
P142
Serial transfer controller
Serial clock counter
Interrupt generator
INTACSI
Divisor selection register 0 (BRGCA0)
2
Serial trigger register 0 (CSIT0)
ATSTP0
ATSTA0
Serial status register 0 (CSIS0)
STBE0
BUSYE0
BUSYLV0
ERRE0
ERRF0
TSF0
3
4
Selector
Selector
fx/6 to fx/32
fx
MASTER0
Automatic data transfer interval specification register 0 (ADTI0)
6-bit counter

(1) Serial I/O shift register 0 (SIOA0)

This is an 8-bit register used to store transmit/receive data in 1-byte transfer mode (bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 0). Writing transmit data to SIOA0 starts the transfer. In addition, after a transfer completion interrupt request (INTACSI) is output (bit 0 (TSF0) of serial status register 0 (CSIS0) = 0), data can be received by reading data from SIOA0.

This register can be written or read by an 8-bit memory manipulation instruction. However, writing to SIOA0 is prohibited when bit 0 (TSF0) of serial status register 0 (CSIS0) = 1

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

Cautions 1. A transfer operation is started by writing to SIOA0. Consequently, when transmission is disabled (bit 3 (TXEA0) of CSIMA0 = 0), write dummy data to the SIOA0 register to start the transfer operation, and then perform a receive operation.

2. Do not write data to SIOA0 while the automatic transmit/receive function is operating.

(2) Automatic data transfer address count register 0 (ADTC0)

This is a register used to indicate buffer RAM addresses during automatic transfer. When automatic transfer is stopped, the data position when transfer stopped can be ascertained by reading ADTC0 register value.

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

$\overline{\text{RESET}}$ input sets this register to 00H. However, reading from ADTC0 is prohibited when bit 0 (TSF0) of serial status register 0 (CSIS0) = 1.

Figure 17-2. Format of Automatic Data Transfer Address Count Register 0 (ADTC0)

Address: FF97H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADTC0	0	0	0	ADTC04	ADTC03	ADTC02	ADTC01	ADTP00

17.3 Registers Controlling Serial Interface CSIA0

Serial interface CSIA0 is controlled by the following six registers.

- Serial operation mode specification register 0 (CSIMA0)
- Serial status register 0 (CSIS0)
- Serial trigger register 0 (CSIT0)
- Divisor selection register 0 (BRGCA0)
- Automatic data transfer address point specification register 0 (ADTP0)
- Automatic data transfer interval specification register 0 (ADTI0)

(1) Serial operation mode specification register 0 (CSIMA0)

This is an 8-bit register used to control the serial transfer 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 17-3. Format of Serial Operation Mode Specification Register 0 (CSIMA0)

Address: FF90H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	0
CSIMA0	CSIAE0	ATE0	ATM0	MASTER0	TXEA0	RXEA0	DIR0	0

CSIAE0	Control of CSIA0 operation enable/disable
0	CSIA0 operation disabled (SOA0: Low level, $\overline{\text{SCKA0}}$: High level)
1	CSIA0 operation enabled

ATE0	Control of automatic transfer operation enable/disable
0	1-byte transfer mode
1	Automatic transfer mode

ATM0	Automatic transfer mode specification
0	Single transfer mode (stops at the address specified by the ADTP0 register)
1	Repeat transfer mode (after transfer is complete, clear the ADTC0 register to 00H to resume transfer)

MASTER0	CSIA0 master/slave mode specification
0	Slave mode (synchronous with $\overline{\text{SCKA0}}$ input clock)
1	Master mode (synchronous with internal clock)

TXEA0	Control of transmit operation enable/disable
0	Transmit operation disabled (SOA0: Low level)
1	Transmit operation enabled

RXEA0	Control of receive operation enable/disable
0	Receive operation disabled
1	Receive operation enabled

DIR0	First bit specification
0	MSB
1	LSB

Cautions 1. When CSIAE0 = 0, the buffer RAM cannot be accessed.

2. When CSIAE0 is changed from 1 to 0, all the registers of the CSIA0 unit are initialized asynchronously. To set CSIAE0 = 1 again, be sure to re-set the registers of the CSIA0 unit.

3. When CSIAE0 is re-set to 1 after CSIAE0 is changed from 1 to 0, it is not guaranteed that the value of the buffer RAM will be retained.

(2) Serial status register 0 (CSIS0)

This is an 8-bit register used to select the input clock and to control the transfer operation of CSIA0.

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

$\overline{\text{RESET}}$ input sets this register to 00H. However, rewriting CSIS0 is prohibited when bit 0 (TSF0).

Figure 17-4. Format of Serial Status Register 0 (CSIS0) (1/2)

Address: FF91H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CSIS0	0	0	STBE0	BUSYE0	BUSYLV0	ERRE0	ERRF0	TSF0

STBE0[Notes 2, 3]	Strobe output enable/disable
0	Strobe output disabled
1	Strobe output enabled

BUSYE0	Busy signal detection enable/disable
0	Busy signal detection disabled (input via BUSY0 pin is ignored)
1	Busy signal detection enabled and transfer wait by busy signal is executed

BUSYLV0[Note 4]	Busy signal active level setting
0	Low level
1	High level

Notes 1. Bits 0 and 1 are read-only.

2. STBE0 is valid only in master mode. In slave mode, 1-byte transfer ends after eight transfer clocks regardless of the STBE0 set value.

3. When STBE0 is set to 1, two transfer clocks are consumed between byte transfers regardless of the setting of automatic data transfer interval specification register 0 (ADTI0). That is, 10 transfer clocks are used for 1-byte transfer even if ADTI0 = 00H is set.

4. In bit error detection by busy input, the active level specified by BUSYLV0 is detected.

Caution Be sure to set bits 6 and 7 to 0.

Figure 17-4. Format of Serial Status Register 0 (CSIS0) (2/2)

ERRE0[Note]	Bit error detection enable/disable
0	Error detection disabled
1	Error detection enabled

ERRF0	Bit error detection flag
0	• Bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) = 0 • At reset input • When transfer is started by setting bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) to 1 or writing to SIOA0.
1	Bit error detected (when ERRE0 = 1, the level specified by BUSYLV0 during the data bit transfer period is detected via BUSY0 pin input).

TSF0	Transfer status detection flag
0	• Bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) = 0 • At reset input • At the end of the specified transfer • When transfer is stopped by setting bit 1 (ATSTP0) of serial trigger register 0 (CSIT0) to 1
1	From the transfer start to the end of the specified transfer

Note The ERRE0 setting is valid even when BUSYE0 = 0.

Caution When TSF0 is 1, rewriting serial operation mode specification register 0 (CSIMA0), serial status register 0 (CSIS0), divisor selection register 0 (BRGCA0), automatic data transfer address point specification register 0 (ADTP0), automatic data transfer interval specification register 0 (ADTI0), and serial I/O shift register 0 (SIOA0) are prohibited. However, these registers can be read and re-written to the same value. In addition, the buffer RAM can be rewritten during transfer.

(3) Serial trigger register 0 (CSIT0)

This is an 8-bit register used to control execution/stop of automatic data transfer.

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

$\overline{\text{RESET}}$ input sets this register to 00H. However, manipulate only when bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) is 1 (manipulation prohibited when ATE0 = 0).

Figure 17-5. Format of Serial Trigger Register 0 (CSIT0)

Address: FF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
CSIT0	0	0	0	0	0	0	ATSTP0	ATSTA0

ATSTP0	Automatic data transfer stop
0	Normal mode
1	Automatic data transfer stopped

ATSTA0	Automatic data transfer start
0	Normal mode
1	Automatic data transfer started

Cautions 1. Even if ATSTP0 or ATSTA0 is set to 1, automatic transfer cannot be started/stopped until 1-byte transfer is complete.

2. ATSTP0 and ATSTA0 retain 1 until immediately before the interrupt signal INTACSI is generated, and then change to 0 automatically.

3. After automatic data transfer is stopped, the data address when the transfer stopped is stored in automatic data transfer address count register 0 (ADTC0). However, since no function to restart automatic data transfer is incorporated, when transfer is stopped by setting ATSTP0 = 1, start automatic data transfer by ATSTA0 after re-setting the registers.

(4) Divisor selection register 0 (BRGCA0)

This is an 8-bit register used to control the serial transfer speed (divisor of CSIA input clock).

This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting BRGCA0 is prohibited.

Figure 17-6. Format of Divisor Selection Register 0 (BRGCA0)

Address: FF93H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
BRGCA0	0	0	0	0	0	0	BRGCA01	BRGCA00

BRGCA01	BRGCA00	CSIA0 input clock (fw) division ratio selection
0	0	fw/6 (1.67 MHz)
0	1	fw/2^3 (1.25 MHz)
1	0	fw/2^4 (625 kHz)
1	1	fw/2^5 (312.5 kHz)

Remark Figures in parentheses apply to operation with fw = 10 MHz, fw = fx (X1 input clock oscillation frequency).

(5) Automatic data transfer address point specification register 0 (ADTP0)

This is an 8-bit register used to specify the buffer RAM address that ends transfer during automatic data transfer (bit 6 (ATE0) of serial operation mode specification register 0 = 1).

This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting ADTP0 is prohibited.

In the 78K0/KF1 Series, 00H to 1FH can be specified because 32 bytes of buffer RAM are incorporated.

Example When ADTP0 is set to 07H
8 bytes of 00H to 07H are transferred.

In repeat transfer mode (bit 5 (ATM0) of CSIMA0 = 1), transfer is performed repeatedly up to the address value set in ADTP0.

Example When 07H is transferred to ADTP0 (repeat transfer mode)
Transfer is repeated as 00H to 07H, 00H to 07H, … .

Figure 17-7. Format of Automatic Data Transfer Address Point Specification Register 0 (ADTP0)

Address: FF94H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTP0	0	0	0	ADTP04	ADTP03	ADTP02	ADTP01	ADTP00

Caution Be sure to set bits 7 to 5 to 0.

The relationship between buffer RAM address values and ADTP0 setting values is shown below.

Table 17-2. Relationship Between Buffer RAM Address Values and ADTP0 Setting Values

Buffer RAM Address Value	ADTP0 Setting Value	Buffer RAM Address Value	ADTP0 Setting Value
FA00H	00H	FA10H	10H
FA01H	01H	FA11H	11H
FA02H	02H	FA12H	12H
FA03H	03H	FA13H	13H
FA04H	04H	FA14H	14H
FA05H	05H	FA15H	15H
FA06H	06H	FA16H	16H
FA07H	07H	FA17H	17H
FA08H	08H	FA18H	18H
FA09H	09H	FA19H	19H
FA0AH	0AH	FA1AH	1AH
FA0BH	0BH	FA1BH	1BH
FA0CH	0CH	FA1CH	1CH
FA0DH	0DH	FA1DH	1DH
FA0EH	0EH	FA1EH	1EH
FA0FH	0FH	FA1FH	1FH

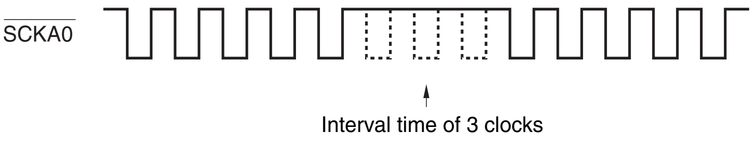
(6) Automatic data transfer interval specification register 0 (ADTI0)

This is an 8-bit register used to specify the interval period between 1-byte transfers during automatic data transfer (bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 1).

Set this register when in master mode (bit 4 (MASTER0) of CSIMA0 = 1) (setting is unnecessary in slave mode). Setting in 1-byte transfer mode (bit 6 (ATE0) of CSIMA0 = 0) is also valid. When the interval time specified by ADTI0 after the end of 1-byte transfer has elapsed, an interrupt request signal (INTACSI) is output. The number of clocks for the interval can be set to between 0 and 63 clocks.

The specified interval time is the transfer clock (specified by divisor selection register 0 (BRGCA0)) multiplied by an integer value.

Example When ADTI0 = 03H



This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting ADTI0 is prohibited.

Figure 17-8. Format of Automatic Data Transfer Interval Specification Register 0 (ADTI0)

Address: FF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTI0	0	0	ADTI05	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00

Caution Because the setting of bit 5 (STBE0) and bit 4 (BUSYE0) of serial status register 0 (CSIS0) takes priority over the ADTI0 setting, the interval time based on the setting of STBE0 and BUSYE0 is generated even when ADTI0 is set to 00H.

Example <1> When STBE = 1, BUSYE = 0: Interval time of two transfer clocks is generated
<2> When STBE = 0, BUSYE = 1: Interval time of one transfer clock is generated
<3> When STBE = 1, BUSYE = 1: Interval time of two transfer clocks is generated

Therefore, setting STBE0 and BUSYE0 to 0 is required to perform no-wait transfer.

17.4 Operation of Serial Interface CSIA0

Serial interface CSIA0 can be used in the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

17.4.1 Operation stop mode

Serial transfer is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P142/$\overline{\text{SCKA0}}$, P143/SIA0, and P144/SOA0 pins can be used as ordinary I/O port pins in this mode.

(1) Register setting

The operation stop mode is set by serial operation mode specification register 0 (CSIMA0).

(a) Serial operation mode specification register 0 (CSIMA0)

This is an 8-bit register used to control the serial transfer 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.

Address: FF90H After reset: 00H R/W

	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	0
CSIMA0	CSIAE0	ATE0	ATM0	MASTER0	TXEA0	RXEA0	DIR0	0

CSIAE0	Control of CSIA0 operation enable/disable
0	CSIA0 operation disabled (SOA0: Low level, $\overline{\text{SCKA0}}$: High level)
1	CSIA0 operation enabled

17.4.2 3-wire serial I/O mode

The one-byte data transmission/reception is executed in the mode in which bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) is set to 0.

The 3-wire serial I/O mode is useful for connecting peripheral I/Os and display controllers having a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

In this mode, communication is executed by using three lines: serial clock ($\overline{\text{SCKA0}}$), serial output (SOA0), and serial input (SIA0) lines.

(1) Register setting

Serial interface CSIA0 is controlled by the following three registers.

- Serial operation mode specification register 0 (CSIMA0)
- Serial status register 0 (CSIS0)
- Divisor selection register 0 (BRGCA0)

(a) Serial operation mode specification register 0 (CSIMA0)

This is an 8-bit register used to control the serial transfer 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.

Address: FF90H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	0
CSIMA0	CSIAE0	ATE0	ATM0	MASTER0	TXEA0	RXEA0	DIR0	0

CSIAE0	Control of CSIA0 operation enable/disable
0	CSIA0 operation disabled (SOA0: Low level, $\overline{\text{SCKA0}}$: High level)
1	CSIA0 operation enabled

ATE0	Control of automatic transfer operation enable/disable
0	1-byte transfer mode
1	Automatic transfer mode

ATM0	Automatic transfer mode specification
0	Single transfer mode (stops at the address specified by the ADTP0 register)
1	Repeat transfer mode (after transfer is complete, clear the ADTC0 register to 00H to resume transfer)

MASTER0	CSIA0 master/slave mode specification
0	Slave mode (synchronous with $\overline{\text{SCKA0}}$ input clock)
1	Master mode (synchronous with internal clock)

TXEA0	Control of transmit operation enable/disable
0	Transmit operation disabled (SOA0: Low level)
1	Transmit operation enabled

RXEA0	Control of receive operation enable/disable
0	Receive operation disabled
1	Receive operation enabled

DIR0	First bit specification
0	MSB
1	LSB

Cautions 1. When CSIAE0 = 0, the buffer RAM cannot be accessed.

2. When CSIAE0 is changed from 1 to 0, all the registers of the CSIA0 unit are initialized asynchronously. To set CSIAE0 = 1 again, be sure to re-set the registers of the CSIA0 unit.

3. When CSIAE0 is re-set to 1 after CSIAE0 is changed from 1 to 0, it is not guaranteed that the value of the buffer RAM will be retained.

(b) Serial status register 0 (CSIS0)

This is an 8-bit register used to select the input clock and to control the transfer operation of CSIA0.

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

$\overline{\text{RESET}}$ input sets this register to 00H. However, rewriting CSIS0 is prohibited when bit 0 (TSF0) is 1.

Address: FF91H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CSIS0	0	0	STBE0	BUSYE0	BUSYLV0	ERRE0	ERRF0	TSF0

STBE0[Notes 2, 3]	Strobe output enable/disable
0	Strobe output disabled
1	Strobe output enabled

BUSYE0	Busy signal detection enable/disable
0	Busy signal detection disabled (input via BUSY0 pin is ignored)
1	Busy signal detection enabled and transfer wait by busy signal is executed

BUSYLV0[Note 4]	Busy signal active level setting
0	Low level
1	High level

Notes 1. Bits 0 and 1 are read-only.

2. STBE0 is valid only in master mode. In slave mode, 1-byte transfer ends after eight transfer clocks regardless of the STBE0 set value.

3. When STBE0 is set to 1, two transfer clocks are consumed between byte transfers regardless of the setting of automatic data transfer interval specification register 0 (ADTI0). That is, 10 transfer clocks are used for 1-byte transfer even if ADTI0 = 00H is set.

4. In bit error detection by busy input, the active level specified by BUSYLV0 is detected.

Caution Be sure to set bits 6 and 7 to 0.

ERRE0[Note]	Bit error detection enable/disable
0	Error detection disabled
1	Error detection enabled

ERRF0	Bit error detection flag
0	• Bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) = 0 • At reset input • When transfer is started by setting bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) to 1 or writing to SIOA0.
1	Bit error detected (when ERRE0 = 1, the level specified by BUSYLV0 during the data bit transfer period is detected via BUSY0 pin input).

TSF0	Transfer status detection flag
0	• Bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) = 0 • At reset input • At the end of the specified transfer • When transfer is stopped by setting bit 1 (ATSTP0) of serial trigger register 0 (CSIT0) to 1
1	From the transfer start to the end of the specified transfer

Note The ERRE0 setting is valid even when BUSYE0 = 0.

Caution When TSF0 is 1, rewriting serial operation mode specification register 0 (CSIMA0), serial status register 0 (CSIS0), divisor selection register 0 (BRGCA0), automatic data transfer address point specification register 0 (ADTP0), automatic data transfer interval specification register 0 (ADTI0), and serial I/O shift register 0 (SIOA0) are prohibited. However, these registers can be read and re-written to the same value. In addition, the buffer RAM can be rewritten during transfer.

(c) Divisor selection register 0 (BRGCA0)

This is an 8-bit register used to control the serial transfer speed (divisor of CSIA input clock).

This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting BRGCA0 is prohibited.

Address: FF93H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
BRGCA0	0	0	0	0	0	0	BRGCA01	BRGCA00

BRGCA01	BRGCA00	CSIA0 input clock (f_W) division ratio selection
0	0	$f_W/6$ (1.67 MHz)
0	1	$f_W/2^3$ (1.25 MHz)
1	0	$f_W/2^4$ (625 kHz)
1	1	$f_W/2^5$ (312.5 kHz)

Remark Figures in parentheses apply to operation with f_W = 10 MHz, f_W = f_X (X1 input clock oscillation frequency)

The relationship between register settings and pins is shown below.

CSIAE0	MASTER0	PM143	P143	PM144	P144	PM142	P142	Serial I/O Shift Register 0 Operation	Serial Clock Counter Operation Control	SIA0/P143 Pin Function	SOA0/P144 Pin Function	$\overline{\text{SCKA0}}$/P142 Pin Function
0	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	Operation stopped	Clear	P143 (CMOS I/O)	P144 (CMOS I/O)	P142 (CMOS I/O)
1	0	1[Note 2]	×[Note 2]	0	0	1	×	Operation enabled	Count operation	SIA0[Note 2] (Input)	SOA0 (CMOS output)	$\overline{\text{SCKA0}}$ (Input)
	1					0	1					$\overline{\text{SCKA0}}$ (CMOS output)

Notes 1. Can be used freely as port function.

2. Can be used as P143 (CMOS I/O) when only transmission is performed (clear bit 2 (RXEA0) of CSIMA0 to 0).

Remark

×:	Don’t care
CSIAE0:	Bit 7 of serial operation mode specification register 0 (CSIMA0)
MASTER0:	Bit 4 of CSIMA0
PM××:	Port mode register
P××:	Port output latch

(3) 1-byte transmission/reception communication operation

(a) 1-byte transmission/reception

When bit 7 (CSIAE0) and bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 1, 0, respectively, if transfer data is written to serial I/O shift register 0 (SIOA0), the data is output via the SOA0 pin in synchronization with the serial clock falling edge, and then input via the SIA0 pin in synchronization with serial clock falling edge, and stored in the SIOA0 register in synchronization with the rising edge 1 clock later.

Data transmission and data reception can be performed simultaneously.

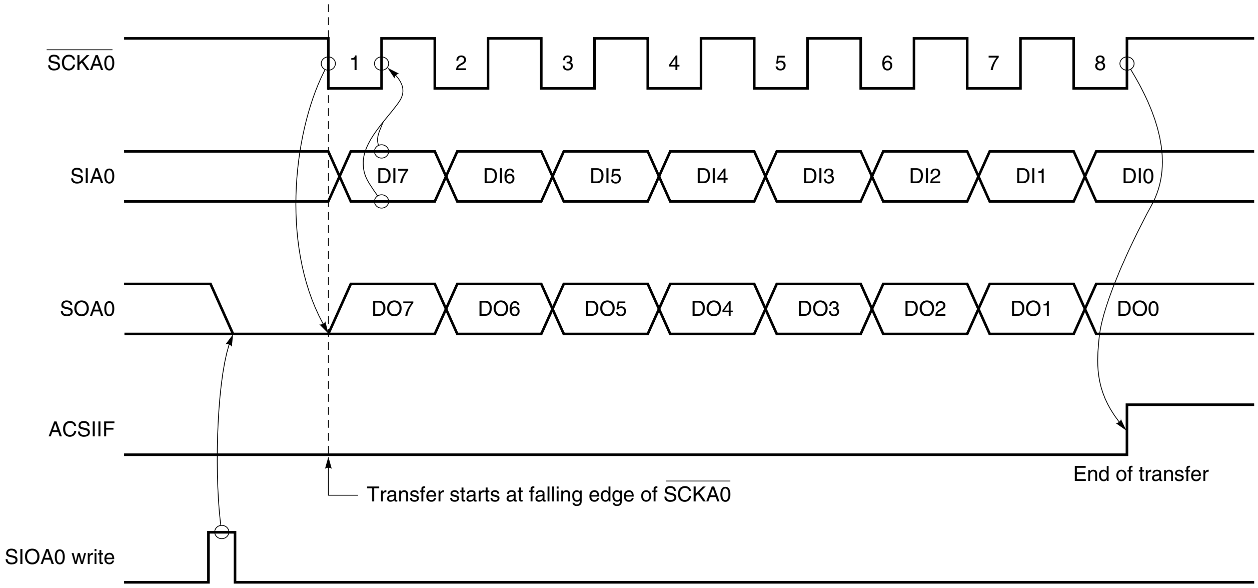
If only reception is to be performed, transfer can only be started by writing a dummy value to the SIOA0 register.

When transfer of 1 byte is complete, an interrupt request signal (INTACSI) is generated.

In 1-byte transmission/reception, the setting of bit 5 (ATM0) of CSIMA0 is invalid.

Be sure to read data after confirming that bit 0 (TSF0) of serial status register 0 (CSIS0) = 0.

Figure 17-9. 3-Wire Serial I/O Mode Timing



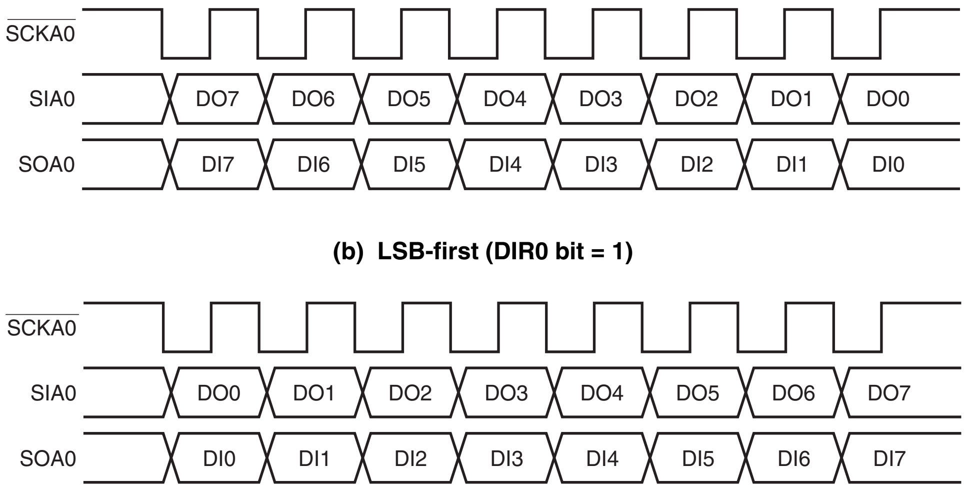
Caution The SOA0 pin becomes low level by an SIOA0 write.

(b) Data format

In the data format, data is changed in synchronization with the $\overline{\text{SCKA0}}$ falling edge as shown below.

The data length is fixed to 8 bits and the data transfer direction can be switched by the specification of bit 1 (DIR0) of serial operation mode specification register 0 (CSIMA0).

Figure 17-10. Format of Transmit/Receive Data



(a) MSB-first (DIR0 bit = 0)

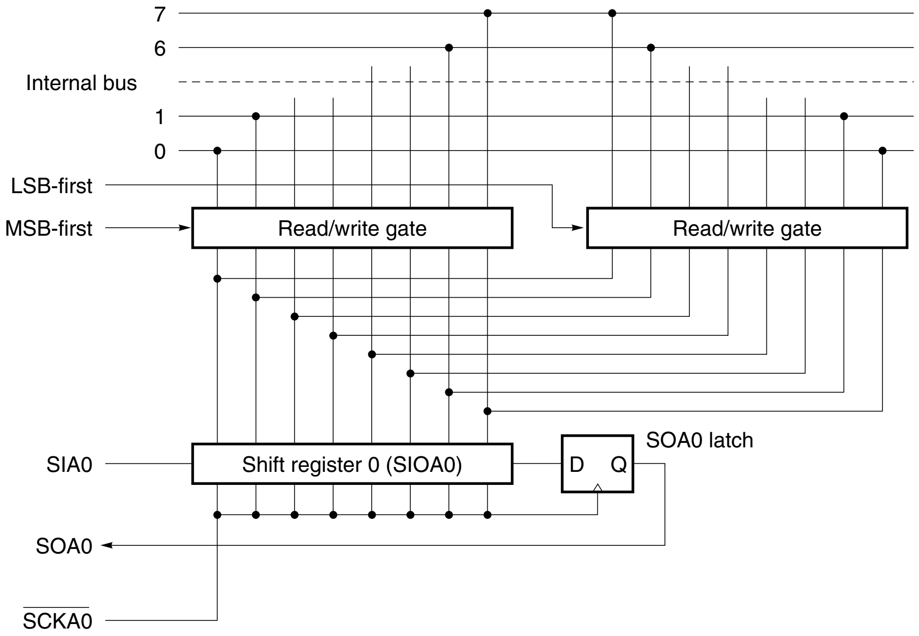
(b) LSB-first (DIR0 bit = 1)

(c) Switching MSB/LSB as start bit

Figure 17-11 shows the configuration of serial I/O shift register 0 (SIOA0) and the internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

Switching MSB/LSB as the start bit can be specified using bit 1 (DIR0) of serial operation mode specification register 0 (CSIMA0).

Figure 17-11. Transfer Bit Order Switching Circuit



Start bit switching is realized by switching the bit order for data written to SIOA0. The SIOA0 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

(d) Transfer start

Serial transfer is started by setting transfer data to serial I/O shift register 0 (SIOA0) when the following two conditions are satisfied.

- Serial interface CSIA0 operation control bit (CSIAE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCKA0}}$ is high level after 8-bit serial transfer.

Caution If CSIAE0 is set to 1 after data is written to SIOA0, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (ACSIIF) is set.

17.4.3 3-wire serial I/O mode with automatic transmit/receive function

Up to 32 bytes of data can be transmitted/received without using software in the mode in which bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) is set to 1. After transfer is started, only data of the set number of bytes stored in RAM in advance can be transmitted, and only data of the set number of bytes can be received and stored in RAM.

In addition, to transmit/receive data continuously, handshake signals (STB0 and BUSY0) generated by hardware are supported. Therefore, connection to peripheral LSIs such as OSD (On Screen Display) LSIs and LCD controller/drivers can be easily realized.

(1) Register setting

Serial interface CSIA0 is controlled by the following six registers.

- Serial operation mode specification register 0 (CSIMA0)
- Serial status register 0 (CSIS0)
- Serial trigger register 0 (CSIT0)
- Divisor selection register 0 (BRGCA0)
- Automatic data transfer address point specification register 0 (ADTP0)
- Automatic data transfer interval specification register 0 (ADTI0)

(a) Serial operation mode specification register 0 (CSIMA0)

This is an 8-bit register used to control the serial transfer 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.

Address: FF90H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	0
CSIMA0	CSIAE0	ATE0	ATM0	MASTER0	TXEA0	RXEA0	DIR0	0

CSIAE0	Control of CSIA0 operation enable/disable
0	CSIA0 operation disabled (SOA0: Low level, $\overline{\text{SCKA0}}$: High level)
1	CSIA0 operation enabled

ATE0	Control of automatic transfer operation enable/disable
0	1-byte transfer mode
1	Automatic transfer mode

ATM0	Automatic transfer mode specification
0	Single transfer mode (stops at the address specified by the ADTP0 register)
1	Repeat transfer mode (after transfer is complete, clear the ADTC0 register to 00H to resume transfer)

MASTER0	CSIA0 master/slave mode specification
0	Slave mode (synchronous with $\overline{\text{SCKA0}}$ input clock)
1	Master mode (synchronous with internal clock)

TXEA0	Control of transmit operation enable/disable
0	Transmit operation disabled (SOA0: Low level)
1	Transmit operation enabled

RXEA0	Control of receive operation enable/disable
0	Receive operation disabled
1	Receive operation enabled

DIR0	First bit specification
0	MSB
1	LSB

Cautions 1. When CSIAE0 = 0, the buffer RAM cannot be accessed.

2. When CSIAE0 is changed from 1 to 0, all the registers of the CSIA0 unit are initialized asynchronously. To set CSIAE0 = 1 again, be sure to re-set the registers of the CSIA0 unit.

3. When CSIAE0 is re-set to 1 after CSIAE0 is changed from 1 to 0, it is not guaranteed that the value of the buffer RAM will be retained.

(b) Serial status register 0 (CSIS0)

This is an 8-bit register used to select the input clock and to control the transfer operation of CSIA0.

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

$\overline{\text{RESET}}$ input sets this register to 00H. However, rewriting CSIS0 is prohibited when bit 0 (TSF0) is 1.

Address: FF91H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CSIS0	0	0	STBE0	BUSYE0	BUSYLV0	ERRE0	ERRF0	TSF0

STBE0[Notes 2, 3]	Strobe output enable/disable
0	Strobe output disabled
1	Strobe output enabled

BUSYE0	Busy signal detection enable/disable
0	Busy signal detection disabled (input via BUSY0 pin is ignored)
1	Busy signal detection enabled and transfer wait by busy signal is executed

BUSYLV0[Note 4]	Busy signal active level setting
0	Low level
1	High level

Notes 1. Bits 0 and 1 are read-only.

2. STBE0 is valid only in master mode. In slave mode, 1-byte transfer ends after eight transfer clocks regardless of the STBE0 set value.

3. When STBE0 is set to 1, two transfer clocks are consumed between byte transfers regardless of the setting of automatic data transfer interval specification register 0 (ADTI0). That is, 10 transfer clocks are used for 1-byte transfer even if ADTI0 = 00H is set.

4. In bit error detection by busy input, the active level specified by BUSYLV0 is detected.

Caution Be sure to set bits 6 and 7 to 0.

ERRE0[Note]	Bit error detection enable/disable
0	Error detection disabled
1	Error detection enabled

ERRF0	Bit error detection flag
0	• Bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) = 0 • At reset input • When transfer is started by setting bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) to 1 or writing to SIOA0.
1	Bit error detected (when ERRE0 = 1, the level specified by BUSYLV0 during the data bit transfer period is detected via BUSY0 pin input).

TSF0	Transfer status detection flag
0	• Bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) = 0 • At reset input • At the end of the specified transfer • When transfer is stopped by setting bit 1 (ATSTP0) of serial trigger register 0 (CSIT0) to 1
1	From the transfer start to the end of the specified transfer

Note The ERRE0 setting is valid even when BUSYE0 = 0.

Caution When TSF0 is 1, rewriting serial operation mode specification register 0 (CSIMA0), serial status register 0 (CSIS0), divisor selection register 0 (BRGCA0), automatic data transfer address point specification register 0 (ADTP0), automatic data transfer interval specification register 0 (ADTI0), and serial I/O shift register 0 (SIOA0) are prohibited. However, these registers can be read and re-written to the same value. In addition, the buffer RAM can be rewritten during transfer.

(c) Serial trigger register 0 (CSIT0)

This is an 8-bit register used to control execution/stop of automatic data transfer.

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

$\overline{\text{RESET}}$ input sets this register to 00H. However, manipulate only when bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) is 1 (manipulation prohibited when ATE0 = 0).

Address: FF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
CSIT0	0	0	0	0	0	0	ATSTP0	ATSTA0

ATSTP0	Automatic data transfer stop
0	Normal mode
1	Automatic data transfer stopped

ATSTA0	Automatic data transfer start
0	Normal mode
1	Automatic data transfer started

Cautions 1. Even if ATSTP0 or ATSTA0 is set to 1, automatic transfer cannot be started/stopped until 1-byte transfer is complete.

2. ATSTP0 and ATSTA0 retain 1 until immediately before the interrupt signal INTACSI is generated, and then change to 0 automatically.

3. After automatic data transfer is stopped, the data address when the transfer stopped is stored in automatic data transfer address count register 0 (ADTC0). However, since no function to restart automatic data transfer is incorporated, when transfer is stopped by setting ATSTP0 = 1, start automatic data transfer by ATSTA0 after re-setting the registers.

(d) Divisor selection register 0 (BRGCA0)

This is an 8-bit register used to control the serial transfer speed (divisor of CSIA input clock).

This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting BRGCA0 is prohibited.

Address: FF93H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
BRGCA0	0	0	0	0	0	0	BRGCA01	BRGCA00

BRGCA01	BRGCA00	CSIA0 input clock (fw) division ratio selection
0	0	fw/6 (1.67 MHz)
0	1	fw/2^3 (1.25 MHz)
1	0	fw/2^4 (625 kHz)
1	1	fw/2^5 (312.5 kHz)

Remark Figures in parentheses apply to operation with fw = 10 MHz, fw = fx (X1 input clock oscillation frequency)

(e) Automatic data transfer address point specification register 0 (ADTP0)

This is an 8-bit register used to specify the buffer RAM address that ends transfer during automatic data transfer (bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 1).

This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting ADTP0 is prohibited.

In the 78K0/KF1 Series, 00H to 1FH can be specified because 32 bytes of buffer RAM are incorporated.

Example When ADTP0 is set to 07H
8 bytes of 00H to 07H are transferred.

In repeat transfer mode (bit 5 (ATM0) of CSIMA0 = 1), transfer is performed repeatedly up to the address value set in ADTP0.

Example When 07H is transferred to ADTP0 (repeat transfer mode)
Transfer is repeated as 00H to 07H, 00H to 07H, … .

Address: FF94H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTP0	0	0	0	ADTP04	ADTP03	ADTP02	ADTP01	ADTP00

Caution Be sure to set bits 7 to 5 to 0.

The relationship between buffer RAM address values and ADTP0 setting values is shown below.

Buffer RAM Address Value	ADTP0 Setting Value	Buffer RAM Address Value	ADTP0 Setting Value
FA00H	00H	FA10H	10H
FA01H	01H	FA11H	11H
FA02H	02H	FA12H	12H
FA03H	03H	FA13H	13H
FA04H	04H	FA14H	14H
FA05H	05H	FA15H	15H
FA06H	06H	FA16H	16H
FA07H	07H	FA17H	17H
FA08H	08H	FA18H	18H
FA09H	09H	FA19H	19H
FA0AH	0AH	FA1AH	1AH
FA0BH	0BH	FA1BH	1BH
FA0CH	0CH	FA1CH	1CH
FA0DH	0DH	FA1DH	1DH
FA0EH	0EH	FA1EH	1EH
FA0FH	0FH	FA1FH	1FH

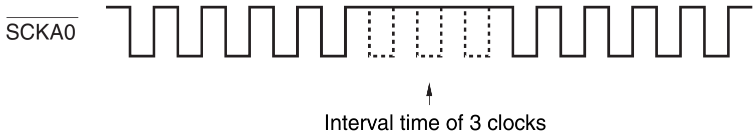
(f) Automatic data transfer interval specification register 0 (ADTI0)

This is an 8-bit register used to specify the interval period between 1-byte transfers during automatic data transfer (bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 1).

Set this register when in master mode (bit 4 (MASTER0) of CSIMA0 = 1) (setting is unnecessary in slave mode). Setting in 1-byte transfer mode (bit 6 (ATE0) of CSIMA0 = 0) is also valid. When the interval time specified by ADTI0 after the end of 1-byte transfer has elapsed, an interrupt request signal (INTACSI) is output. The number of clocks for the interval can be set to between 0 an 63 clocks.

The specified interval time is the transfer clock (specified by divisor selection register 0 (BRGCA0)) multiplied by an integer value.

Example When ADTI0 = 03H



This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting ADTI0 is prohibited.

Address: FF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTI0	0	0	ADTI05	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00

Caution Because the setting of bit 5 (STBE0) and bit 4 (BUSYE0) of serial status register 0 (CSIS0) takes priority over the ADTI0 setting, the interval time based on the setting of STBE0 and BUSYE0 is generated even when ADTI0 is set to 00H.

Example <1> When STBE = 1, BUSYE = 0: Interval time of two transfer clocks is generated

<2> When STBE = 0, BUSYE = 1: Interval time of one transfer clock is generated

<3> When STBE = 1, BUSYE = 1: Interval time of two transfer clocks is generated

Therefore, setting STBE0 and BUSYE0 to 0 is required to perform no-wait transfer.

(3) Automatic transmit/receive data setting

(a) Transmit data setting

<1> Write transmit data from the least significant address FA00H of buffer RAM (up to FA1FH at maximum). The transmit data should be in the order from lower address to higher address.

<2> Set the automatic data transfer address point specification register 0 (ADTP0) to the value obtained by subtracting 1 from the number of transmit data bytes.

(b) Automatic transmission/reception mode setting

<1> Set CSIAE0 and ATE0 of serial operating mode specification register 0 (CSIMA0) to 1.
<2> Set RXEA0 and TXEA0 of CSIMA0 to 1.
<3> Set a data transfer interval in automatic data transfer interval specification register 0 (ADTI0).
<4> Set bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) to 1.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data indicated by automatic data transfer address count register 0 (ADTC0) is transferred to SIOA0, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address indicated by ADTC0.
- ADTC0 is incremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTC0 incremental output matches the set value of automatic data transfer address point specification register 0 (ADTP0) (end of automatic transmission/reception). However, if bit 5 (ATM0) of CSIMA0 is set to 1 (repeat mode), ADTC0 is cleared after a match between ADTP0 and ADTC0, and then repeated transmission/reception is started.
- When automatic transmission/reception is terminated, TSF0 is cleared to 0.

(4) Automatic transmission/reception communication operation

(a) Automatic transmission/reception mode

Automatic transmission/reception can be performed using buffer RAM.
The data stored in the buffer RAM is output from the SOA0 pin via the SIOA0 register in synchronization with the serial clock falling edge by performing (a) and (b) in **(3) Automatic transmit/receive data setting**.
The data is then input from the SIA0 pin via the SIOA0 register in synchronization with the serial clock falling edge and the receive data is stored in the buffer RAM in synchronization with the rising edge 1 clock later.
Data transfer ends if bit 0 (TSF0) of serial status register 0 (CSIS0) is set to 1 when any of the following conditions is met.

- Reset by setting bit 7 (CSIAE0) of the CSIMA0 register to 0
- Transfer of 1 byte is complete by setting bit 1 (ATSTP0) of the CSIT0 register to 1
- Transfer of 1 byte is complete when bit 1 (ERRF0) of the CSIS0 register becomes 1 while bit 2 (ERRE0) = 1
- Transfer of the range specified by the ADTP0 register is complete

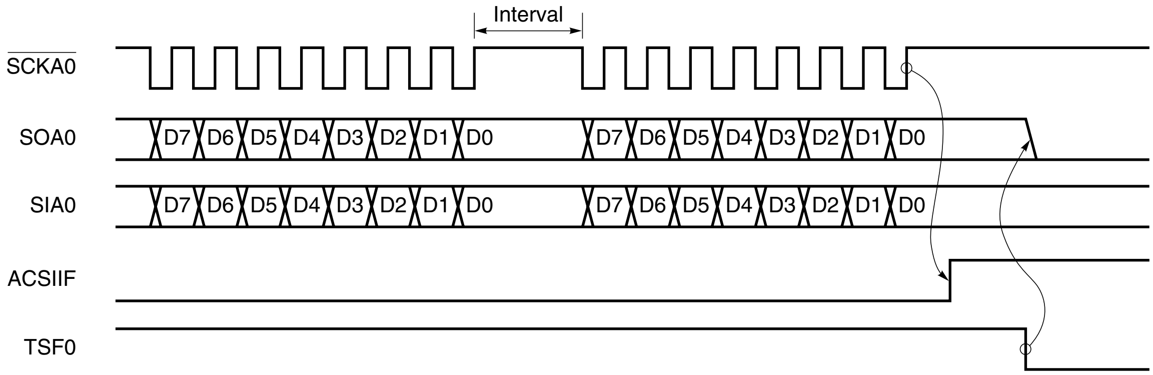
At this time, an interrupt request signal (INTACSI) is generated except when the CSIAE0 bit = 0.

If a transfer is terminated in the middle, transfer starting from the remaining data is not possible. Read automatic data transfer address count register 0 (ADTC0) to confirm how much of the data has already been transferred, set the transfer data again, and then re-execute transfer.

In addition, when busy control and strobe control are not performed, the BUSY0/BUZ/INTP7/P141 and STB0/P145 pins can be used as ordinary I/O port pins.

Figure 17-12 shows the operation timing in automatic transmission/reception mode and Figure 17-13 shows the operation flowchart. Figure 17-14 shows the operation of internal buffer RAM when 6 bytes of data are transmitted/received.

Figure 17-12. Automatic Transmission/Reception Mode Operation Timings



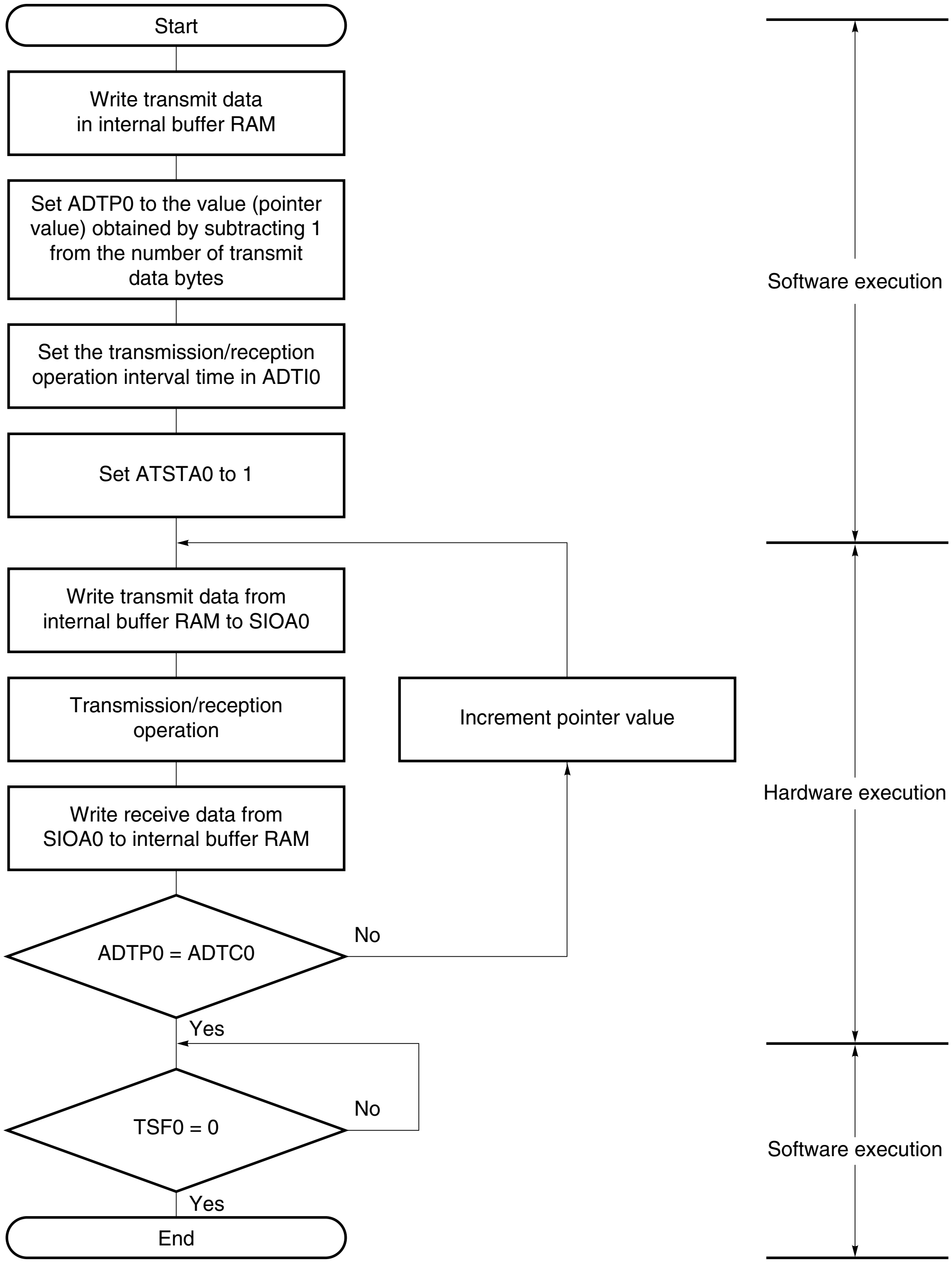
Cautions 1. Because, in the automatic transmission/reception mode, the automatic transmit/receive function writes/reads data to/from the internal buffer RAM after 1-byte transmission/reception, an interval is inserted until the next transmission/reception. As the buffer RAM write/read is performed at the same time as CPU processing, the interval is dependent upon the value of automatic data transfer interval specification register 0 (ADTI0) and the set values of bits 5 and 4 (STBE0, BUSYE0) of serial status register 0 (CSIS0) (see (6) Automatic transmit/receive interval time).

2. When TSF0 is cleared, the SOA0 pin becomes low level.

Remark ACSIIF: Interrupt request flag

TSF0: Bit 0 of serial status register 0 (CSIS0)

Figure 17-13. Automatic Transmission/Reception Mode Flowchart


Start
Write transmit data in internal buffer RAM
Set ADTP0 to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI0
Set ATSTA0 to 1
Write transmit data from internal buffer RAM to SIOA0
Transmission/reception operation
Write receive data from SIOA0 to internal buffer RAM
ADTP0 = ADTC0
No
Increment pointer value
Yes
TSF0 = 0
No
Yes
End
Software execution
Hardware execution
Software execution

ADTP0:	Automatic data transfer address point specification register 0
ADTI0:	Automatic data transfer interval specify register 0
ATSTA0:	Bit 0 of serial trigger register 0 (CSIT0)
SIOA0:	Serial I/O shift register 0
ADTC0:	Automatic data transfer address count register 0
TSF0:	Bit 0 of serial status register 0 (CSIS0)

In 6-byte transmission/reception (ATM0 = 0, RXEA0 = 1, TXEA0 = 1) in automatic transmission/reception mode, internal buffer RAM operates as follows.

(i) Before transmission/reception (see Figure 17-14 (a).)

When bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1, transmit data 1 (T1) is transferred from the internal buffer RAM to SIOA0. When transmission of the first byte is completed, receive data 1 (R1) is transferred from SIOA0 to the buffer RAM, and automatic data transfer address count register 0 (ADTC0) is incremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIOA0.

(ii) 4th byte transmission/reception point (see Figure 17-14 (b).)

Transmission/reception of the third byte is completed, and transmit data 4 (T4) is transferred from the internal buffer RAM to SIOA0. When transmission of the fourth byte is completed, the receive data 4 (R4) is transferred from SIOA0 to the internal buffer RAM, and ADTC0 is incremented.

(iii) Completion of transmission/reception (see Figure 17-14 (c).)

When transmission of the sixth byte is completed, receive data 6 (R6) is transferred from SIOA0 to the internal buffer RAM, and the interrupt request flag (ACSIIF) is set (INTACSI generation).

Figure 17-14. Internal Buffer RAM Operation in 6-Byte Transmission/Reception (in Automatic Transmission/Reception Mode) (1/2)

(a) Before transmission/reception

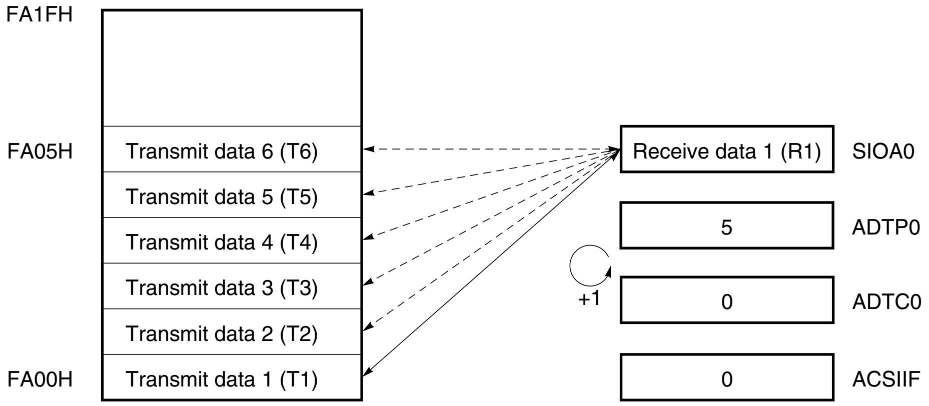
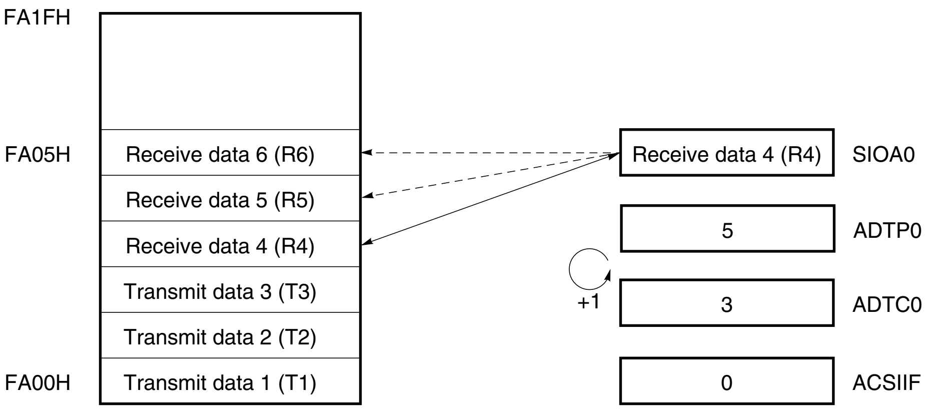
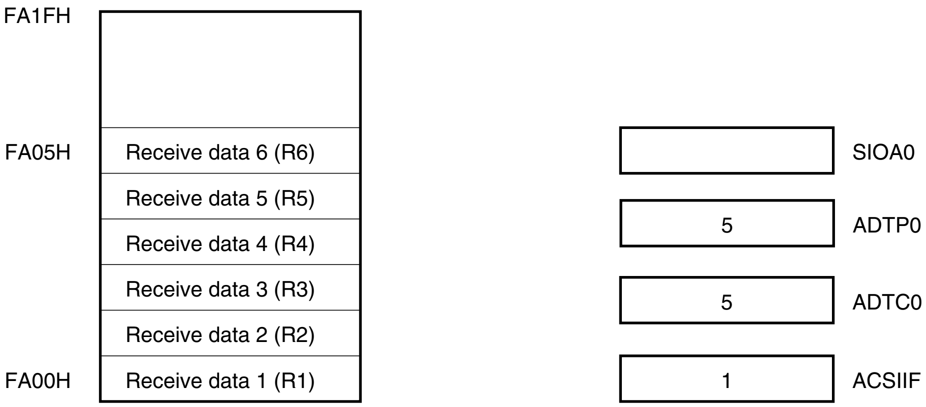


Figure 17-14. Internal Buffer RAM Operation in 6-Byte Transmission/Reception (in Automatic Transmission/Reception Mode) (2/2)

(b) 4th byte transmission/reception



(c) Completion of transmission/reception



(b) Automatic transmission mode

In this mode, the specified number of 8-bit unit data are transmitted.

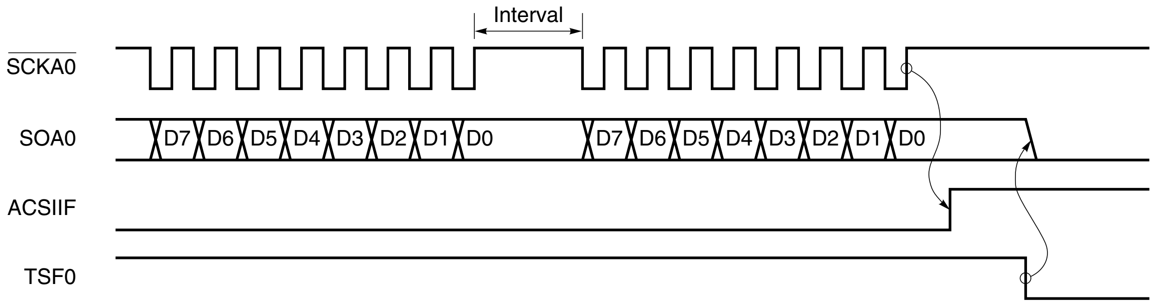
Serial transfer is started when bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1 while bit 7 (CSIAE0), bit 6 (ATE0), and bit 3 (TXEA0) of serial operating mode specification register 0 (CSIMA0) are set to 1.

When the final byte has been transmitted, an interrupt request flag (ACSIIF) is set. However, judge the termination of automatic transmission and reception, not by ACSIIF but by bit 0 (TSF0) of serial status register 0 (CSIS0).

If a receive operation, busy control and strobe control are not executed, the BUSY0/BUZ/INTP7/P141 and STB0/P145 pins can be used as normal I/O port pins.

Figure 17-15 shows the automatic transmission mode operation timing, and Figure 17-16 shows the operation flowchart. Figure 17-17 shows the operation of the internal buffer RAM when 6 bytes of data are transmitted or received.

Figure 17-15. Automatic Transmission Mode Operation Timing



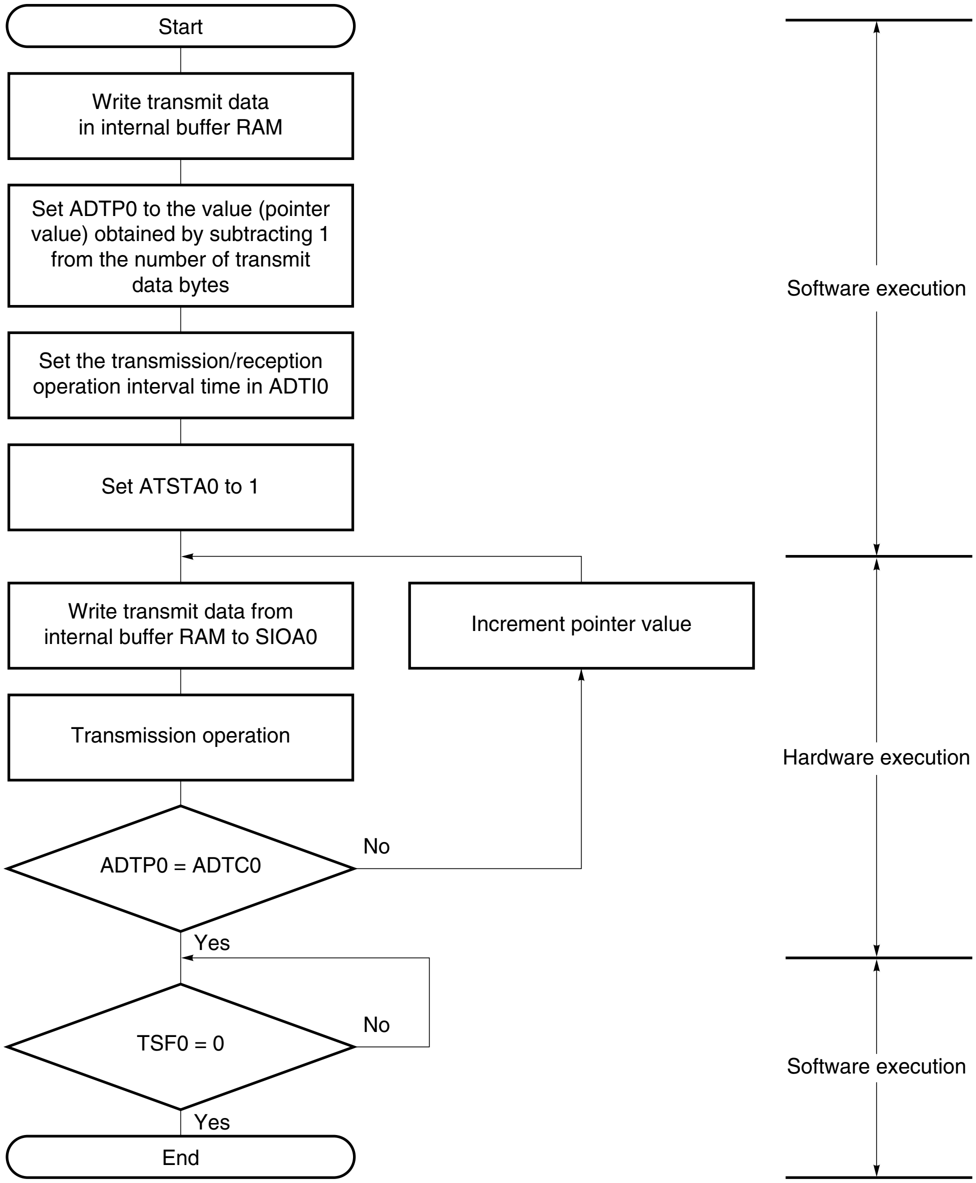
Cautions 1. Because, in the automatic transmission mode, the automatic transmit/receive function reads data from the internal buffer RAM after 1-byte transmission, an interval is inserted until the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the interval is dependent upon the value of automatic data transfer interval specification register 0 (ADTI0) and the set values of bits 5 and 4 (STBE0, BUSYE0) of serial status register 0 (CSIS0) (see (6) Automatic transmit/receive interval time).

2. When TSF0 is cleared, the SOA0 pin becomes low level.

Remark ACSIIF: Interrupt request flag

TSF0: Bit 0 of serial status register 0 (CSIS0)

Figure 17-16. Automatic Transmission Mode Flowchart


Start
Write transmit data in internal buffer RAM
Set ADTP0 to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI0
Set ATSTA0 to 1
Write transmit data from internal buffer RAM to SIOA0
Increment pointer value
Transmission operation
ADTP0 = ADTC0
No
Yes
TSF0 = 0
No
Yes
End
Software execution
Hardware execution
Software execution

ADTP0:	Automatic data transfer address point specification register 0
ADTI0:	Automatic data transfer interval specification register 0
ATSTA0:	Bit 0 of serial trigger register 0 (CSIT0)
SIOA0:	Serial I/O shift register 0
ADTC0:	Automatic data transfer address count register 0
TSF0:	Bit 0 of serial status register 0 (CSIS0)

In 6-byte transmission (ATM0 = 0, RXEA0 = 0, TXEA0 = 1, ATE0 = 1) in automatic transmission mode, internal buffer RAM operates as follows.

(i) Before transmission (see Figure 17-17 (a).)

When bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1, transmit data 1 (T1) is transferred from the internal buffer RAM to SIOA0. When transmission of the first byte is completed, automatic data transfer address count register 0 (ADTC0) is incremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIOA0.

(ii) 4th byte transmission point (see Figure 17-17 (b).)

Transmission of the third byte is completed, and transmit data 4 (T4) is transferred from the internal buffer RAM to SIOA0. When transmission of the fourth byte is completed, ADTC0 is incremented.

(iii) Completion of transmission (see Figure 17-17 (c).)

When transmission of the sixth byte is completed, the interrupt request flag (ACSIIF) is set (INTACSI generation).

Figure 17-17. Internal Buffer RAM Operation in 6-Byte Transmission (in Automatic Transmission Mode) (1/2)

(a) Before transmission

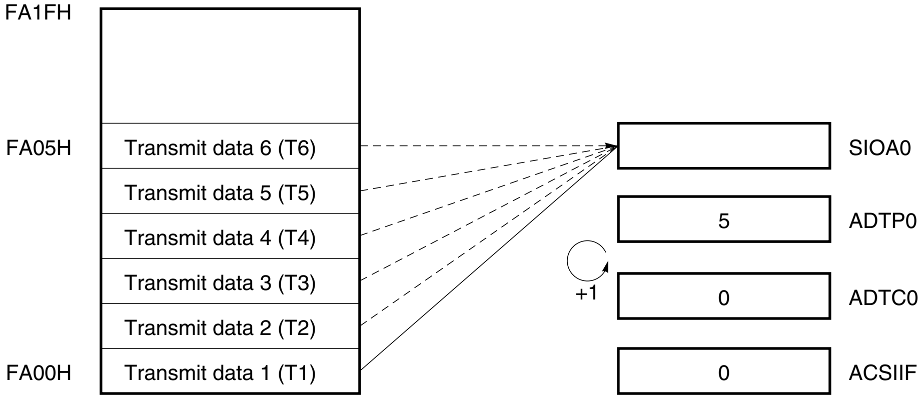
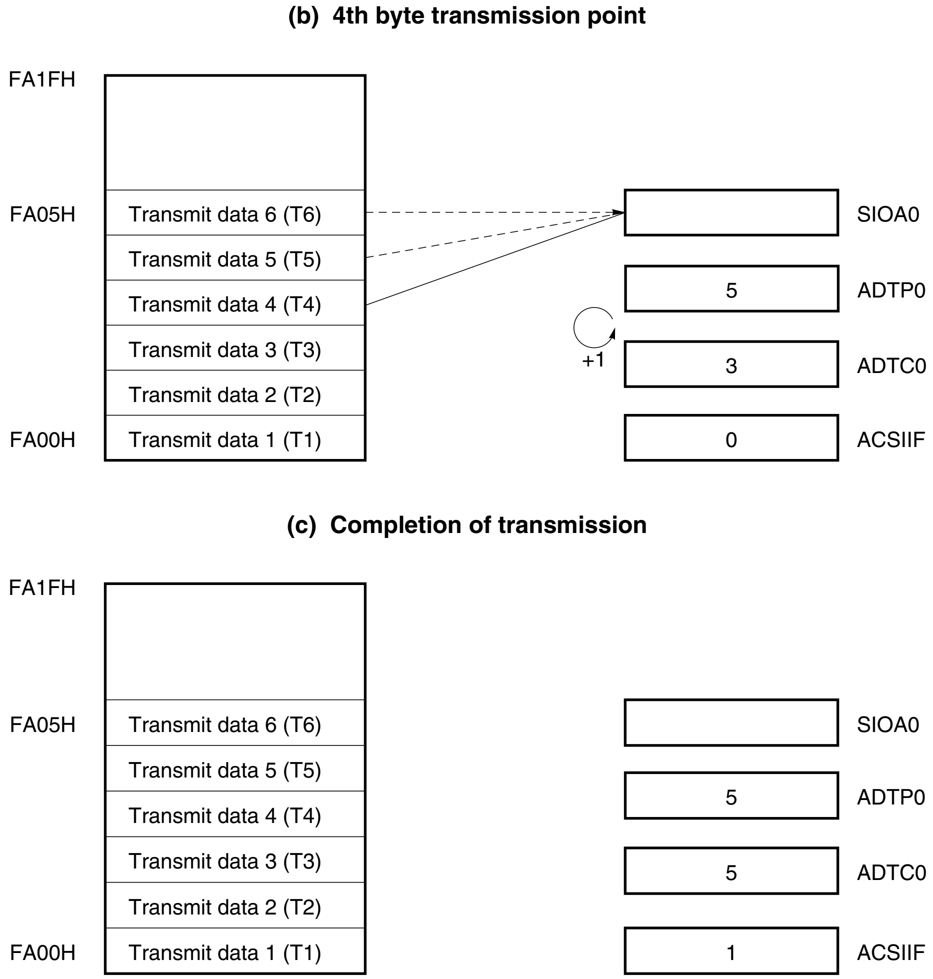


Figure 17-17. Internal Buffer RAM Operation in 6-Byte Transmission (in Automatic Transmission Mode) (2/2)



(b) 4th byte transmission point

(c) Completion of transmission

(c) Repeat transmission mode

In this mode, data stored in the internal buffer RAM is transmitted repeatedly.

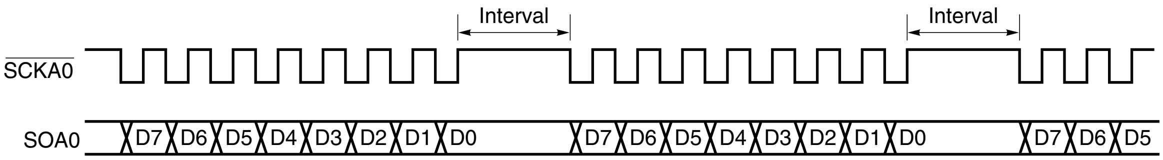
Serial transfer is started when bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1 while bit 7 (CSIAE0), bit 6 (ATE0), bit 5 (ATM0), and bit 3 (TXEA0) of serial operating mode specification register 0 (CSIMA0) are set to 1.

Unlike the basic transmission mode, after the final byte (data in address FA1FH) has been transmitted, the interrupt request flag (ACSIIF) is not set, the automatic data transfer address count register 0 (ADTC0) is reset to 0, and the internal buffer RAM contents are transmitted again.

When a reception operation, busy control and strobe control are not performed, the BUSY0/BUZ/INTP7/P141 and STB0/P145 pins can be used as ordinary I/O port pins.

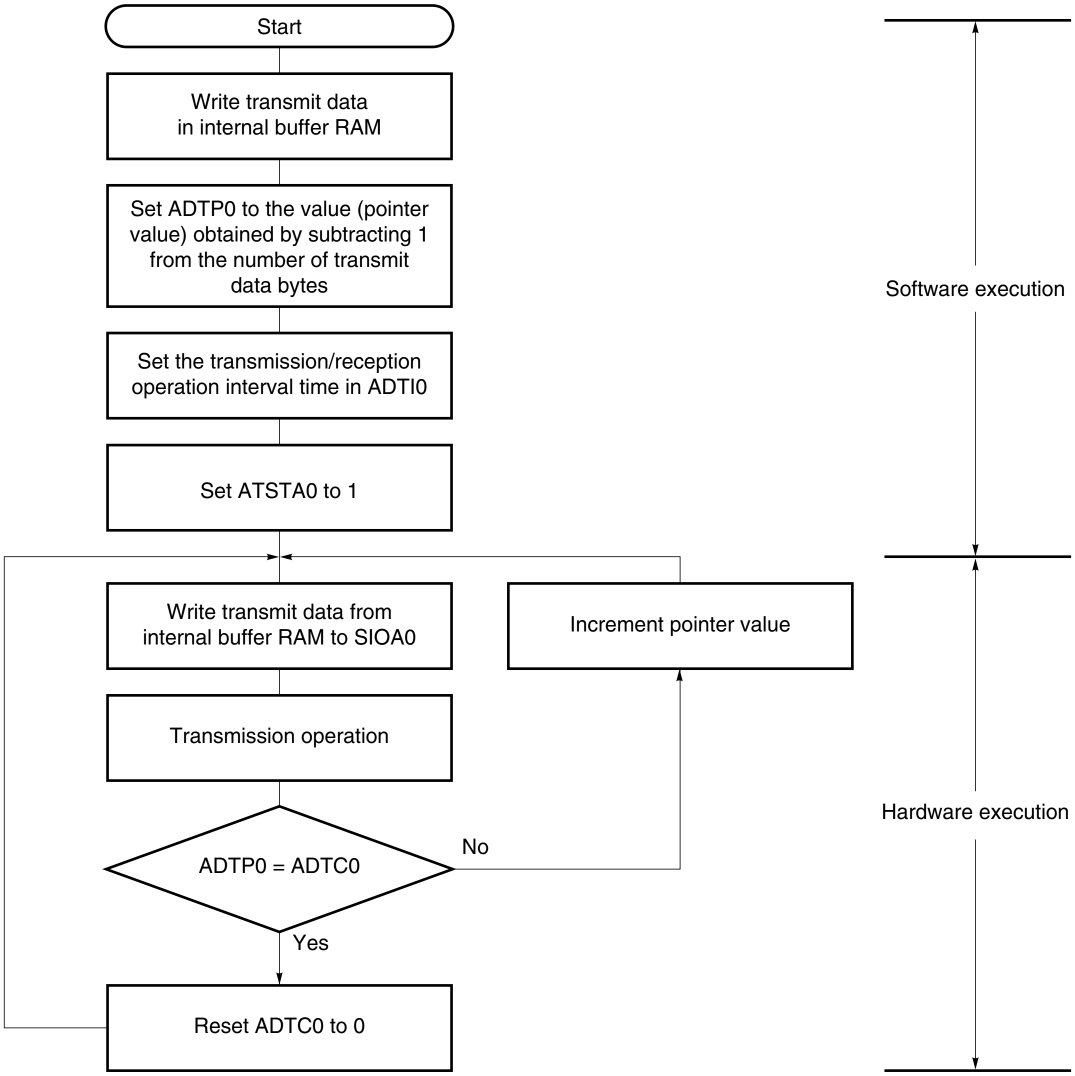
The repeat transmission mode operation timing is shown in Figure 17-18, and the operation flowchart in Figure 17-19. Figure 17-20 shows the operation of the internal buffer RAM when 6 bytes of data are transmitted in the repeat transmission mode.

Figure 17-18. Repeat Transmission Mode Operation Timing



Caution Because, in the repeat transmission mode, a read is performed on the buffer RAM after the transmission of one byte, the interval is included in the period up to the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the interval is dependent upon automatic data transfer interval specification register 0 (ADTI0) and the set values of bits 5 and 4 (STBE0, BUSYE0) of serial status register 0 (CSIS0) (see (6) Automatic transmit/receive interval time).

Figure 17-19. Repeat Transmission Mode Flowchart


Start
Write transmit data in internal buffer RAM
Set ADTP0 to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI0
Set ATSTA0 to 1
Write transmit data from internal buffer RAM to SIOA0
Increment pointer value
Transmission operation
ADTP0 = ADTC0
No
Yes
Reset ADTC0 to 0
Software execution
Hardware execution

ADTP0:	Automatic data transfer address point specification register 0
ADTI0:	Automatic data transfer interval specification register 0
ATSTA0:	Bit 0 of serial trigger register 0 (CSIT0)
SIOA0:	Serial I/O shift register 0
ADTC0:	Automatic data transfer address count register 0

In 6-byte transmission (ATM0 = 1, RXEA0 = 0, TXEA0 = 1, ATE0 = 1) in repeat transmission mode, internal buffer RAM operates as follows.

(i) Before transmission (see Figure 17-20 (a).)

When bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1, transmit data 1 (T1) is transferred from the internal buffer RAM to SIOA0. When transmission of the first byte is completed, automatic data transfer address count register 0 (ADTC0) is incremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIOA0.

(ii) Upon completion of transmission of 6 bytes (see Figure 17-20 (b).)

When transmission of the sixth byte is completed, the interrupt request flag (ACSIIF) is not set.
ADTC0 is reset to 0.

(iii) 7th byte transmission point (see Figure 17-20 (c).)

Transmit data 1 (T1) is transferred from the internal buffer RAM to SIOA0 again. When transmission of the first byte is completed, ADTC0 is incremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIOA0.

Figure 17-20. Internal Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmission Mode) (1/2)

(a) Before transmission

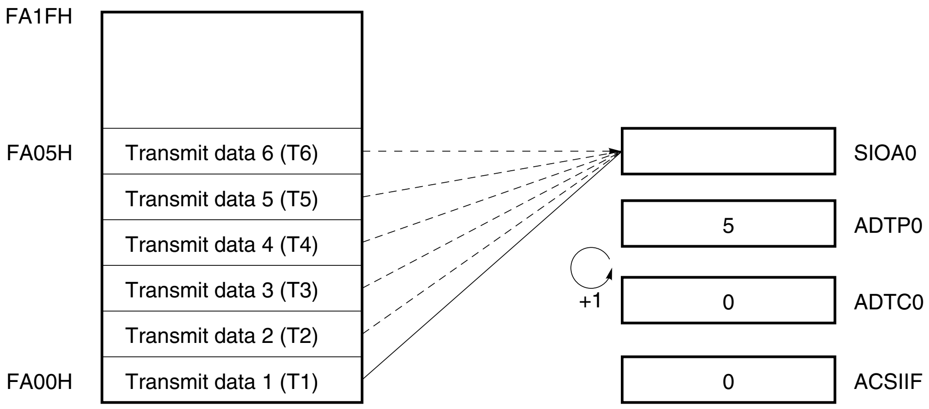
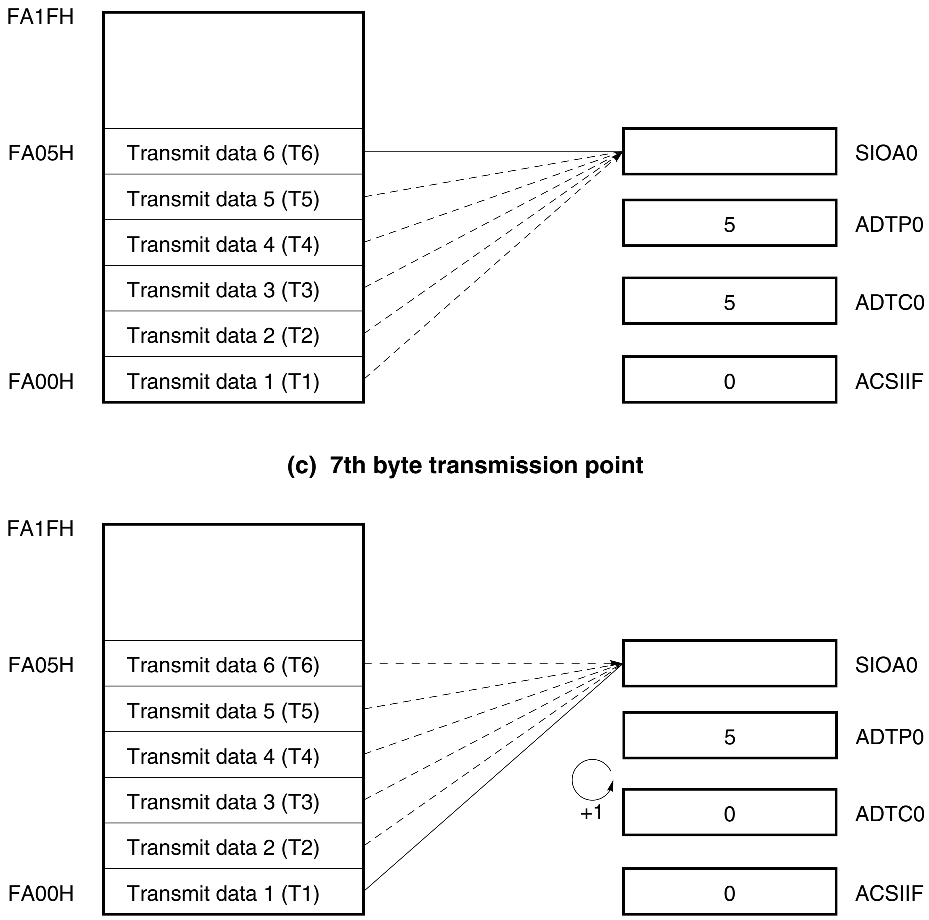


Figure 17-20. Internal Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmission Mode) (2/2)



(b) Upon completion of transmission of 6 bytes

(c) 7th byte transmission point

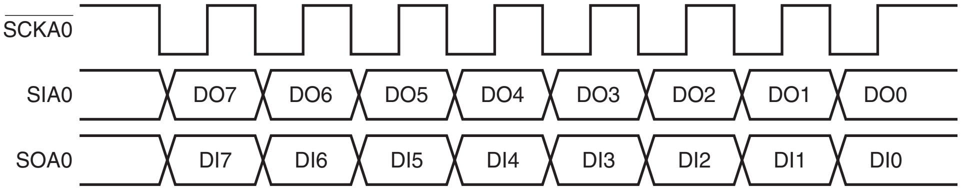
(d) Data format

In the data format, data is changed in synchronization with the $\overline{\text{SCKA0}}$ falling edge as shown below.

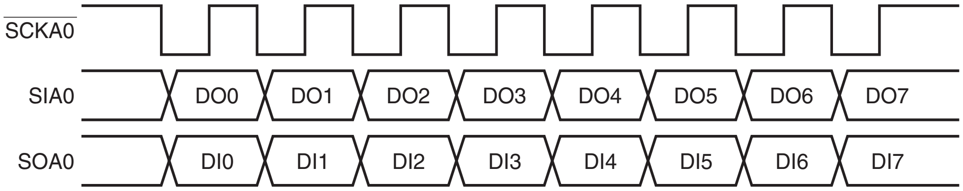
The data length is fixed to 8 bits and the data transfer direction can be switched by the specification of bit 1 (DIR0) of serial operation mode specification register 0 (CSIMA0).

Figure 17-21. Format of CSIA0 Transmit/Receive Data

(a) MSB-first (DIR0 bit = 0)



(b) LSB-first (DIR0 bit = 1)



(e) Automatic transmission/reception suspension and restart

Automatic transmission/reception can be temporarily suspended by setting bit 1 (ATSTP0) of serial trigger register 0 (CSIT0) to 1.

During 8-bit data transfer, the transmission/reception is not suspended. It is suspended upon completion of 8-bit data transfer.

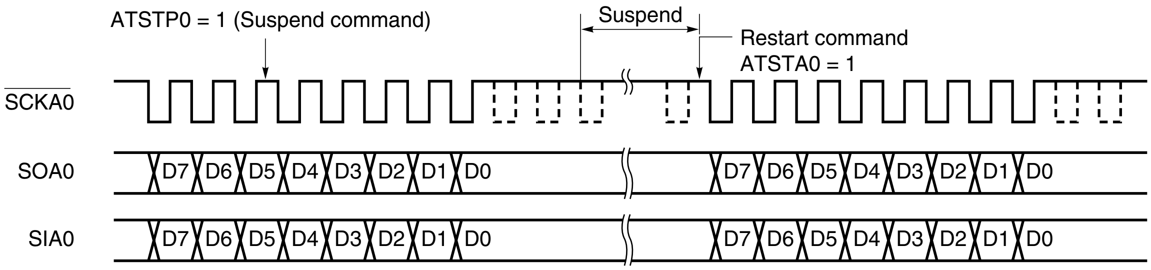
When suspended, bit 0 (TSF0) of serial status register 0 (CSIS0) is set to 0 after transfer of the 8th bit, and all the port pins that function alternately as (P141/BUZ/BUSY0/INTP7, P142/$\overline{\text{SCKA0}}$, P143/SIA0, P144/SOA0, and P145/STB0) are set to the port mode.

To restart automatic transmission/reception, set bit 0 (ATSTA0) of CSIT0 to 1. The remaining data can be transmitted in this way.

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer is suspended and the HALT mode is set if during 8-bit data transfer. When the HALT mode is cleared, automatic transmission/reception is restarted from the suspended point.

2. When suspending automatic transmission/reception, do not change the operating mode to 3-wire serial I/O mode while TSF0 = 1.

Figure 17-22. Automatic Transmission/Reception Suspension and Restart



ATSTP0: Bit 1 of serial trigger register 0 (CSIT0)

ATSTA0: Bit 0 of CSIT0

(5) Synchronization control

Busy control and strobe control are functions used to synchronize transmission/reception between the master device and a slave device.

By using these functions, a shift in bits being transmitted or received can be detected.

(a) Busy control option

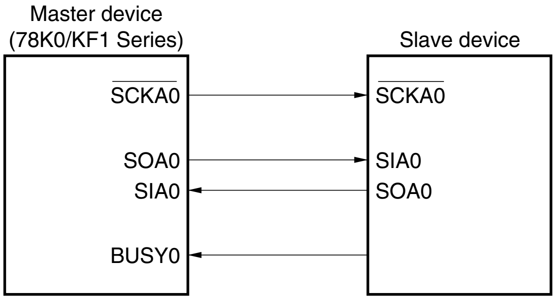
Busy control is a function to keep the serial transmission/reception by the master device waiting while the busy signal output by a slave device to the master is active.

When using this busy control option, the following conditions must be satisfied.

- Bit 6 (ATE0) of serial operating mode specification register 0 (CSIMA0) is set to 1.
- Bit 4 (BUSYE0) of serial status register 0 (CSIS0) is set to 1.

Figure 17-23 shows the system configuration of the master device and slave device when the busy control option is used.

Figure 17-23. System Configuration When Busy Control Option Is Used



The master device inputs the busy signal output by the slave device to the BUSY0/BUZ/INTP7/P141 pin. The master device samples the input busy signal in synchronization with the falling of the serial clock. Even if the busy signal becomes active while 8-bit data is being transmitted or received, transmission/reception by the master is not kept waiting. If the busy signal is active at the rising edge of the serial clock one clock after completion of transmission/reception of the 8-bit data, the busy input becomes valid. After that, the master transmission/reception is kept waiting while the busy signal is active.

The active level of the busy signal is set by bit 3 (BUSYLV0) of CSIS0.

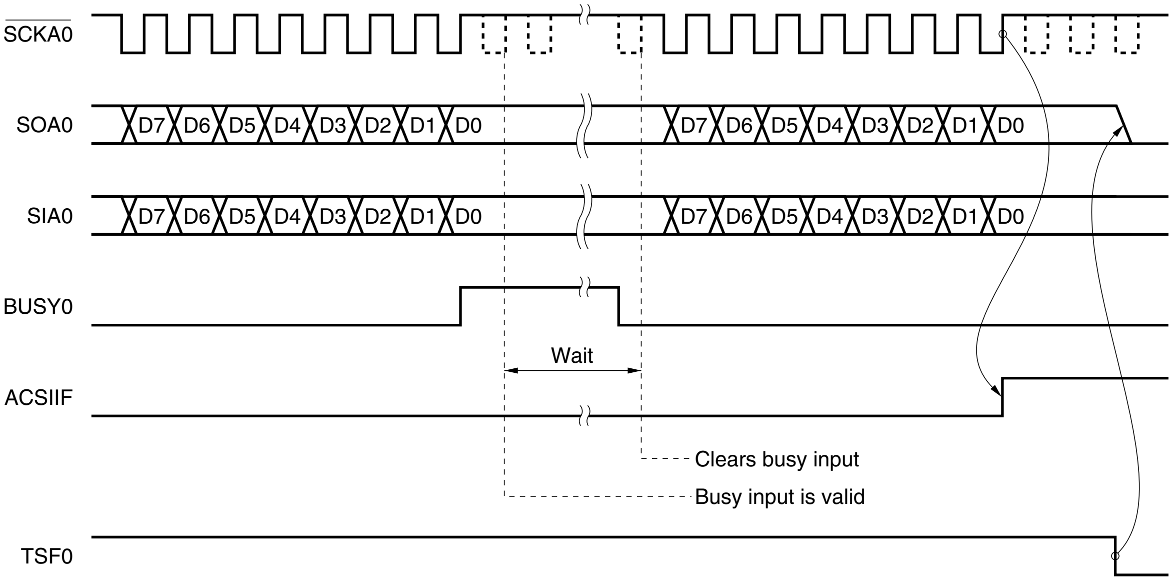
BUSYLV0 = 1: Active-high
BUSYLV0 = 0: Active-low

When using the busy control option, select the internal clock as the serial clock. Control with the busy signal cannot be implemented with the external clock.

Figure 17-24 shows the operation timing when the busy control option is used.

Caution Busy control cannot be used simultaneously with the interval time control function of automatic data transfer interval specification register 0 (ADTI0). If used, busy control is invalid.

Figure 17-24. Operation Timing When Busy Control Option Is Used (When BUSYLV0 = 1)



Caution **If the TSF0 is cleared, the SOA0 pin goes low.**

Remark ACSIIF: Interrupt request flag
TSF0: Bit 0 of serial status register 0 (CSIS0)

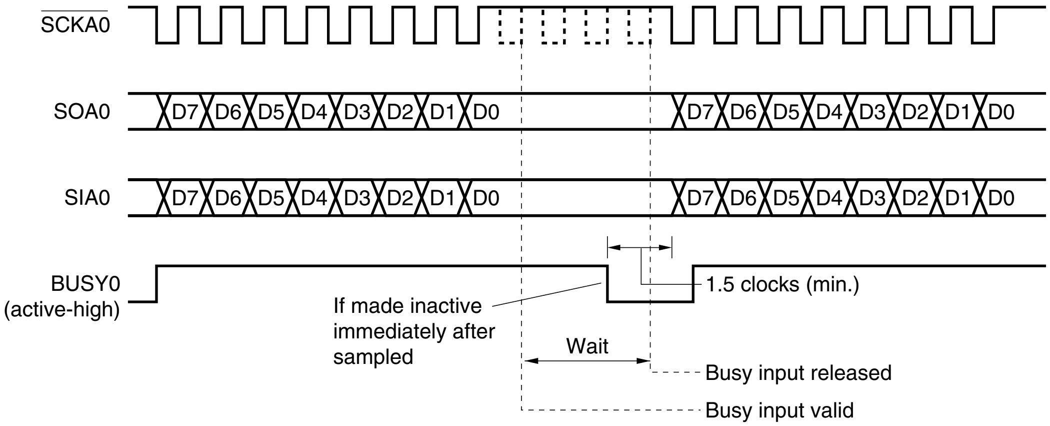
When the busy signal becomes inactive, waiting is released. If the sampled busy signal is inactive, transmission/reception of the next 8-bit data is started at the falling edge of the next serial clock.

Because the busy signal is asynchronous with the serial clock, it takes up to 1 clock until the busy signal is sampled, even if made inactive by the slave. It takes 0.5 clock until data transfer is started after the busy signal was sampled.

To accurately release waiting, the slave must keep the busy signal inactive at least for the duration of 1.5 clock.

Figure 17-25 shows the timing of the busy signal and releasing the waiting. This figure shows an example in which the busy signal is active as soon as transmission/reception has been started.

Figure 17-25. Busy Signal and Wait Release (When BUSYLV0 = 1)



(b) Busy & strobe control option

Strobe control is a function used to synchronize data transmission/reception between the master and slave devices. The master device outputs the strobe signal from the STB0/P145 pin when 8-bit transmission/reception has been completed. By this signal, the slave device can determine the timing of the end of data transmission. Therefore, synchronization is established even if a bit shift occurs because noise is superimposed on the serial clock, and transmission of the next byte is not affected by the bit shift.

To use the strobe control option, the following conditions must be satisfied:

- Bit 6 (ATE0) of the serial operating mode specification register 0 (CSIMA0) is set to 1.
- Bit 5 (STBE0) of serial status register 0 (CSIS0) is set to 1.

Usually, the busy control and strobe control options are simultaneously used as handshake signals. In this case, the strobe signal is output from the STB0/P145 pin, the BUSY0/BUZ/INTP7/P141 pin can be sampled to keep transmission/reception waiting while the busy signal is input.

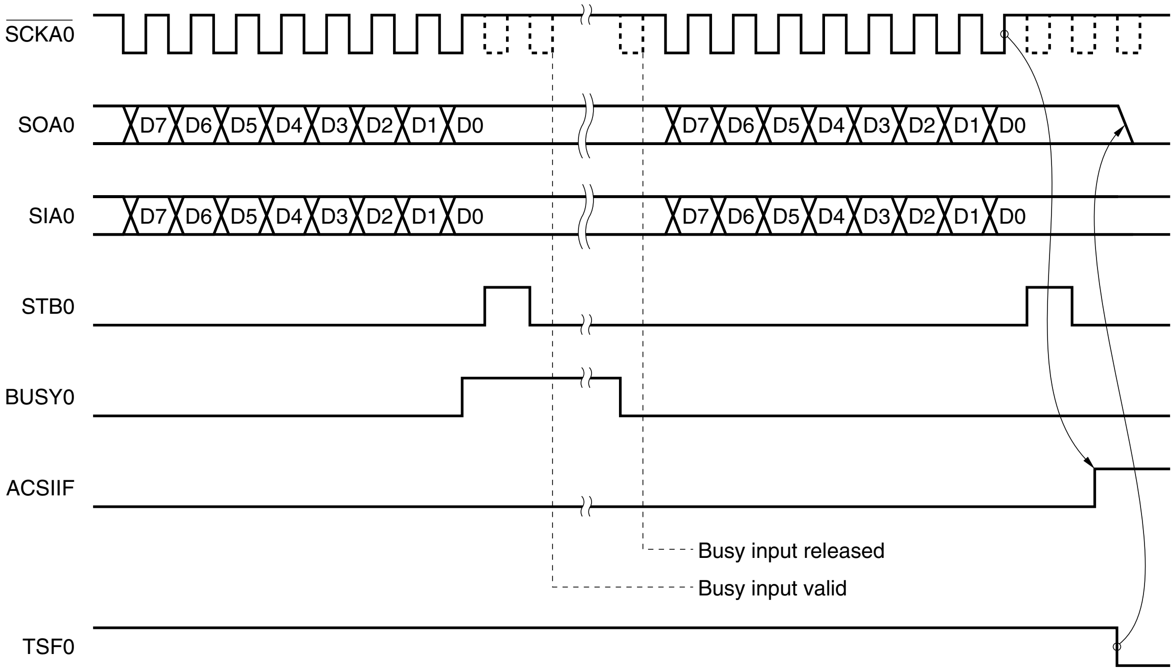
A high level lasting for one transfer clock is output from the STB0/P145 pin in synchronization with the falling edge of the ninth serial clock as the strobe signal. The busy signal is detected at the rising edge of the serial clock two clocks after 8-bit data transmission/reception completion.

When the strobe control option is not used, the P145/STB0 pin can be used as a normal I/O port pin.

Figure 17-26 shows the operation timing when the busy & strobe control options are used.

When the strobe control option is used, the interrupt request flag (ACSIIF) that is set on completion of transmission/reception is set after the strobe signal is output.

Figure 17-26. Operation Timing When Busy & Strobe Control Options Are Used (When BUSYLV0 = 1)



Caution When TSF0 is cleared, the SOA0 pin goes low.

Remark ACSIIF: Interrupt request flag
TSF0: Bit 0 of serial status register 0 (CSIS0)

(c) Bit shift detection by busy signal

During automatic transmission/reception, a bit shift of the serial clock of the slave device may occur because noise is superimposed on the serial clock signal output by the master device. Unless the strobe control option is used at this time, the bit shift affects transmission of the next byte. In this case, the master can detect the bit shift by checking the busy signal during transmission by using the busy control option.

A bit shift is detected by using the busy signal as follows:

The slave outputs the busy signal after the rising of the eighth serial clock during data transmission/reception (to not keep transmission/reception waiting by the busy signal at this time, make the busy signal inactive within 2 clocks).

The master samples the busy signal in synchronization with the falling edge of the serial clock if bit 2 (ERRE0) of serial status register 0 (CSIS0) is set to 1. If a bit shift does not occur, all the eight serial clocks that have been sampled are inactive. If the sampled serial clocks are active, it is assumed that a bit shift has occurred, error processing is executed (by setting bit 1 (ERRF0) of serial status register 0 (CSIS0) to 1, and transfer is suspended and an interrupt request signal (INTACSI) is output).

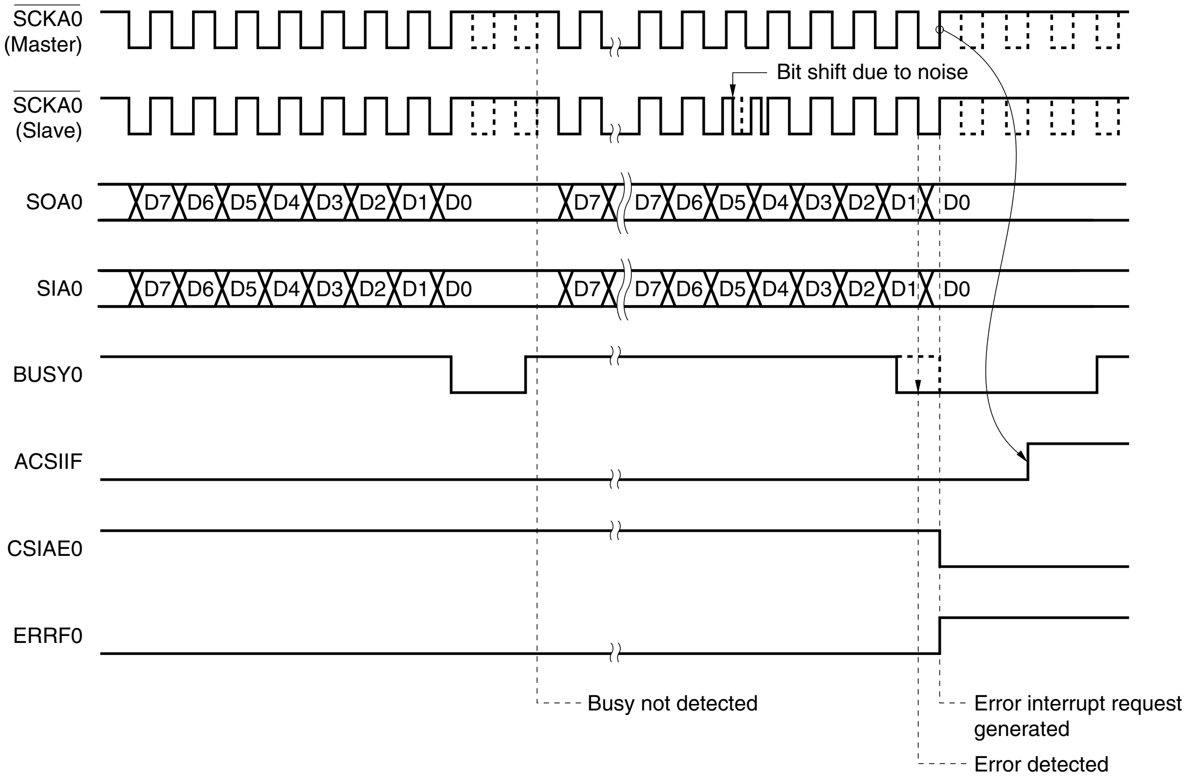
Although transfer is suspended after completion of 1-byte data transfer, slave signal output, wait due to the busy signal, and wait due to the interval time specified by ADTI0 are not executed.

If ERRE0 = 0, ERRF0 cannot become 1 even if a bit shift occurs.

Figure 17-27 shows the operation timing of the bit shift detection function by the busy signal.

Remark The bit error function is valid both in the master mode and slave mode. The setting of ERRE0 is valid even when BUSYE0 = 0.

Figure 17-27. Operation Timing of Bit Shift Detection Function by Busy Signal (When BUSYLV0 = 0)



ACSIIF: Interrupt request flag

CSIAE0: Bit 7 of serial operating mode specification register 0 (CSIMA0)

ERRF0: Bit 1 of serial status register 0 (CSIS0)

(6) Automatic transmit/receive interval time

When using the automatic transmit/receive function, the read/write operations from/to the internal buffer RAM are performed after transmitting/receiving one byte. Therefore, an interval is inserted before the next transmit/receive operation.

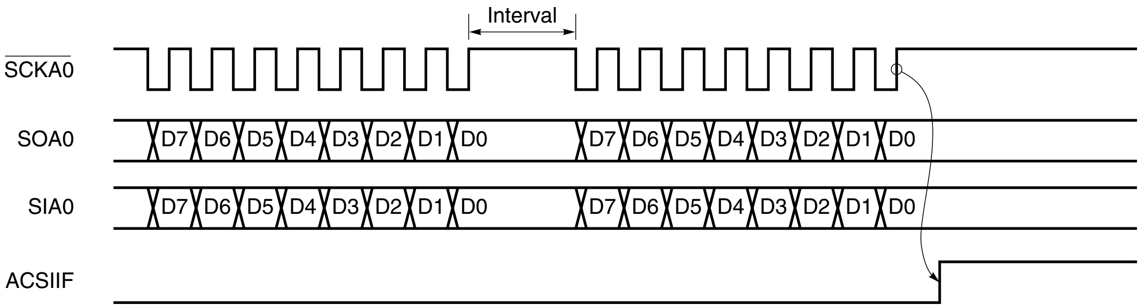
Since the read/write operations from/to the buffer RAM are performed in parallel with the CPU processing when using the automatic transmit/receive function by the internal clock, the interval depends on the value which is set in the automatic data transfer interval specification register 0 (ADTI0) and bits 5 and 4 (STBE0, BUSYE0) of the serial status register 0 (CSIS0).

When ADTI0 is set to 00H, an interval time based on the to STBE0 and BUSYE0 settings is generated.

For example, when ADTI0 = 00H and STBE0 = BUSYE0 = 1, an interval time of two clocks is generated. If an interval time of two clocks or more is set by ADTI0, the interval time set by ADTI0 is generated regardless of the STBE0 and BUSYE0 settings.

Example <1> When STBE0 = 1, BUSYE0 = 0: Interval time of two transfer clocks is generated
<2> When STBE0 = 0, BUSYE0 = 1: Interval time of one transfer clock is generated
<3> When STBE0 = 1, BUSYE0 = 1: Interval time of two transfer clocks is generated

Figure 17-28. Automatic Data Transmit/Receive Interval Time



ACSIIF: Interrupt request flag

CHAPTER 18 MULTIPLIER/DIVIDER

18.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 16 bits = 32 bits, 16-bit remainder (division)

18.2 Configuration of Multiplier/Divider

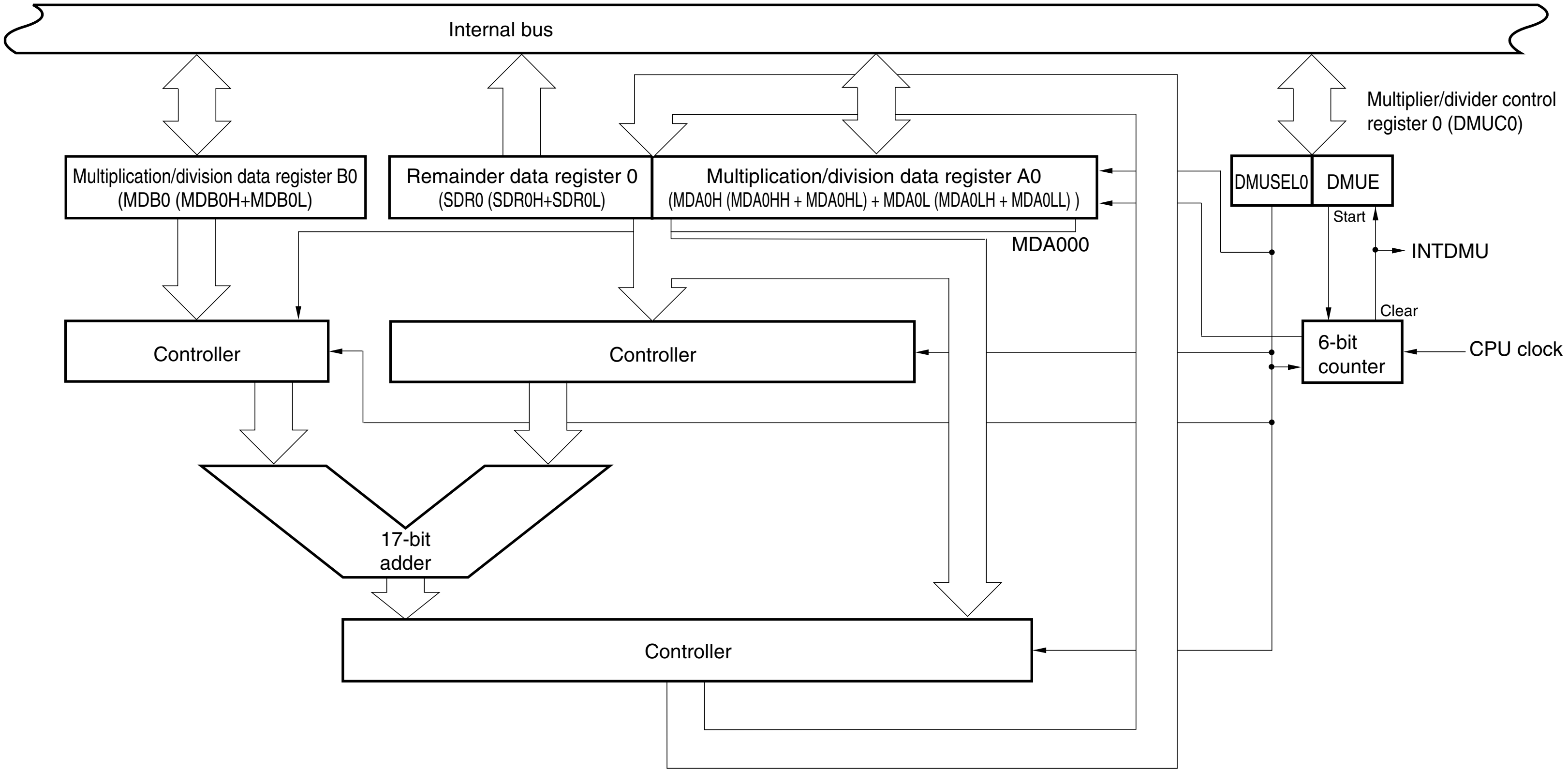
The multiplier/divider consists of the following hardware.

Table 18-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Remainder data register 0 (SDR0) Multiplication/division data registers A0 (MDA0H, MDA0L) Multiplication/division data registers B0 (MDB0)
Control register	Multiplier/divider control register 0 (DMUC0)

Figure 18-1 shows the block diagram of the multiplier/divider.

Figure 18-1. Block Diagram of Multiplier/Divider


Internal bus
Multiplier/divider control register 0 (DMUC0)
Multiplication/division data register B0 (MDB0 (MDB0H+MDB0L)
Remainder data register 0 (SDR0 (SDR0H+SDR0L)
Multiplication/division data register A0 (MDA0H (MDA0HH + MDA0HL) + MDA0L (MDA0LH + MDA0LL))
MDA000
DMUSEL0
DMUE
Start
INTDMU
Clear
6-bit counter
CPU clock
Controller
Controller
17-bit adder
Controller

(1) Remainder data register 0 (SDR0)

SDR0 is a 16-bit register that stores a remainder. This register stores 0 in the multiplication mode and the remainder of an operation result in the division mode.

This register can be read by an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 0000H.

Figure 18-2. Format of Remainder Data Register 0 (SDR0)

Address: FF60H, FF61H After reset: 0000H R

Symbol	FF61H								FF60H							
SDR0	SDR 015	SDR 014	SDR 013	SDR 012	SDR 011	SDR 010	SDR 009	SDR 008	SDR 007	SDR 006	SDR 005	SDR 004	SDR 003	SDR 002	SDR 001	SDR 000

Cautions 1. The value read from SDR0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1) is not guaranteed.

2. SDR0 is reset when the operation is started (when DMUE is set to 1).

(2) Multiplication/division data register A0 (MDA0H, MDA0L)

MDA0 is a register that stores 16-bit multiplier A in the multiplication mode, and a 32-bit dividend in the division mode (higher 16 bits: MDA0H, lower 16 bits: MDA0L, MDA0H+MDA0L: MDA0).

The functions of MDA0 when an operation is executed are shown in the table below.

Table 18-2. Functions of MDA0 During Operation Execution

DMUSEL0[Note]	Operation Mode	Setting	Operation Result
0	Division mode	Dividend	Division result (quotient)
1	Multiplication mode	Higher 10 bits: 0, Lower 16 bits: Multiplier A	Multiplication result (product)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

```
     <Multiplier A>          <Multiplier B>          <Product>
MDA0 (bits 15 to 0) × MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0)
```

- Register configuration during division

```
     <Dividend>              <Divisor>               <Quotient>              <Remainder>
MDA0 (bits 31 to 0) ÷ MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0) … SDR0 (bits 15 to 0)
```

MDA0 fetches the calculation result as soon as the clock is input, when bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is set to 1.

MDA0H and MDA0L can be set by an 8-bit or 16-bit memory instruction.

$\overline{\text{RESET}}$ input sets this register to 0000H.

Figure 18-3. Format of Multiplication/Division Data Register A0 (MDA0H, MDA0L)

Address: FF62H, FF63H, FF64H, FF65H After reset: 0000H, 0000H R/W

Symbol	FF65H (MDA0HH)								FF64H (MDA0HL)							
MDA0H	MDA 031	MDA 030	MDA 029	MDA 028	MDA 027	MDA 026	MDA 025	MDA 024	MDA 023	MDA 022	MDA 021	MDA 020	MDA 019	MDA 018	MDA 017	MDA 016

Symbol	FF63H (MDA0LH)								FF62H (MDA0LL)							
MDA0L	MDA 015	MDA 014	MDA 013	MDA 012	MDA 011	MDA 010	MDA 009	MDA 008	MDA 007	MDA 006	MDA 005	MDA 004	MDA 003	MDA 002	MDA 001	MDA 000

Cautions 1. MDA0H is reset to 0 when an operation is started in the multiplication mode (when multiplier/divider control register 0 (DMUC0) is set to 81H).

2. Do not change the value of MDA0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

3. The value read from MDA0 during operation processing (while DMUE is 1) is not guaranteed.

(3) Multiplication/division data register B0 (MDB0)

MDB0 is a register that stores a 16-bit multiplier B in the multiplication mode and a 16-bit divisor in the division mode.

This register can be set by an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 0000H.

Figure 18-4. Format of Multiplication/Division Data Register B0 (MDB0)

Address: FF66H, FF67H After reset: 0000H R/W

Symbol	FF67H								FF66H							
MDB0	MDB 015	MDB 014	MDB 013	MDB 012	MDB 011	MDB 010	MDB 009	MDB 008	MDB 007	MDB 006	MDB 005	MDB 004	MDB 003	MDB 002	MDB 001	MDB 000

Cautions 1. Do not change the value of MDB0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

2. Do not set MDB0 to 0000H in the division mode. If set, undefined operation results are stored in MDA0 and SDR0.

18.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by multiplier/divider control register 0 (DMUC0).

(1) Multiplier/divider control register 0 (DMUC0)

DMUC0 is an 8-bit register that controls the operation of the multiplier/divider.
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 18-5. Format of Multiplier/Divider Control Register 0 (DMUC0)

Address: FF68H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0

DMUE[Note]	Operation start/stop
0	Stops operation
1	Starts operation

DMUSEL0	Operation mode (multiplication/division) selection
0	Division mode
1	Multiplication mode

Note When DMUE is set to 1, the operation is started. DMUE is automatically cleared to 0 after the operation is complete.

Cautions 1. If DMUE is cleared to 0 during operation processing (when DMUE is 1), the operation result is not guaranteed. If the operation is completed while the clearing instruction is being executed, the operation result is guaranteed, provided that the interrupt flag is set.

2. Do not change the value of DMUSEL0 during operation processing (while DMUE is 1). If it is changed, undefined operation results are stored in multiplication/division data register A0 (MDA0) and remainder data register 0 (SDR0).

3. If DMUE is cleared to 0 during operation processing (while DMUE is 1), the operation processing is stopped. To execute the operation again, set multiplication/division data register A0 (MDA0), multiplication/division data register B0 (MDB0), and multiplier/divider control register 0 (DMUC0), and start the operation (by clearing DMUE to 0).

18.4 Operations of Multiplier/Divider

18.4.1 Multiplication operation

(1) Multiplication operation

- Initial setting

1. Set operation data to multiplication/division data register A0L (MDA0L) and multiplication/division data register B0 (MDB0).
2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 1. The internal clock will start (operation will start).

- During operation

3. The operation will be completed when 16 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers during operation, and therefore the read values of these registers are not guaranteed).

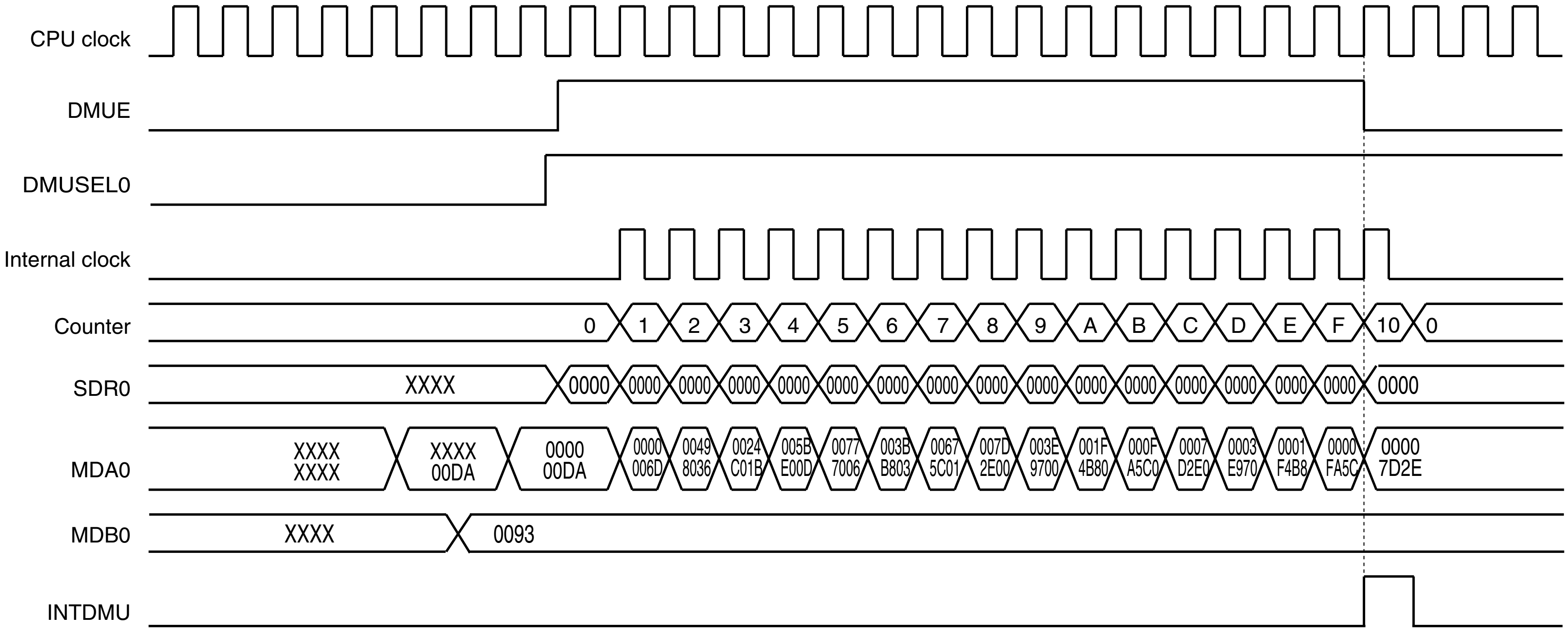
- End of operation

4. After the operation, an interrupt request signal (INTDMU) is generated.
5. The operation result data is stored in the MDA0L and MDA0H registers.
6. DMUE is cleared to 0 (end of operation).

- Next operation

7. To execute multiplication next, start from the initial setting in **18.4.1 Multiplication operation (1)**.
8. To execute division next, start from the initial setting in **18.4.2 Division operation (1)**.

Figure 18-6. Timing Chart of Multiplication Operation (00DAH × 0093H)



(2) Multiplication operation → multiplication operation

<End of first operation>

1. After the operation, an interrupt request signal (INTDMU) is generated.
2. The operation result data is stored in multiplication/division data register A0 (MDA0L and MDA0H).
3. Bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is cleared to 0 (end of operation).

<Starting second operation>

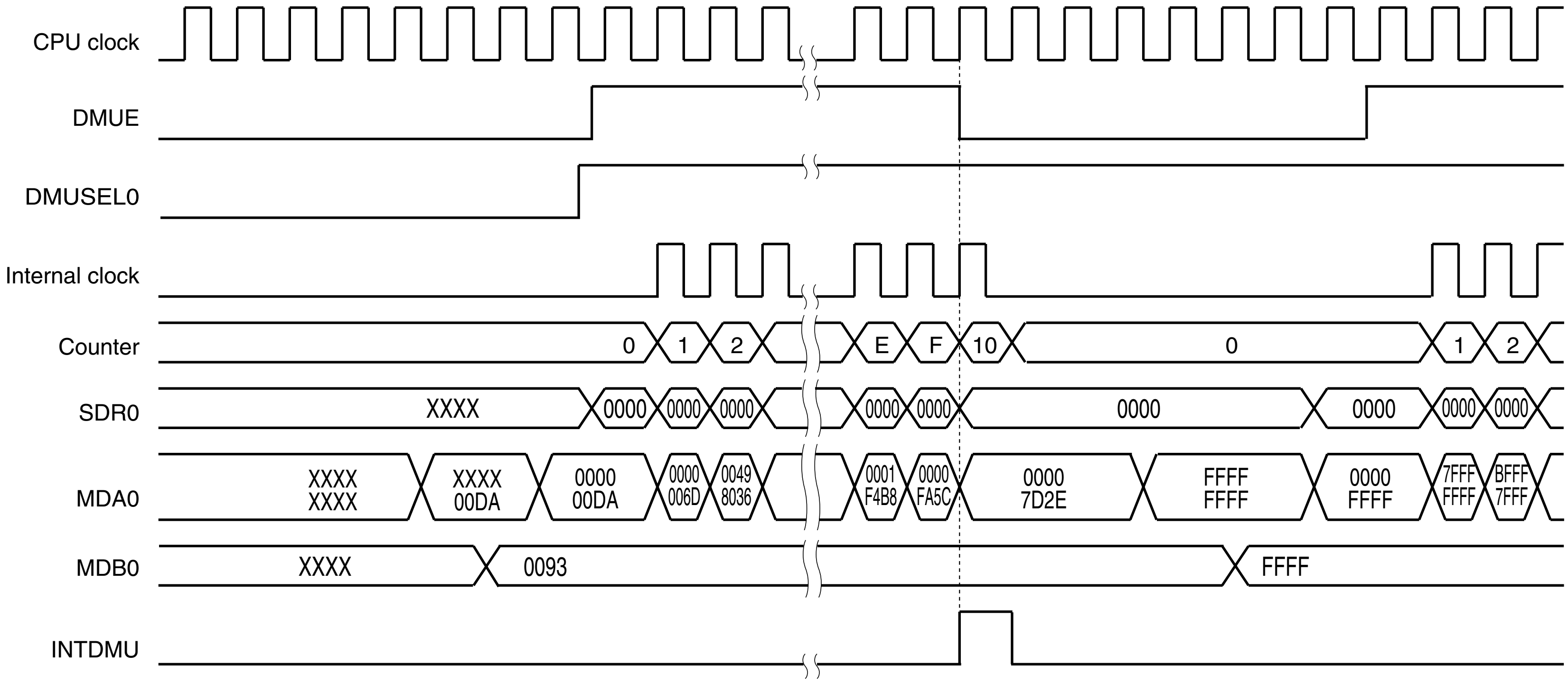
- Initial setting

4. Set operation data to multiplication/division data register A0L (MDA0L) and multiplication/division data register B0 (MDB0).
5. Set bits 0 (DMUSEL0) and 7 (DMUE) of DMUC0 to 1. The internal clock will start (operation will start).

- During operation

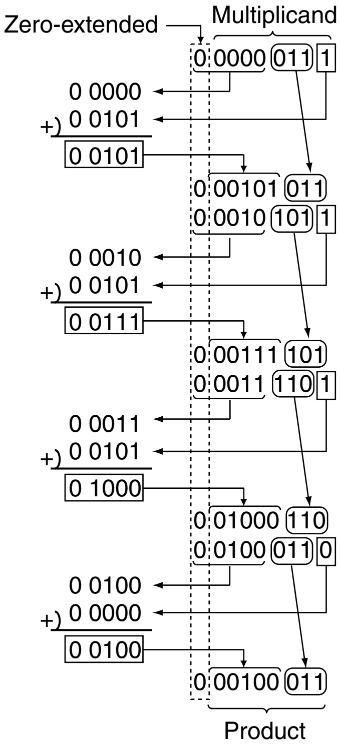
6. The operation will be completed when 16 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers during operation, and therefore the read values of these registers are not guaranteed).

Figure 18-7. Timing Chart When Multiplication Is Executed Successively (00DAH × 0093H → FFFFH × FFFFH)



As an example, the calculation of “4 bits × 4 bits (0111B × 0101B)” is shown below.

Figure 18-8. Example of Multiplication Operation by Multiplier/Divider (4 Bits × 4 Bits (0111B × 0101B))



1. If the least significant bit of the multiplicand register whose most significant bit is zero-extended is 1, the multiplier with the most significant bit zero-extended is added to the higher 5 bits of the multiplicand register.
2. If the least significant bit of the multiplicand register whose most significant bit is zero-extended is 0, zero is added to the higher 5 bits of the multiplicand register.
3. After addition, the lower 4 bits of the multiplicand register are shifted 1 bit to the right, and the product with the most significant bit of the addition result zero-extended is stored in the higher 5 bits.

18.4.2 Division operation

(1) Division operation

- Initial setting

1. Set operation data to multiplication/division data register A0 (MDA0L and MDA0H) and multiplication/division data register B0 (MDB0).
2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 0 and 1, respectively. The internal clock will start (operation will start).

- During operation

3. The operation will be completed when 32 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers and remainder data register 0 (SDR0) during operation, and therefore the read values of these registers are not guaranteed).

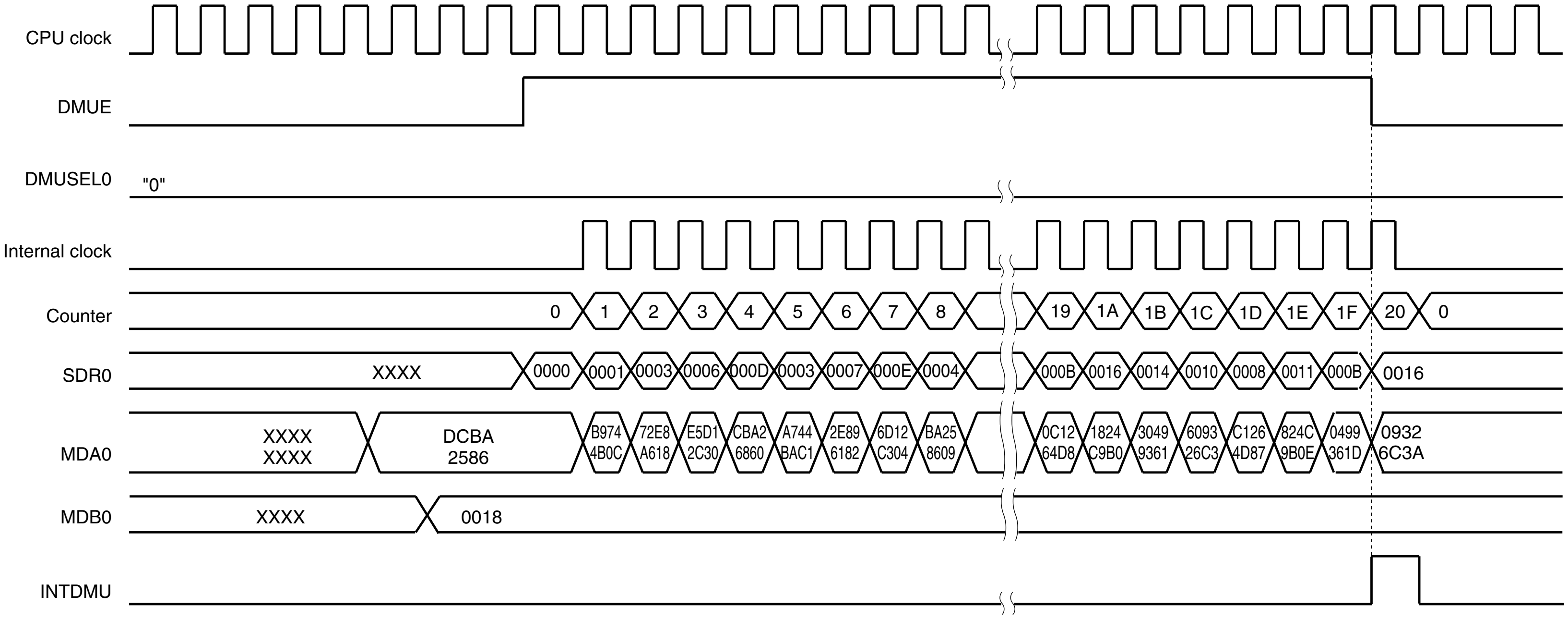
- End of operation

4. After the operation, an interrupt request signal (INTDMU) is generated.
5. The result data is stored in the MDA0L, MDA0H, and SDR0 registers.
6. DMUE is cleared to 0 (end of operation).

- Next operation

7. To execute multiplication next, start from the initial setting in **18.4.1 Multiplication operation (1)**.
8. To execute division next, start from the initial setting in **18.4.2 Division operation (1)**.

Figure 18-9. Timing Chart of Division Operation (DCBA2586H ÷ 0018H)



(2) Division operation → division operation

<End of first operation>

1. After the operation, an interrupt request signal (INTDMU) is generated.
2. The result data is stored in multiplication/division data register A0 (MDA0L and MDA0H) and remainder data register 0 (SDR0).
3. Bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is cleared to 0 (end of operation).

<Starting second operation>

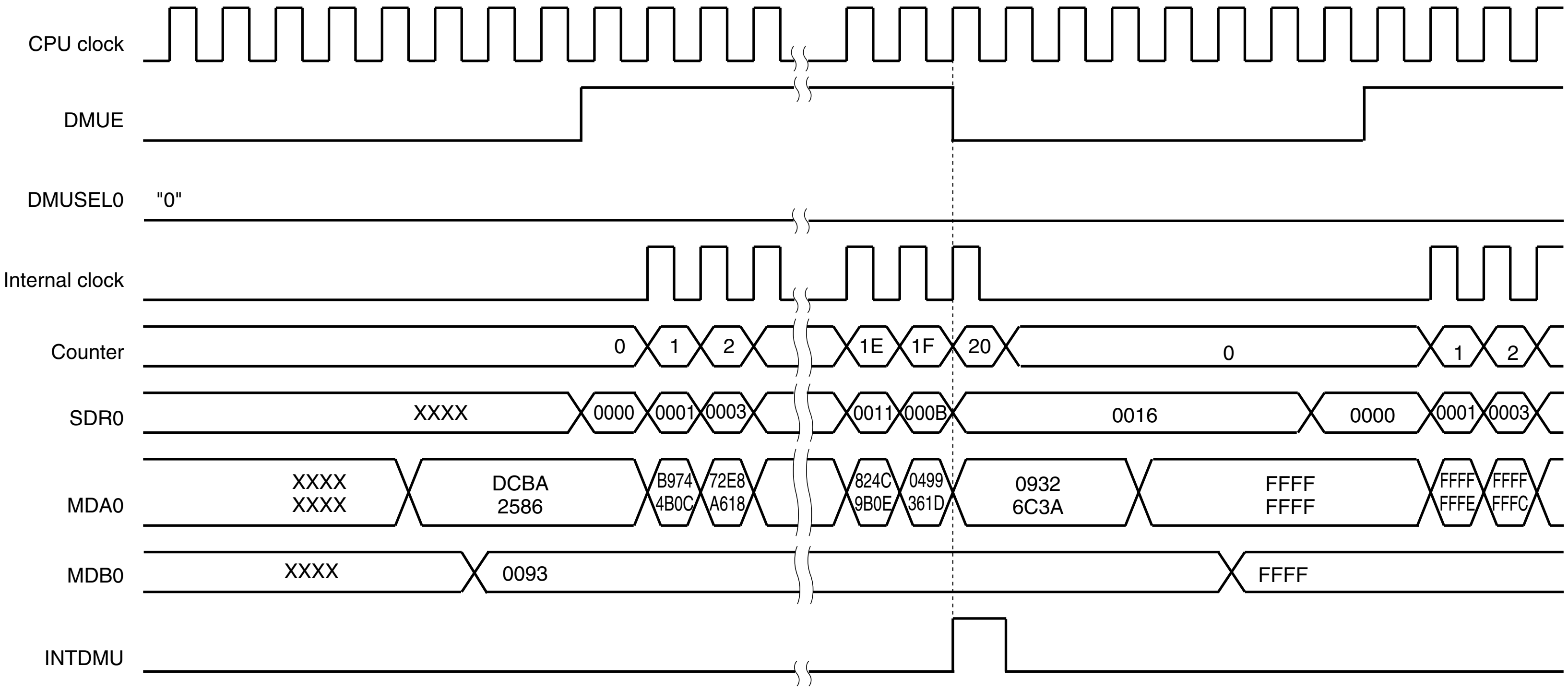
- Initial setting

4. Set operation data to the MDA0L, MDA0H, and MDB0 registers.
5. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 0 and 1, respectively. The internal clock will start (operation will start).

- During operation

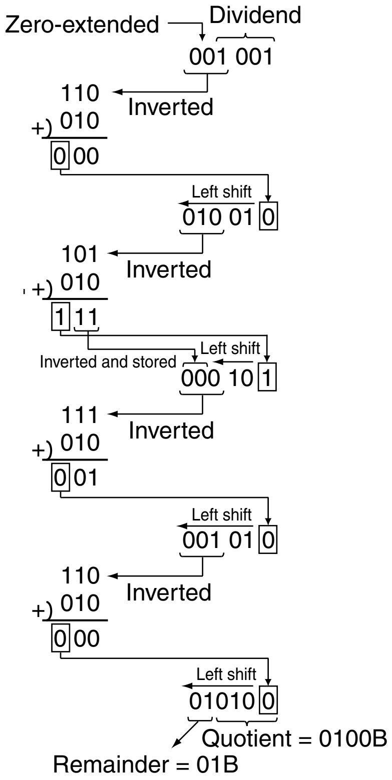
6. The operation will be completed when 32 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L, MDA0H, and SDR0 registers during operation, and therefore the read values of these registers are not guaranteed).

Figure 18-10. Timing Chart When Division Is Executed Successively (DCBA2586 ÷ 0018H → FFFFFFFFH ÷ FFFFH)



As an example, the calculation of “4 bits ÷ 2 bits (1001B ÷ 10B)” is shown below.

Figure 18-11. Example of Division Operation by Multiplier/Divider (4 Bits ÷ 2 Bits (1001B ÷ 10B))



1. The inverted value of the higher 3 bits of the dividend register whose most significant bit is extended by “00” is added to the divisor whose most significant bit is zero-extended.
2. After addition, the dividend register is shifted 1 bit to the left and “0” is stored in the least significant bit if the most significant bit is 0.
3. If the most significant bit is 1 after addition, the dividend register is shifted 1 bit to the left, and the inverted addition result, except the most significant bit, is stored in the higher 2 bits of the dividend register. 1 is stored in the least significant bit.
4. The higher 2 bits of the dividend register indicate the remainder, while the lower 4 bits indicate the quotient.
5. After addition, the lower 4 bits of the dividend register are shifted 1 bit to the right, and the addition result whose most significant bit is zero-extended is stored in the higher 5 bits.

CHAPTER 19 INTERRUPT FUNCTIONS

19.1 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, PR1H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupts with the same priority are generated simultaneously, each interrupt is serviced according to its predetermined priority (see **Table 19-1**).

A standby release signal is generated.

Nine external interrupt requests and 20 (17 in the μPD780143 and 780144) 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.

19.2 Interrupt Sources and Configuration

A total of 30 (27 in the μPD780143 and 780144) interrupt sources exist for maskable and software interrupts (see **Table 19-1**).

Table 19-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTLVI	Low-voltage detection	Internal	0004H	(A)
	1	INTP0	Pin input edge detection	External	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6	UART6 reception error generation	Internal	0012H	(A)
	8	INTSR6	End of UART6 reception		0014H	
	9	INTST6	End of UART6 transmission		0016H	
	10	INTCSI10/ INTST0	End of CSI10 transfer/end of UART0 transmission		0018H	
	11	INTTMH1	Match between TMH1 and CRH1 (when compare register is specified)		001AH	
	12	INTTMH0	Match between TMH0 and CRH0 (when compare register is specified)		001CH	
	13	INTTM50	Match between TM50 and CR50 (when compare register is specified)		001EH	
	14	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		0020H	
	15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	
	16	INTAD	End of A/D conversion		0024H	
	17	INTSR0	End of UART0 reception or reception error generation		0026H	
	18	INTWTI	Watch timer reference time interval signal		0028H	
	19	INTTM51	Match between TM51 and CR51 (when compare register is specified)		002AH	

Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 19-1.

Table 19-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	20	INTKR	Key interrupt detection	External	002CH	(C)
	21	INTWT	Watch timer overflow	Internal	002EH	(A)
	22	INTP6	Pin input edge detection	External	0030H	(B)
	23	INTP7			0032H	
	24	INTDMU	End of multiply/divide operation	Internal	0034H	(A)
	25	INTCSI11[Note 3]	End of CSI11 transfer		0036H	
	26	INTTM001[Note 3]	Match between TM01 and CR001 (when compare register is specified), TI011 pin valid edge detection (when capture register is specified)		0038H	
	27	INTTM011[Note 3]	Match between TM01 and CR011 (when compare register is specified), TI001 pin valid edge detection (when capture register is specified)		003AH	
	28	INTACSI	End of CSIA0 transfer		003CH	
Software	–	BRK	BRK instruction execution	–	003EH	(D)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on reset			
		LVI	Low-voltage detection			
		Clock monitor	X1 oscillation stop detection			
		WDT	WDT overflow			

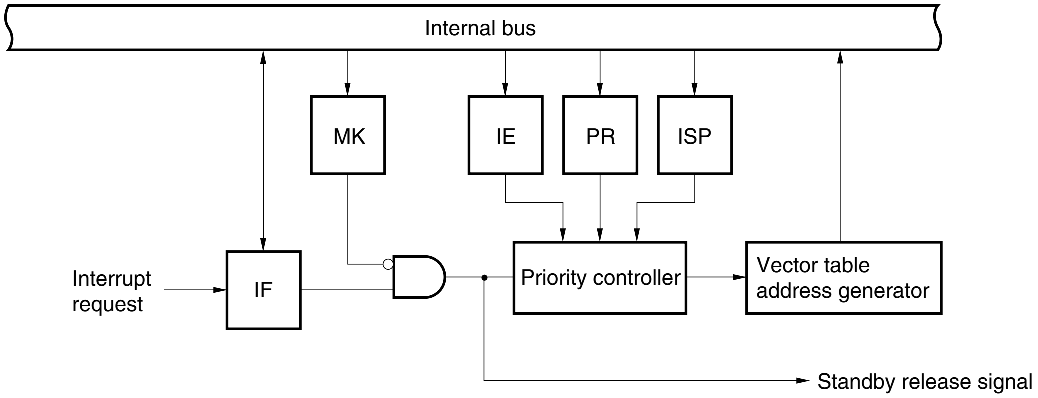
Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 19-1.

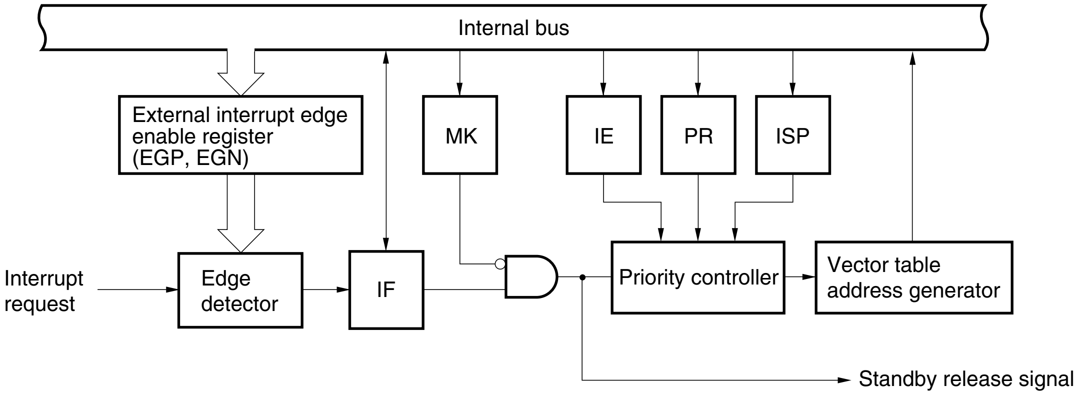
3. The interrupt sources INTCSI11, INTTM001, and INTTM011 are available only in the μPD780146, 780148, and 78F0148.

Figure 19-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



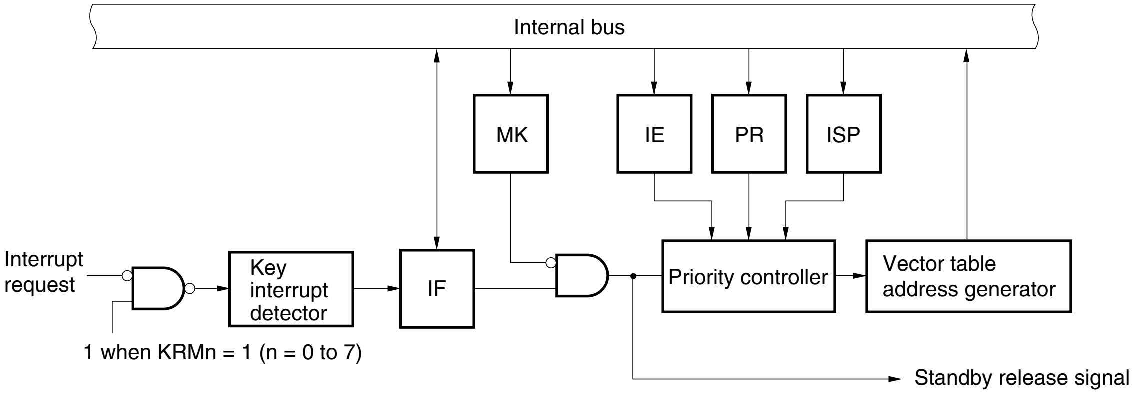
(B) External maskable interrupt (INTP0 to INTP7)



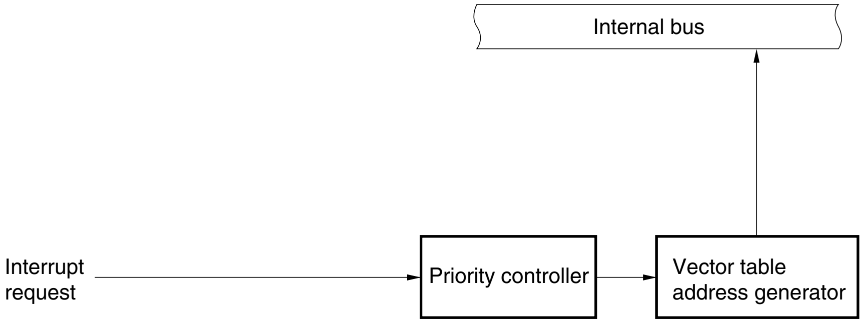
IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag

Figure 19-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag
KRM:	Key return mode register

19.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L, IF1H)
- Interrupt mask flag register (MK0L, MK0H, MK1L, MK1H)
- Priority specification flag register (PR0L, PR0H, PR1L, PR1H)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 19-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources

Interrupt Request	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
INTST6	STIF6		STMK6		STPR6	
INTCSI10	DUALIF0[Note 1]		DUALMK0[Note 1]		DUALPR0[Note 1]	
INTST0						
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
INTSR0	SRIF0		SRMK0		SRPR0	
INTWTI	WTIIF		WTIMK		WTIPR	
INTTM51	TMIF51		TMMK51		TMPR51	
INTKR	KRIF		KRMK		KRPR	
INTWR	WTIF		WTMK		WTPR	
INTP6	PIF6		PMK6		PPR6	
INTP7	PIF7		PMK7		PPR7	
INTSR0	SRIF0		SRMK0		SRPR0	
INTDMU	DMUIF	IF1H	DMUMK	MK1H	DMUPR	PR1H
INTCSI11[Note 2]	CSIIF11[Note 2]		CSIMK11[Note 2]		CSIPR11[Note 2]	
INTTM001[Note 2]	TMIF001[Note 2]		TMMK001[Note 2]		TMPR001[Note 2]	
INTTM011[Note 2]	TMIF011[Note 2]		TMMK011[Note 2]		TMPR011[Note 2]	
INTACSI	ACSIIF		ACSIMK		ACSIPR	

Notes 1. If either of the two types of interrupt sources is generated, these flags are set (1).

2. μPD780146, 780148, and 78F0148 only.

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H)

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.

IF0L, IF0H, IF1L, and IF1H are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, and IF1L and IF1H are combined to form 16-bit registers IF0 and IF1, they are set by a 16-bit memory manipulation instruction.

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

Figure 19-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)

Address: FFE0H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	DUALIF0	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
IF1L	PIF7	PIF6	WTIF	KRIF	TMIF51	WTIIF	SRIF0	ADIF

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
IF1H	0[Note 1]	0[Note 1]	0[Note 1]	ACSIIF	TMIF011[Note 2]	TMIF001[Note2]	CSIIF11[Note 2]	DMUIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Notes 1. Be sure to set bits 5 to 7 of IF1H to 0.

2. μPD780146, 780148, and 78F0148 only. Be sure to set these bits to 0 in the μPD780143 and 780144.

Cautions 1. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

2. When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, and MK1H are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, and MK1L and MK1H are combined to form 16-bit registers MK0 and MK1, they are set by a 16-bit memory manipulation instruction.

★ $\overline{\text{RESET}}$ input sets MK0L, MK0H, and MK1L to FFH and MK1H to DFH.

Figure 19-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)

Address: FFE4H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	DUALMK0	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	PMK7	PMK6	WTMK	KRMK	TMMK51	WTIMK	SRMK0	ADMK

Address: FFE7H After reset: DFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1H	1[Note 1]	1[Note 1]	0[Note 1]	ACSIMK	TMMK011[Note 2]	TMMK001[Note 2]	CSIMK11[Note 2]	DMUMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

★ **Notes 1.** Be sure to set bits 6 and 7 of MK1H to 1 and bit 5 to 0.

2. μPD780146, 780148, and 78F0148 only. Be sure to set these bits to 1 in the μPD780143 and 780144.

(3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, PR1L, and PR1H are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H, and PR1L and PR1H are combined to form 16-bit registers PR0 and PR1, they are set by a 16-bit memory manipulation instruction.

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

Figure 19-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)

Address: FFE8H After reset: FFH R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	DUALPRO	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR1L	PPR7	PPR6	WTPR	KRPR	TMPR51	WTIPR	SRPR0	ADPR

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR1H	1[Note 1]	1[Note 1]	1[Note 1]	ACSIPR	TMPR011[Note 2]	TMPR001[Note2]	CSIPR11[Note 2]	DMUPR

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Notes 1. Be sure to set bits 5 to 7 of PR1H to 1.

2. μPD780146, 780148, and 78F0148 only. Be sure to set these bits to 1 in the μPD780143 and 780144.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP7.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 19-5. 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	EGP7	EPG6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 19-3 shows the ports corresponding to EGPn and EGNn.

Table 19-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3
EGP4	EGN4	P33	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6
EGP7	EGN7	P141	INTP7

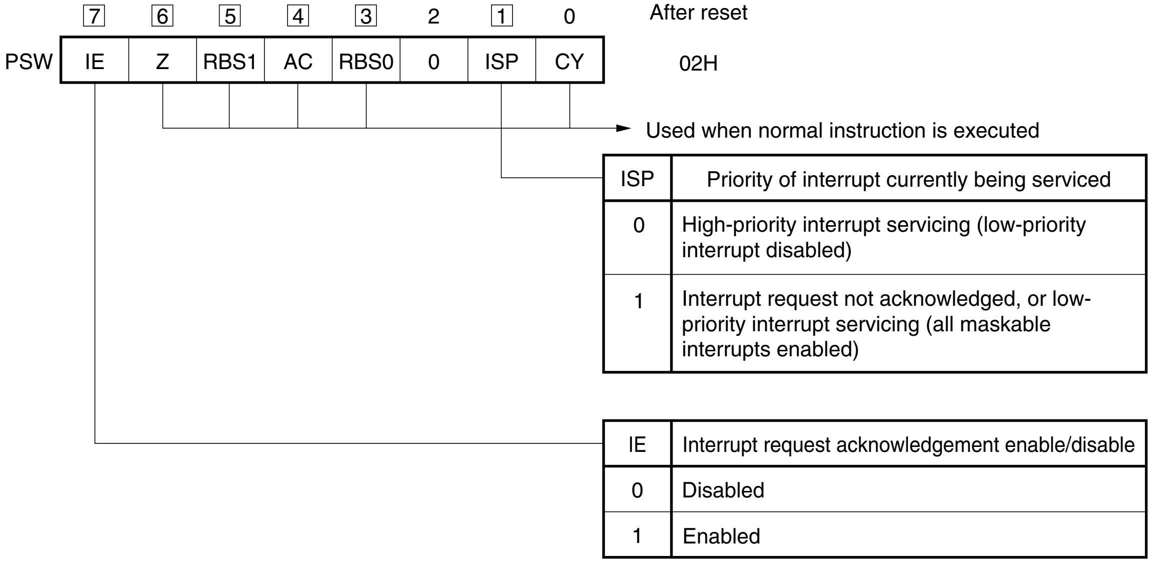
(5) Program status word (PSW)

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 19-6. 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 acknowledgement enable/disable
0	Disabled
1	Enabled

19.4 Interrupt Servicing Operations

19.4.1 Maskable interrupt request acknowledgement

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 19-4 below.

For the interrupt request acknowledgement timing, see **Figures 19-8** and **19-9**.

Table 19-4. Time from Generation of Maskable Interrupt Request Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

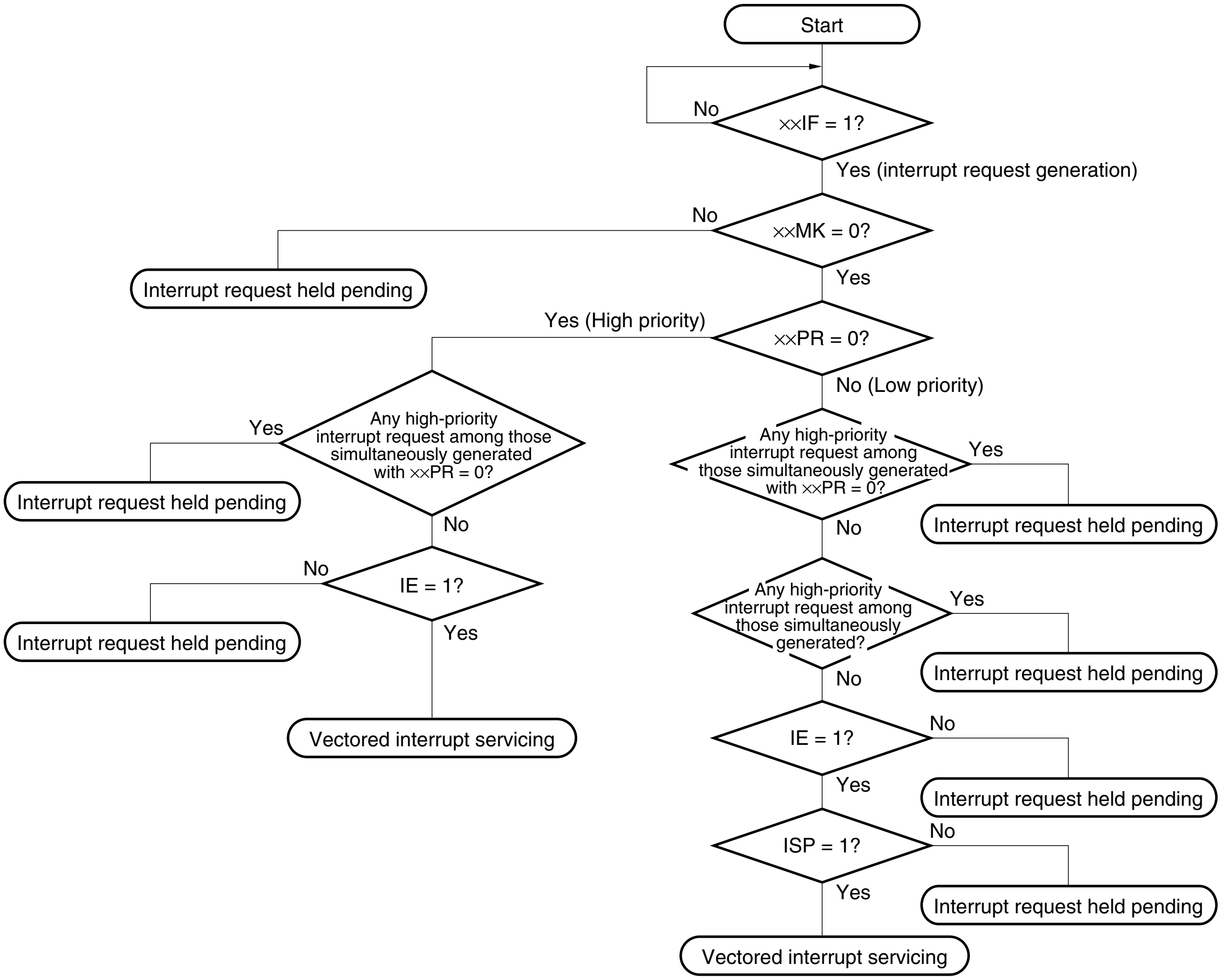
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19-7 shows the interrupt request acknowledgement 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 19-7. Interrupt Request Acknowledgement Processing Algorithm


Start
××IF = 1?
No
Yes (interrupt request generation)
××MK = 0?
No
Interrupt request held pending
Yes
××PR = 0?
Yes (High priority)
No (Low priority)
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
IE = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
Any high-priority interrupt request among those simultaneously generated?
Yes
Interrupt request held pending
No
IE = 1?
No
Interrupt request held pending
Yes
ISP = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing

××IF: Interrupt request flag

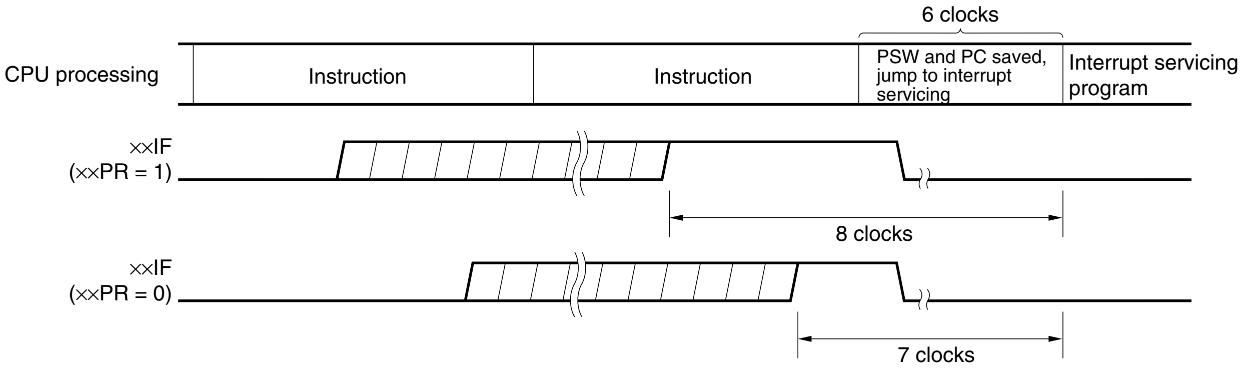
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgement of maskable interrupt request (1 = Enable, 0 = Disable)

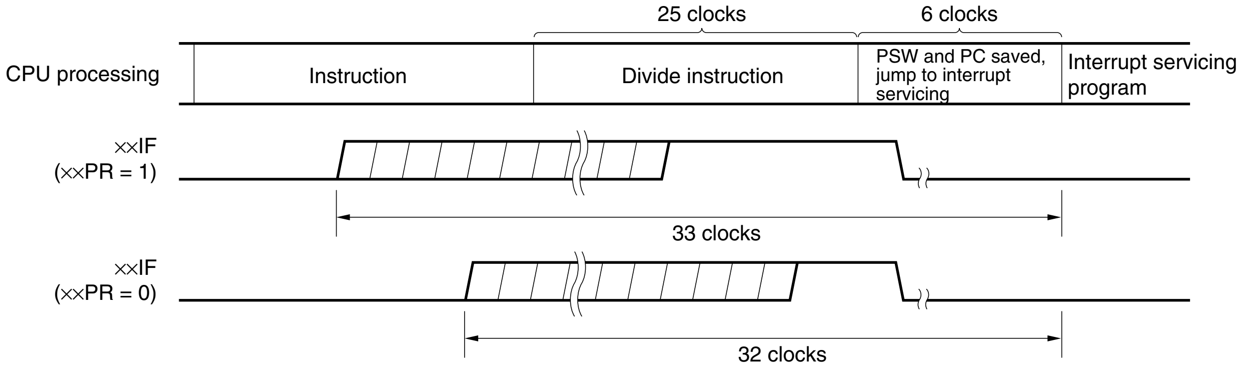
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 19-8. Interrupt Request Acknowledgement Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 19-9. Interrupt Request Acknowledgement Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

19.4.2 Software interrupt request acknowledgement

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.

19.4.3 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 acknowledgement enabled state is selected (IE = 1). Also, when an interrupt request is acknowledged, interrupt request acknowledgement 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 acknowledgement.

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 one main processing instruction execution.

Table 19-5 shows interrupt requests enabled for multiple interrupt servicing and Figure 19-10 shows multiple interrupt servicing examples.

Table 19-5. Interrupt Request Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request Interrupt Being Serviced		Maskable 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. 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 acknowledgement is disabled.

IE = 1: Interrupt request acknowledgement is enabled.

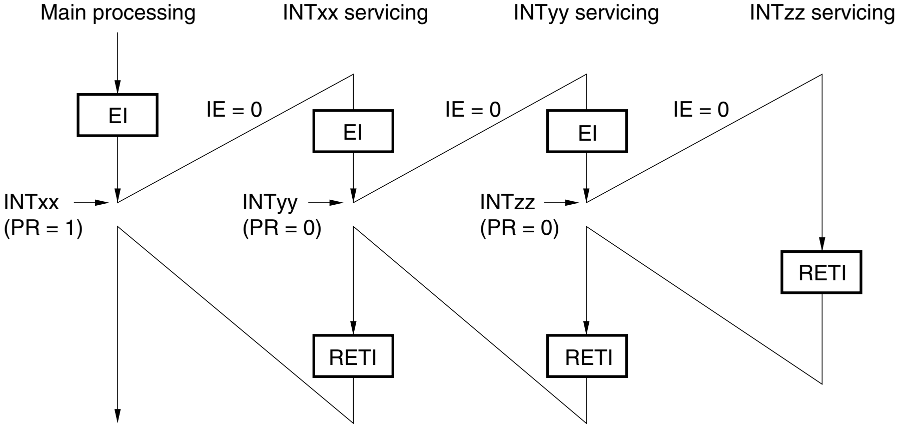
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

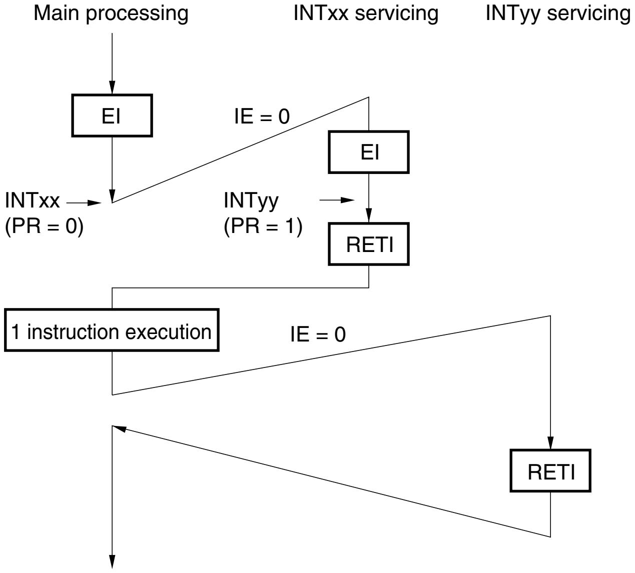
Figure 19-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgement.

Example 2. Multiple interrupt servicing does not occur due to priority control

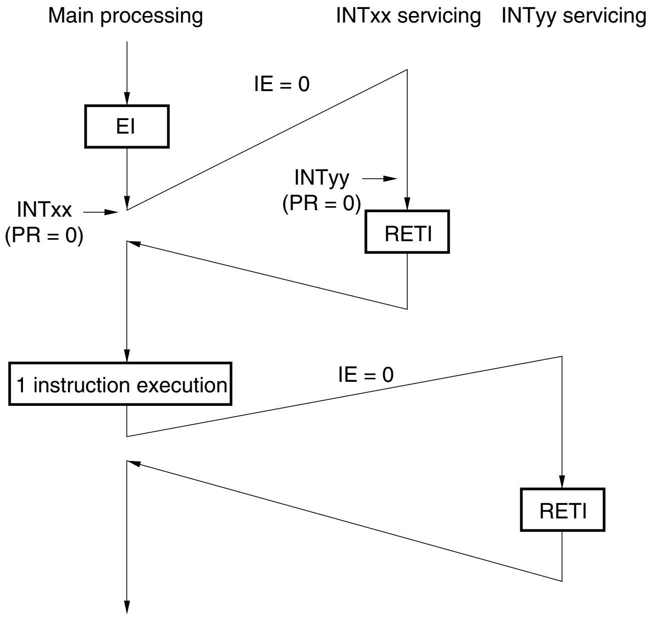


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0:	Higher priority level
PR = 1:	Lower priority level
IE = 0:	Interrupt request acknowledgement disabled

Figure 19-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledgement disabled

19.4.4 Interrupt request hold

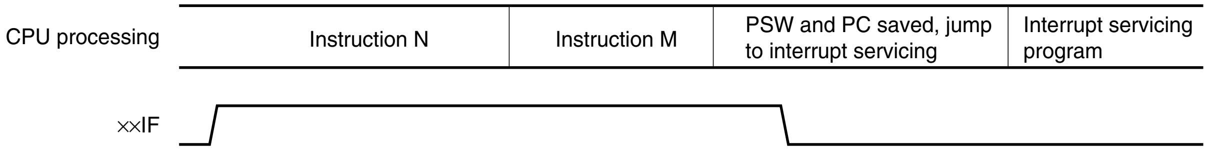
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgement is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, and PR1H 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 19-11 shows the timing at which interrupt requests are held pending.

Figure 19-11. Interrupt Request Hold


CPU processing
Instruction N
Instruction M
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF

Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 20 KEY INTERRUPT FUNCTION

20.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pins (KR0 to KR7).

Table 20-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRM0	Controls KR0 signal in 1-bit units.
KRM1	Controls KR1 signal in 1-bit units.
KRM2	Controls KR2 signal in 1-bit units.
KRM3	Controls KR3 signal in 1-bit units.
KRM4	Controls KR4 signal in 1-bit units.
KRM5	Controls KR5 signal in 1-bit units.
KRM6	Controls KR6 signal in 1-bit units.
KRM7	Controls KR7 signal in 1-bit units.

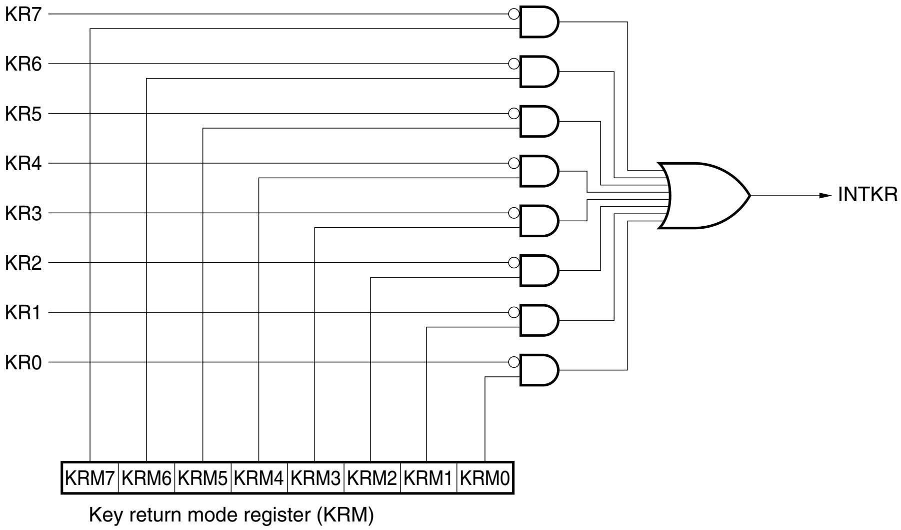
20.2 Configuration of Key Interrupt

The key interrupt consists of the following hardware.

Table 20-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 20-1. Block Diagram of Key Interrupt



20.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register controls the KRM0 to KRM7 bits using the KR0 to KR7 signals, respectively.

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

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

Figure 20-2. Format of Key Return Mode Register (KRM)

Address: FF6EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. If any of the KRM0 to KRM7 bits used is set to 1, set bits 0 to 7 (PU70 to PU77) of the corresponding pull-up resistor register 7 (PU7) to 1.

2. If KRM is changed, the interrupt request flag may be set. Therefore, disable interrupts and then change the KRM register. Clear the interrupt request flag and enable interrupts.

3. The bits not used in the key interrupt mode can be used as normal ports.

CHAPTER 21 STANDBY FUNCTION

21.1 Standby Function and Configuration

21.1.1 Standby function

Table 21-1. Relationship Between HALT Mode, STOP Mode, and Clock

	X1 Input Clock	Ring-OSC Clock	Subsystem Clock	CPU Clock
HALT mode	Oscillation continues	Oscillation continues	Oscillation continues	Operation stopped
STOP mode	Oscillation stopped	Oscillation continues	Oscillation continues	Operation stopped

The standby function is designed to reduce the power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped, but the system clock oscillator continues oscillating. In this mode, current consumption is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the X1 input clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU power consumption.

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.

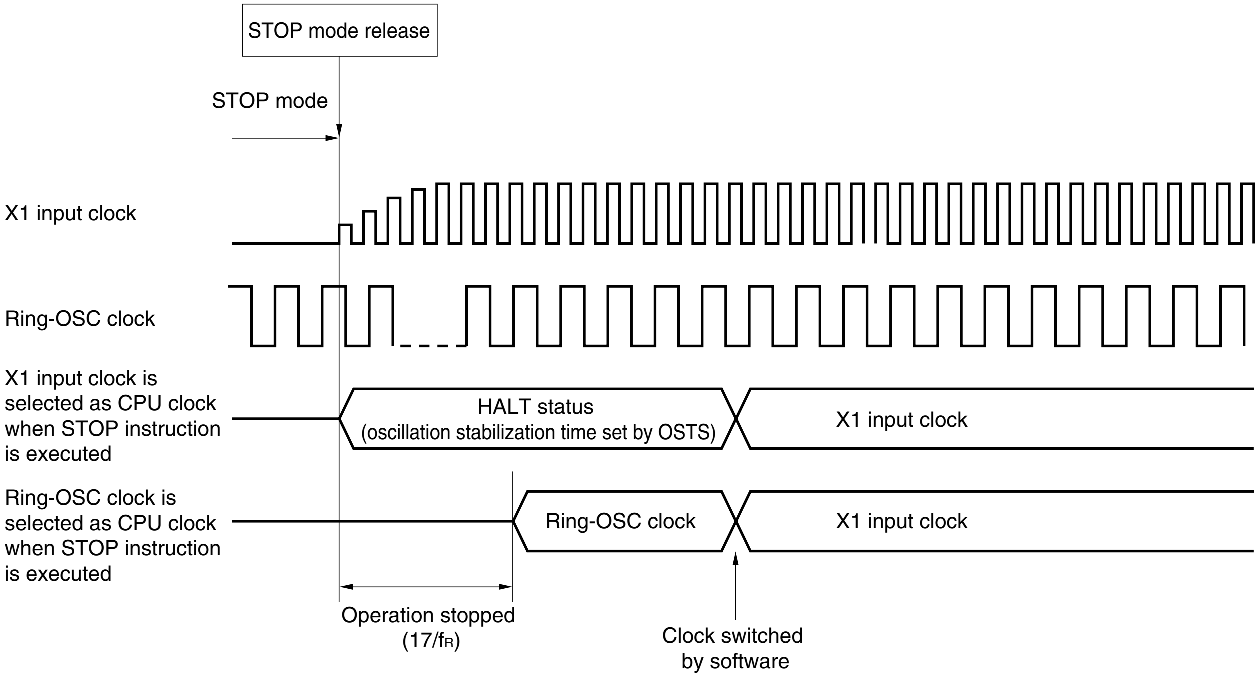
Cautions 1. STOP mode can be used only when operating on the X1 input clock or Ring-OSC clock. HALT mode can be used when operating on the X1 input clock, Ring-OSC clock, or subsystem clock. However, when the STOP instruction is executed during Ring-OSC clock operation, the X1 oscillator stops, but Ring-OSC oscillator does not stop.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation before executing STOP instruction.

3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

4. Ring-OSC clock oscillation cannot be stopped in the STOP mode. However, when the Ring-OSC clock is used as the CPU clock, the CPU operation is stopped for 17/f_R (s) after STOP mode is released.

Figure 21-1. Operation Timing When STOP Mode Is Released


STOP mode release
STOP mode
X1 input clock
Ring-OSC clock
X1 input clock is selected as CPU clock when STOP instruction is executed
HALT status
(oscillation stabilization time set by OSTS)
X1 input clock
Ring-OSC clock is selected as CPU clock when STOP instruction is executed
Ring-OSC clock
X1 input clock
Operation stopped
(17/fR)
Clock switched by software

21.1.2 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)

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

This is the status register of the X1 input clock oscillation stabilization time counter. If the Ring-OSC clock is used as the CPU clock, the X1 input clock oscillation stabilization time can be checked.

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

★ $\overline{\text{RESET}}$ input, STOP instruction, MSTOP = 1, and MCC = 1 clear OSTC to 00H.

Figure 21-2. 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}/fx min. (204.8 µs min.)
1	1	0	0	0	2^{13}/fx min. (819.2 µs min.)
1	1	1	0	0	2^{14}/fx min. (1.64 ms min.)
1	1	1	1	0	2^{15}/fx min. (3.27 ms min.)
1	1	1	1	1	2^{16}/fx min. (6.55 ms min.)

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

Remarks 1. Values in parentheses are for operation with fx = 10 MHz.

2. fx: X1 input clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 oscillation stabilization wait time when STOP mode is released. The wait time set by OSTS is valid only after STOP mode is released when the X1 input clock is selected as the CPU clock. After STOP mode is released when the Ring-OSC clock is selected, check the oscillation stabilization time using OSTC.

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

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

Figure 21-3. 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

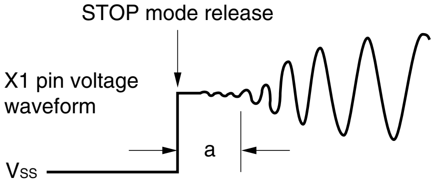
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	1	2^{11}/fx (204.8 μs)
0	1	0	2^{13}/fx (819.2 μs)
0	1	1	2^{14}/fx (1.64 ms)
1	0	0	2^{15}/fx (3.27 ms)
1	0	1	2^{16}/fx (6.55 ms)
Other than above			Setting prohibited

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

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

The X1 oscillation stabilization time counter counts only during the oscillation stabilization time set by OSTS. Therefore, note that only the statuses during the oscillation stabilization time set by OSTS are set to OSTC after STOP mode has been released.

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



Remarks 1. Values in parentheses are for operation with fx = 10 MHz.

2. fx: X1 input clock oscillation frequency

21.2 Standby Function Operation

21.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the X1 input clock, Ring-OSC clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 21-2. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on X1 Input Clock				When HALT Instruction Is Executed While CPU Is Operating on Ring-OSC Clock			
		When Ring-OSC Oscillation Continues		When Ring-OSC Oscillation Stopped[Note 1]		When X1 Input Clock Oscillation Continues		When X1 Input Clock Oscillation Stopped	
Item		When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used
System clock		The X1 oscillator, Ring-OSC oscillator, and subsystem clock oscillator are able to oscillate. Clock supply to the CPU is stopped.							
CPU		Operation stopped							
Port (latch)		Status before HALT mode was set is retained							
16-bit timer/event counter 00		Operable				Operation stopped			
16-bit timer/event counter 01[Note 2]		Operable				Operation stopped			
8-bit timer/event counter 50		Operable				Operable only when TI50 is selected as the count clock			
8-bit timer/event counter 51		Operable				Operable only when TI51 is selected as the count clock			
8-bit timer H0		Operable				Operable only when TO50 is selected as the count clock during 8-bit timer/event counter 50 operation			
8-bit timer H1		Operable				Operable only when $f_R/2^7$ is selected as the count clock			
Watch timer		Operable	Operable[Note 3]	Operable	Operable[Note 3]	Operable[Note 4]	Not operable	Operable[Note 4]	Not operable
Watchdog timer	Ring-OSC cannot be stopped[Note 5]	Operable		–		Operable			
	Ring-OSC can be stopped[Note 5]	Operation stopped							
★ A/D converter		Operable				Not operable			
Serial interface	UART0	Operable				Operable only when TO50 is selected as the serial clock during TM50 operation			
	UART6	Operable							
	CSI10	Operable				Operable only when external $\overline{\text{SCK10}}$ is selected as the serial clock			
	CSI11[Note 2]	Operable				Operable only when external $\overline{\text{SCK11}}$ is selected as the serial clock			
	CSIA0	Operable				Operation stopped			
Clock monitor		Operable		Operation stopped		Operable		Operation stopped	
Multiplier/divider		Operable				Operation stopped			
Power-on-clear function[Note 6]		Operable							
Low-voltage detection function		Operable							
External interrupt		Operable							

Notes 1. When “Stopped by software” is selected for Ring-OSC by a mask option and Ring-OSC is stopped by software (for mask options, see **CHAPTER 27 MASK OPTIONS**).

2. μPD780146, 780148, and 78F0148 only.

3. Operable when the X1 input clock is selected.

4. Operable when the subsystem clock is selected.

5. “Ring-OSC cannot be stopped” or “Ring-OSC can be stopped by software” can be selected by a mask option.

6. When “POC used” is selected by a mask option.

Table 21-2. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock			
		When X1 Input Clock Oscillation Continues		When X1 Input Clock Oscillation Stopped	
Item		When Ring-OSC Oscillation Continues	When Ring-OSC Oscillation Stopped[Note 1]	When Ring-OSC Oscillation Continues	When Ring-OSC Oscillation Stopped[Note 1]
System clock		The X1 oscillator, Ring-OSC oscillator, and subsystem clock oscillator are able to oscillate. Clock supply to the CPU is stopped.			
CPU		Operation stopped			
Port (latch)		Status before HALT mode was set is retained			
16-bit timer/event counter 00		Operable		Operation stopped	
16-bit timer/event counter 01[Note 2]		Operable		Operation stopped	
8-bit timer/event counter 50		Operable		Operable only when TI50 is selected as the count clock	
8-bit timer/event counter 51		Operable		Operable only when TI51 is selected as the count clock	
8-bit timer H0		Operable		Operable only when TO50 is selected as the count clock during 8-bit timer/event counter 50 operation	
8-bit timer H1		Operable	Operable only when the X1 input clock is selected as the count clock	Operable only when $f_R/2^7$ is selected as the count clock	Operation stopped
Watch timer		Operable		Operable only when subsystem clock is selected	
Watchdog timer	Ring-OSC cannot be stopped[Note 3]	Operable	–	Operable	–
	Ring-OSC can be stopped[Note 3]	Operation stopped			
★ A/D converter		Operable		Not operable	
Serial interface	UART0	Operable		Operable only when TO50 is selected as the serial clock during TM50 operation	
	UART6	Operable			
	CSI10	Operable		Operable only when external clock is selected as the serial clock	
	CSI11[Note 2]	Operable		Operable only when external clock is selected as the serial clock	
	CSIA0	Operable		Operation stopped	
Clock monitor		Operable	Operation stopped		
Multiplier/divider		Operable		Operation stopped	
Power-on-clear function[Note 4]		Operable			
Low-voltage detection function		Operable			
External interrupt		Operable			

Notes 1. When "Stopped by software" is selected for Ring-OSC by a mask option and Ring-OSC is stopped by software (for mask options, see **CHAPTER 27 MASK OPTIONS**).

2. μPD780146, 780148, and 78F0148 only.

3. "Ring-OSC cannot be stopped" or "Ring-OSC can be stopped by software" can be selected by a mask option.

4. When "POC used" is selected by a mask option.

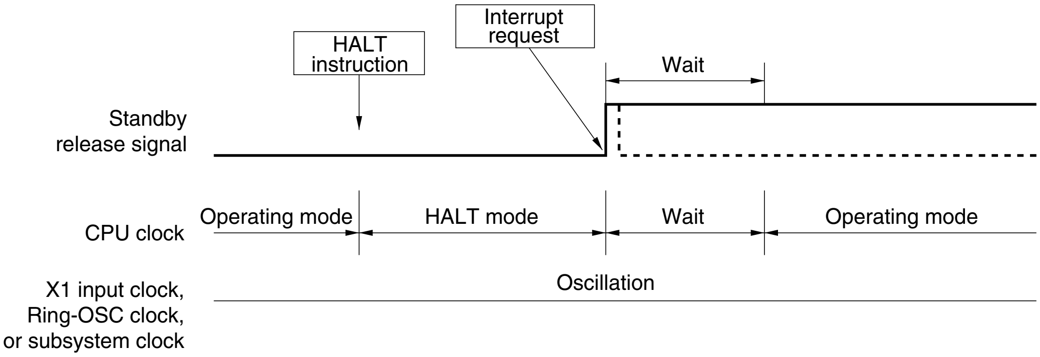
(2) HALT mode release

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 acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 21-4. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

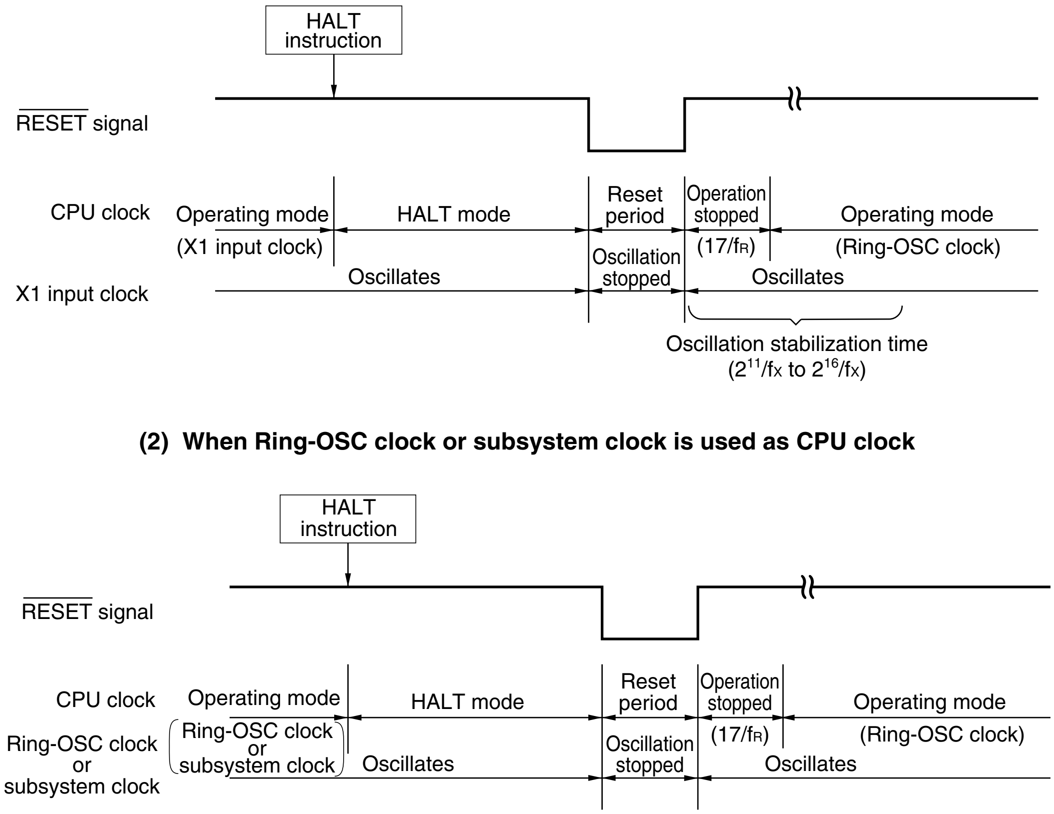
2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

(b) Release by $\overline{\text{RESET}}$ input

When the $\overline{\text{RESET}}$ signal is input, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 21-5. HALT Mode Release by $\overline{\text{RESET}}$ Input



(1) When X1 input clock is used as CPU clock

(2) When Ring-OSC clock or subsystem clock is used as CPU clock

Remarks 1. fx: X1 input clock oscillation frequency

2. fR: Ring-OSC clock oscillation frequency

Table 21-3. Operation After HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: Don't care

21.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the X1 input clock or Ring-OSC clock.

Caution Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 21-4. Operating Statuses in STOP Mode

STOP Mode Setting		When STOP Instruction Is Executed While CPU Is Operating on X1 Input Clock				When STOP Instruction Is Executed While CPU Is Operating on Ring-OSC Clock	
		When Ring-OSC Oscillation Continues		When Ring-OSC Oscillation Stopped[Note 1]			
Item		When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used
System clock		Only X1 oscillator oscillation is stopped. Clock supply to the CPU is stopped.					
CPU		Operation stopped					
Port (latch)		Status before STOP mode was set is retained					
16-bit timer/event counter 00		Operation stopped					
16-bit timer/event counter 01[Note 2]		Operation stopped					
8-bit timer/event counter 50		Operable only when TI50 is selected as the count clock					
8-bit timer/event counter 51		Operable only when TI51 is selected as the count clock					
8-bit timer H0		Operable only when TO50 is selected as the count clock during 8-bit timer/event counter 50 operation					
8-bit timer H1		Operable[Note 3]		Operation stopped		Operable[Note 3]	
Watch timer		Operable[Note 4]	Operation stopped	Operable[Note 4]	Operation stopped	Operable[Note 4]	Operation stopped
Watchdog timer	Ring-OSC cannot be stopped[Note 5]	Operable		–		Operable	
	Ring-OSC can be stopped[Note 5]	Operation stopped					
A/D converter		Operation stopped					
Serial interface	UART0	Operable only when TO50 is selected as the serial clock during TM50 operation					
	UART6						
	CSI10	Operable only when external $\overline{\text{SCK10}}$ is selected as the serial clock					
	CSI11[Note 2]	Operable only when external $\overline{\text{SCK11}}$ is selected as the serial clock					
	CSIA0	Operation stopped					
Clock monitor		Operation stopped					
Multiplier/divider		Operation stopped					
Power-on-clear function[Note 6]		Operable					
Low-voltage detection function		Operable					
External interrupt		Operable					

Notes 1. When “Stopped by software” is selected for Ring-OSC by a mask option and Ring-OSC is stopped by software (for mask options, see **CHAPTER 27 MASK OPTIONS**).

2. µPD780146, 780148, and 78F0148 only.

3. Operation continues only when $f_R/2^7$ is selected as the count clock.

4. Operable when the subsystem clock is selected.

5. “Ring-OSC cannot be stopped” or “Ring-OSC can be stopped by software” can be selected by a mask option.

6. When “POC used” is selected by a mask option.

(2) STOP mode release

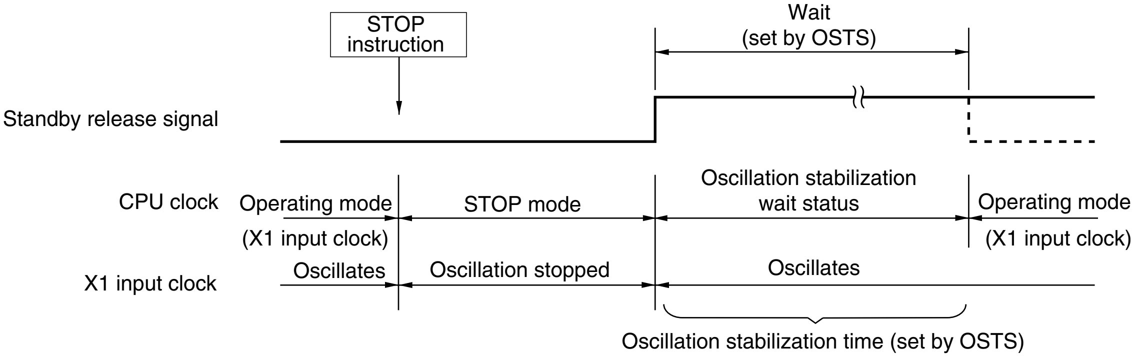
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

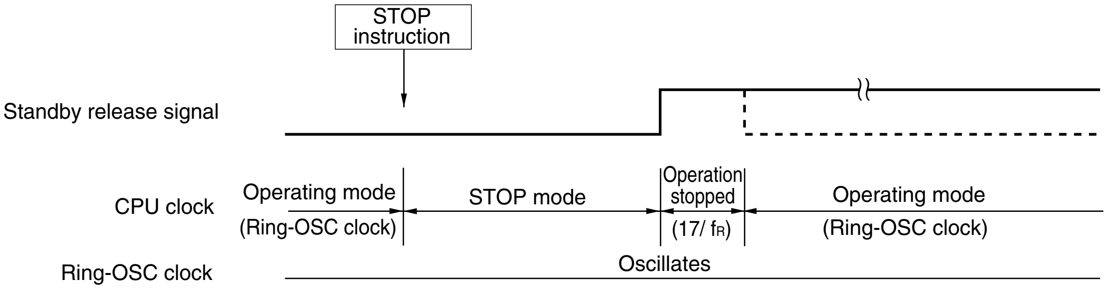
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 21-6. STOP Mode Release by Interrupt Request Generation

(1) When X1 input clock is used as CPU clock


STOP instruction
Wait (set by OSTS)
Standby release signal
CPU clock
Operating mode (X1 input clock)
STOP mode
Oscillation stabilization wait status
Operating mode (X1 input clock)
X1 input clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stabilization time (set by OSTS)

(2) When Ring-OSC clock is used as CPU clock


STOP instruction
Standby release signal
CPU clock
Operating mode (Ring-OSC clock)
STOP mode
Operation stopped (17/ fR)
Operating mode (Ring-OSC clock)
Ring-OSC clock
Oscillates

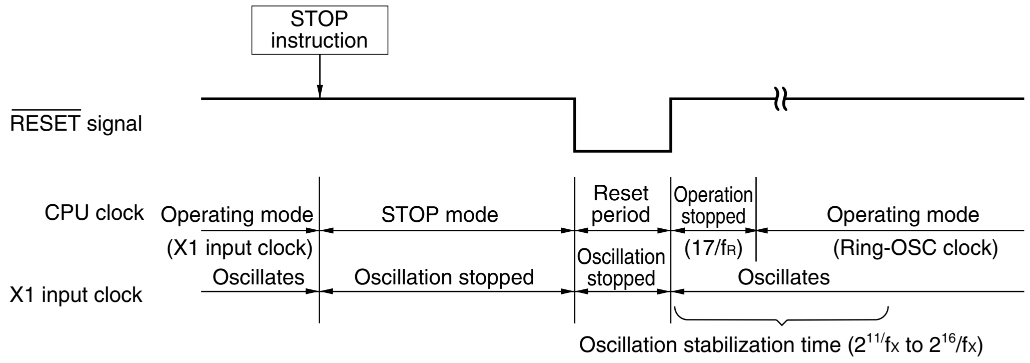
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

When the $\overline{\text{RESET}}$ signal is input, STOP mode is released and a reset operation is performed after the oscillation stabilization time has elapsed.

Figure 21-7. STOP Mode Release by $\overline{\text{RESET}}$ Input

(1) When X1 input clock is used as CPU clock


STOP instruction
$\overline{\text{RESET}}$ signal
CPU clock
X1 input clock
Operating mode (X1 input clock)
STOP mode
Reset period
Operation stopped (17/fR)
Operating mode (Ring-OSC clock)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (2^{11}/fx to 2^{16}/fx)

(2) When Ring-OSC clock is used as CPU clock

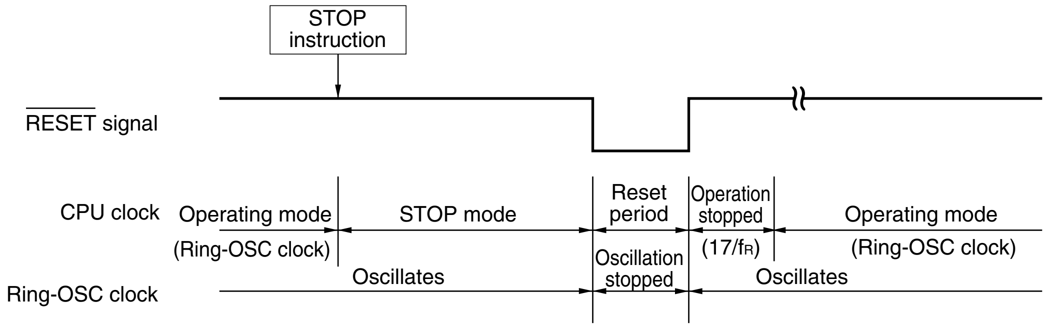

STOP instruction
$\overline{\text{RESET}}$ signal
CPU clock
Ring-OSC clock
Operating mode (Ring-OSC clock)
STOP mode
Reset period
Operation stopped (17/fR)
Operating mode (Ring-OSC clock)
Oscillates
Oscillation stopped
Oscillates

Table 21-5. Operation After STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	STOP mode held
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: Don't care

CHAPTER 22 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 X1 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, X1 clock oscillation stop is detected by the clock monitor, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Table 22-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release, except for P130, which is low-level output.

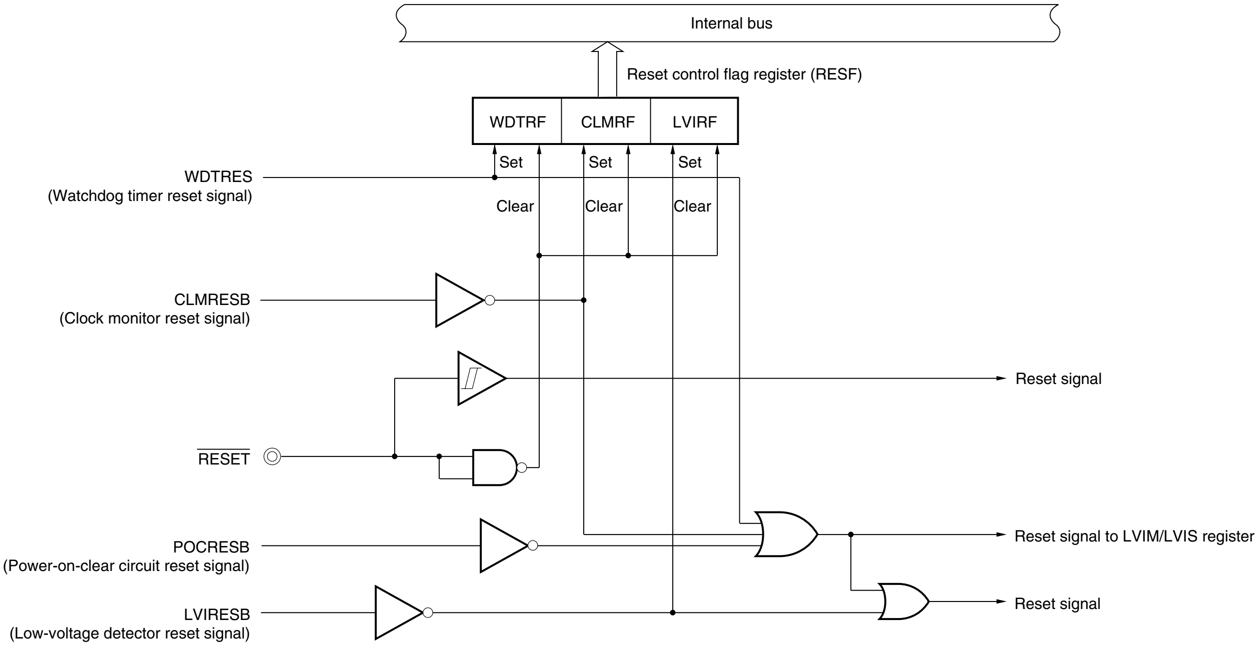
When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for $17/f_R$ (s). A reset generated by the watchdog timer and clock monitor sources is automatically released after the reset, and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for $17/f_R$ (s) (see **Figures 22-2** to **22-4**). Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} > V_{POC}$ or $V_{DD} > V_{LVI}$ after the reset, and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for $17/f_R$ (s) (see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 25 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 X1 input clock and Ring-OSC clock stop oscillating.

3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance, except for P130, which is set to low-level output.

Figure 22-1. Block Diagram of Reset Function


Internal bus
Reset control flag register (RESF)
WDTRF
CLMRF
LVIRF
Set
Set
Set
Clear
Clear
Clear
WDTRES
(Watchdog timer reset signal)
CLMRESB
(Clock monitor reset signal)
Reset signal
RESET
POCRESB
(Power-on-clear circuit reset signal)
Reset signal to LVIM/LVIS register
LVIRESB
(Low-voltage detector reset signal)
Reset signal

Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level selection register

Figure 22-2. Timing of Reset by $\overline{\text{RESET}}$ Input

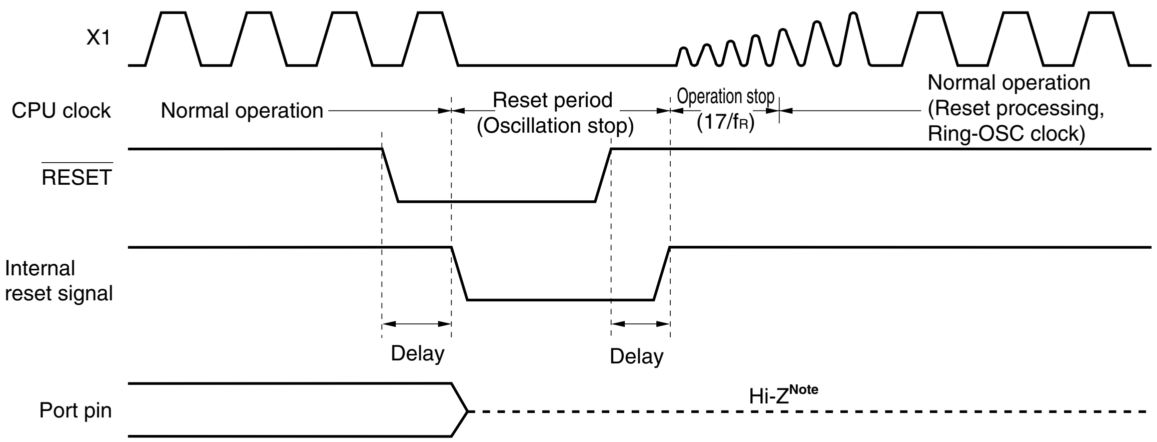
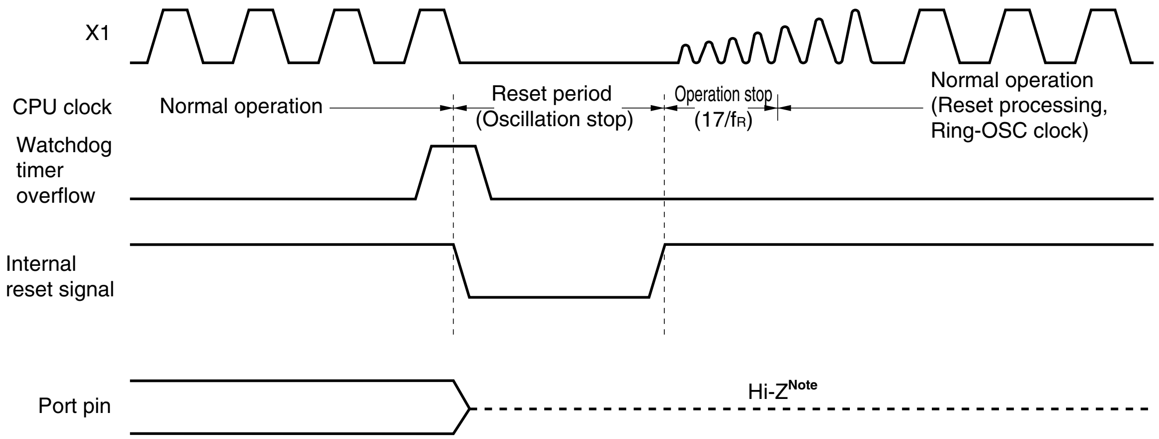
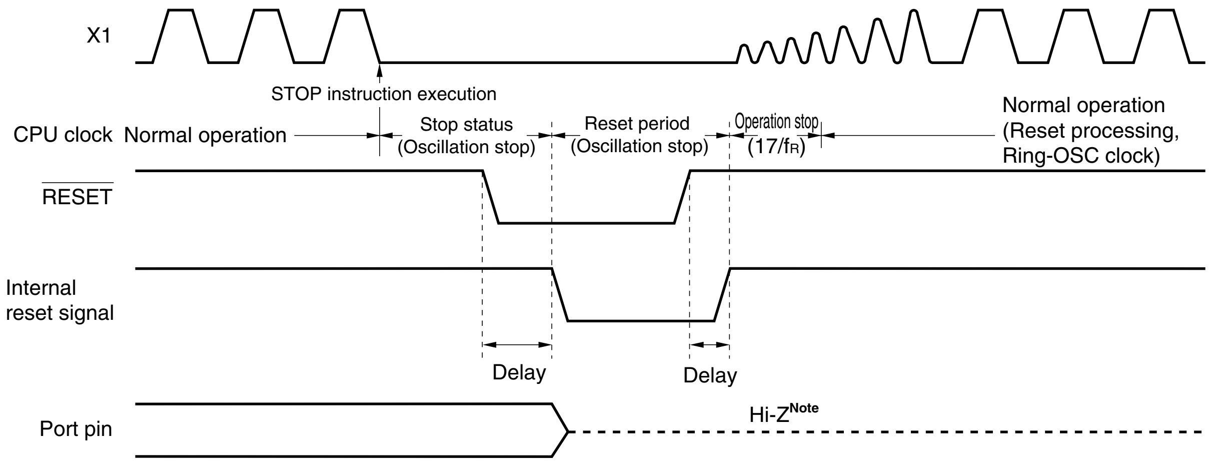


Figure 22-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

Figure 22-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note The port pins become high impedance, except for P130, which is set to low-level output.

Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 25 LOW-VOLTAGE DETECTOR**.

Table 22-1. Hardware Statuses After Reset (1/3)

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]
Ports (P0 to P7, P12 to P14) (output latches)		00H
Port mode registers (PM0, PM1, PM3 to PM7, PM12, PM14)		FFH
Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU12, PU14)		00H
Input switch control register (ISC)		00H
Internal memory size switching register (IMS)		CFH
Internal expansion RAM size switching register (IXS)		0CH
Memory expansion mode register (MEM)		00H
Memory expansion wait setting register (MM)		10H
Processor clock control register (PCC)		00H
Ring-OSC mode register (RCM)		00H
Main clock mode register (MCM)		00H
Main OSC control register (MOC)		00H
Oscillation stabilization time select register (OSTS)		05H
Oscillation stabilization time counter status register (OSTC)		00H
16-bit timer/event counters 00, 01[Note 3]	Timer counters 00, 01 (TM00, TM01)	0000H
	Capture/compare registers 000, 010, 001, 011 (CR000, CR010, CR001, CR011)	0000H
	Mode control registers 00, 01 (TMC00, TMC01)	00H
	Prescaler mode registers 00, 01 (PRM00, PRM01)	00H
	Capture/compare control registers 00, 01 (CRC00, CRC01)	00H
	Timer output control registers 00, 01 (TOC00, TOC01)	00H
8-bit timer/event counters 50, 51	Timer counters 50, 51 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	00H
	Timer clock selection registers 50, 51 (TCL50, TCL51)	00H
	Mode control registers 50, 51 (TMC50, TMC51)	00H
8-bit timers H0, H1	Compare registers 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	Mode registers (TMHMD0, TMHMD1)	00H
	Carrier control register 1 (TMCYC1)[Note 4]	00H
Watch timer	Operation mode register (WTM)	00H
Clock output/buzzer output controller	Clock output selection register (CKS)	00H

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. 16-bit timer/event counter 01 is available only for the μPD780146, 780148, and 78F0148.

4. 8-bit timer H1 only.

Table 22-1. Hardware Statuses After Reset (2/3)

Hardware		Status After Reset
Watchdog timer	Mode register (WDTM)	67H
	Enable register (WDTE)	9AH
A/D converter	Conversion result register (ADCR)	Undefined
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	Power-fail comparison mode register (PFM)	00H
	Power-fail comparison threshold register (PFT)	00H
Serial interface UART0	Receive buffer register 0 (RXB0)	FFH
	Transmit shift register 0 (TXS0)	FFH
	Asynchronous serial interface operation mode register 0 (ASIM0)	01H
	Baud rate generator control register 0 (BRGC0)	1FH
Serial interface UART6	Receive buffer register 6 (RXB6)	FFH
	Transmit buffer register 6 (TXB6)	FFH
	Asynchronous serial interface operation mode register 6 (ASIM6)	01H
	Asynchronous serial interface reception error status register 6 (ASIS6)	00H
	Asynchronous serial interface transmission status register 6 (ASIF6)	00H
	Clock selection register 6 (CKSR6)	00H
	Baud rate generator control register 6 (BRGC6)	FFH
	Asynchronous serial interface control register 6 (ASICL6)	16H
Serial interfaces CSI10, CSI11 Note	Transmit buffer registers 10, 11 (SOTB10, SOTB11)	Undefined
	Serial I/O shift registers 10, 11 (SIO10, SIO11)	Undefined
	Serial operation mode registers 10, 11 (CSIM10, CSIM11)	00H
	Serial clock selection registers 10, 11 (CSIC10, CSIC11)	00H
Serial interface CSIA0	Shift register 0 (SIOA0)	00H
	Operation mode specification register 0 (CSIMA0)	00H
	Status register 0 (CSIS0)	00H
	Trigger register 0 (CSIT0)	00H
	Divisor selection register 0 (BRGCA0)	03H
	Automatic data transfer address point specification register 0 (ADTP0)	00H
	Automatic data transfer interval specification register 0 (ADTI0)	00H
	Automatic data transfer address count register 0 (ADTC0)	00H
Multiplier/divider	Remainder data register 0 (SDR0)	0000H
	Multiplication/division data register A0 (MDA0H, MDA0L)	0000H
	Multiplication/division data register B0 (MDB0)	0000H
	Multiplier/divider control register 0 (DMUC0)	00H
Key interrupt	Key return mode register (KRM)	00H
Clock monitor	Mode register (CLM)	00H

Note Serial interface CSI11 is available only for the μPD780146, 780148, and 78F0148.

Table 22-1. Hardware Statuses After Reset (3/3)

Hardware		Status After Reset
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, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	Mask flag registers 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	Priority specification flag registers 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	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.

Reset Source Register	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by CLM	Reset by LVI
RESF	See **Table 22-2**.				
LVIM	Cleared (00H)	Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS					

22.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/KF1 Series. 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 22-5. 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 22-2.

Table 22-2. RESF Status When Reset Request Is Generated

Reset Source Flag	$\overline{\text{RESET}}$ input	Reset by POC	Reset by WDT	Reset by CLM	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held	Held
CLMRF			Held	Set (1)	Held
LVIRF			Held	Held	Set (1)

CHAPTER 23 CLOCK MONITOR

23.1 Functions of Clock Monitor

The clock monitor samples the X1 input clock using the on-chip Ring-OSC, and generates an internal reset signal when the X1 input 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 22 RESET FUNCTION**.

The clock monitor automatically stops under the following conditions.

- In STOP mode and during the oscillation stabilization time
- ★ When the X1 input clock is stopped by software (MSTOP = 1 or MCC = 1)
- During the oscillation stabilization time after reset is released
- When the Ring-OSC clock is stopped

Remark MSTOP: Bit 7 of the main OSC control register (MOC)

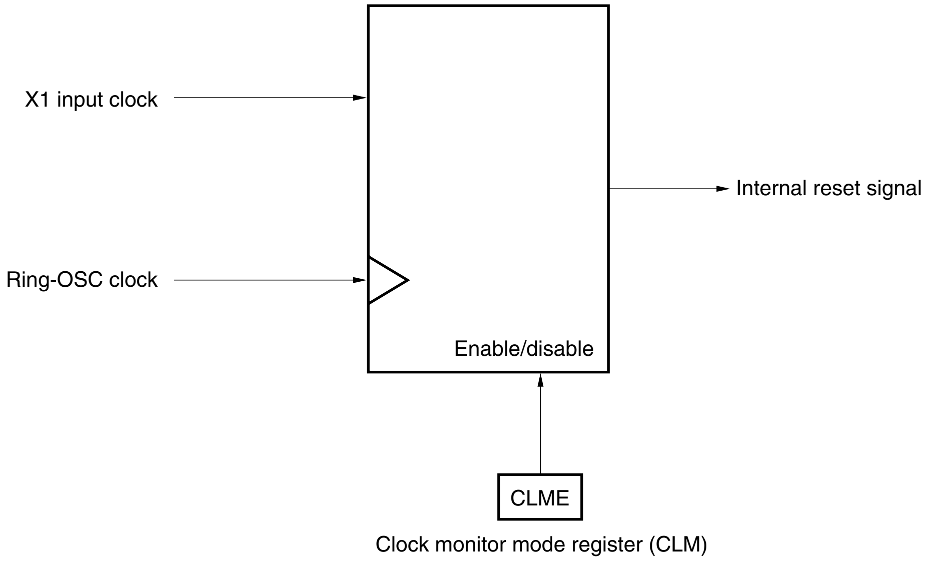
23.2 Configuration of Clock Monitor

Clock monitor consists of the following hardware.

Table 23-1. Configuration of Clock Monitor

Item	Configuration
Control register	Clock monitor mode register (CLM)

Figure 23-1. Block Diagram of Clock Monitor



23.3 Registers Controlling Clock Monitor

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 23-2. Format of Clock Monitor Mode Register (CLM)

Address: FFA9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	$\boxed{0}$
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. CLMRF is read by software and then automatically cleared to 0. CLMRF is cleared under the following conditions.

- **$\overline{\text{RESET}}$ input**
- **Internal reset signal generation by POC**
- **After read by software**

23.4 Operation of Clock Monitor

This section explains the functions of the clock monitor. The start and stop conditions are as follows.

<Start condition>

When bit 0 (CLME) of the clock monitor mode register (CLM) is set to operation enabled (1).

<Stop condition>

- In STOP mode and during the oscillation stabilization time
- During the oscillation stabilization time after reset is released
- ★ When the X1 input clock is stopped by software (MSTOP = 1 or MCC = 1)
- When the Ring-OSC clock is stopped

Remark MSTOP: Bit 7 of the main OSC control register (MOC)

Table 23-2. Operation Status of Clock Monitor (When CLME = 1)

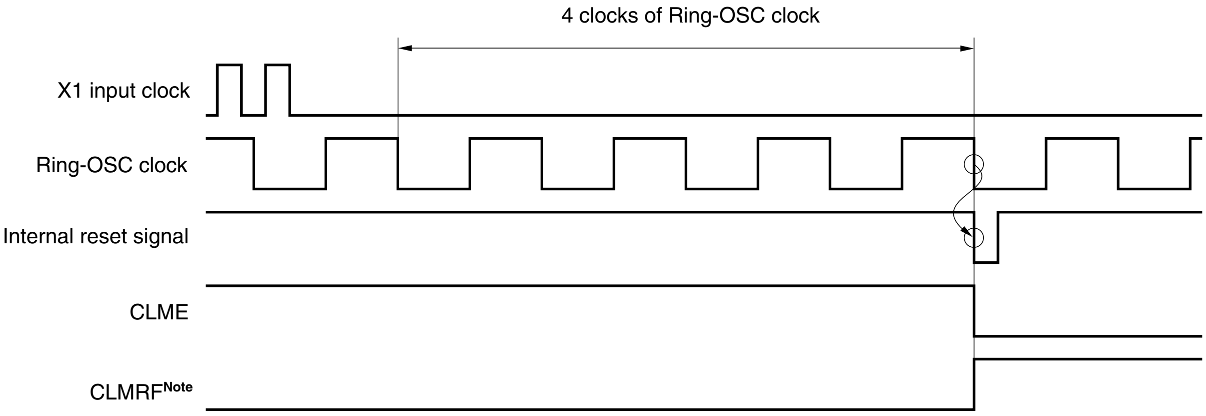
CPU Operation Clock	Operation Mode	X1 Input Clock Status	Ring-OSC Clock Status	Clock Monitor Status
X1 input clock	STOP mode	Stopped	Oscillating	Stopped
			Stopped[Note]	
	$\overline{\text{RESET}}$ input		Oscillating	
			Stopped[Note]	
	HALT mode	Oscillating	Oscillating	Operating
			Stopped[Note]	Stopped
Ring-OSC clock	STOP mode	Stopped	Oscillating	Stopped
	$\overline{\text{RESET}}$ input			
	HALT mode	Oscillating		Operating
		Stopped		Stopped

Note The Ring-OSC clock is stopped only when the “Ring-OSC can be stopped by software” is selected by a mask option. If “Ring-OSC cannot be stopped” is selected, the Ring-OSC clock cannot be stopped.

The clock monitor timing is as shown in Figure 23-3.

Figure 23-3. Timing of Clock Monitor (1/3)

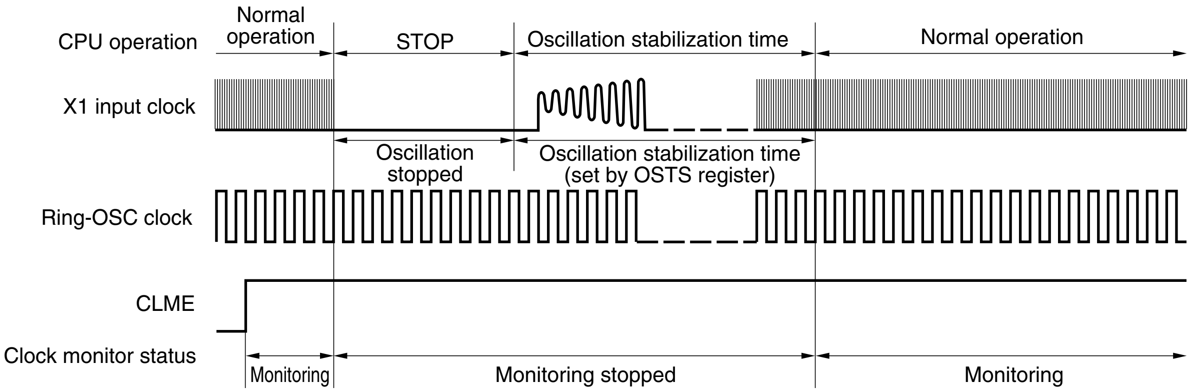
(1) When internal reset is executed by oscillation stop of X1 input clock



Note CLMRF is read by software and then automatically cleared to 0. CLMRF is cleared under the following conditions.

- $\overline{\text{RESET}}$ input
- Internal reset signal generation by POC
- After read by software

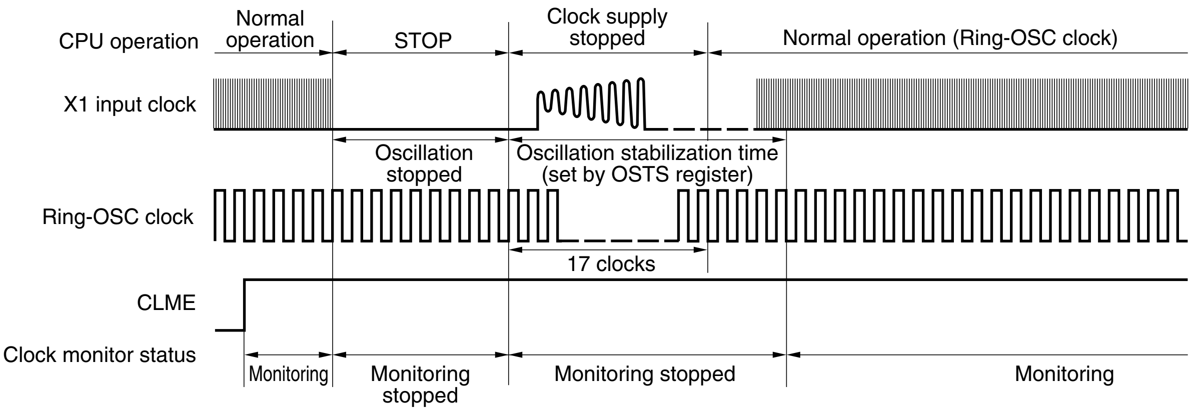
(2) Clock monitor status after STOP mode is released
(CLME = 1 is set when CPU clock operates on X1 input clock and before entering STOP mode)



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 X1 input clock oscillation stabilization time. Monitoring is stopped in STOP mode and during the oscillation stabilization time.

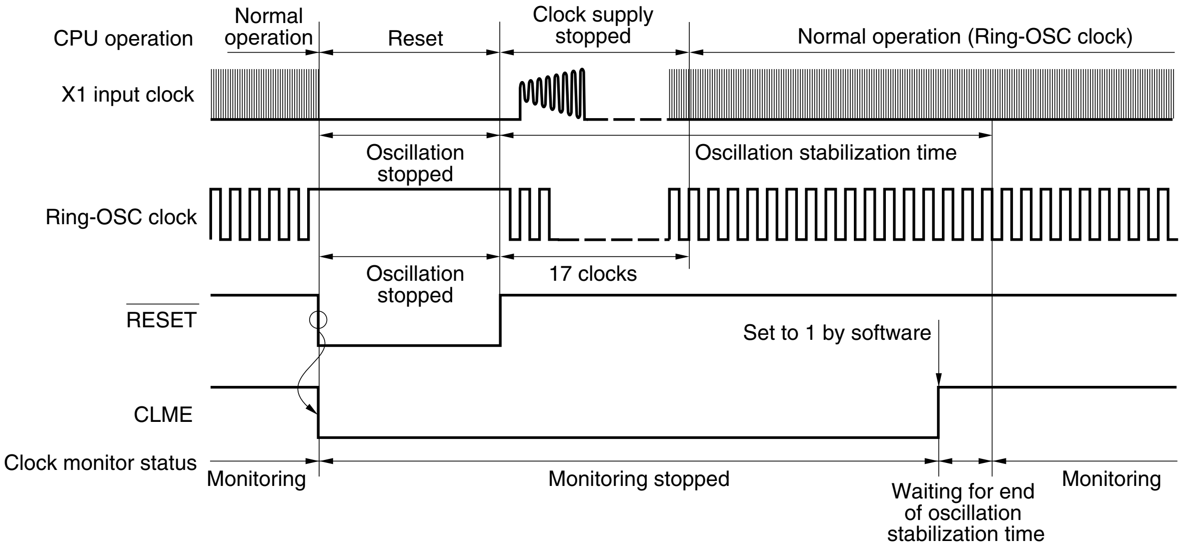
Figure 23-3. Timing of Clock Monitor (2/3)

(3) Clock monitor status after STOP mode is released (CLME = 1 is set when CPU clock operates on Ring-OSC clock and before entering STOP mode)



When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before entering STOP mode, monitoring automatically starts at the end of the X1 input clock oscillation stabilization time. Monitoring is stopped in STOP mode and during the oscillation stabilization time.

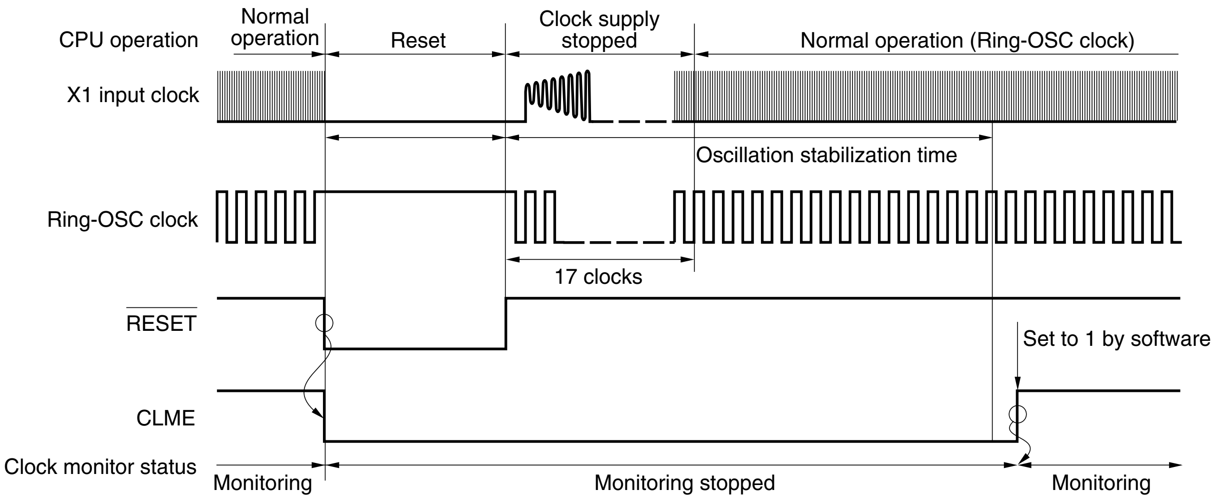
(4) Clock monitor status after $\overline{\text{RESET}}$ input (CLME = 1 is set after $\overline{\text{RESET}}$ input and during X1 input clock oscillation stabilization time)



$\overline{\text{RESET}}$ 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 of the X1 input clock, monitoring is not performed until the oscillation stabilization time of the X1 input clock ends. Monitoring is automatically started at the end of the oscillation stabilization time.

Figure 23-3. Timing of Clock Monitor (3/3)

(5) Clock monitor status after $\overline{\text{RESET}}$ input (CLME = 1 is set after $\overline{\text{RESET}}$ input and at the end of X1 input 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 of the X1 input clock, monitoring is started.

CHAPTER 24 POWER-ON-CLEAR CIRCUIT

24.1 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}), and generates internal reset signal when $V_{DD} < V_{POC}$.
- The following can be selected by a mask option.
  - POC disabled
  - POC used (detection voltage: V_{POC} = 2.85 V ±0.15 V)
  - POC used (detection voltage: V_{POC} = 3.5 V ±0.2 V)

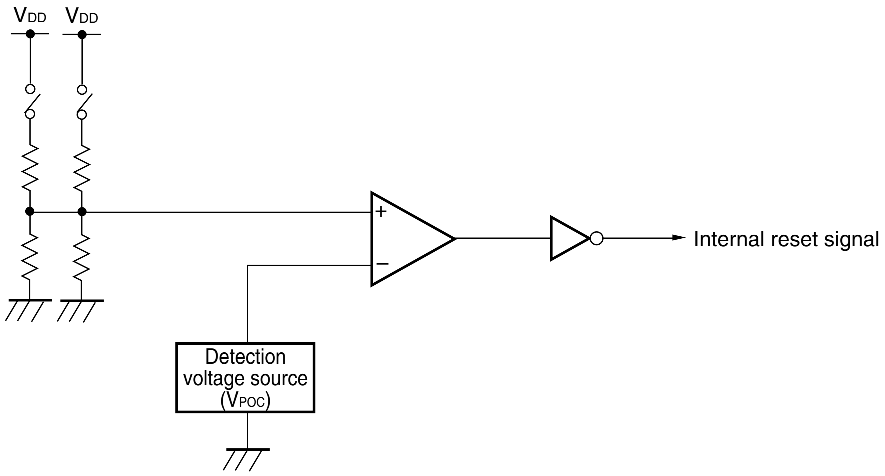
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset cause is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), low-voltage-detection (LVI) circuit, or clock monitor. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT, LVI, or the clock monitor.
For details of the RESF, refer to **CHAPTER 22 RESET FUNCTION**.

24.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 24-1.

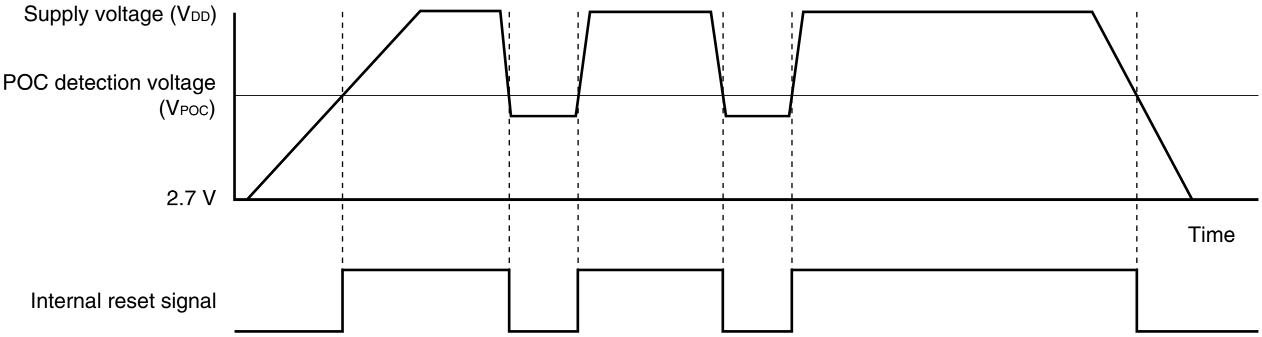
Figure 24-1. Block Diagram of Power-on-Clear Circuit



24.3 Operation of Power-on-Clear Circuit

In the power-on-clear circuit, the supply voltage (V_{DD}) and detection voltage (V_{POC}) are compared, and when V_{DD} < V_{POC}, an internal reset signal is generated.

Figure 24-2. Timing of Internal Reset Signal Generation in Power-on-Clear Circuit



★ 24.4 Cautions for Power-on-Clear Circuit

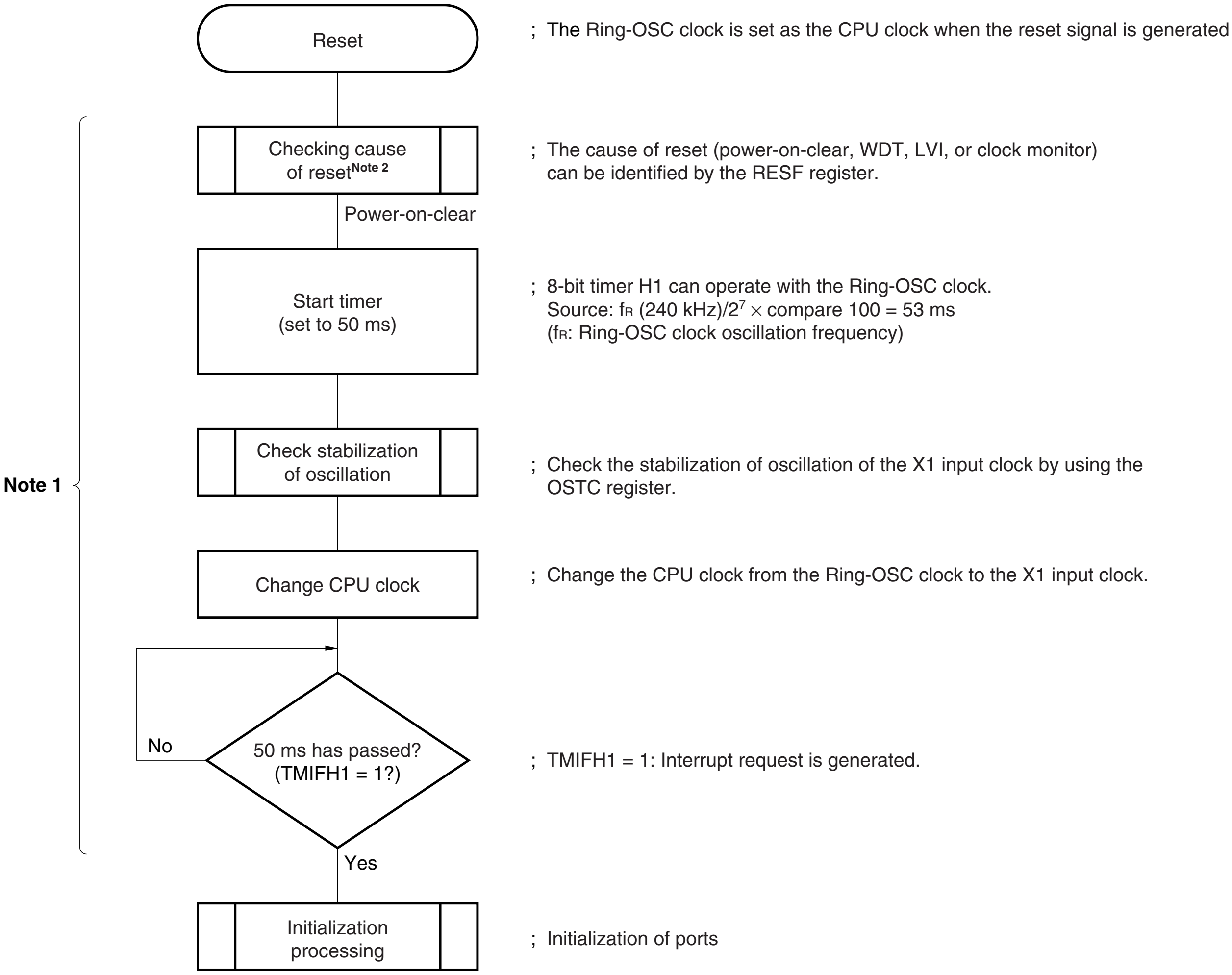
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 24-3. Example of Software Processing After Release of Reset (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

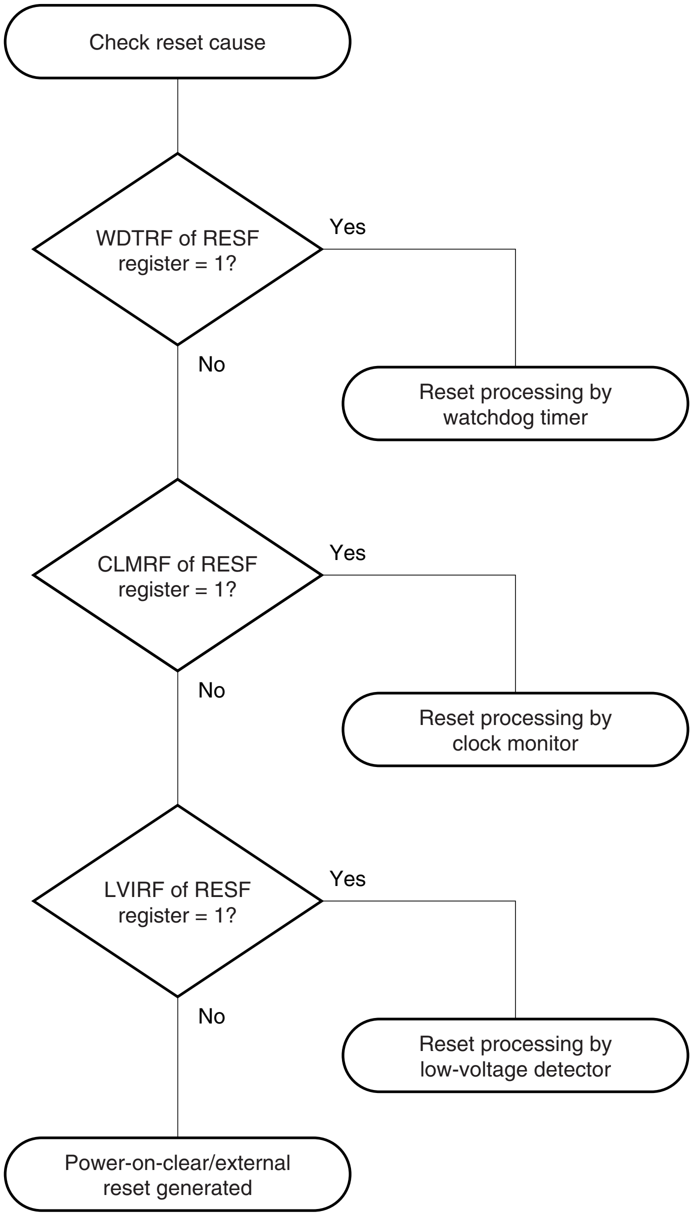

Reset
; The Ring-OSC clock is set as the CPU clock when the reset signal is generated
Checking cause of reset[Note 2]
; The cause of reset (power-on-clear, WDT, LVI, or clock monitor) can be identified by the RESF register.
Power-on-clear
Start timer (set to 50 ms)
; 8-bit timer H1 can operate with the Ring-OSC clock.
Source: f_R (240 kHz)/2^7 × compare 100 = 53 ms
(f_R: Ring-OSC clock oscillation frequency)
Check stabilization of oscillation
; Check the stabilization of oscillation of the X1 input clock by using the OSTC register.
Note 1
Change CPU clock
; Change the CPU clock from the Ring-OSC clock to the X1 input clock.
No
50 ms has passed? (TMIFH1 = 1?)
; TMIFH1 = 1: Interrupt request is generated.
Yes
Initialization processing
; Initialization of ports

Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 24-3. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause


Check reset cause
WDTRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
CLMRF of RESF register = 1?
Yes
No
Reset processing by clock monitor
LVIRF of RESF register = 1?
Yes
No
Reset processing by low-voltage detector
Power-on-clear/external reset generated

CHAPTER 25 LOW-VOLTAGE DETECTOR

25.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has following functions.

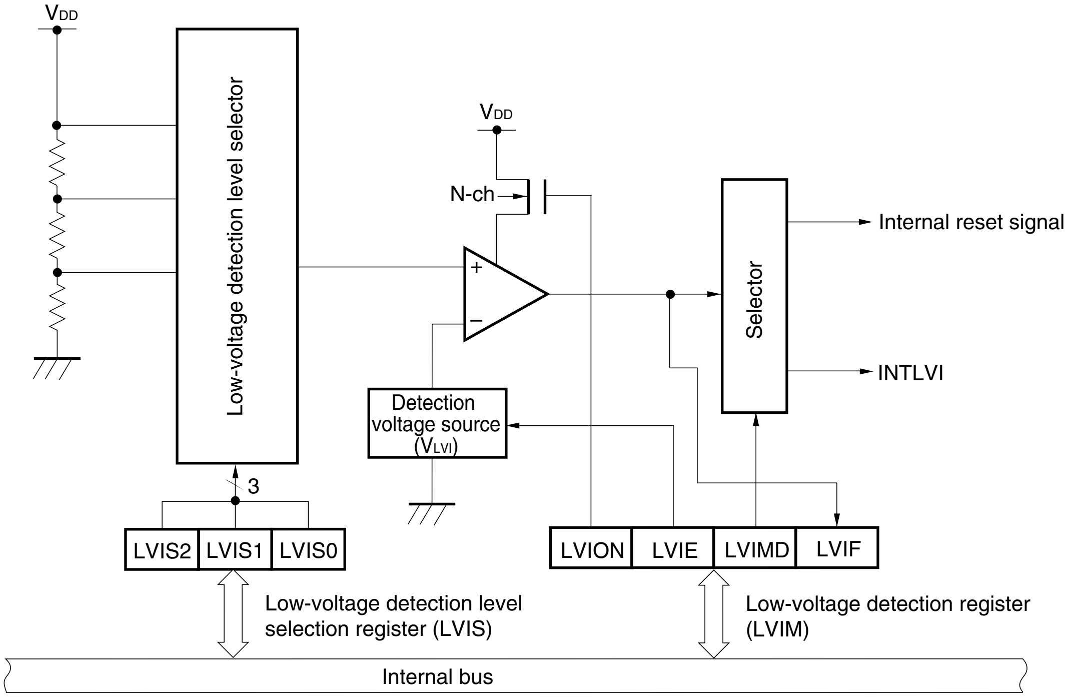
- Compares supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an internal interrupt signal or internal reset signal when $V_{DD} < V_{LVI}$.
- Detection levels (five levels) of supply voltage can be changed by software.
- Interrupt or reset function can be selected by software.
- 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 22 RESET FUNCTION**.

25.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown below.

Figure 25-1. Block Diagram of Low-Voltage Detector


VDD
Low-voltage detection level selector
VDD
N-ch
+
–
Detection voltage source (VLVI)
Selector
Internal reset signal
INTLVI
3
LVIS2
LVIS1
LVIS0
LVION
LVIE
LVIMD
LVIF
Low-voltage detection level selection register (LVIS)
Low-voltage detection register (LVIM)
Internal bus

25.3 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.

Figure 25-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFBEH After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	$\boxed{4}$	3	2	$\boxed{1}$	$\boxed{0}$
LVIM	LVION	0	0	LVIE	0	0	LVIMD	LVIF

LVION[Notes 1, 2]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVIE[Notes 1, 3, 4]	Specifies reference voltage generator
0	Disables operation
1	Enables operation

LVIMD[Note 1]	Low-voltage detection operation mode selection
0	Generates interrupt signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})
1	Generates internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})

LVIF[Note 5]	Low-voltage detection flag
0	Supply voltage (V_{DD}) > detection voltage (V_{LVI}), or when operation is disabled
1	Supply voltage (V_{DD}) < detection voltage (V_{LVI})

Notes 1. LVION, LVIE, and LVIMD are cleared to 0 at a reset other than an LVI reset. These are not cleared to 0 at an LVI reset.

2. 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.

3. When LVIE is set to 1, a reference voltage generator operation in the LVI circuit is started. Use software to instigate a wait of at least 2 ms from when LVIE is set to 1 until LVION is set to 1.

4. If “use POC” is selected by a mask option, leave LVIE as 0. A wait time (2 ms) until LVION is set to 1 is not necessary.

5. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

Caution To stop LVI, follow either of the procedures below.

- **When using 8-bit manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0 first and then clear LVIE 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.

★ **Figure 25-3. 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)[Note]
1	0	1	V_{LVI5} (3.3 V ±0.15 V)[Note]
1	1	0	V_{LVI6} (3.1 V ±0.15 V)[Note]
1	1	1	Setting prohibited

Note When the detection voltage of the POC circuit is specified as V_{POC} = 3.5 V ±0.2 V by a mask option, do not select V_{LVI4} to V_{LVI6} as the LVI detection voltage. Even if V_{LVI4} to V_{LVI6} are selected, POC circuit has priority.

25.4 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 4 (LVIE) of the low-voltage detection register (LVIM) to 1 (enables reference voltage generator operation).

<4> Use software to instigate a wait of at least 2 ms.

<5> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<6> Use software to instigate a wait of at least 0.2 ms.

<7> Confirm that "supply voltage (V_{DD}) > detection voltage (V_{LVI})" at bit 0 (LVIF) of LVIM.

<8> Set bit 1 (LVIMD) of LVIM to 1 (generates internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})).

Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <5>.

2. If "use POC" is selected by a mask option, procedures <3> and <4> are not required.

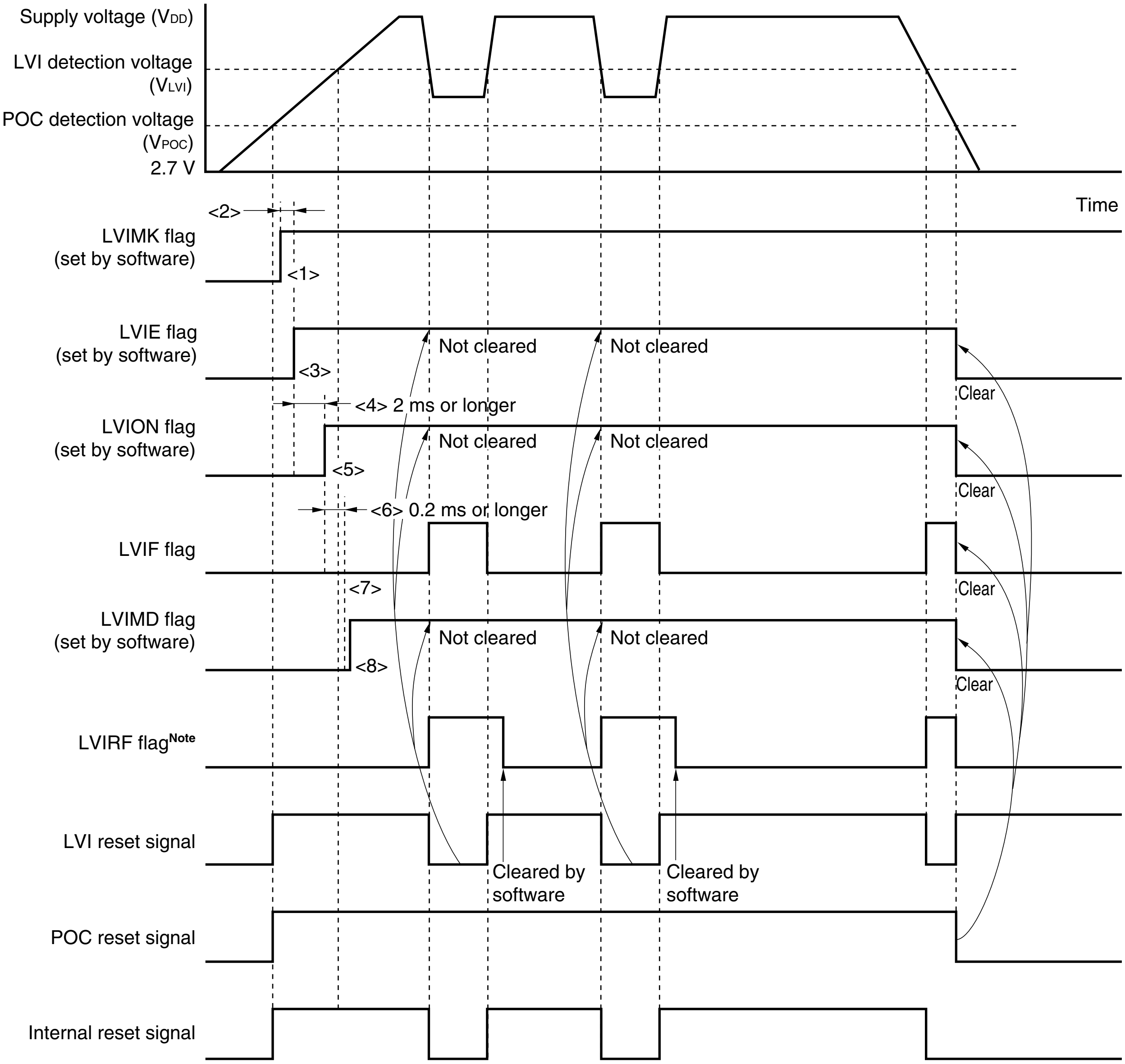
3. If supply voltage (V_{DD}) > detection voltage (V_{LVI}) when LVIM is set to 1, an internal reset signal is not generated.

- 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, LVION to 0, and LVIE to 0 in that order.

Figure 25-4. Timing of Low-Voltage Detector Internal Reset Signal Generation



Note LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, refer to **CHAPTER 22 RESET FUNCTION**.

Remark <1> to <8> in Figure 25-4 above correspond to <1> to <8> in the description of “when starting operation” in **25.4 (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 4 (LVIE) of the low-voltage detection register (LVIM) to 1 (enables reference voltage generator operation).

<4> Use software to instigate a wait of at least 2 ms.

<5> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<6> Use software to instigate a wait of at least 0.2 ms.

<7> Confirm that "supply voltage (V_{DD}) > detection voltage (V_{LVI})" at bit 0 (LVIF) of LVIM.

<8> Clear the interrupt request flag of LVI (LVIIF) to 0.

<9> Release the interrupt mask flag of LVI (LVIMK).

<10> Execute the EI instruction (when vector interrupts are used).

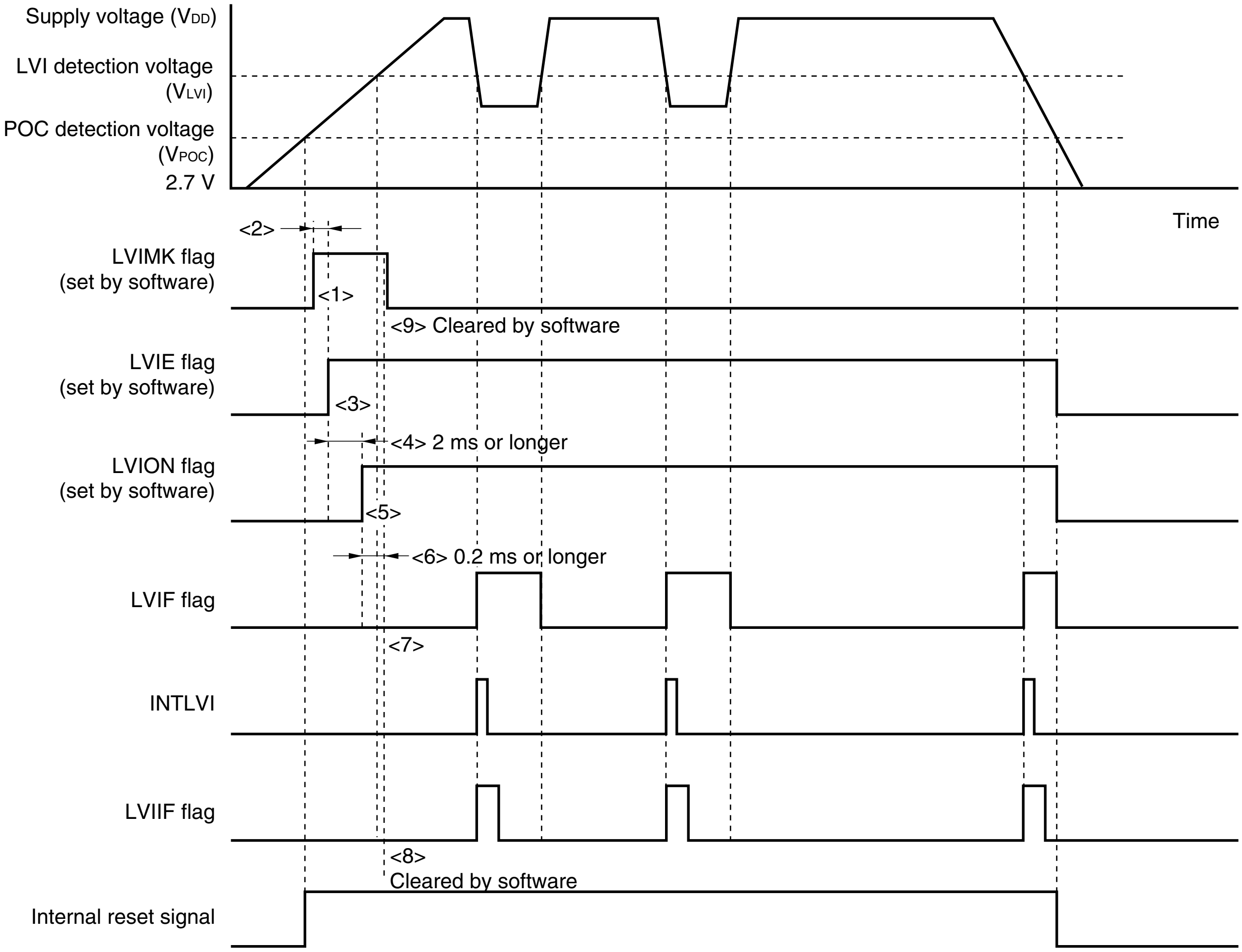
Caution If "use POC" is selected by a mask option, procedures <3> and <4> are not required.

- 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 first, and then clear LVIE to 0.

Figure 25-5. Timing of Low-Voltage Detector Interrupt Signal Generation


Supply voltage (V_{DD})
LVI detection voltage (V_{LVI})
POC detection voltage (V_{POC})
2.7 V
Time
<2>
LVIMK flag (set by software)
<1>
<9> Cleared by software
LVIE flag (set by software)
<3>
<4> 2 ms or longer
LVION flag (set by software)
<5>
<6> 0.2 ms or longer
LVIF flag
<7>
INTLVI
LVIIF flag
<8>
Cleared by software
Internal reset signal

Remark <1> to <9> in Figure 25-5 above correspond to <1> to <9> in the description of “when starting operation” in **25.4 (2) When used as interrupt**.

★ 25.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

(1) When used as reset

The system may be repeatedly reset and released from the reset status.

In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take action (2) below.

In this system, take the following actions.

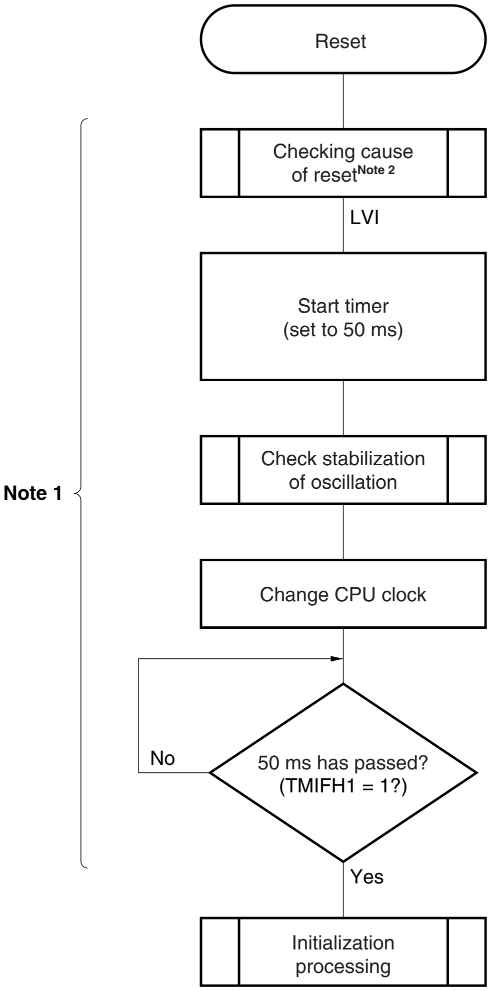
<Action>

(1) When used as reset

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 25-6. Example of Software Processing After Release of Reset (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage


Reset
Checking cause of reset[Note 2]
LVI
Start timer (set to 50 ms)
Check stabilization of oscillation
Change CPU clock
No
50 ms has passed? (TMIFH1 = 1?)
Yes
Initialization processing
Note 1

; The Ring-OSC clock is set as the CPU clock when the reset signal is generated

; The cause of reset (power-on-clear, WDT, LVI, or clock monitor) can be identified by the RESF register.

; 8-bit timer H1 can operate with the Ring-OSC clock.
Source: f_R (240 kHz)/2^7 × compare 100 = 53 ms
(f_R: Ring-OSC clock oscillation frequency)

; Check the stabilization of oscillation of the X1 input clock by using the OSTC register.

; Change the CPU clock from the Ring-OSC clock to the X1 input clock.

; TMIFH1 = 1: Interrupt request is generated.

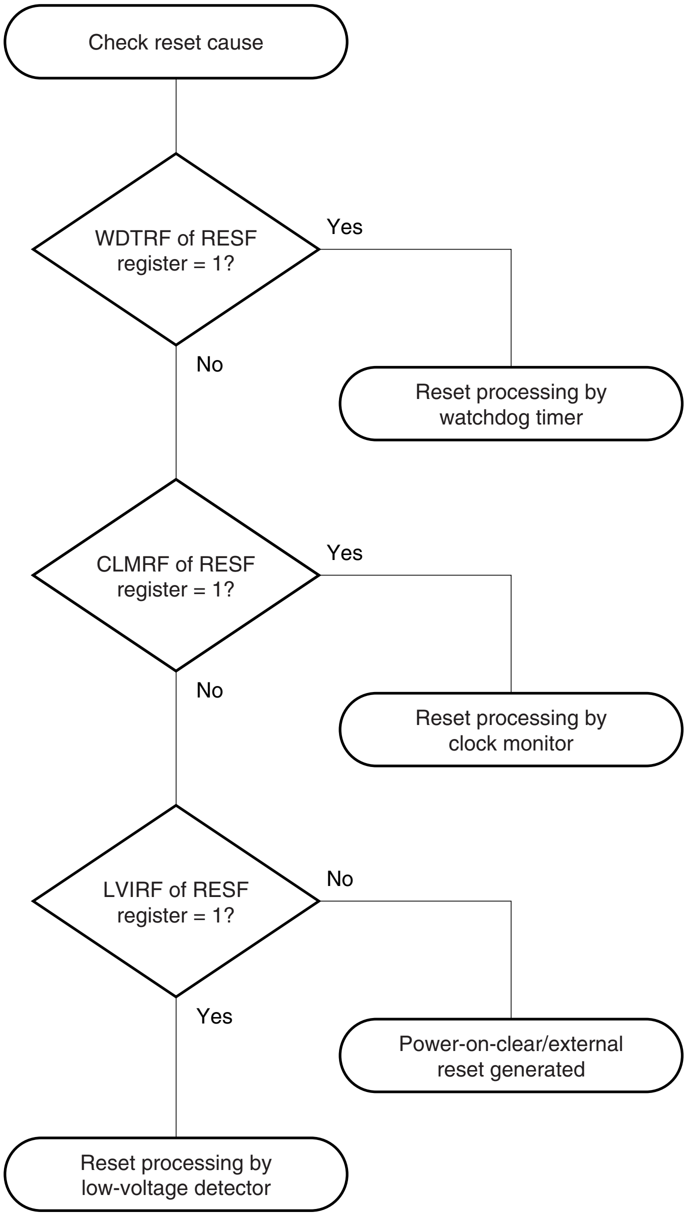
; Initialization of ports

Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 25-6. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause


Check reset cause
WDTRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
CLMRF of RESF register = 1?
Yes
No
Reset processing by clock monitor
LVIRF of RESF register = 1?
No
Yes
Power-on-clear/external reset generated
Reset processing by low-voltage detector

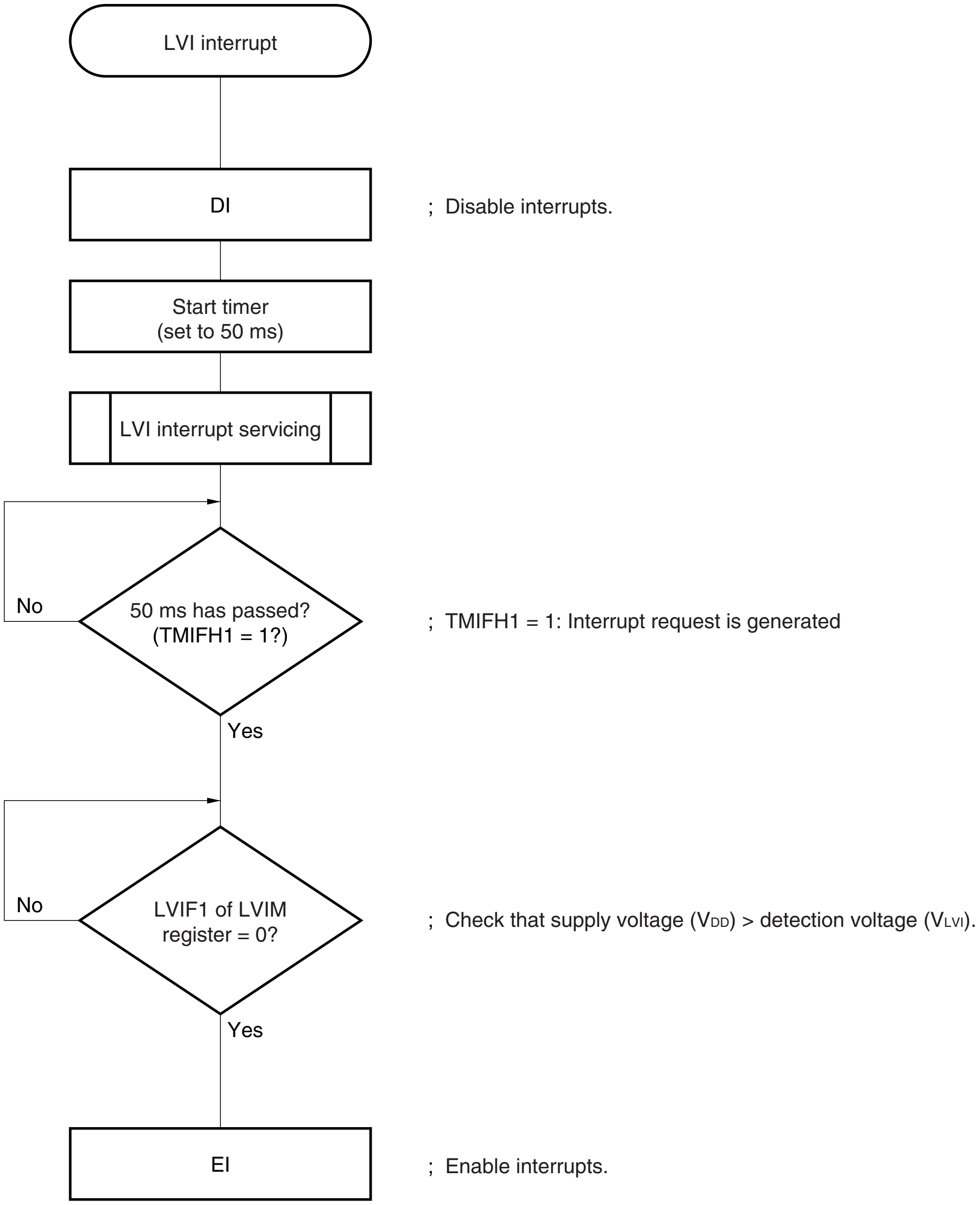
(2) When used as interrupt

Disable interrupts (DI) in the servicing routine of the LVI interrupt, and check to see if “supply voltage (V_{DD}) > detection voltage (V_{LVI})”, by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Then enable interrupts (EI).

In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, disable interrupts (DI), 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).

Figure 25-7. Example of Software Processing of LVI Interrupt

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



26.1 Outline of Regulator

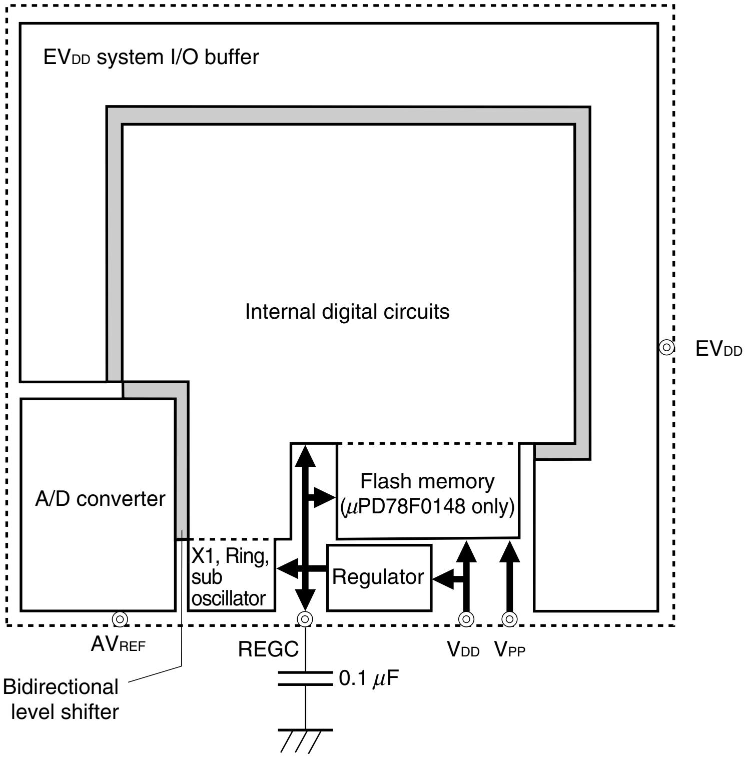
The 78K0/KF1 Series includes a circuit to realize low-voltage operation inside the device. To stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a 0.1 μF capacitor.

★ The regulator of the 78K0/KF1 Series stops operating in the following cases.

- During the reset period
- In STOP mode
- In HALT mode when the CPU is operating on the subsystem clock

Figure 26-1 shows the block diagram of the periphery of the regulator.

Figure 26-1. Block Diagram of Regulator Periphery



Remark To use the CPU at high speed (f_{XP} = 10 MHz, V_{DD} = 4.0 to 5.5 V), connect the REGC pin directly to V_{DD} and use at the same potential as the V_{DD} pin.

CHAPTER 27 MASK OPTIONS

Mask ROM versions are provided with the following mask options.

1. Power-on-clear (POC) circuit
   - POC cannot be used
   - POC used (detection voltage: V_{POC} = 2.85 V ±0.15 V)
   - POC used (detection voltage: V_{POC} = 3.5 V ±0.2 V)
2. Ring-OSC
   - Cannot be stopped
   - Can be stopped by software
3. Pull-up resistor of P60 to P63 pins
   - Pull-up resistor can be incorporated in 1-bit units

     (Pull-up resistors are not available for the flash memory versions.)

Flash memory versions that support the mask options of the mask ROM versions are as follows.

Table 27-1. Flash Memory Versions Supporting Mask Options of Mask ROM Versions

Mask Option		Flash Memory Version
POC Circuit	Ring-OSC	
POC cannot be used	Cannot be stopped	μPD78F0148M1
	Can be stopped by software	μPD78F0148M2
POC used (V_{POC} = 2.85 V ±0.15 V)	Cannot be stopped	μPD78F0148M3
	Can be stopped by software	μPD78F0148M4
POC used (V_{POC} = 3.5 V ±0.2 V)	Cannot be stopped	μPD78F0148M5
	Can be stopped by software	μPD78F0148M6

CHAPTER 28 μPD78F0148

The μPD78F0148 is provided as the flash memory version of the 78K0/KF1 Series.

The μPD78F0148 replaces the internal mask ROM of the μPD780148 with flash memory to which a program can be written, erased, and overwritten while mounted on the board. Table 28-1 lists the differences between the μPD78F0148 and the mask ROM versions.

Table 28-1. Differences Between μPD78F0148 and Mask ROM Versions

Item	μPD78F0148	Mask ROM Versions
Internal ROM configuration	Flash memory	Mask ROM
Internal ROM capacity	60 KB[Note]	μPD780143: 24 KB μPD780144: 32 KB μPD780146: 48 KB μPD780148: 60 KB
Internal expansion RAM capacity	1024 bytes[Note]	μPD780143: None μPD780144: None μPD780146: 1024 bytes μPD780148: 1024 bytes
IC pin	None	Available
V_{PP} pin	Available	None
Electrical specifications	Refer to **CHAPTER 30 ELECTRICAL SPECIFICATIONS (TARGET VALUES)**.	

Note The same capacity as the mask ROM versions can be specified by means of the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM versions.

28.1 Internal Memory Size Switching Register

The μPD78F0148 allows 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 Be sure to set the value of the relevant mask ROM version at initialization.

Figure 28-1. 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
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	1	1	0	24 KB
1	0	0	0	32 KB
1	1	0	0	48 KB
1	1	1	1	60 KB
Other than above				Setting prohibited

The IMS settings required to obtain the same memory map as mask ROM versions are shown in Table 28-2.

Table 28-2. Internal Memory Size Switching Register Settings

Target Mask ROM Versions	IMS Setting
μPD780143	C6H
μPD780144	C8H
μPD780146	CCH
μPD780148	CFH

Caution When using a mask ROM version, be sure to set the value indicated in Table 28-2 to IMS.

28.2 Internal Expansion RAM Size Switching Register

This register is used to set the internal expansion RAM capacity via software.

This register is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IXS to 0CH.

Caution Be sure to set the value of the relevant mask ROM version at initialization.

Figure 28-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
0	1	1	0	0	0 bytes
0	1	0	1	0	1024 bytes
Other than above					Setting prohibited

The IXS settings required to obtain the same memory map as mask ROM versions are shown in Table 28-3.

Table 28-3. Internal Expansion RAM Size Switching Register Settings

Target Mask ROM Versions	IXS Setting
μPD780143	0CH
μPD780144	0CH
μPD780146	0AH
μPD780148	0AH

Caution When using a mask ROM version, be sure to set the value indicated in Table 28-3 to IXS.

28.3 Flash Memory Programming

On-board writing of flash memory (with device mounted on target system) is supported.

On-board writing is performed after connecting a dedicated flash programmer (Flashpro III (FL-PR3, PG-FP3)) to the host machine and target system.

Moreover, writing to flash memory can also be performed using a flash memory writing adapter connected to Flashpro III.

Remark FL-PR3 is a product of Naito Densei Machida Mfg. Co., Ltd.

28.3.1 Selection of communication mode

Writing to flash memory is performed using Flashpro III and serial communication. Select the communication mode for writing from Table 28-4. For the selection of the communication mode, a format like the one shown in Figure 28-3 is used. The communication mode is selected according to the number of V_{PP} pulses shown in Table 28-4.

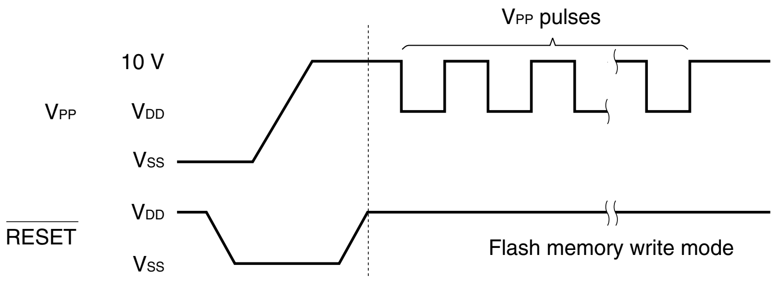
Table 28-4. Communication Mode List

Communication Mode	Number of Channels	Pin Used[Note]	Number of V_{PP} Pulses
3-wire serial I/O	1	$\overline{\text{SCK10}}$/TxD0/P10 SI10/RxD0/P11 SO10/P12	0
		$\overline{\text{SCK10}}$/TxD0/P10 SI10/RxD0/P11 SO10/P12 HS/P15/TOH0	3
UART (UART0)	1	TxD0/$\overline{\text{SCK10}}$/P10 RxD0/SI10/P11	8
		TxD0/$\overline{\text{SCK10}}$/P10 RxD0/SI10/P11 HS/P15/TOH0	11
UART (UART6)	1	TxD6/P13 RxD6/P14	9

Note After shifting to flash memory programming mode, all pins not used for flash memory programming are set to the same state as after reset. Therefore, since all ports become output high-impedance, pin processing, such as connecting to V_{DD} or V_{SS} via a resistor is required if the output high-impedance state is not acknowledged by external devices.

Caution Be sure to select the number of V_{PP} pulses shown in Table 28-4 for the communication mode.

Figure 28-3. Communication Mode Selection Format



28.3.2 Flash memory programming function

Flash memory writing is performed via command and data transmit/receive operations using the selected communication mode. The main functions are listed in Table 28-5.

Table 28-5. Main Functions of Flash Memory Programming

Function	Description
Reset	Used to detect write stop and transmission synchronization.
Batch verify	Compares entire memory contents and input data.
Batch erase	Erases the entire memory contents.
Batch blank check	Checks the erase status of the entire memory.
High-speed write	Performs writing to flash memory according to write start address and number of write data (bytes).
Continuous write	Performs successive write operations using the data input with high-speed write operation.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Delete time setting	Inputs the memory delete time.
Baud rate setting	Sets the communication rate when the UART method is used.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

28.3.3 Connecting Flashpro III

The connection between Flashpro III and the μPD78F0148 differs depending on the communication mode (3-wire serial I/O or UART). Figures 28-4 to 28-8 show the connection diagrams of each case.

Figure 28-4. Connection of Flashpro III in 3-Wire Serial I/O Mode

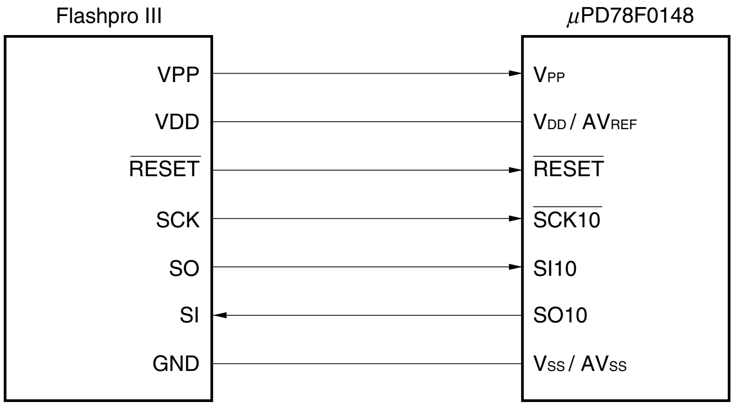


Figure 28-5. Connection of Flashpro III in 3-Wire Serial I/O Mode (Using Handshake)

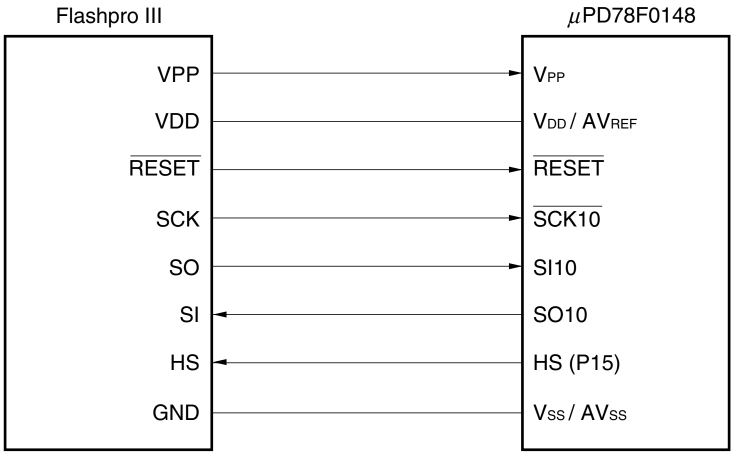


Figure 28-6. Connection of Flashpro III in UART (UART0) Mode

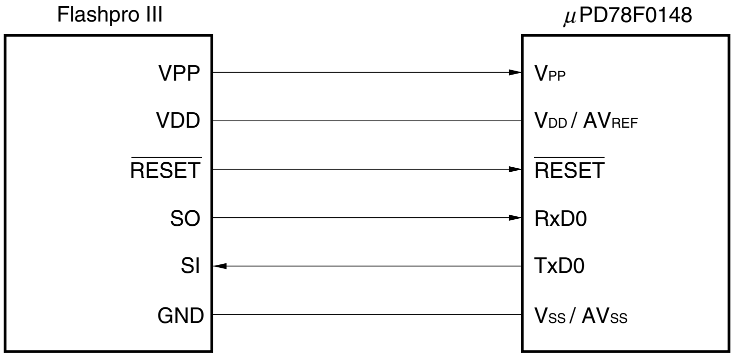


Figure 28-7. Connection of Flashpro III in UART (UART0) Mode (Using Handshake)

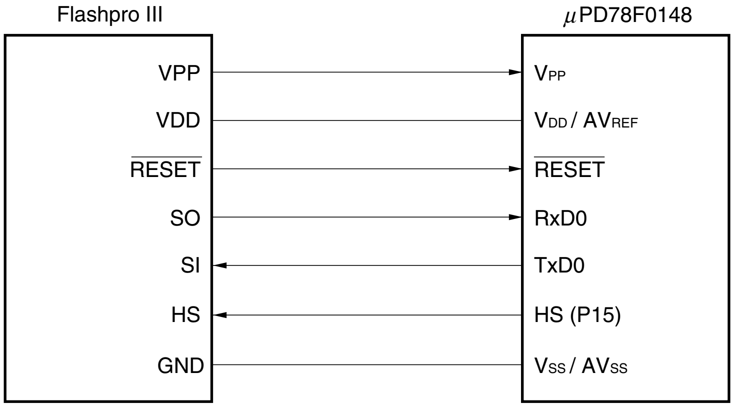
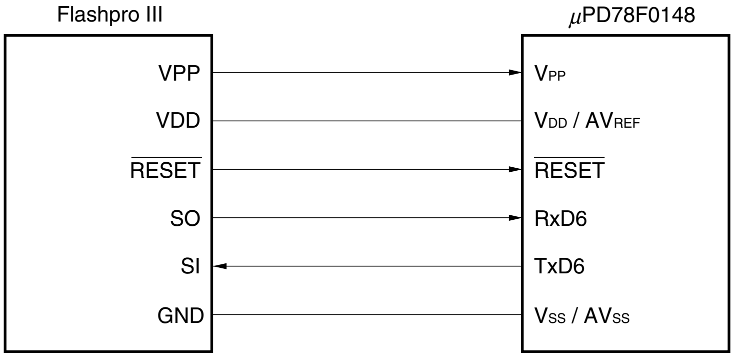


Figure 28-8. Connection of Flashpro III in UART (UART6) Mode



28.3.4 Connection on adapter for flash memory writing

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 28-9. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O Mode

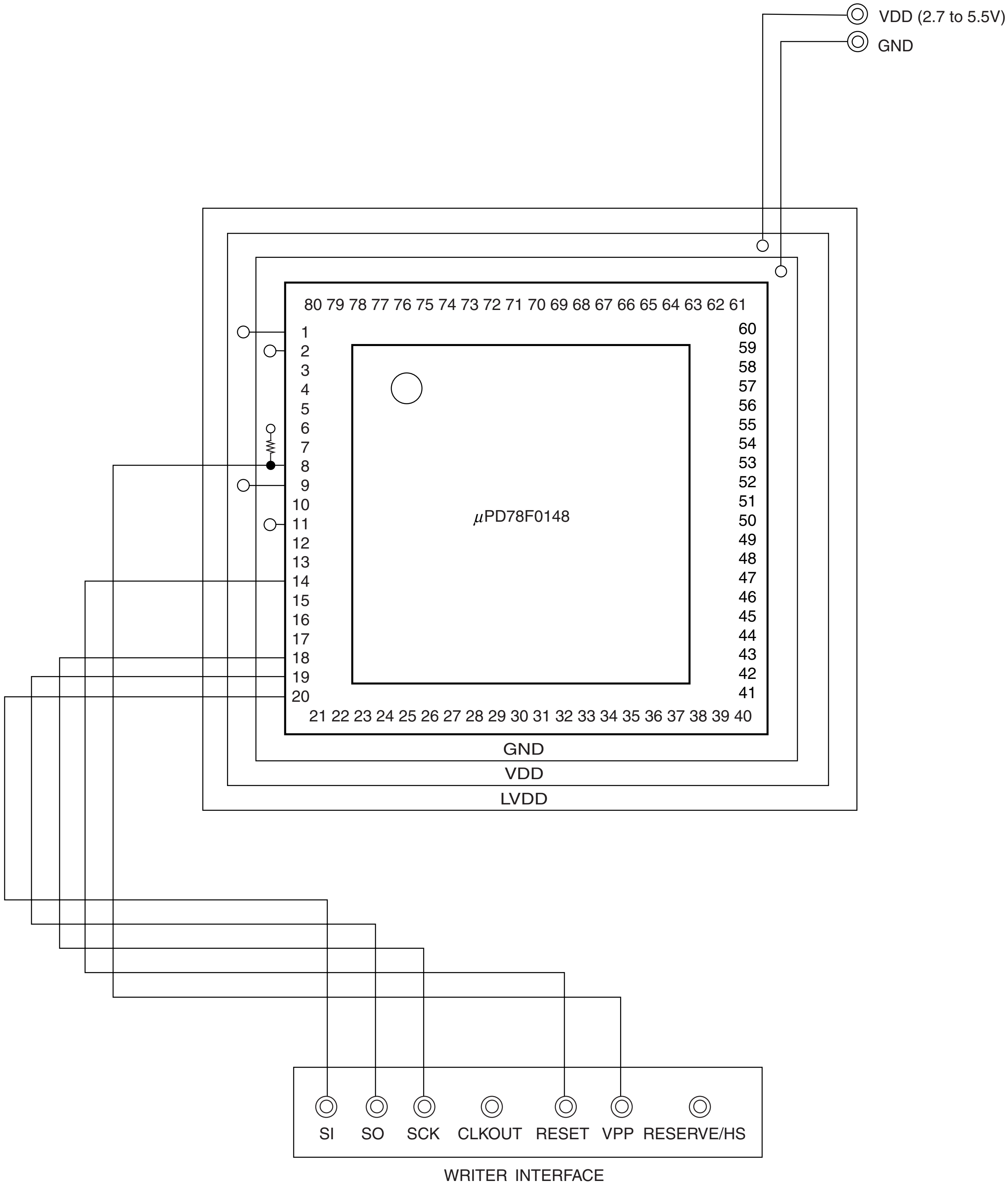


Figure 28-10. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O Mode (Using Handshake)

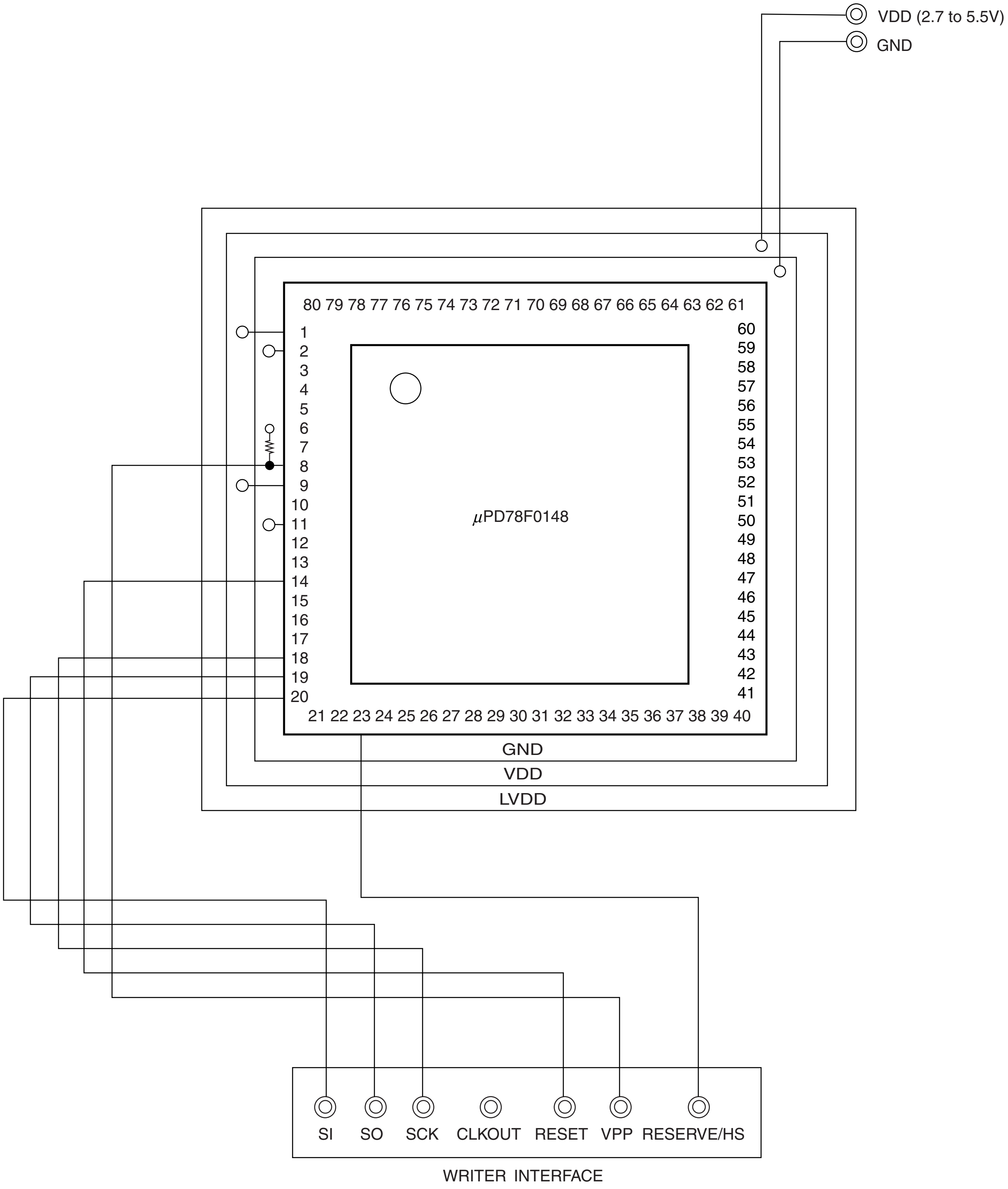


Figure 28-11. Example of Wiring Adapter for Flash Memory Writing in UART (UART0) Mode

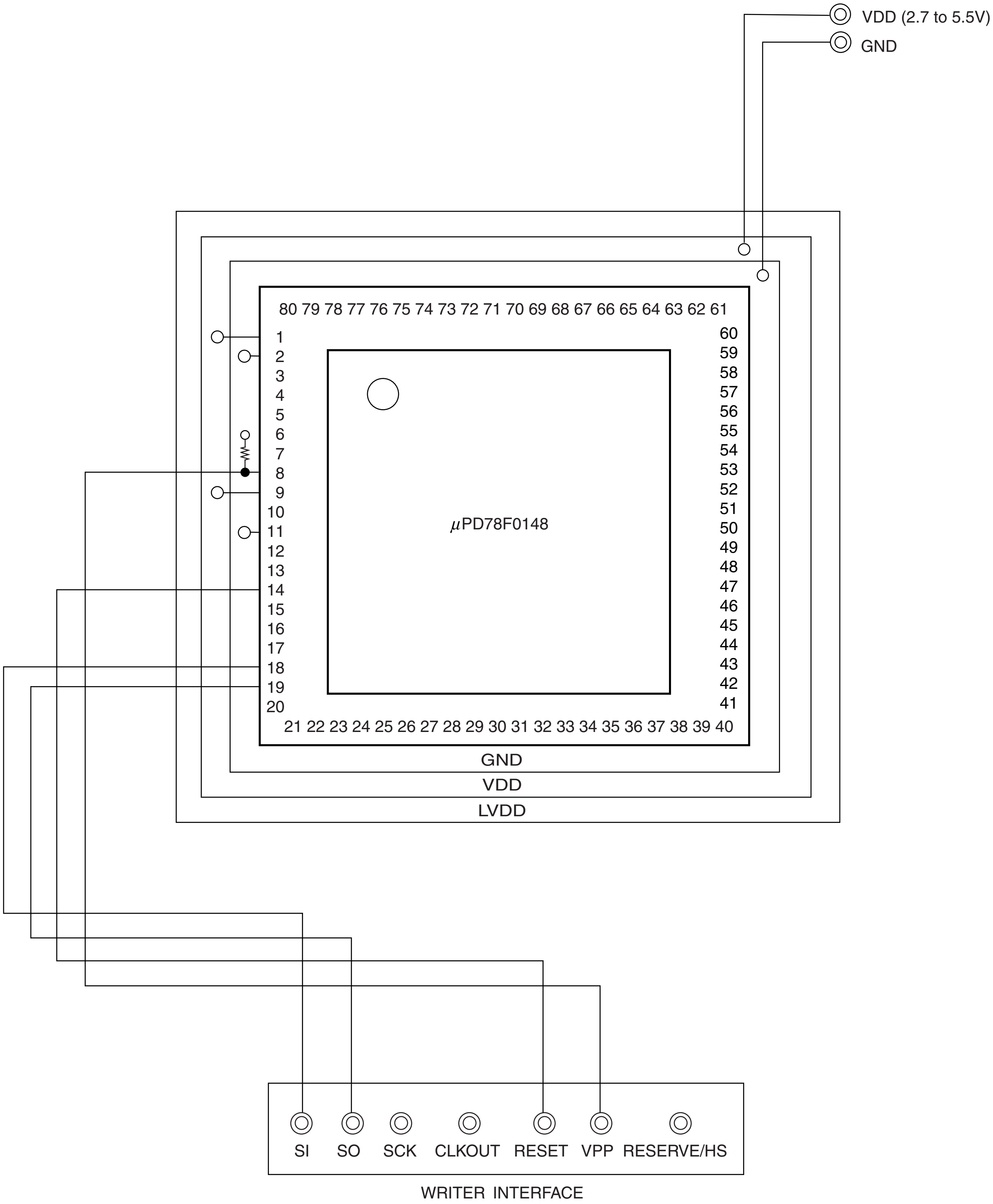


Figure 28-12. Example of Wiring Adapter for Flash Memory Writing in UART (UART0) Mode (Using Handshake)

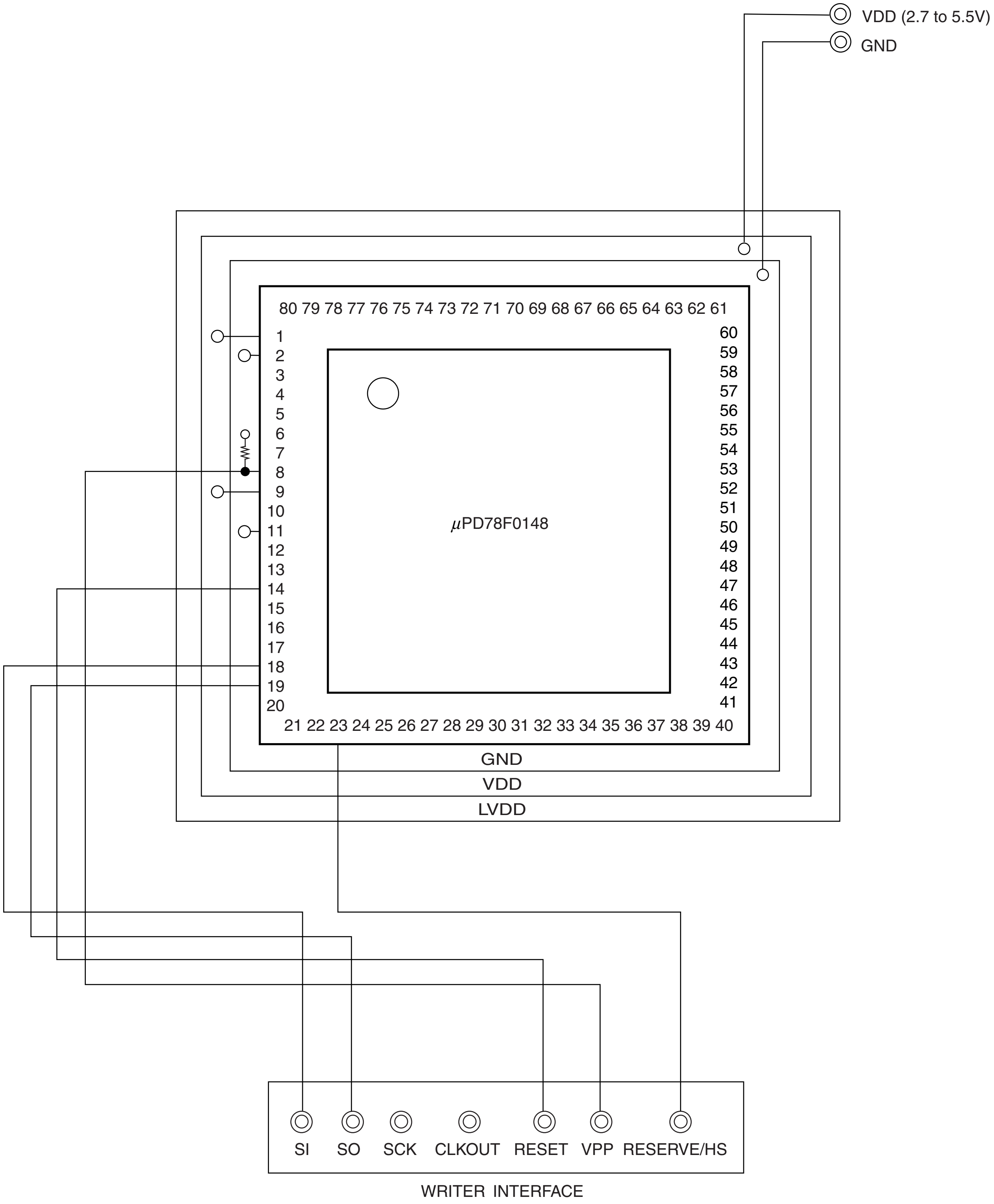
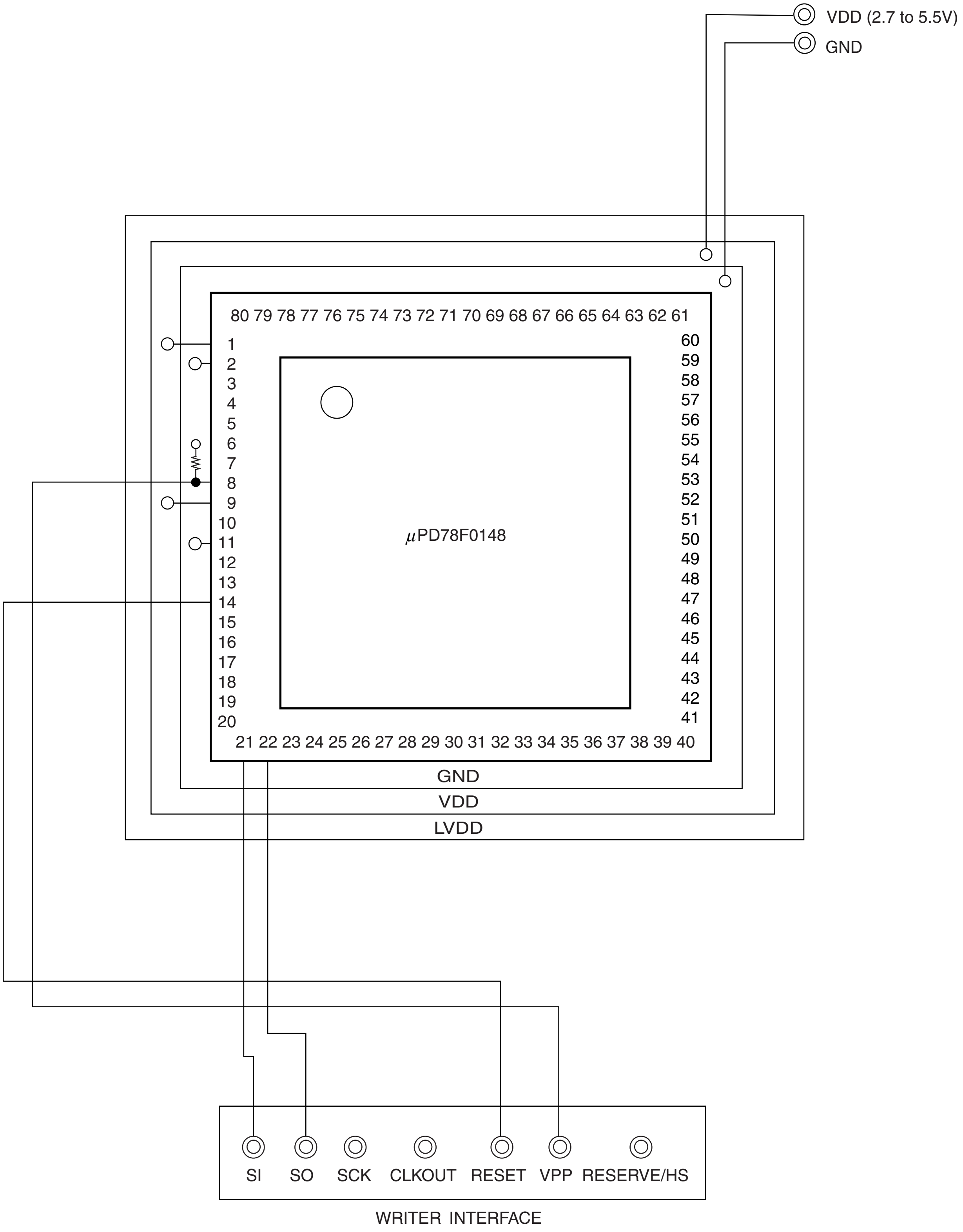


Figure 28-13. Example of Wiring Adapter for Flash Memory Writing in UART (UART6) Mode



CHAPTER 29 INSTRUCTION SET

This chapter lists each instruction set of the 78K0/KF1 Series in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

29.1 Conventions Used in Operation List

29.1.1 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 29-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol[Note] Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007FH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-5 Special Function Register List**.

29.1.2 Description of operation column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
RBS:	Register bank select flag
IE:	Interrupt request enable flag
NMIS:	Non-maskable interrupt servicing flag
( ):	Memory contents indicated by address or register contents in parentheses
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
∧:	Logical product (AND)
∨:	Logical sum (OR)
⊻:	Exclusive logical sum (exclusive OR)
‾:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

29.1.3 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

29.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte			
		saddr, #byte	3	6	7	(saddr) ← byte			
		sfr, #byte	3	–	7	sfr ← byte			
		A, r Note 3	1	2	–	A ← r			
		r, A Note 3	1	2	–	r ← A			
		A, saddr	2	4	5	A ← (saddr)			
		saddr, A	2	4	5	(saddr) ← A			
		A, sfr	2	–	5	A ← sfr			
		sfr, A	2	–	5	sfr ← A			
		A, !addr16	3	8	9 + n	A ← (addr16)			
		!addr16, A	3	8	9 + m	(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 + n	A ← (DE)			
		[DE], A	1	4	5 + m	(DE) ← A			
		A, [HL]	1	4	5 + n	A ← (HL)			
		[HL], A	1	4	5 + m	(HL) ← A			
		A, [HL + byte]	2	8	9 + n	A ← (HL + byte)			
		[HL + byte], A	2	8	9 + m	(HL + byte) ← A			
		A, [HL + B]	1	6	7 + n	A ← (HL + B)			
		[HL + B], A	1	6	7 + m	(HL + B) ← A			
		A, [HL + C]	1	6	7 + n	A ← (HL + C)			
		[HL + C], A	1	6	7 + m	(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 + n + m	A ↔ (addr16)			
		A, [DE]	1	4	6 + n + m	A ↔ (DE)			
		A, [HL]	1	4	6 + n + m	A ↔ (HL)			
		A, [HL + byte]	2	8	10 + n + m	A ↔ (HL + byte)			
		A, [HL + B]	2	8	10 + n + m	A ↔ (HL + B)			
		A, [HL + C]	2	8	10 + n + m	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.

3. n is the number of waits when the external memory expansion area is read.

4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp[Note 3]	1	4	–	AX ← rp			
		rp, AX[Note 3]	1	4	–	rp ← AX			
		AX, !addr16	3	10	12 + 2n	AX ← (addr16)			
		!addr16, AX	3	10	12 + 2m	(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 + n	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9 + n	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.

3. n is the number of waits when the external memory expansion area is read.

4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5 + n	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9 + n	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9 + n	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9 + n	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.

3. n is the number of waits when the external memory expansion area is read.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	4	–	A ← A ∨ r	×		
		r, A	2	4	–	r ← r ∨ A	×		
		A, saddr	2	4	5	A ← A ∨ (saddr)	×		
		A, !addr16	3	8	9 + n	A ← A ∨ (addr16)	×		
		A, [HL]	1	4	5 + n	A ← A ∨ (HL)	×		
		A, [HL + byte]	2	8	9 + n	A ← A ∨ (HL + byte)	×		
		A, [HL + B]	2	8	9 + n	A ← A ∨ (HL + B)	×		
		A, [HL + C]	2	8	9 + n	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 + n	A ← A ⊻ (addr16)	×		
		A, [HL]	1	4	5 + n	A ← A ⊻ (HL)	×		
		A, [HL + byte]	2	8	9 + n	A ← A ⊻ (HL + byte)	×		
		A, [HL + B]	2	8	9 + n	A ← A ⊻ (HL + B)	×		
		A, [HL + C]	2	8	9 + n	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 + n	A – (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A – (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	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.

3. n is the number of waits when the external memory expansion area is read.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7 + n	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 + n + m	(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.

3. n is the number of waits when the external memory expansion area is read.

4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7 + n	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 + n	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 + n	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 + n + m	(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 + n + m	(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.

3. n is the number of waits when the external memory expansion area is read.

4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← (PC + 3)$_H$, (SP – 2) ← (PC + 3)$_L$, PC ← addr16, SP ← SP – 2			
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)$_H$, (SP – 2) ← (PC + 2)$_L$, PC$_{15-11}$ ← 00001, PC$_{10-0}$ ← addr11, SP ← SP – 2			
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)$_H$, (SP – 2) ← (PC + 1)$_L$, PC$_H$ ← (00000000, addr5 + 1), PC$_L$ ← (00000000, addr5), SP ← SP – 2			
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)$_H$, (SP – 3) ← (PC + 1)$_L$, PC$_H$ ← (003FH), PC$_L$ ← (003EH), SP ← SP – 3, IE ← 0			
	RET		1	6	–	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), SP ← SP + 2			
	RETI		1	6	–	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
	RETB		1	6	–	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1			
		rp	1	4	–	(SP – 1) ← rp$_H$, (SP – 2) ← rp$_L$, SP ← SP – 2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R	R	R
		rp	1	4	–	rp$_H$ ← (SP + 1), rp$_L$ ← (SP), SP ← SP + 2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	PC$_H$ ← A, PC$_L$ ← X			
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11 + n	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 + n	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 + n + m	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.

3. n is the number of waits when the external memory expansion area is read.

4. m is the number of waits when the external memory expansion area is written.

29.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand First Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand First Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

CHAPTER 30 ELECTRICAL SPECIFICATIONS (TARGET VALUES)

These specifications are only target values, and may not be satisfied by mass-produced products.

★ **Absolute Maximum Ratings (T_A = 25°C) (1/2)**

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	EV_{DD}			–0.3 to +6.5	V
	REGC			–0.3 to +6.5	V
	V_{SS}			–0.3 to +0.3	V
	EV_{SS}			–0.3 to +0.3	V
	AV_{REF}			–0.3 to V_{DD} + 0.3[Note 1]	V
	AV_{SS}			–0.3 to +0.3	V
	V_{PP}	μPD78F0148 only **Note 2**		–0.3 to +10.5	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P20 to P27, P30 to P33, P40 to P47, P50 to P57, P60, P61, P64 to P67, P70 to P77, P120, P140 to P145, X1, X2, XT1, XT2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3[Note 1]	V
	V_{I2}	P62, P63	N-ch open drain	–0.3 to +13	V
			On-chip pull-up resistor	–0.3 to V_{DD} + 0.3[Note 1]	V
	V_{I3}	V_{PP} in flash programming mode (μPD78F0148 only)		–0.3 to +10.5	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3[Note 1]	V
Analog input voltage	V_{AN}			AV_{SS} – 0.3 to AV_{REF} + 0.3[Note 1] and –0.3 to V_{DD} + 0.3[Note 1]	V
Output current, high	I_{OH}	Per pin		–10	mA
		Total of all pins –60 mA	P00 to P06, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P142 to P145	–30	mA
			P10 to P17, P30 to P33, P120, P130, P140, P141	–30	mA

Note 1. Must be 6.5 V or lower.
(Refer to **Note 2** on the next page.)

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions		Ratings	Unit
Output current, low	I_{OL}	Per pin	P00 to P06, P10 to P17, P30 to P33, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P130, P140 to P145	20	mA
			P60 to P63	30	mA
		Total of all pins 70 mA	P00 to P06, P40 to P47, P50 to P57, P60, P61, P64 to P67, P70 to P77, P142 to P145	35	mA
			P10 to P17, P30 to P33, P62, P63, P120, P130, P140, P141	35	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
Storage temperature	T_{stg}	μPD780143, 780144, 780146, 780148		–65 to +150	°C
		μPD78F0148		–40 to +125	

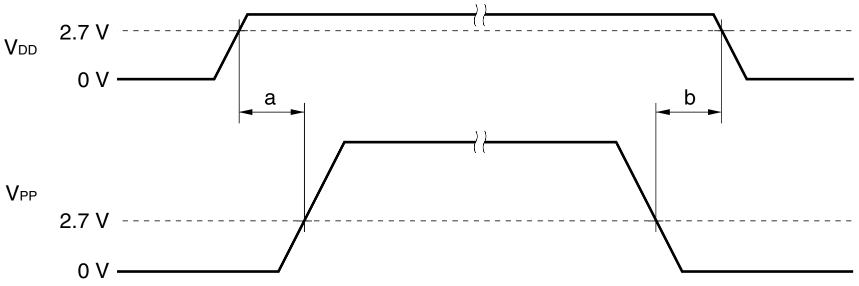
Note 2. Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- **When supply voltage rises**

  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (2.7 V) of the operating voltage range (15 μs if the supply voltage is dropped by the regulator) (see a in the figure below).

- **When supply voltage drops**

  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (2.7 V) of the operating voltage range of V_{DD} (see b in the figure below).

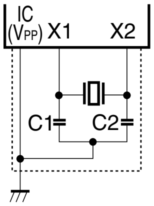
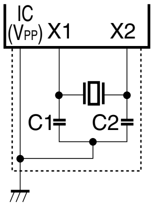
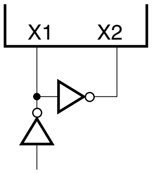


Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure 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.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_{XP})[Note 1]	When a capacitor is connected to the REGC pin[Note 2]	3.3 V ≤ V_{DD} < 5.5 V	2.0		8.38	MHz
				2.7 V ≤ V_{DD} < 3.3 V	2.0		5.0	
			When the REGC pin is directly connected to V_{DD}	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	
Crystal resonator		Oscillation frequency (f_{XP})[Note 1]	When a capacitor is connected to the REGC pin[Note 2]	3.3 V ≤ V_{DD} < 5.5 V	2.0		8.38	MHz
				2.7 V ≤ V_{DD} < 3.3 V	2.0		5.0	
			When the REGC pin is directly connected to V_{DD}	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[Note 3]		X1 input frequency (f_{XP})[Note 1]	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_{XPH}, t_{XPL})	4.0 V ≤ V_{DD} ≤ 5.5 V		46		500	ns
			3.3 V ≤ V_{DD} < 4.0 V		56		500	
			2.7 V ≤ V_{DD} < 3.3 V		96		500	

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. When the REGC pin is connected to V_{SS} via a 0.1 μF capacitor.

3. Connect the REGC pin directly to V_{DD}.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the Ring-OSC after reset is released, check the oscillation stabilization time of the X1 input clock using the oscillation stabilization time status register (OSTC). Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

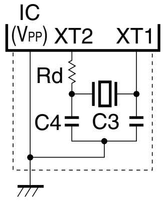
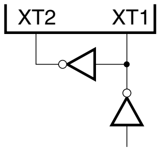
Ring-OSC Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
On-chip Ring-OSC oscillator	Oscillation frequency (f_R)		120	240	480	kHz

Subsystem Clock Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})[Note]		32	32.768	35	kHz
External clock		XT1 input frequency (f_{XT})[Note]		32		38.5	kHz
★		XT1 input high-/low-level width (t_{XTH}, t_{XTL})		12		15	μs

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (1/4)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
	Output current, high	I_{OH}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			–5	mA
★			Total of P10 to P17, P30 to P33, P120, P130, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
★			Total of P00 to P06, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P142 to P145	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
			All pins	2.7 V ≤ V_{DD} < 4.0 V			–10	mA
	Output current, low	I_{OL}	Per pin for P00 to P06, P10 to P17, P30 to P33, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P130, P140 to P145	4.0 V ≤ V_{DD} ≤ 5.5 V			10	mA
			Per pin for P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V			15	mA
			Total of P10 to P17, P30 to P33, P62, P63, P120, P130, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
			Total of P00 to P06, P40 to P47, P50 to P57, P60, P61, P64 to P67, P70 to P77, P142 to P145	4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
			All pins	2.7 V ≤ V_{DD} < 4.0 V			10	mA
	Input voltage, high	V_{IH1}	P12, P13, P15, P40 to P47, P50 to P57, P64 to P67, P144, P145		$0.7V_{DD}$		V_{DD}	V
		V_{IH2}	P00 to P06, P10, P11, P14, P16, P17, P30 to P33, P70 to P77, P120, P140 to P143, $\overline{\text{RESET}}$		$0.8V_{DD}$		V_{DD}	V
★		V_{IH3}	P20 to P27[Note]		$0.7AV_{REF}$		AV_{REF}	V
		V_{IH4}	P60, P61		$0.7V_{DD}$		V_{DD}	V
		V_{IH5}	P62, P63		$0.7V_{DD}$		12	V
		V_{IH6}	X1, X2, XT1, XT2		V_{DD} – 0.5		V_{DD}	V
	Input voltage, low	V_{IL1}	P12, P13, P15, P40 to P47, P50 to P57, P64 to P67, P144, P145		0		$0.3V_{DD}$	V
		V_{IL2}	P00 to P06, P10, P11, P14, P16, P17, P30 to P33, P70 to P77, P120, P140 to P143, $\overline{\text{RESET}}$		0		$0.2V_{DD}$	V
★		V_{IL3}	P20 to P27[Note]		0		$0.3AV_{REF}$	V
		V_{IL4}	P60, P61		0		$0.3V_{DD}$	V
		V_{IL5}	P62, P63		0		$0.3V_{DD}$	V
		V_{IL6}	X1, X2, XT1, XT2		0		0.4	V

Note When used as A/D converter analog input pins, set AV_{REF} = V_{DD}.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/4)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
★	Output voltage, high	V_{OH}	Total of P10 to P17, P30 to P33, P120, P130, P140, P141 I_{OH} = –25 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –5 mA	V_{DD} – 1.0			V
★			Total of P00 to P06, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P142 to P145 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_{OL1}	Total of P10 to P17, P30 to P33, P62, P63, P120, P130, P140, P141 I_{OL} = 30 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 10 mA			1.3	V
			Total of P00 to P06, P40 to P47, P50 to P57, P60, P61, P64 to P67, P70 to P77, P142 to P145 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
		V_{OL2}	P60 to P63	I_{OL} = 15 mA			2.0	V
★	Input leakage current, high	I_{LIH1}	V_I = V_{DD}	P00 to P06, P10 to P17, P30 to P33, P40 to P47, P50 to P57, P60, P61, P64 to P67, P70 to P77, P120, P140 to P145, $\overline{\text{RESET}}$			3	μA
★			V_I = AV_{REF}	P20 to P27			3	μA
		I_{LIH2}	V_I = V_{DD}	X1, X2, XT1, XT2			20	μA
★		I_{LIH3}	V_I = 12 V	P62, P63 (N-ch open drain)			3	μA
★	Input leakage current, low	I_{LIL1}	V_I = 0 V	P00 to P06, P10 to P17, P20 to P27, P30 to P33, P40 to P47, P50 to P57, P60, P61, P64 to P67, P70 to P77, P120, P140 to P145, $\overline{\text{RESET}}$			–3	μA
★		I_{LIL2}		X1, X2, XT1, XT2			–20	μA
★		I_{LIL3}		P62, P63 (N-ch open drain)			-3^{Note}	μA
	Output leakage current, high	I_{LOH}	V_O = V_{DD}				3	μA
	Output leakage current, low	I_{LOL}	V_O = 0 V				–3	μA
	Pull-up resistance value	R_L	V_I = 0 V		10	30	100	kΩ
	V_{PP} supply voltage (μPD78F0148 only)	V_{PP1}	In normal operation mode		0		0.2V_{DD}	V

Note If there is no on-chip pull-up resistor for P62 and P63 (specified by a mask option) and if port 6 has been set to input mode when a read instruction is executed to read from port 6, a low-level input leakage current of up to –45 μA flows during only one cycle. At all other times, the maximum leakage current is –3 μA.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

★

DC Characteristics (3/4): μPD78F0148

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	X1 crystal oscillation operating mode[Note 2]	f_{XP} = 10 MHz V_{DD} = 5.0 V ±10%[Notes 3, 7]	When A/D converter is stopped		13.5	29.7	mA
				When A/D converter is operating[Note 9]		14.5	31.9	mA
			f_{XP} = 8.38 MHz V_{DD} = 5.0 V ±10%[Notes 3, 8]	When A/D converter is stopped		9.5	19	mA
				When A/D converter is operating[Note 9]		10.5	21	mA
			f_{XP} = 5 MHz V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		5	10	mA
				When A/D converter is operating[Note 9]		7	14	mA
	I_{DD2}	X1 crystal oscillation HALT mode	f_{XP} = 10 MHz V_{DD} = 5.0 V ±10%[Notes 3, 7]	When peripheral functions are stopped		1.7	3.4	mA
				When peripheral functions are operating			9.6	mA
			f_{XP} = 8.38 MHz V_{DD} = 5.0 V ±10%[Notes 3, 8]	When peripheral functions are stopped		1	2	mA
				When peripheral functions are operating			4.5	mA
			f_{XP} = 5 MHz V_{DD} = 3.0 V ±10%[Note 3]	When peripheral functions are stopped		0.5	1	mA
				When peripheral functions are operating			2	mA
	I_{DD3}	Ring-OSC operating mode[Note 4]	V_{DD} = 5.0 V ±10%			0.7	2.1	mA
			V_{DD} = 3.0 V ±10%			0.4	1.2	mA
	I_{DD4}	32.768 kHz crystal oscillation operating mode[Notes 4, 6]	V_{DD} = 5.0 V ±10%			115	230	μA
			V_{DD} = 3.0 V ±10%			95	190	μA
	I_{DD5}	32.768 kHz crystal oscillation HALT mode[Notes 4, 6]	V_{DD} = 5.0 V ±10%			30	60	μA
			V_{DD} = 3.0 V ±10%			6	18	μA
	I_{DD6}	STOP mode	V_{DD} = 5.0 V ±10%	POC: OFF, RING: OFF		0.1	30	μA
				POC: OFF, RING: ON		14	58	μA
				POC: ON[Note 5], RING: OFF		3.5	35.5	μA
				POC: ON[Note 5], RING: ON		17.5	63.5	μA
			V_{DD} = 3.0 V ±10%	POC: OFF, RING: OFF		0.05	10	μA
				POC: OFF, RING: ON		7.5	25	μA
				POC: ON[Note 5], RING: OFF		3.5	15.5	μA
				POC: ON[Note 5], RING: ON		11	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. I_{DD1} includes peripheral operation current.

3. When PCC = 00H.

4. When main system clock is stopped.

5. Including when LVIE (bit 4 of LVIM) = 1 with POC-OFF selected by a mask option.

6. When POC-OFF (including LVIE = 0) is selected by a mask option and Ring-OSC oscillation is stopped.

7. When the REGC pin is directly connected to V_{DD}.

8. When the REGC pin is connected to V_{SS} via a 0.1 μF capacitor.

9. Including the current that flows through the AV_{REF} pin.

★ DC Characteristics (4/4): μPD780143, 780144, 780146, and 780148

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	X1 crystal oscillation operating mode[Note 2]	f_{XP} = 10 MHz V_{DD} = 5.0 V ±10%[Notes 3, 7]	When A/D converter is stopped		6	15	mA
				When A/D converter is operating[Note 9]		7	17.5	mA
			f_{XP} = 8.38 MHz V_{DD} = 5.0 V ±10%[Notes 3, 8]	When A/D converter is stopped		4	8	mA
				When A/D converter is operating[Note 9]		5	10	mA
			f_{XP} = 5 MHz V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		3	6	mA
				When A/D converter is operating[Note 9]		4	8	mA
	I_{DD2}	X1 crystal oscillation HALT mode	f_{XP} = 10 MHz V_{DD} = 5.0 V ±10%[Notes 3, 7]	When peripheral functions are stopped		1.7	3.4	mA
				When peripheral functions are operating			9.6	mA
			f_{XP} = 8.38 MHz V_{DD} = 5.0 V ±10%[Notes 3, 8]	When peripheral functions are stopped		1	2	mA
				When peripheral functions are operating			4.5	mA
			f_{XP} = 5 MHz V_{DD} = 3.0 V ±10%[Note 3]	When peripheral functions are stopped		0.5	1	mA
				When peripheral functions are operating			2	mA
	I_{DD3}	Ring-OSC operating mode[Note 4]	V_{DD} = 5.0 V ±10%			0.3	0.9	mA
			V_{DD} = 3.0 V ±10%			0.19	0.57	mA
	I_{DD4}	32.768 kHz crystal oscillation operating mode[Notes 4, 6]	V_{DD} = 5.0 V ±10%			45	90	μA
			V_{DD} = 3.0 V ±10%			25	50	μA
	I_{DD5}	32.768 kHz crystal oscillation HALT mode[Notes 4, 6]	V_{DD} = 5.0 V ±10%			30	60	μA
			V_{DD} = 3.0 V ±10%			6	18	μA
	I_{DD6}	STOP mode	V_{DD} = 5.0 V ±10%	POC: OFF, RING: OFF		0.1	30	μA
				POC: OFF, RING: ON		14	58	μA
				POC: ON[Note 5], RING: OFF		3.5	35.5	μA
				POC: ON[Note 5], RING: ON		17.5	63.5	μA
			V_{DD} = 3.0 V ±10%	POC: OFF, RING: OFF		0.05	10	μA
				POC: OFF, RING: ON		7.5	25	μA
				POC: ON[Note 5], RING: OFF		3.5	15.5	μA
				POC: ON[Note 5], RING: ON		11	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. I_{DD1} includes peripheral operation current.

3. When PCC = 00H.

4. When main system clock is stopped.

5. Including when LVIE (bit 4 of LVIM) = 1 with POC-OFF selected by a mask option.

6. When POC-OFF (including LVIE = 0) is selected by a mask option and Ring-OSC oscillation is stopped.

7. When the REGC pin is directly connected to V_{DD}.

8. When the REGC pin is connected to V_{SS} via a 0.1 μF capacitor.

9. Including the current that flows through the AV_{REF} pin.

AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock operation	X1 input clock	**Note 1**	3.3 V ≤ V_{DD} ≤ 5.5 V	0.238		16	µs
					2.7 V ≤ V_{DD} < 3.3 V	0.4		16	µs
				Note 2	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
			Ring-OSC clock			4.17	8.33	16.67	µs
		Subsystem clock operation				114	122	125	µs
TI000, TI010, TI001[Note 3], TI011[Note 3] 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 4]			µs
		2.7 V ≤ V_{DD} < 4.0 V				2/f_{sam} + 0.2[Note 4]			µs
TI50, TI51 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V						10	MHz
		2.7 V ≤ V_{DD} < 4.0 V						5	MHz
TI50, TI51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	4.0 V ≤ V_{DD} ≤ 5.5 V				50			ns
		2.7 V ≤ V_{DD} < 4.0 V				100			ns
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}					1			µs
Key return input low-level width	t_{KR}	4.0 V ≤ V_{DD} ≤ 5.5 V				50			ns
		2.7 V ≤ V_{DD} < 4.0 V				100			ns
$\overline{\text{RESET}}$ low-level width	t_{RST}					10			µs

Notes 1. When the REGC pin is connected to V_{SS} via a 0.1 µF capacitor.

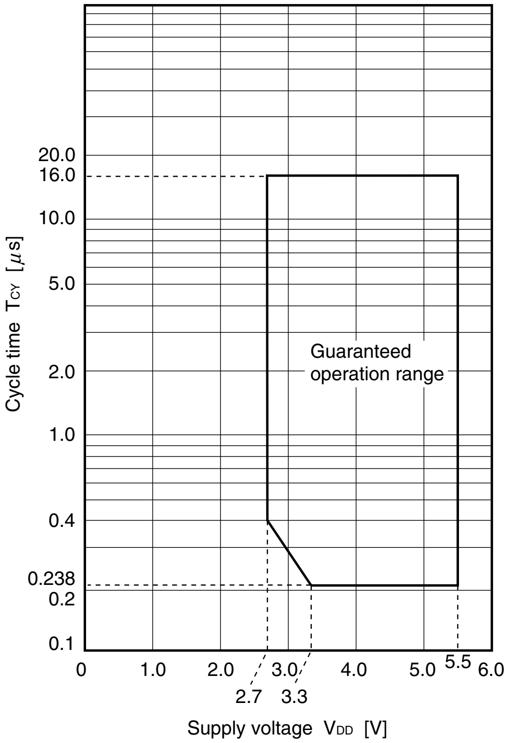
2. When the REGC pin is directly connected to V_{DD}.

3. µPD780146, 780148, and 78F0148 only.

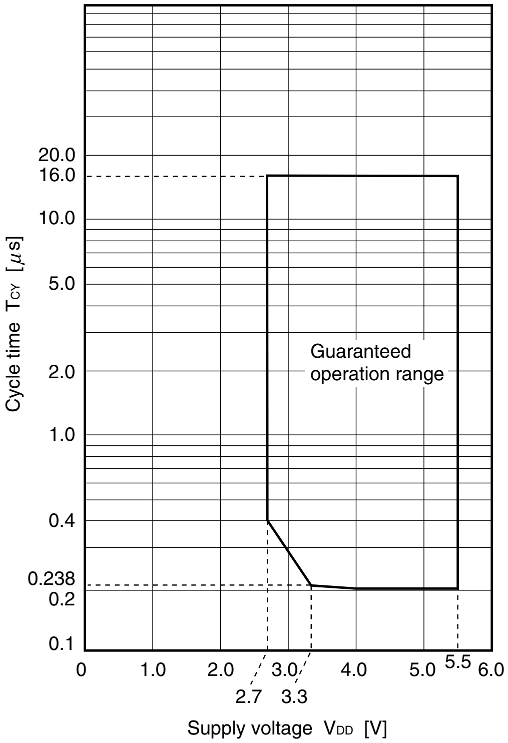
4. Selection of f_{sam} = f_{XP}, f_{XP}/4, f_{XP}/256, or f_{XP}, f_{XP}/16, f_{XP}/64 is possible using bits 0 and 1 (PRM000, PRM001 or PRM010, PRM011) of prescaler mode registers 00 and 01 (PRM00, PRM01). Note that when selecting the TI000, TI010, TI001, or TI011 valid edge as the count clock, f_{sam} = f_{XP}.

T_{CY} vs. V_{DD} (X1 Input Clock Operation)

(a) When REGC pin is connected to V_{SS} via 0.1 μF capacitor



(b) When REGC pin is directly connected to V_{DD}



(2) Read/write operation

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(1/2)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
ASTB high-level width	t_{ASTH}		$0.3t_{CY}$		ns
Address setup time	t_{ADS}		20		ns
Address hold time	t_{ADH}		6		ns
Data input time from address	t_{ADD1}			$(2 + 2n)t_{CY} - 54$	ns
	t_{ADD2}			$(3 + 2n)t_{CY} - 60$	ns
Address output time from $\overline{RD}$↓	t_{RDAD}		0	100	ns
Data input time from $\overline{RD}$↓	t_{RDD1}			$(2 + 2n)t_{CY} - 87$	ns
	t_{RDD2}			$(3 + 2n)t_{CY} - 93$	ns
Read data hold time	t_{RDH}		0		ns
$\overline{RD}$ low-level width	t_{RDL1}		$(1.5 + 2n)t_{CY} - 33$		ns
	t_{RDL2}		$(2.5 + 2n)t_{CY} - 33$		ns
Input time from $\overline{RD}$↓ to $\overline{WAIT}$↓	t_{RDWT1}			$t_{CY} - 43$	ns
	t_{RDWT2}			$t_{CY} - 43$	ns
Input time from $\overline{WR}$↓ to $\overline{WAIT}$↓	t_{WRWT}			$t_{CY} - 25$	ns
$\overline{WAIT}$ low-level width	t_{WTL}		$(0.5 + 2n)t_{CY} + 10$	$(2 + 2n)t_{CY}$	ns
Write data setup time	t_{WDS}		60		ns
Write data hold time	t_{WDH}		6		ns
$\overline{WR}$ low-level width	t_{WRL1}		$(1.5 + 2n)t_{CY} - 15$		ns
Delay time from ASTB↓ to $\overline{RD}$↓	t_{ASTRD}		6		ns
Delay time from ASTB↓ to $\overline{WR}$↓	t_{ASTWR}		$2t_{CY} - 15$		ns
Delay time from $\overline{RD}$↑ to ASTB↑ at external fetch	t_{RDAST}		$0.8t_{CY} - 15$	$1.2t_{CY}$	ns
Address hold time from $\overline{RD}$↑ at external fetch	t_{RDADH}		$0.8t_{CY} - 15$	$1.2t_{CY} + 30$	ns
Write data output time from $\overline{RD}$↑	t_{RDWD}		40		ns
Write data output time from $\overline{WR}$↓	t_{WRWD}		10	60	ns
Address hold time from $\overline{WR}$↑	t_{WRADH}		$0.8t_{CY} - 15$	$1.2t_{CY} + 30$	ns
Delay time from $\overline{WAIT}$↑ to $\overline{RD}$↑	t_{WTRD}		$0.8t_{CY}$	$2.5t_{CY} + 25$	ns
Delay time from $\overline{WAIT}$↑ to $\overline{WR}$↑	t_{WTWR}		$0.8t_{CY}$	$2.5t_{CY} + 25$	ns

Caution T_{CY} can only be used at 0.238 μs (MIN).

Remarks 1. $t_{CY} = T_{CY}/4$

2. n indicates the number of waits.

3. C_L = 100 pF (C_L indicates the load capacitance of the AD0 to AD7, A8 to A15, $\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, and ASTB pins.)

(2) Read/write operation

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(2/2)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
ASTB high-level width	t_{ASTH}		$0.3t_{CY}$		ns
Address setup time	t_{ADS}		30		ns
Address hold time	t_{ADH}		10		ns
Input time from address to data	t_{ADD1}			$(2 + 2n)t_{CY} - 108$	ns
	t_{ADD2}			$(3 + 2n)t_{CY} - 120$	ns
Output time from $\overline{RD}$↓ to address	t_{RDAD}		0	200	ns
Input time from $\overline{RD}$↓ to data	t_{RDD1}			$(2 + 2n)t_{CY} - 148$	ns
	t_{RDD2}			$(3 + 2n)t_{CY} - 162$	ns
Read data hold time	t_{RDH}		0		ns
$\overline{RD}$ low-level width	t_{RDL1}		$(1.5 + 2n)t_{CY} - 40$		ns
	t_{RDL2}		$(2.5 + 2n)t_{CY} - 40$		ns
Input time from $\overline{RD}$↓ to $\overline{WAIT}$↓	t_{RDWT1}			$t_{CY} - 75$	ns
	t_{RDWT2}			$t_{CY} - 60$	ns
Input time from $\overline{WR}$↓ to $\overline{WAIT}$↓	t_{WRWT}			$t_{CY} - 50$	ns
$\overline{WAIT}$ low-level width	t_{WTL}		$(0.5 + 2n)t_{CY} + 10$	$(2 + 2n)t_{CY}$	ns
Write data setup time	t_{WDS}		60		ns
Write data hold time	t_{WDH}		10		ns
$\overline{WR}$ low-level width	t_{WRL1}		$(1.5 + 2n)t_{CY} - 30$		ns
Delay time from ASTB↓ to $\overline{RD}$↓	t_{ASTRD}		10		ns
Delay time from ASTB↓ to $\overline{WR}$↓	t_{ASTWR}		$2t_{CY} - 30$		ns
Delay time from $\overline{RD}$↑ to ASTB↑ at external fetch	t_{RDAST}		$0.8t_{CY} - 30$	$1.2t_{CY}$	ns
Hold time from $\overline{RD}$↑ to address at external fetch	t_{RDADH}		$0.8t_{CY} - 30$	$1.2t_{CY} + 60$	ns
Write data output time from $\overline{RD}$↑	t_{RDWD}		40		ns
Write data output time from $\overline{WR}$↓	t_{WRWD}		20	120	ns
Hold time from $\overline{WR}$↑ to address	t_{WRADH}		$0.8t_{CY} - 30$	$1.2t_{CY} + 60$	ns
Delay time from $\overline{WAIT}$↑ to $\overline{RD}$↑	t_{WTRD}		$0.5t_{CY}$	$2.5t_{CY} + 50$	ns
Delay time from $\overline{WAIT}$↑ to $\overline{WR}$↑	t_{WTWR}		$0.5t_{CY}$	$2.5t_{CY} + 50$	ns

Caution T_{CY} can only be used at 0.4 μs (MIN).

Remarks 1. $t_{CY} = T_{CY}/4$

2. n indicates the number of waits.

3. C_L = 100 pF (C_L indicates the load capacitance of the AD0 to AD7, A8 to A15, $\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, and ASTB pins.)

(3) Serial interface

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) UART mode (UART6, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbs

(b) UART mode (UART0, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbs

(c) 3-wire serial I/O mode (master mode, $\overline{\text{SCK1n}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		3.3 V ≤ V_{DD} < 4.0 V	240			ns
		2.7 V ≤ V_{DD} < 3.3 V	400			ns
$\overline{\text{SCK1n}}$ high-/low-level width	t_{KH1}, t_{KL1}		t_{KCY1}/2–10			ns
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK1}		30			ns
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO1}	C = 100 pF[Note]			30	ns

Note C is the load capacitance of the $\overline{\text{SCK1n}}$ and SO1n output lines.

(d) 3-wire serial I/O mode (slave mode, $\overline{\text{SCK1n}}$... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ cycle time	t_{KCY2}		400			ns
$\overline{\text{SCK1n}}$ high-/low-level width	t_{KH2}, t_{KL2}		t_{KCY2}/2			ns
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK2}		80			ns
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO2}	C = 100 pF[Note]			120	ns

Note C is the load capacitance of the SO1n output line.

Remark n = 0: μPD780143, 780144
n = 0, 1: μPD780146, 780148, 78F0148

(e) 3-wire serial I/O mode with automatic transmit/receive function ($\overline{\text{SCKA0}}$... internal clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKA0}}$ cycle time	t_{KCY3}	4.0 V ≤ V_{DD} ≤ 5.5 V		600			ns
		2.7 V ≤ V_{DD} < 4.0 V		1200			ns
$\overline{\text{SCKA0}}$ high-/low-level width	t_{TH3}, t_{TL3}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY3}2–50			ns
		2.7 V ≤ V_{DD} < 4.0 V		t_{KCY3}/2–100			ns
SIA0 setup time (to $\overline{\text{SCKA0}}$↑)	t_{SIK3}			100			ns
SIA0 hold time (from $\overline{\text{SCKA0}}$↑)	t_{KSI3}			300			ns
Delay time from $\overline{\text{SCKA0}}$↓ to SOA0 output	t_{KSO3}	C = 100 pF[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V			200	ns
			2.7 V ≤ V_{DD} < 4.0 V			300	
Time from $\overline{\text{SCKA0}}$↑ to STB0↑	t_{SBD}			t_{KCY3}/2–100			ns
Strobe signal high-level width	t_{SBW}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY3}–30			ns
		2.7 V ≤ V_{DD} < 4.0 V		t_{KCY3}–60			ns
Busy signal setup time (to busy signal detection timing)	t_{BYS}			100			ns
Busy signal hold time (from busy signal detection timing)	t_{BYH}	4.0 V ≤ V_{DD} ≤ 5.5 V		100			ns
		2.7 V ≤ V_{DD} < 4.0 V		150			ns
Time from busy inactive to $\overline{\text{SCKA0}}$↓	t_{SPS}					2t_{KCY3}	ns

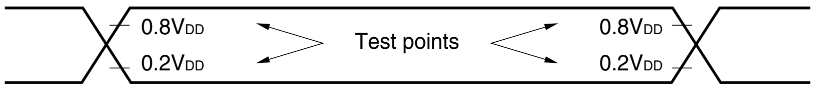
Note C is the load capacitance of the $\overline{\text{SCKA0}}$ and SOA0 output lines.

(f) 3-wire serial I/O mode with automatic transmit/receive function ($\overline{\text{SCKA0}}$ ... external clock input)

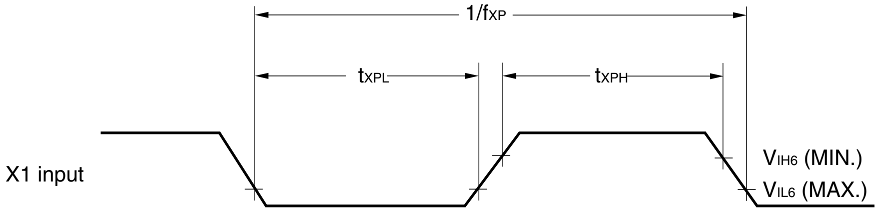
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKA0}}$ cycle time	t_{KCY4}	4.0 V ≤ V_{DD} ≤ 5.5 V		600			ns
		2.7 V ≤ V_{DD} < 4.0 V		1200			ns
$\overline{\text{SCKA0}}$ high-/low-level width	t_{KH4}, t_{KL4}	4.0 V ≤ V_{DD} ≤ 5.5 V		300			ns
		2.7 V ≤ V_{DD} < 4.0 V		600			ns
SIA0 setup time (to $\overline{\text{SCKA0}}$↑)	t_{SIK4}			100			ns
SIA0 hold time (from $\overline{\text{SCKA0}}$↑)	t_{KSI4}			300			ns
Delay time from $\overline{\text{SCKA0}}$↓ to SOA0 output	t_{KSO4}	C = 100 pF[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V			200	ns
			2.7 V ≤ V_{DD} < 4.0 V			300	ns
$\overline{\text{SCKA0}}$ rise/fall time	t_{R4}, t_{F4}	When external device expansion function is used				120	ns
		When external device expansion function is not used				1000	ns

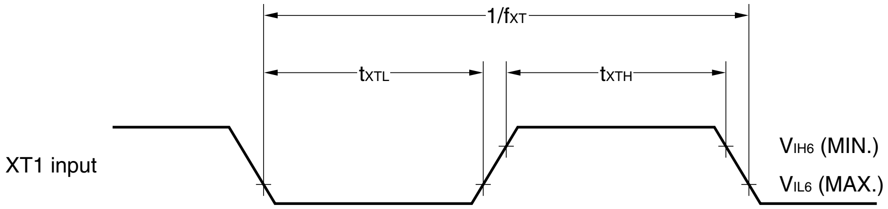
Note C is the load capacitance of the SOA0 output line.

AC Timing Test Points (Excluding X1 Input)

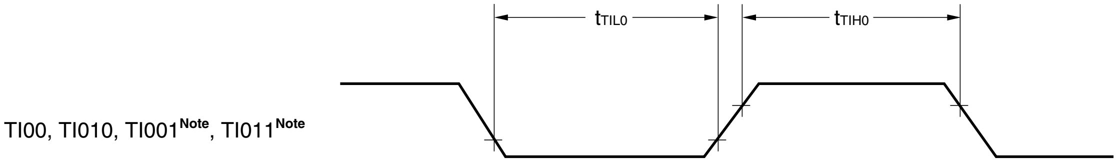


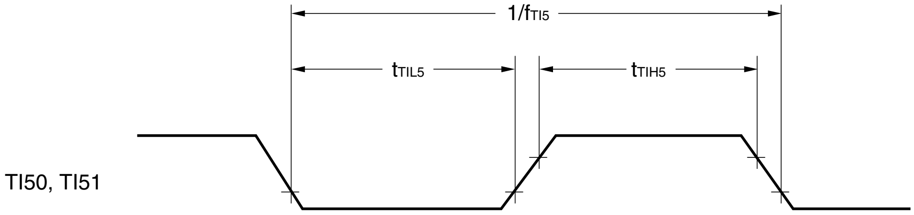
Clock Timing



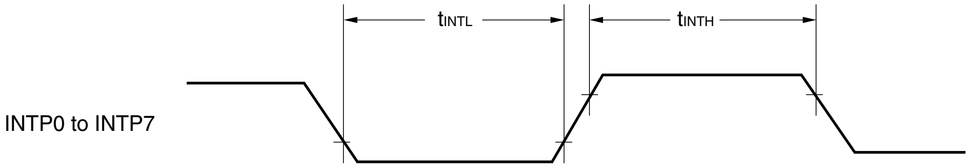


TI Timing



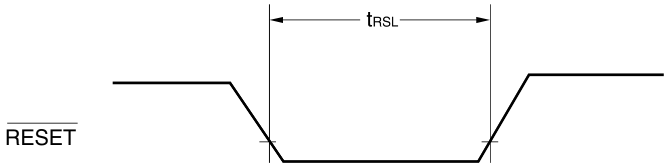


Interrupt Request Input Timing



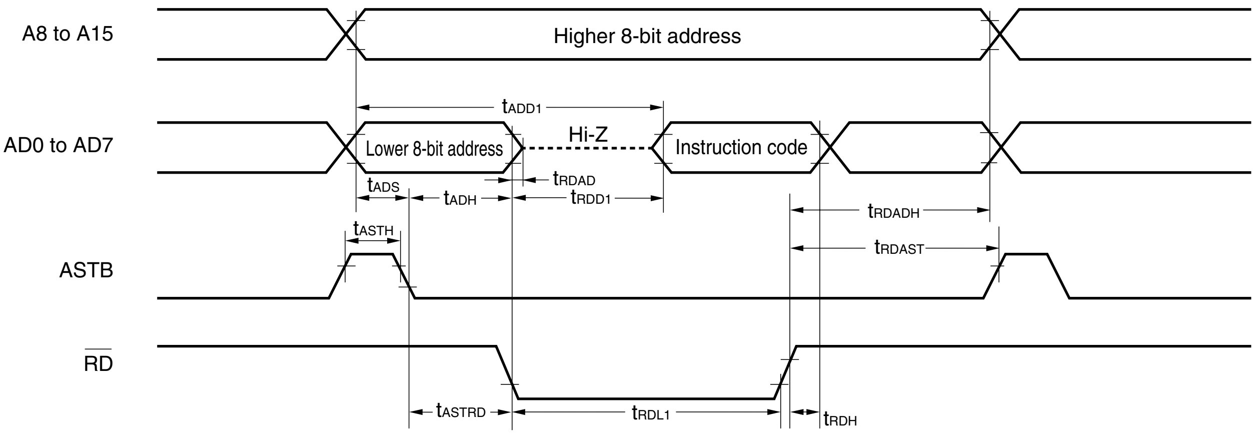
Note μPD780146, 780148, and 78F0148 only.

$\overline{\text{RESET}}$ Input Timing

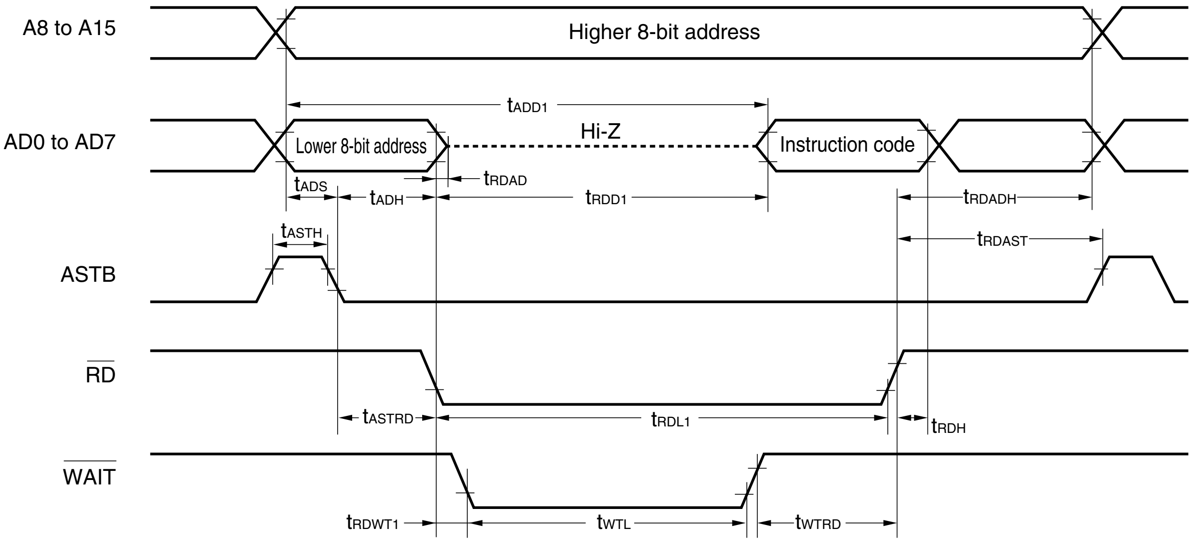


Read/Write Operation

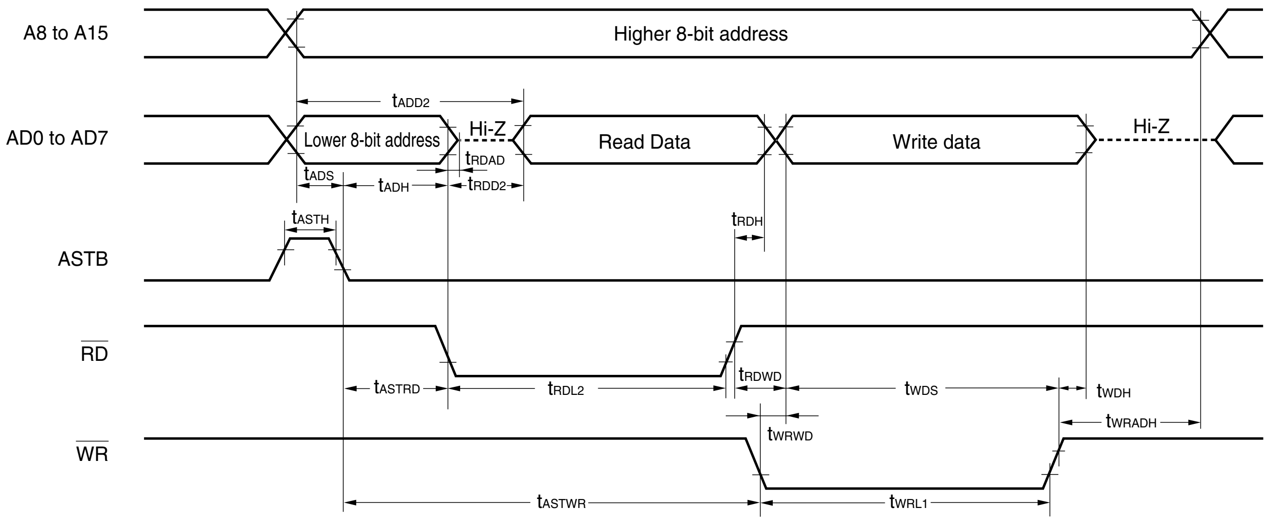
External fetch (no wait):



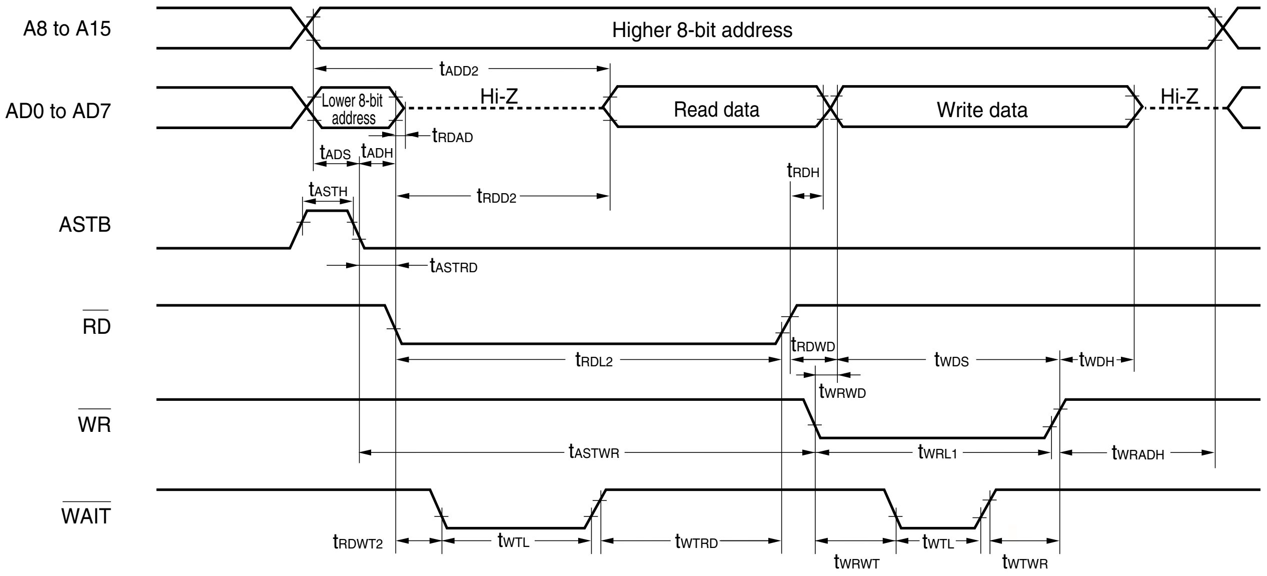
External fetch (wait insertion):



External data access (no wait):

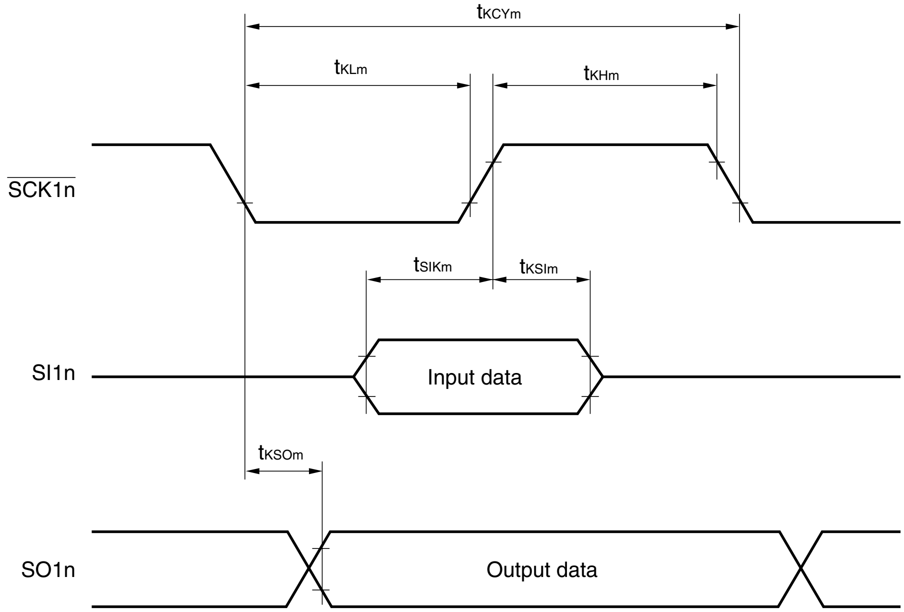


External data access (wait insertion):



Serial Transfer Timing

3-wire serial I/O mode:

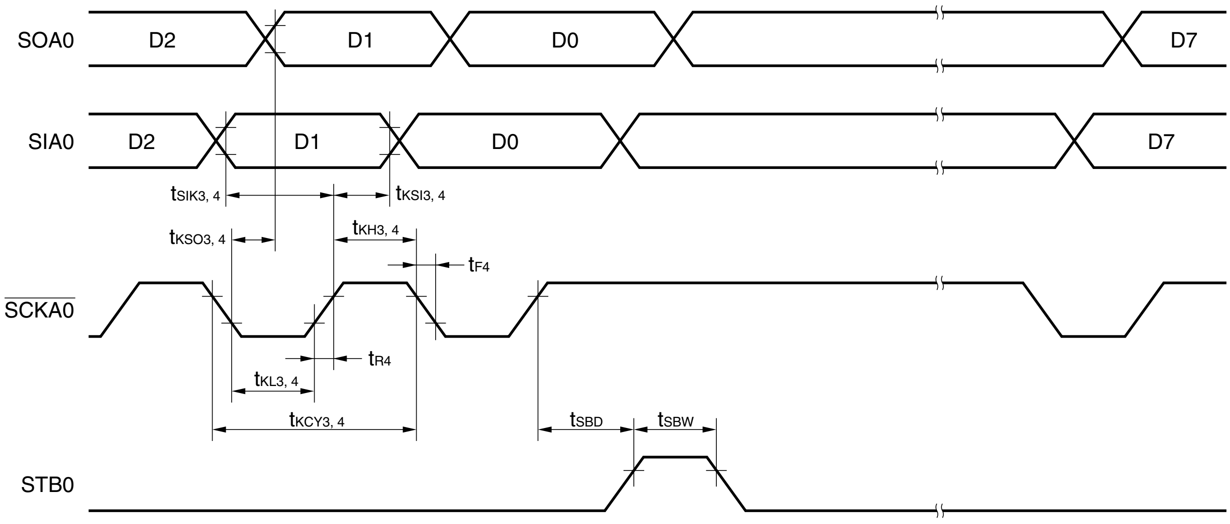


Remark m = 1, 2

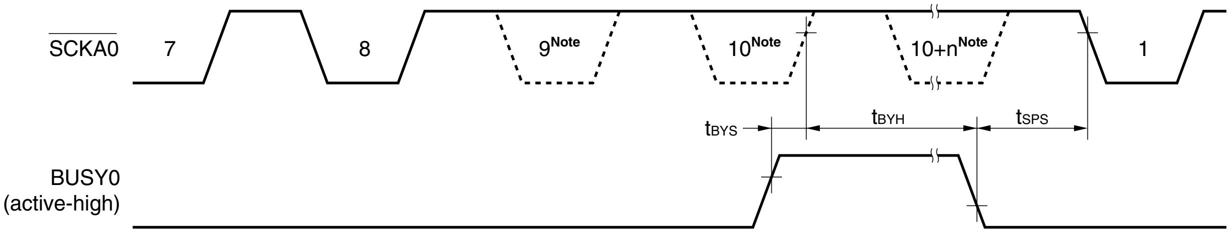
n = 0: μPD780143, 780144

n = 0, 1: μPD780146, 780148, 78F0148

3-wire serial I/O mode with automatic transmit/receive function:



3-wire serial I/O mode with automatic transmit/receive function (busy processing):



Note The signal is not actually driven low here; it is shown as such to indicate the timing.

★ A/D Converter Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V		±0.3	±0.6	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	14		100	μs
		2.7 V ≤ AV_{REF} < 4.0 V	17		100	μs
Zero-scale error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Full-scale error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Integral non-linearity error[Note 1]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
Differential non-linearity error[Note 1]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

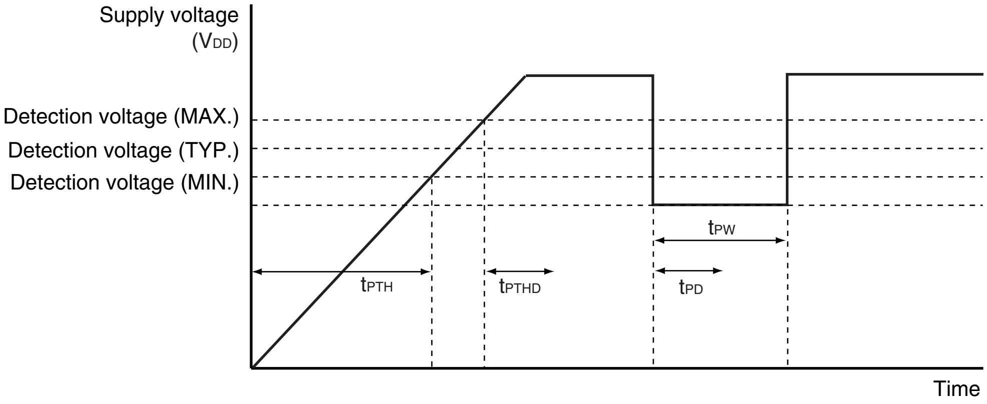
2. This value is indicated as a ratio (%FSR) to the full-scale value.

★ POC Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}	Mask option = 3.5 V	3.3	3.5	3.7	V
	V_{POC1}	Mask option = 2.85 V	2.7	2.85	3.0	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → 2.7 V	0.0015		1500	ms
		V_{DD}: 0 V → 3.3 V	0.002		1800	ms
Response delay time 1[Note]	t_{PTHD}	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2[Note]	t_{PD}	When power supply falls, V_{DD} = 1.7 V			1.0	ms
Minimum pulse width	t_{PW}		0.2			ms

Note Time required from voltage detection to reset release.

POC Circuit Timing



LVI Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		4.1	4.3	4.5	V
	V_{LVI1}		3.9	4.1	4.3	V
	V_{LVI2}		3.7	3.9	4.1	V
	V_{LVI3}		3.5	3.7	3.9	V
	V_{LVI4}		3.3	3.5	3.7	V
★	V_{LVI5}		3.15	3.3	3.45	V
★	V_{LVI6}		2.95	3.1	3.25	V
Response time[Note 1]	t_{LD}			0.2	2.0	ms
Minimum pulse width	t_{LW}		0.2			ms
★ Reference voltage stabilization wait time[Note 2]	t_{LWAIT0}			0.5	2.0	ms
Operation stabilization wait time[Note 3]	t_{LWAIT1}			0.1	0.2	ms

Notes 1. Time required from voltage detection to interrupt output or $\overline{\text{RESET}}$ output.

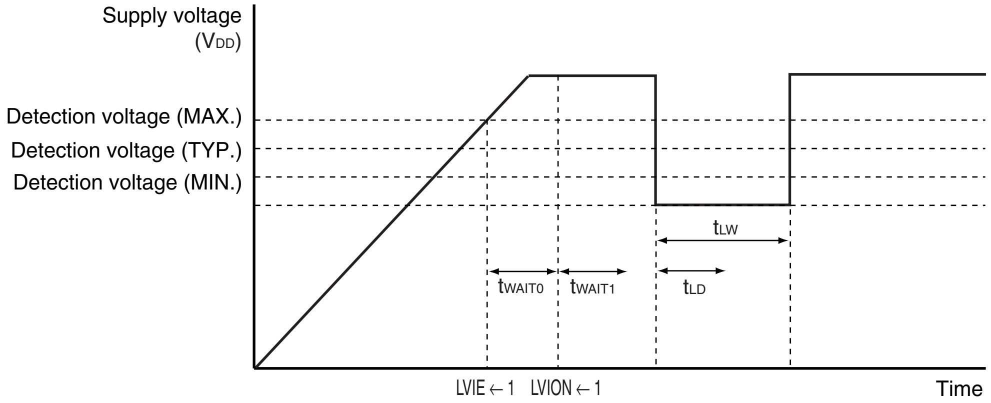
2. Time required from setting LVIE to 1 to reference voltage stabilization when POC = OFF is selected by the POC mask option.

3. 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. $V_{POCn} < V_{LVIm}$ (n = 0 and 1, 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
Data retention supply voltage	V_{DDDR}		1.6		5.5	V
Release signal set time	t_{SREL}		0			μs

★ Flash Memory Programming Characteristics: μPD78F0148

(T_A = +10 to +60°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(1) Write erase characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} supply voltage	V_{PP2}	During flash memory programming	9.7	10.0	10.3	V
V_{DD} supply current	I_{DD}	When V_{PP} = V_{PP2}, f_{XP} = 10 MHz, V_{DD} = 5.5 V			37	mA
V_{PP} supply current	I_{PP}	V_{PP} = V_{PP2}			100	mA
Step erase time[Note 1]	T_{er}		0.199	0.2	0.201	s
Overall erase time[Note 2]	T_{era}	When step erase time = 0.2 s			20	s/chip
Writeback time[Note 3]	T_{wb}		49.4	50	50.6	ms
Number of writebacks per 1 writeback command[Note 4]	C_{wb}	When writeback time = 50 ms			60	Times
Number of erases/writebacks	C_{erwb}				16	Times
Step write time[Note 5]	T_{wr}		48	50	52	μs
Overall write time per word[Note 6]	T_{wrw}	When step write time = 50 μs (1 word = 1 byte)	48		520	μs
Number of rewrites per chip[Note 7]	C_{erwr}	1 erase + 1 write after erase = 1 rewrite			20	Times

Notes 1. The recommended setting value of the step erase time is 0.2 s.

2. The prewrite time before erasure and the erase verify time (writeback time) are not included.

3. The recommended setting value of the writeback time is 50 ms.

4. Writeback is executed once by the issuance of the writeback command. Therefore, the number of retries must be the maximum value minus the number of commands issued.

5. The recommended setting value of the step write time is 50 μs.

6. The actual write time per word is 100 μs longer. The internal verify time during or after a write is not included.

7. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

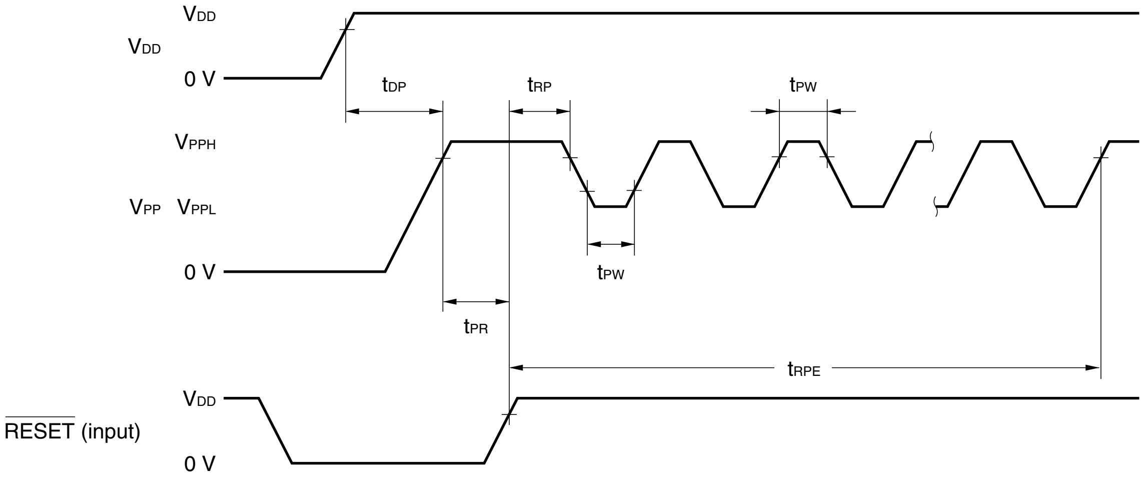
Example: P: Write, E: Erase
Shipped product → P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

Remark The range of the operating clock during flash memory programming is the same as the range during normal operation.

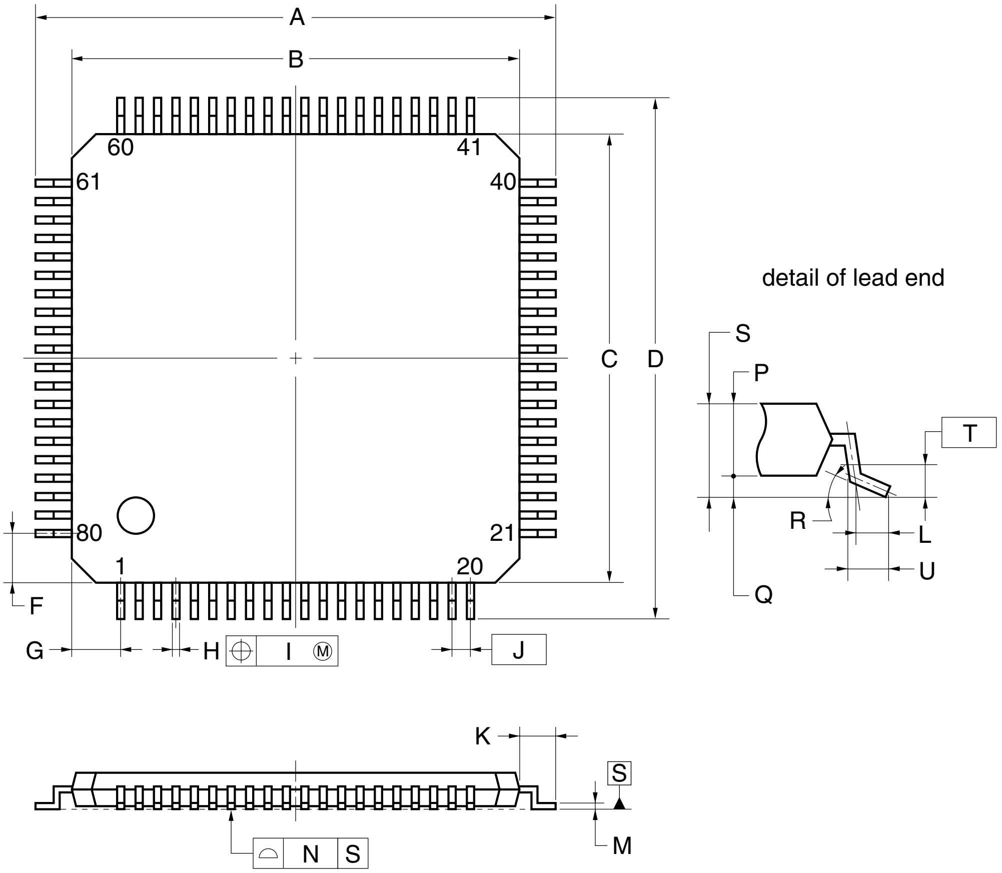
(2) Serial write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Set time from V_{DD}↑ to V_{PP}↑	t_{DP}		10			μs
Release time from V_{PP}↑ to $\overline{\text{RESET}}$↑	t_{PR}		10			μs
V_{PP} pulse input start time from $\overline{\text{RESET}}$↑	t_{RP}		2			ms
V_{PP} pulse high-/low-level width	t_{PW}		8			μs
V_{PP} pulse input end time from $\overline{\text{RESET}}$↑	t_{RPE}				20	ms
V_{PP} pulse low-level input voltage	V_{PPL}		$0.8V_{DD}$		$1.2V_{DD}$	V
V_{PP} pulse high-level input voltage	V_{PPH}		9.7	10.0	10.3	V

Flash Write Mode Setting Timing



80-PIN PLASTIC TQFP (FINE PITCH) (12x12)



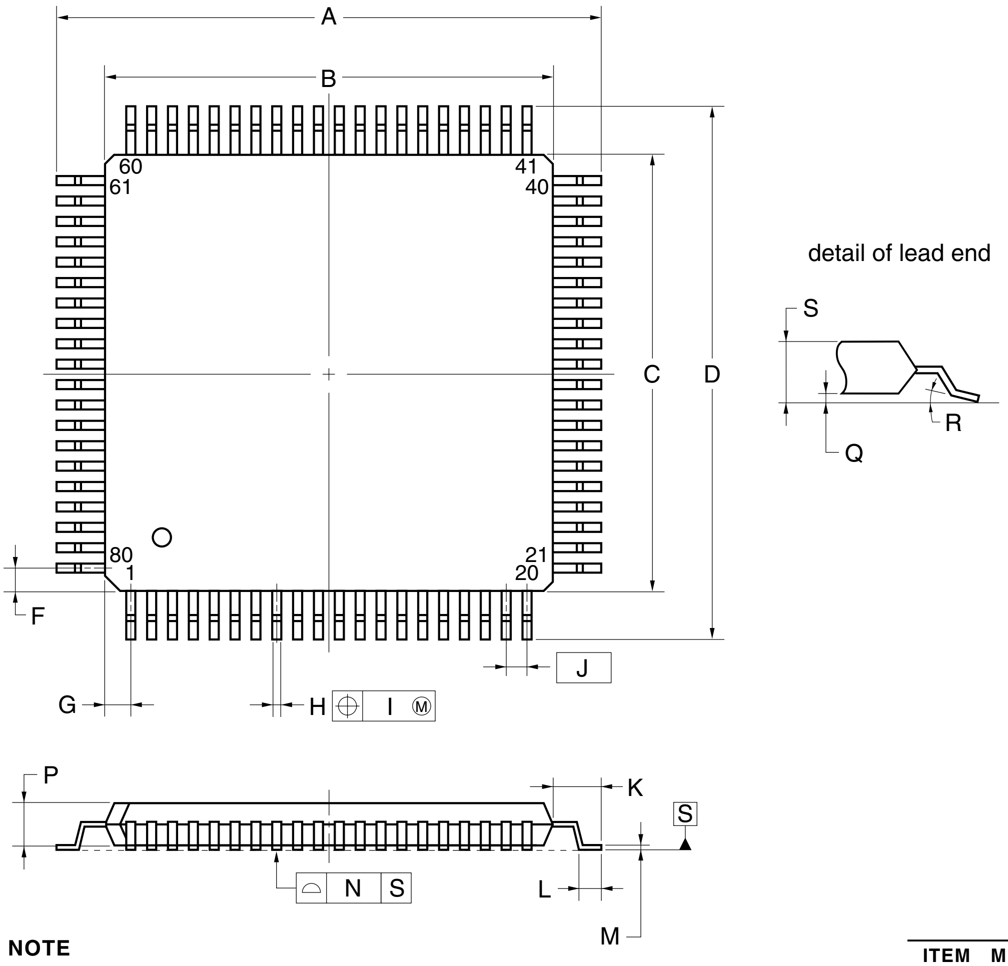
NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	14.0±0.2
B	12.0±0.2
C	12.0±0.2
D	14.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5
M	0.145±0.05
N	0.08
P	1.0
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.1±0.1
T	0.25
U	0.6±0.15

P80GK-50-9EU-1

80-PIN PLASTIC QFP (14x14)



NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	17.20±0.20
B	14.00±0.20
C	14.00±0.20
D	17.20±0.20
F	0.825
G	0.825
H	0.32±0.06
I	0.13
J	0.65 (T.P.)
K	1.60±0.20
L	0.80±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	1.40±0.10
Q	0.125±0.075
R	$3°^{+7°}_{-3°}$
S	1.70 MAX.

P80GC-65-8BT-1

★

CHAPTER 32 CAUTIONS FOR WAIT

32.1 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 32-1**). This must be noted when real-time processing is performed.

32.2 Peripheral Hardware That Generates Wait

Table 32-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

Table 32-1. Registers That Generate Wait and Number of CPU Wait Clocks

<table>
<tr><th>Peripheral Hardware</th><th>Register</th><th>Access</th><th>Number of Wait Clocks</th></tr>
<tr><td>Watchdog timer</td><td>WDTM</td><td>Write</td><td>3 clocks (fixed)</td></tr>
<tr><td>Serial interface UART0</td><td>ASIS0</td><td>Read</td><td>1 clock (fixed)</td></tr>
<tr><td>Serial interface UART6</td><td>ASIS6</td><td>Read</td><td>1 clock (fixed)</td></tr>
<tr><td rowspan="6">A/D converter</td><td>ADM</td><td>Write</td><td rowspan="4">2 to 5 clocks[Note]
(when ADM.5 flag = “1”)
2 to 9 clocks[Note]
(when ADM.5 flag = “0”)</td></tr>
<tr><td>ADS</td><td>Write</td></tr>
<tr><td>PFM</td><td>Write</td></tr>
<tr><td>PFT</td><td>Write</td></tr>
<tr><td>ADCR</td><td>Read</td><td>1 to 5 clocks
(when ADM.5 flag = “1”)
1 to 9 clocks
(when ADM.5 flag = “0”)</td></tr>
<tr><td colspan="3">&lt;Calculating maximum number of wait clocks>
$\{(1/f_{MACRO}) \times 2/(1/f_{CPU})\} + 1$
*The result after the decimal point is truncated if it is less than t_{CPUL} after it has been multiplied by (1/f_{CPU}), and is rounded up if it exceeds t_{CPUL}.
f_{MACRO}: Macro operating frequency
(When bit 5 (FR2) of ADM = “1”: $f_X/2$, when bit 5 (FR2) of ADM = “0”: $f_X/2^2$)
f_{CPU}: CPU clock frequency
t_{CPUL}: Low-level width of CPU clock</td></tr>
</table>

Note No wait cycle is generated for the CPU if the number of wait clocks calculated by the above expression is 1.

Remarks 1. The clock is the CPU clock (f_{CPU}).

2. When the CPU is operating on the subsystem clock and the X1 input clock is stopped, do not access the registers listed above using an access method in which a wait request is issued.

32.3 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 32-2. 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: X1 input clock frequency

t_{CPUL}: Low-level width of CPU clock

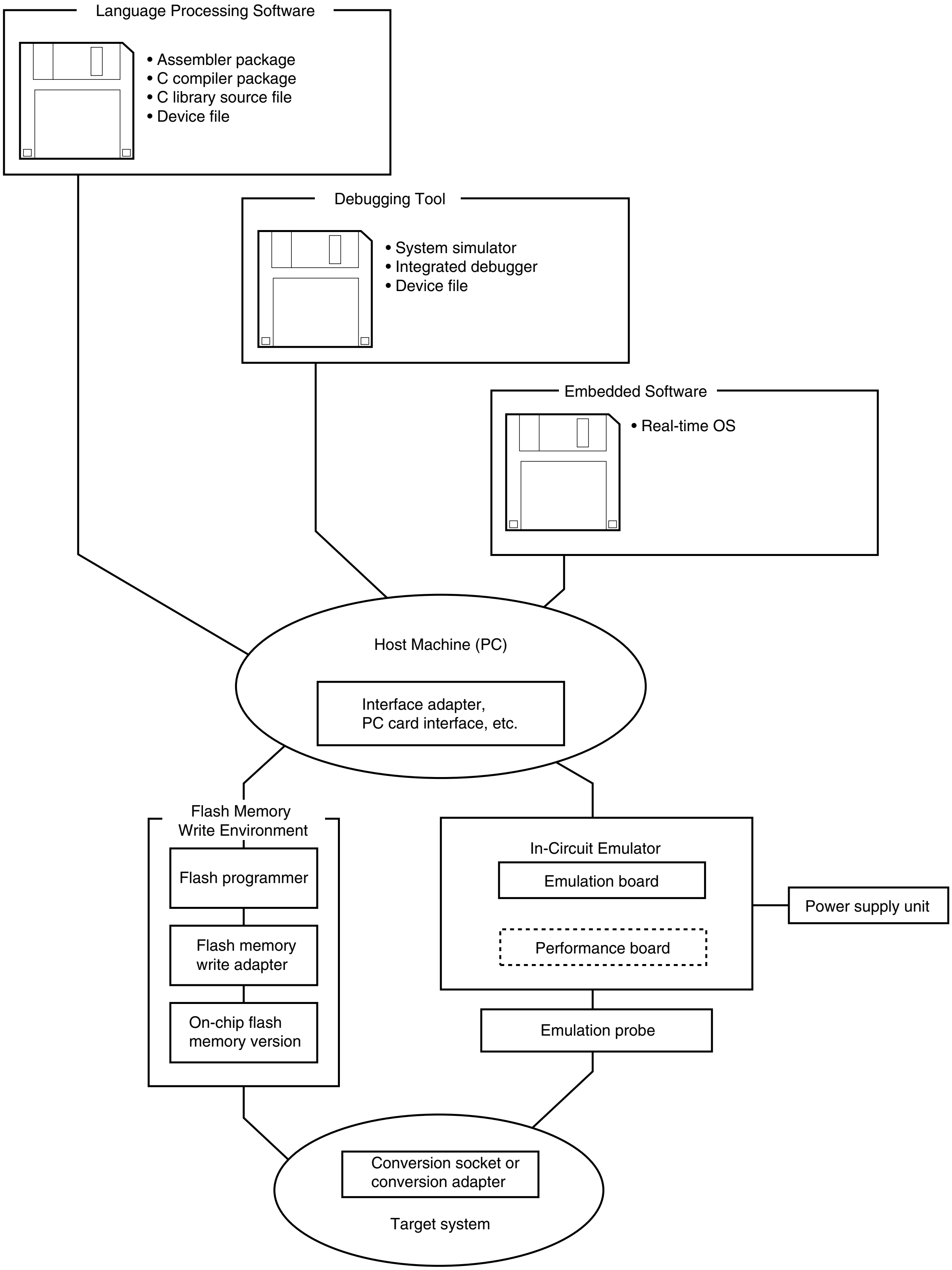
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0/KF1 Series. 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, 98, 2000
  - Windows NT™ Ver 4.0

Figure A-1. Development Tool Configuration



Remark The item in the broken-line box differs according to the development environment. See **A.4.1 Hardware**.

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: *μ*S××××SP78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

*μ*S××××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 (DF780148) (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
DF780148[Notes 1, 2] 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××××DF780148
CC78K0-L[Note 3] 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 working environment does not depend on any particular operating system.
	Part number: µS××××CC78K0-L

Notes 1. The DF780148 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0.

2. Under development

3. 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××××DF780148
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 Flash Memory Writing Tools

Flashpro III (part number: FL-PR3, PG-FP3) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-80GK-9EU FA-80GC-8BT Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro III. • FA-80GK-9EU: For 80-pin plastic TQFP (GK-9EU type) • FA-80GC-8BT: For 80-pin plastic QFP (GC-8BT type)
Flashpro III controller	Program to control Flashpro III from a PC. Provided with Flashpro III.

Remark FL-PR3, FA-80GK-9EU, and FA-80GC-8BT are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

A.4 Debugging Tools

A.4.1 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-780148-NS-EM1[Note] 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-80GK Emulation probe		This probe is used to connect the in-circuit emulator to a target system and is designed for use with 80-pin plastic TQFP (GK-9EU type).
	TGK-080SDW Conversion adapter	This conversion socket connects the NP-80GK to a target system board designed for an 80-pin plastic TQFP (GK-9EU type).
NP-80GC Emulation probe		This emulation probe is used to connect the in-circuit emulator and target system, and is designed for an 80-pin plastic QFP (GC-8BT type).
	EV-9200GC-80 Conversion socket	This conversion socket is used to connect the NP-80GC and target system board to which 80-pin plastic QFP (GC-8BT type) can be connected.
NP-80GC-TQ NP-H80GC-TQ Emulation probe		This emulation probe is used to connect the in-circuit emulator and target system, and is designed for an 80-pin plastic QFP (GC-8BT type).
	TGC-080SBP Conversion adapter	This conversion adapter is used to connect the NP-80GC-TQ or NP-H80GC-TQ and a target system board to which an 80-pin plastic QFP (GC-8BT type) can be connected.

Note Under development

Remarks 1. NP-80GK, NP-80GC, NP-80GC-TQ, and NP-H80GC-TQ are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

2. TGK-080SDW and TGC-080SBP are products of TOKYO ELETECH CORPORATION.
For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

3. EV-9200GC-80 is sold in five-device units.

4. TGK-080SDW and TGC-080SBP are sold in individual units.

A.4.2 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 (DF780148) (sold separately).
	Part number: µS××××SM78K0
ID78K0-NS Integrated debugger (supporting in-circuit emulator IE-78K0-NS(-A))	This debugger is a control program used to debug 78K/0 Series microcontrollers. It adopts a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif™. It also 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. In addition, by incorporating function modules such as a task debugger and system performance analyzer, the efficiency of debugging programs that run on real-time OSs can be improved. 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
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)	

APPENDIX B EMBEDDED SOFTWARE

The following embedded products are available for efficient development and maintenance of the 78K0/KF1 Series.

Real-Time OS

RX78K/0 Real-time OS	The RX78K/0 is a real-time OS conforming to the μITRON specifications. A tool (configurator) for generating the nucleus of the RX78K/0 and multiple information tables is supplied. Used in combination with an assembler package (RA78K/0) and device file (DF780148) (both sold separately). **<Precaution when using RX78K/0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS prompt when using it in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution To purchase the RX78K/0, first fill in the purchase application form and sign the user agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Object source program for mass production

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)[Note]	3.5-inch 2HD FD
BB13		Windows (English version)[Note]	
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

Note Can also be operated in DOS environment.

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)

NEC

Facsimile Message

From:

____
Name

____
Company

____
Tel. FAX

____
Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

P.R. China
NEC Electronics Shanghai, Ltd.
Fax: +86-21-6841-1137

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: ______________________________

Document number: ______________________________ Page number: ______________

____

____

____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑