To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

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

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

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

User's Manual

μPD780988 Subseries

8-Bit Single-Chip Microcontrollers

μPD780982
μPD780983
μPD780984
μPD780986
μPD780988
μPD78F0988A

μPD780982(A)
μPD780983(A)
μPD780984(A)
μPD780986(A)
μPD780988(A)
μPD78F0988A(A)

Document No. U13029EJ7V1UD00 (7th edition)
Date Published August 2005 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① **VOLTAGE APPLICATION WAVEFORM AT INPUT PIN**

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② **HANDLING OF UNUSED INPUT PINS**

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

③ **PRECAUTION AGAINST ESD**

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ **STATUS BEFORE INITIALIZATION**

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

⑤ **POWER ON/OFF SEQUENCE**

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

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

⑥ **INPUT OF SIGNAL DURING POWER OFF STATE**

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

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

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

- **The information in this document is current as of July, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

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

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

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

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

(Note)

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

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

M8E 02. 11-1

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 (1/2)

Page	Description
U13029JJ6V0UD00 → U13029JJ7V0UD00	
Throughout	• Addition of package 64-pin plastic LQFP (14 x 14) μPD780982GC-×××-8BS, 780983GC-×××-8BS, 780984GC-×××-8BS μPD780986GC-×××-8BS, 780988GC-×××-8BS, 78F0988AGC-8BS μPD780982GC(A)-×××-8BS, 780983GC(A)-×××-8BS, 780984GC(A)-×××-8BS μPD780986GC(A)-×××-8BS, 780988GC(A)-×××-8BS • Change of power supply voltage range as shown below. V_{DD} = 4.0 to 5.5 V → V_{DD} = 3.0 to 5.5 V (expanded-specification products), V_{DD} = 4.0 to 5.5 V (conventional products) • Change of system clock oscillation frequency (f_X) as shown below. f_X = 8.38 MHz → f_X = 12 MHz (expanded-specification products only), f_X =8.38 MHz • Change of minimum instruction execution time
p.26	Addition of **1.1 Expanded-Specification Products and Conventional Products**
	1.6 Pin Configuration (Top View)
p.30	• Addition of **Cautions 2** and **3** to 64-pin plastic SDIP (19.05 mm (750))
p.31	• Addition of **Cautions 2** and **3** to 64-pin plastic QFP (14 x 14), 64-pin plastic LQFP (14 x 14)
p.56	**3.1.2 Internal data memory space** Addition of description on **(1) Internal high-speed RAM** and **(2) Internal expansion RAM**
p.99	Modification of **Table 5-2 Relationship Between CPU Clock and Minimum Instruction Execution Time**
p.105	Modification of **Figure 5-5 Switching Between System Clock and CPU Clock**
p.117	Modification of **Figure 6-9 Format of Prescaler Mode Register 00**
p.118	Modification of **Figure 6-10 Format of Prescaler Mode Register 01**
p.123	Addition of **Figure 6-16 Configuration Diagram of PPG Output**
p.123	Addition of **Figure 6-17 PPG Output Operation Timing**
p.146	Modification of **Figure 7-7 Format of Timer Clock Select Register 50**
p.147	Modification of **Figure 7-8 Format of Timer Clock Select Register 51**
p.147	Modification of **Figure 7-9 Format of Timer Clock Select Register 52**
p.164	Modification of **Figure 8-2 Format of Inverter Timer Control Register 7**
p.173	Modification of **Table 9-1 Loop Detection Time of Watchdog Timer**
p.174	Modification of **Table 9-2 Interval Time**
p.175	Modification of **Figure 9-2 Format of Watchdog Timer Clock Select Register**
p.177	Modification of **Figure 9-4 Format of Oscillation Stabilization Time Select Register**
p.178	Modification of **Table 9-4 Loop Detection Time of Watchdog Timer**
p.179	Modification of **Table 9-5 Interval Time of Interval Timer**
p.204	**11.2 Configuration of A/D Converter** Addition of register figure to **(2) A/D conversion result register 0 (ADCR0)**
p.206	Modification of **Figure 11-2 Format of A/D Converter Mode Register 0**
p.214	**11.5 Notes on A/D Converter** Addition of **(6) Input impedance of ANI0 to ANI7 pins**
p.232	Modification of **Figure 12-9 Format of Baud Rate Generator Control Register 0**
p.233	Modification of **Figure 12-10 Format of Baud Rate Generator Control Register 1**

Major Revisions in This Edition (2/2)

Page	Description
 p.234 p.235	**12.4.2 Asynchronous serial interface (UART) mode** • Modification of description • Modification of **(1) Register setting (c) Baud rate generator control registers 0, 1 (BRGC00, BRGC01)**
p.240	Modification of **Table 12-2 Relationship Between Source Clock of 5-Bit Counter and Value of m (with UART00)**
p.240	Modification of **Table 12-3 Relationship Between Source Clock of 5-Bit Counter and Value of m (with UART01)**
p.241	Modification of **Table 12-4 Relationship Between System Clock and Baud Rate**
p.248	Addition of **Remark** to **12.4.3 Infrared data transfer mode**
p.249	Modification of **Table 12-7 Baud Rate That Can Be Set in Infrared Data Transfer Mode**
p.254	Modification of **Figure 13-2 Format of Serial Operation Mode Register 3**
p.280	Addition of **Caution** to **15.1 External Device Expansion Function**
p.284	Change of R/W to W in **Figure 15-2 Format of Memory Expansion Mode Register**
p.294	Modification of **Figure 16-1 Format of Oscillation Stabilization Time Select Register**
p.297	Modification of **Figure 16-3 Releasing HALT Mode by $\overline{\text{RESET}}$ Input**
p.300	Modification of **Figure 16-5 Releasing STOP Mode by $\overline{\text{RESET}}$ Input**
p.308	Revision of descriptions on flash memory programming as **18.3 Flash Memory Features**
p.329	**18.4.5 Entry RAM area** Modification of **(c) Write time data**
p.357	Addition of **CHAPTER 20 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)**
p.377	Addition of **CHAPTER 21 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**
p.396	Addition of **CHAPTER 22 PACKAGE DRAWINGS**
p.399	Addition of **CHAPTER 23 RECOMMENDED SOLDERING CONDITIONS**
p.402	Modification of **APPENDIX A DEVELOPMENT TOOLS**
p.414	Addition of **APPENDIX B NOTES ON DESIGNING TARGET SYSTEM**
U13029JJ7V0UD00 → U13029JJ7V1UD00	
p.28	Modification of **1.4 Ordering Information**
p.399	Modification of **CHAPTER 23 RECOMMENDED SOLDERING CONDITIONS**

The mark ★ shows major revised points.

INTRODUCTION

Target Readers

This manual is intended for users who wish to understand the functions of the µPD780988 Subseries and to design and develop application systems and programs using these microcontrollers.

Purpose

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

Organization

The µPD780988 Subseries User's Manual is divided into two parts: this manual and instructions (common to the 78K/0 Series).

µPD780988 Subseries User's Manual (This manual)	78K/0 Series User's Manual Instructions
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupt functions	• Explanation of instruction
• Other on-chip peripheral functions	
• Electrical specifications	

How to Read This Manual

It is assumed that the reader of this manual has general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

□ To those who use this manual as the manual of the µPD780982(A), 780983(A), 780984(A), 780986(A), 780988(A), and 78F0988A(A):

→ Unless there are functional differences, the µPD780982, 780983, 780984, 780986, 780988, and 78F0988A are treated as representative devices, therefore, when this is used as a manual for the µPD780982(A), 780983(A), 780984(A), 780986(A), 780988(A), and 78F0988A(A) read the product names as µPD780982(A), 780983(A), 780984(A), 780986(A), 780988(A), and 78F0988A(A).

□ To understand the functions in general:

→ Read this manual in the order of the contents.

□ How to interpret register format:

→ The bit name of a bit whose number is encircled is defined as a reserved word in the RA78K0, and in the header file sfrbit.in the CC78K0.

□ When you know a register name and want to confirm its details:

→ Read **APPENDIX C REGISTER INDEX**.

□ To know the µPD789830 Subseries instruction functions in detail:

→ Refer to **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 applications that require 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 representation:	$\overline{\times\times\times}$ (overscore over pin or signal name)
	Note:	Footnote for item marked with **Note** in the text
	Caution:	Information requiring particular attention
	Remark:	Supplementary information
	Numerical representation:	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.
μPD780988 Subseries User's Manual	This manual
μPD780988 Subseries Inverter Control Application Note	U13119E
78K/0 Series Instructions User's Manual	U12326E
78K/0 Series Basics (I) Application Note	U12704E

Documents Related to Development Software Tools (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
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
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 for designing.**

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

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-78K0-NS-PA Performance Board	To be prepared
IE-78001-R-A In-Circuit Emulator	U14142E
IE-78K0-R-EX1 In-Circuit Emulator	To be prepared

Documents Related to Flash Memory Writing

Document Name	Document No.
PG-FP3 Flash Memory Programmer User's Manual	U13502E
PG-FP4 Flash Memory Programmer User's Manual	U15260E

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products and Packages -	X13769X
Semiconductor Device 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

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

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

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

Figure No. Title Page

LIST OF FIGURES (3/6)

Figure No. Title Page

LIST OF FIGURES (4/6)

Figure No. Title Page

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL

★ 1.1 Expanded-Specification Products and Conventional Products

The expanded-specification product and conventional product refer to the following products.

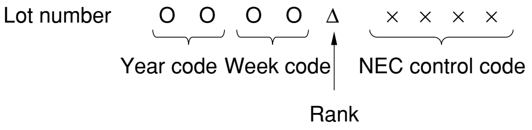
Expanded-specification product: Products with a rank[Note] other than K

- Mask ROM versions for which orders were received after December 1, 2001.
- Flash memory versions that were shipped after January 1, 2002.

Conventional product: Products with rank[Note] K

- Products other than the above expanded specification products.

Note The rank is indicated by the 5th digit from the left in the lot number marked on the package.



Expanded-specification products and conventional products differ in the power supply voltage range and operating frequency ratings. The differences are shown in Table 1-1.

Table 1-1. Differences Between Expanded-Specification Products and Conventional Products

Power Supply Voltage (V$_{DD}$)	Guaranteed Operating Speed (Operating Frequency)	
	Conventional Products	Expanded-Specification Products
4.5 to 5.5 V	8.38 MHz (0.238 μs)	12 MHz (0.166 μs)
4.0 to 5.5 V	8.38 MHz (0.238 μs)	8.38 MHz (0.238 μs)
3.0 to 5.5 V	–	8.38 MHz (0.238 μs)

Remark The parenthesized values indicates the minimum instruction execution time.

1.2 Features

- Internal ROM and RAM

Item Part Number	Program Memory		Data Memory	
	Internal ROM	Flash Memory	Internal High-Speed RAM	Internal Expansion RAM
μPD780982	16 KB	—	1,024 bytes	—
μPD780983	24 KB	—		—
μPD780984	32 KB	—		—
μPD780986	48 KB	—		1,024 bytes
μPD780988	60 KB	—		
μPD78F0988A	—	60 KB[Note 1]		1,024 bytes[Note 2]

Notes 1. 16, 24, 32, 48, or 60 KB are selectable by using the internal memory size switching register (IMS).

2. 0 or 1,024 bytes are selectable by using the internal expansion RAM size switching register (IXS).

- Less EMI (Electro Magnetic Interference) noise than existing μPD78014 and 78018F Subseries
- External memory expansion space: 256 bytes (except μPD780988)
- ★ Minimum instruction execution time: 0.166 μs (@ f_X = 12 MHz operation[Note]), 0.238 μs (@ f_X = 8.38 MHz operation)
- Instruction set suitable for system control
  - · Bit processing in entire address space
  - · Multiply/divide instructions
- I/O ports: 47
- A/D converter
  - · 10-bit resolution × 8 channels
- Serial interface: 3 channels
  - · UART mode: 2 channels
  - · 3-wire serial I/O mode: 1 channel
- Timer: 7 channels

· 10-bit inverter control timer:	1 channel
· 16-bit timer/event counter:	2 channels
· 8-bit timer/event counter:	3 channels
· Watchdog timer:	1 channel

- Vectored interrupts: 26
- ★ Power supply voltage: V_{DD} = 3.0 to 5.5 V (expanded-specification products)
  V_{DD} = 4.0 to 5.5 V (conventional products)

Note Expanded-specification products only.

1.3 Applications

Motor control for inverter air conditioners, washing machines, refrigerators, etc.

★ 1.4 Ordering Information

- Mask ROM products

Part Number	Package	Quality Grade
μPD780982CW-×××	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780982CW-×××-A	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780982GC-×××-8BS	64-pin plastic LQFP (14 x 14)	Standard
μPD780982GC-×××-8BS-A	64-pin plastic LQFP (14 x 14)	Standard
μPD780983CW-×××	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780983CW-×××-A	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780983GC-×××-8BS	64-pin plastic LQFP (14 x 14)	Standard
μPD780983GC-×××-8BS-A	64-pin plastic LQFP (14 x 14)	Standard
μPD780984CW-×××	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780984CW-×××-A	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780984GC-×××-8BS	64-pin plastic LQFP (14 x 14)	Standard
μPD780984GC-×××-8BS-A	64-pin plastic LQFP (14 x 14)	Standard
μPD780986CW-×××	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780986CW-×××-A	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780986GC-×××-8BS	64-pin plastic LQFP (14 x 14)	Standard
μPD780986GC-×××-8BS-A	64-pin plastic LQFP (14 x 14)	Standard
μPD780988CW-×××	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780988CW-×××-A	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD780988GC-×××-8BS	64-pin plastic LQFP (14 x 14)	Standard
μPD780988GC-×××-8BS-A	64-pin plastic LQFP (14 x 14)	Standard
μPD780982GC(A)-×××-8BS	64-pin plastic LQFP (14 x 14)	Special
μPD780983GC(A)-×××-8BS	64-pin plastic LQFP (14 x 14)	Special
μPD780984GC(A)-×××-8BS	64-pin plastic LQFP (14 x 14)	Special
μPD780986GC(A)-×××-8BS	64-pin plastic LQFP (14 x 14)	Special
μPD780988GC(A)-×××-8BS	64-pin plastic LQFP (14 x 14)	Special

Remarks 1. ××× indicates ROM code suffix.

2. Products that have the part numbers suffixed by “-A” are lead-free products.

Please refer to **Quality Grades on NEC Semiconductor Devices (C11531E)** published by NEC Corporation to know the specification of the quality grades of the devices and applications.

- Flash Memory products

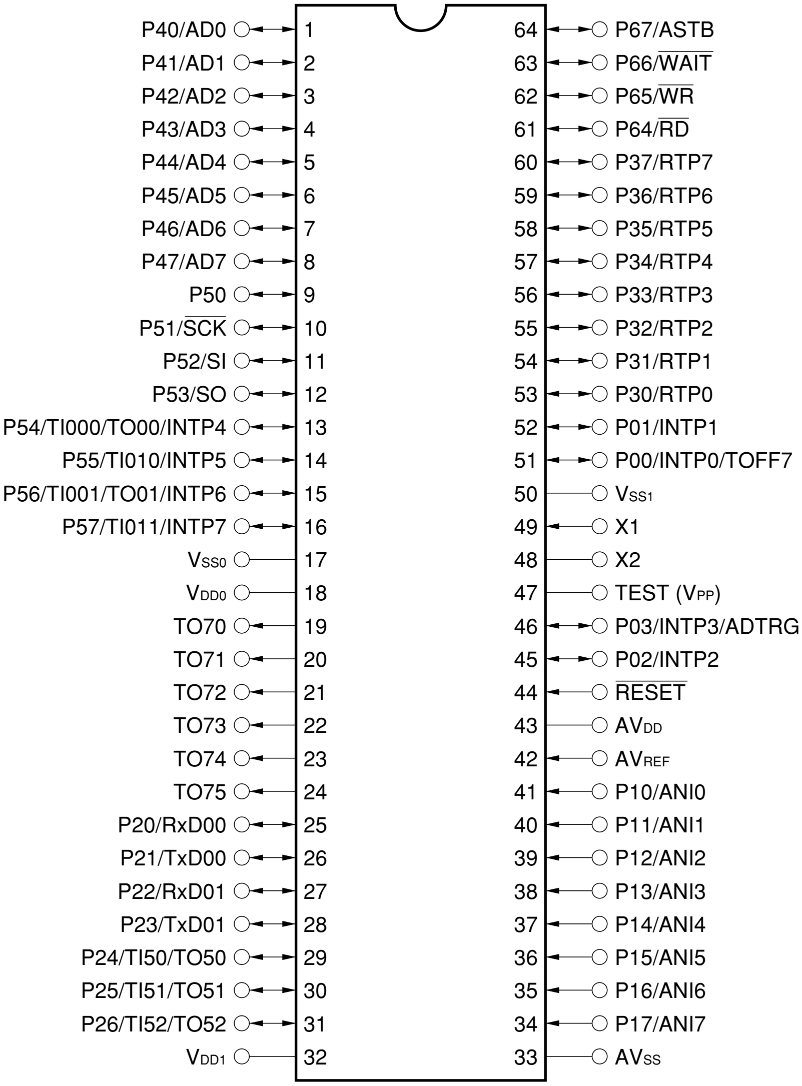
Part Number	Package	Quality Grade
μPD78F0988ACW	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD78F0988ACW-A	64-pin plastic SDIP (19.05 mm (750))	Standard
μPD78F0988AGC-AB8	64-pin plastic QFP (14 x 14)	Standard
μPD78F0988AGC-AB8-A	64-pin plastic QFP (14 x 14)	Standard
μPD78F0988AGC-8BS	64-pin plastic LQFP (14 x 14)	Standard
μPD78F0988AGC-8BS-S	64-pin plastic LQFP (14 x 14)	Standard
μPD78F0988AGC(A)-AB8	64-pin plastic QFP (14 x 14)	Special

Remark Products that have the part numbers suffixed by “-A” are lead-free products.

Please refer to **Quality Grades on NEC Semiconductor Devices (C11531E)** published by NEC Corporation to know the specification of the quality grades of the devices and applications.

1.5 Pin Configuration (Top View)

- **64-pin plastic SDIP (19.05 mm (750))**



Cautions 1. Connect the TEST pin directly to V_{SS0}.

★ **2. Connect the V_{PP} pin directly to the V_{SS0} pin in normal operation mode.**

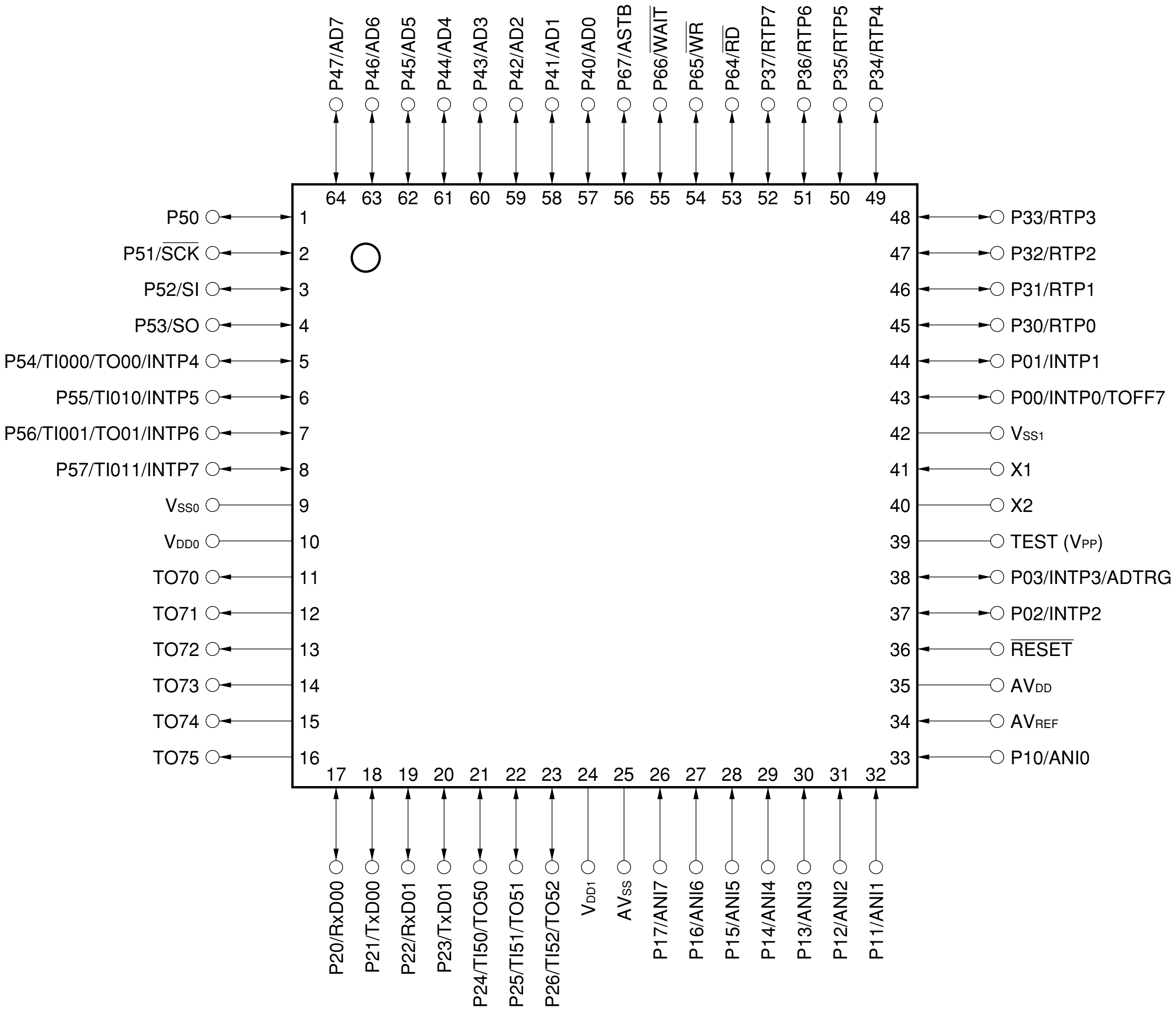
★ **3. Connect the V_{PP} pin to V_{SS0} via a 10 kΩ pull-down resistor in the flash memory writing mode.**

4. The 64-pin plastic SDIP (19.5 mm (750)) package is not supported for special quality grade products.

Remarks 1. The pin connection in parentheses is for the μPD78F0988A.

2. When the μPD780988 Subseries is used in applications where the noise generated inside the microcontroller needs to be reduced, the implementation of noise reduction measures, such as supplying voltage to V_{DD0} and V_{DD1} individually and connecting V_{SS0} and V_{SS1} to different ground lines, is recommended.

- **64-pin plastic QFP (14 x 14)**
- **64-pin plastic LQFP (14 x 14)**



Cautions 1. Connect the TEST pin directly to V_{SS0}.

★ **2. Connect the V_{PP} pin directly to the V_{SS0} pin in normal operation mode.**

★ **3. Connect the V_{PP} pin to V_{SS0} via a 10 kΩ pull-down resistor in the flash memory writing mode.**

Remarks 1. The pin connection in parentheses is for the μPD78F0988A.

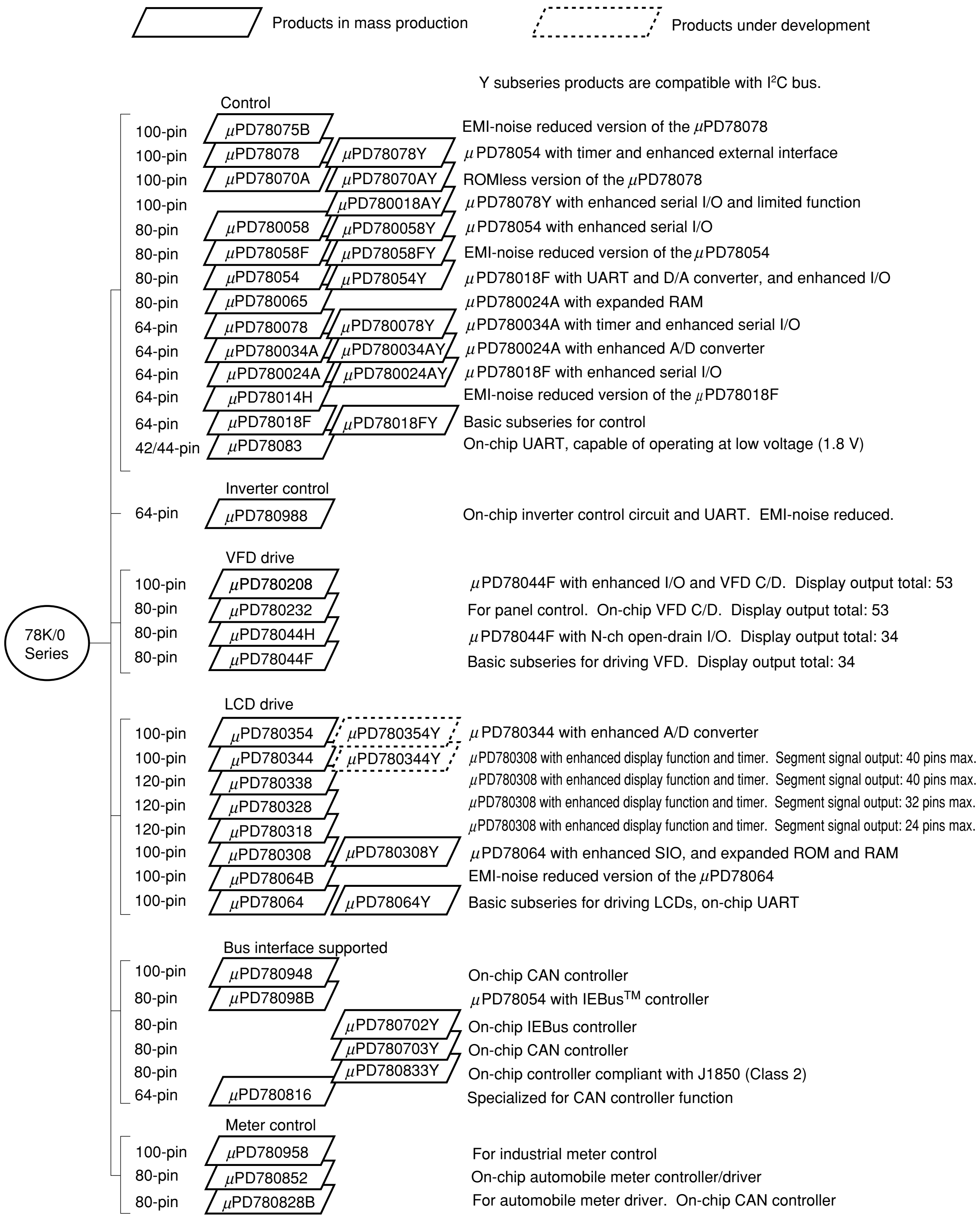
2. When the μPD780988 Subseries is used in applications where the noise generated inside the microcontroller needs to be reduced, the implementation of noise reduction measures, such as supplying voltage to V_{DD0} and V_{DD1} individually and connecting V_{SS0} and V_{SS1} to different ground lines, is recommended.

AD0 to AD7:	Address/data bus
ADTRG:	AD trigger input
ANI0 to ANI7:	Analog input
ASTB:	Address strobe
AV_{DD}:	Analog power supply
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog ground
INTP0 to INTP7:	External interrupt input
P00 to P03:	Port 0
P10 to P17:	Port 1
P20 to P26:	Port 2
P30 to P37:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P64 to P67:	Port 6
$\overline{\text{RD}}$:	Read strobe
$\overline{\text{RESET}}$:	Reset
RTP0 to RTP7:	Real-time port

RxD00, RxD01:	Receive data
$\overline{\text{SCK}}$:	Serial clock
SI:	Serial input
SO:	Serial output
TEST:	Test
TI000, TI001,	
TI010, TI011,	
TI50 to TI52:	Timer input
TO00, TO01,	
TO50 to TO52,	
TO70 to TO75:	Timer output
TOFF7:	Timer output off
TxD00, TxD01:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{PP}:	Programming power supply
V_{SS0}, V_{SS1}:	Ground
$\overline{\text{WAIT}}$:	Wait
$\overline{\text{WR}}$:	Write strobe
X1, X2:	Crystal

★ 1.6 78K/0 Series Lineup

The products in the 78K/0 Series are listed below. The names enclosed in boxes are subseries names.



Y subseries products are compatible with I^{2}C bus.

	Control		
100-pin	μPD78075B		EMI-noise reduced version of the μPD78078
100-pin	μPD78078	μPD78078Y	μPD78054 with timer and enhanced external interface
100-pin	μPD78070A	μPD78070AY	ROMless version of the μPD78078
100-pin		μPD780018AY	μPD78078Y with enhanced serial I/O and limited function
80-pin	μPD780058	μPD780058Y	μPD78054 with enhanced serial I/O
80-pin	μPD78058F	μPD78058FY	EMI-noise reduced version of the μPD78054
80-pin	μPD78054	μPD78054Y	μPD78018F with UART and D/A converter, and enhanced I/O
80-pin	μPD780065		μPD780024A with expanded RAM
64-pin	μPD780078	μPD780078Y	μPD780034A with timer and enhanced serial I/O
64-pin	μPD780034A	μPD780034AY	μPD780024A with enhanced A/D converter
64-pin	μPD780024A	μPD780024AY	μPD78018F with enhanced serial I/O
64-pin	μPD78014H		EMI-noise reduced version of the μPD78018F
64-pin	μPD78018F	μPD78018FY	Basic subseries for control
42/44-pin	μPD78083		On-chip UART, capable of operating at low voltage (1.8 V)
	Inverter control		
64-pin	μPD780988		On-chip inverter control circuit and UART. EMI-noise reduced.
	VFD drive		
100-pin	μPD780208		μPD78044F with enhanced I/O and VFD C/D. Display output total: 53
80-pin	μPD780232		For panel control. On-chip VFD C/D. Display output total: 53
80-pin	μPD78044H		μPD78044F with N-ch open-drain I/O. Display output total: 34
80-pin	μPD78044F		Basic subseries for driving VFD. Display output total: 34
	LCD drive		
100-pin	μPD780354	μPD780354Y	μPD780344 with enhanced A/D converter
100-pin	μPD780344	μPD780344Y	μPD780308 with enhanced display function and timer. Segment signal output: 40 pins max.
120-pin	μPD780338		μPD780308 with enhanced display function and timer. Segment signal output: 40 pins max.
120-pin	μPD780328		μPD780308 with enhanced display function and timer. Segment signal output: 32 pins max.
120-pin	μPD780318		μPD780308 with enhanced display function and timer. Segment signal output: 24 pins max.
100-pin	μPD780308	μPD780308Y	μPD78064 with enhanced SIO, and expanded ROM and RAM
100-pin	μPD78064B		EMI-noise reduced version of the μPD78064
100-pin	μPD78064	μPD78064Y	Basic subseries for driving LCDs, on-chip UART
	Bus interface supported		
100-pin	μPD780948		On-chip CAN controller
80-pin	μPD78098B		μPD78054 with IEBus$^{\text{TM}}$ controller
80-pin		μPD780702Y	On-chip IEBus controller
80-pin		μPD780703Y	On-chip CAN controller
80-pin		μPD780833Y	On-chip controller compliant with J1850 (Class 2)
64-pin	μPD780816		Specialized for CAN controller function
	Meter control		
100-pin	μPD780958		For industrial meter control
80-pin	μPD780852		On-chip automobile meter controller/driver
80-pin	μPD780828B		For automobile meter driver. On-chip CAN controller

Remark VFD (Vacuum Fluorescent Display) is referred to as FIP$^{\text{TM}}$ (Fluorescent Indicator Panel) in some documents, but the functions of the two are the same.

The major functional differences among the subseries are listed below.

- Non-Y subseries

Subseries Name	Function	ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	8-Bit D/A	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
			8-Bit	16-Bit	Watch	WDT							
Control	μPD78075B	32 K to 40 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	88	1.8 V	√
	μPD78078	48 K to 60 K											
	μPD78070A	–									61	2.7 V	
	μPD780058	24 K to 60 K	2 ch							3 ch (time-division UART: 1 ch)	68	1.8 V	
	μPD78058F	48 K to 60 K								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16 K to 60 K										2.0 V	
	μPD780065	40 K to 48 K							–	4 ch (UART: 1 ch)	60	2.7 V	
	μPD780078	48 K to 60 K		2 ch			–	8 ch		3 ch (UART: 2 ch)	52	1.8 V	
	μPD780034A	8 K to 32 K		1 ch						3 ch (UART: 1 ch)	51		
	μPD780024A						8 ch	–					
	μPD78014H									2 ch	53		
	μPD78018F	8 K to 60 K											
	μPD78083	8 K to 16 K		–	–					1 ch (UART: 1 ch)	33		–
Inverter control	μPD780988	16 K to 60 K	3 ch	**Note**	–	1 ch	–	8 ch	–	3 ch (UART: 2 ch)	47	4.0 V	√
VFD drive	μPD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	2 ch	74	2.7 V	–
	μPD780232	16 K to 24 K	3 ch	–	–		4 ch				40	4.5 V	
	μPD78044H	32 K to 48 K	2 ch	1 ch	1 ch		8 ch			1 ch	68	2.7 V	
	μPD78044F	16 K to 40 K								2 ch			
LCD drive	μPD780354	24 K to 32 K	4 ch	1 ch	1 ch	1 ch	–	8 ch	–	3 ch (UART: 1 ch)	66	1.8 V	–
	μPD780344						8 ch	–					
	μPD780338	48 K to 60 K	3 ch	2 ch			–	10 ch	1 ch	2 ch (UART: 1 ch)	54		
	μPD780328										62		
	μPD780318										70		
	μPD780308	48 K to 60 K	2 ch	1 ch			8 ch	–	–	3 ch (time-division UART: 1 ch)	57	2.0 V	
	μPD78064B	32 K								2 ch (UART: 1 ch)			
	μPD78064	16 K to 32 K											
Bus interface supported	μPD780948	60 K	2 ch	2 ch	1 ch	1 ch	8 ch	–	–	3 ch (UART: 1 ch)	79	4.0 V	√
	μPD78098B	40 K to 60 K		1 ch					2 ch		69	2.7 V	–
	μPD780816	32 K to 60 K		2 ch			12 ch		–	2 ch (UART: 1 ch)	46	4.0 V	
Meter control	μPD780958	48 K to 60 K	4 ch	2 ch	–	1 ch	–	–	–	2 ch (UART: 1 ch)	69	2.2 V	–
Dash-board control	μPD780852	32 K to 40 K	3 ch	1 ch	1 ch	1 ch	5 ch	–	–	3 ch (UART: 1 ch)	56	4.0 V	–
	μPD780828B	32 K to 60 K									59		

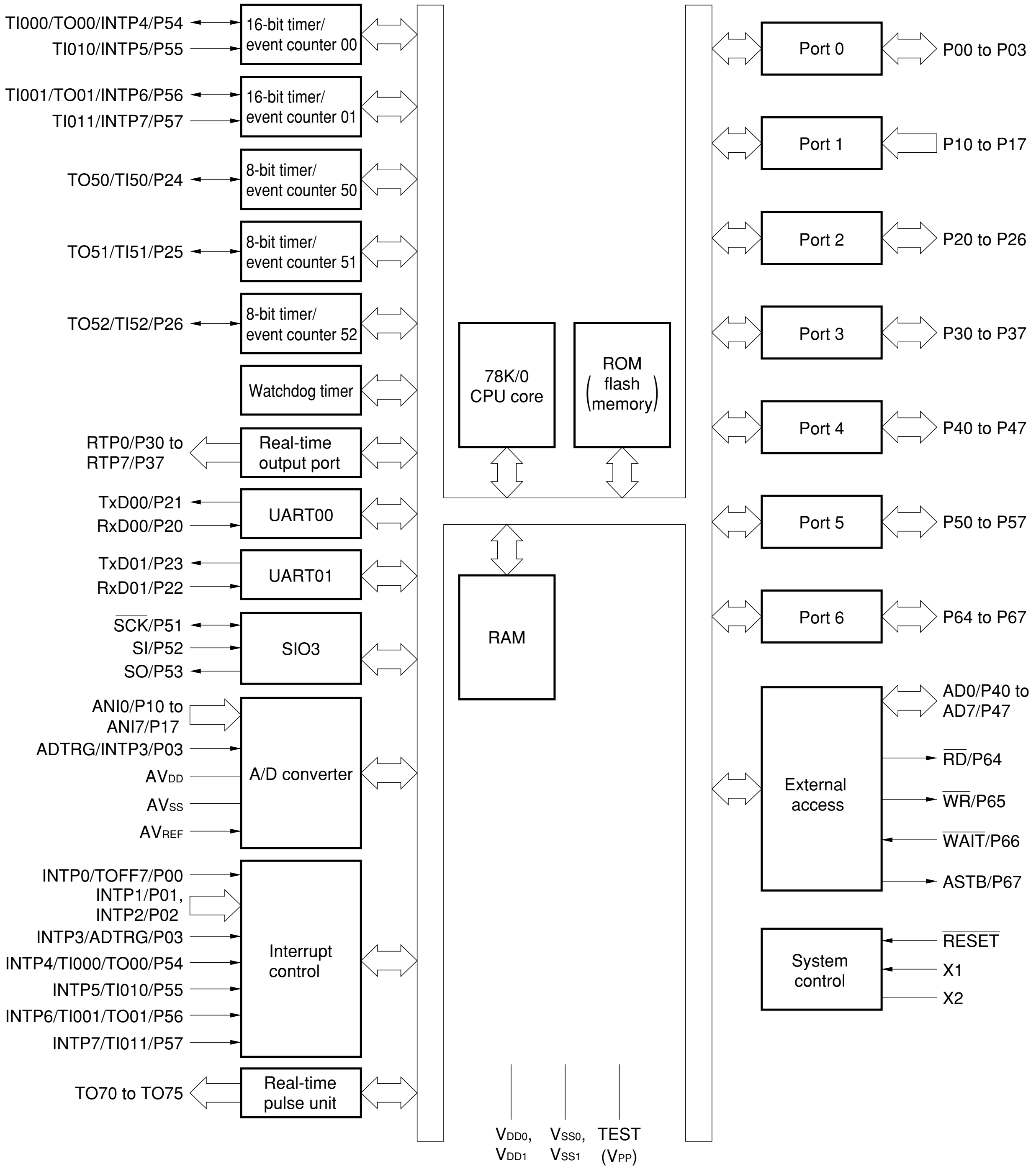
Note 16-bit timer: 2 channels
10-bit timer: 1 channel

- Y subseries

Subseries Name	Function	ROM Capacity (Bytes)	Timer				8-Bit	10-Bit	8-Bit	Serial Interface	I/O	V_{DD} MIN. Value	External
			8-Bit	16-Bit	Watch	WDT	A/D	A/D	D/A				Expansion
Control	μPD78078Y	48 K to 60 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch, I^{2}C: 1 ch)	88	1.8 V	√
	μPD78070AY	–									61	2.7 V	
	μPD780018AY	48 K to 60 K							–	3 ch (I^{2}C: 1 ch)	88		
	μPD780058Y	24 K to 60 K	2 ch						2 ch	3 ch (time-division UART: 1 ch, I^{2}C: 1 ch)	68	1.8 V	
	μPD78058FY	48 K to 60 K								3 ch (UART: 1 ch, I^{2}C: 1 ch)	69	2.7 V	
	μPD78054Y	16 K to 60 K										2.0 V	
	μPD780078Y	48 K to 60 K		2 ch			–	8 ch	–	4 ch (UART: 2 ch, I^{2}C: 1 ch)	52	1.8 V	
	μPD780034AY	8 K to 32 K		1 ch						3 ch (UART: 1 ch, I^{2}C: 1 ch)	51		
	μPD780024AY						8 ch	–					
	μPD78018FY	8 K to 60 K								2 ch (I^{2}C: 1 ch)	53		
LCD drive	μPD780354Y	24 K to 32 K	4 ch	1 ch	1 ch	1 ch	–	8 ch	–	4 ch (UART: 1 ch, I^{2}C: 1 ch)	66	1.8 V	–
	μPD780344Y						8 ch	–					
	μPD780308Y	48 K to 60 K	2 ch							3 ch (time-division UART: 1 ch, I^{2}C: 1 ch)	57	2.0 V	
	μPD78064Y	16 K to 32 K								2 ch (UART: 1 ch, I^{2}C: 1 ch)			
Bus interface supported	μPD780701Y	60 K	3 ch	2 ch	1 ch	1 ch	16 ch	–	–	4 ch (UART: 1 ch, I^{2}C: 1 ch)	67	3.5 V	–
	μPD780703Y												
	μPD780833Y										65	4.5 V	

Remark Functions other than the serial interface are common to both the Y and non-Y subseries.

1.7 Block Diagram



Remarks 1. The internal ROM and RAM capacities differ depending on the product.

2. The pin connection in parentheses is for the μPD78F0988A.

1.8 Functional Outline

Item		Part Number μPD780982	μPD780983	μPD780984	μPD780986	μPD780988	μPD78F0988A
Internal memory	ROM	Mask ROM					Flash memory
		16 KB	24 KB	32 KB	48 KB	60 KB	60 KB[Note 1]
	High-speed RAM	1024 bytes					
	Expansion RAM	None			1024 bytes		1024 bytes[Note 2]
Memory space		64 KB					
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)					
★ Minimum instruction execution time		On-chip minimum instruction execution time variable function • Expanded-specification products 0.166 μs/0.33 μs/0.66 μs/1.3 μs/2.6 μs (@ 12 MHz operation with system clock, V_{DD} = 4.5 to 5.5 V) 0.238 μs/0.48 μs/0.96 μs/1.9 μs/3.8 μs (@ 8.38 MHz operation with system clock) • Conventional products 0.238 μs/0.48 μs/0.96 μs/1.9 μs/3.8 μs (@ 8.38 MHz operation with system clock)					
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjust, etc.					
I/O ports		Total: 47 • CMOS inputs: 8 • CMOS I/O: 39					
Real-time output ports		• 8 bits × 1 or 4 bits × 2 • 6 bits × 1 or 4 bits × 1					
A/D converter		• 10-bit resolution × 8 channels					
Serial interface		• UART mode: 2 channels • 3-wire serial I/O mode: 1 channel					
Timer		• 16-bit timer/event counter: 2 channels • 8-bit timer/event counter: 3 channels • 10-bit inverter control timer: 1 channel • Watchdog timer: 1 channel					
Timer outputs		11 (General-purpose outputs: 5, inverter control outputs: 6)					
Vectored interrupt sources	Maskable	Internal: 16, external: 8					
	Non-maskable	Internal: 1					
	Software	1					
★ Power supply voltage		• V_{DD} = 3.0 to 5.5 V (expanded-specification products) • V_{DD} = 4.0 to 5.5 V (conventional products)					
Operating ambient temperature		T_A = –40 to +85°C					
Package		• 64-pin plastic SDIP (19.05 mm (750))[Note 3] • 64-pin plastic QFP (14 x 14) ★ • 64-pin plastic LQFP (14 x 14)					

Notes 1. The capacity of the flash memory can be changed using the internal memory size select register (IMS).

2. The capacity of the internal expansion RAM can be changed using the internal expansion RAM size select register (IXS).

3. Standard quality grade products only.

The table below shows the outline of timer/event counters (for details, refer to **CHAPTER 6 16-BIT TIMER/EVENT COUNTER, CHAPTER 7 8-BIT TIMER/EVENT COUNTER, CHAPTER 8 10-BIT INVERTER CONTROL TIMER, CHAPTER 9 WATCHDOG TIMER**).

		16-Bit Timer/ Event Counter	8-Bit Timer/ Event Counter	10-Bit Inverter Control Timer	Watchdog Timer
Operation Mode	Interval timer	2 channels	3 channels	1 channel	1 channel[Note]
	External event counter	√	√	–	–
Function	Timer output	√	√	√	–
	PWM output	–	√	–	–
	PPG output	√	–	–	–
	Pulse width measurement	√	–	–	–
	Square-wave output	√	√	–	–
	Interrupt request	√	√	√	√

Note The watchdog timer can perform either the watchdog timer function or the interval timer function.

1.9 Differences Between Standard Quality Grade Products and (A) Products

The differences between standard grade products (μPD780982, 780983, 780984, 780986, 780988, 78F0988A) and (A) products (μPD780982(A), 780983(A), 780984(A), 780986(A), 780988(A), 78F0988A(A)) are shown in Table 1-2.

Table 1-2. Differences Between Standard Quality Grade Products and (A) Products

Part Number Item	Standard Products	(A) Products
Quality grade	Standard	Special
★ Package	• 64-pin plastic SDIP (19.05 mm (750)) • 64-pin plastic QFP (14 x 14) • 64-pin plastic LQFP (14 x 14)	• 64-pin plastic QFP (14 x 14) • 64-pin plastic LQFP (14 x 14)

1.10 Differences Between Flash Memory Products μPD78F0988A and μPD78F0988

Table 1-3 shows the differences between the μPD78F0988A and μPD78F0988 (old product).

Table 1-3. Differences Between μPD78F0988A and μPD78F0988

Part Number Item	μPD78F0988A	μPD78F0988 (Old Product)
Flash memory area	2 areas 0: 0 to 1FFFH 1: 2000H to EFFFH	3 areas 0: 0 to 1FFFH 1: 2000H to 7FFFH 2: 8000H to EFFFH
Quality grade	• Standard • Special 64-pin plastic QFP (14 x 14)	Standard

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0 4-bit I/O port Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP0/TOFF7
P01				INTP1
P02				INTP2
P03				INTP3/ADTRG
P10 to P17	Input	Port 1 8-bit input only port	Input	ANI0 to ANI7
P20	I/O	Port 2 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	RxD00
P21				TxD00
P22				RxD01
P23				TxD01
P24				TI50/TO50
P25				TI51/TO51
P26				TI52/TO52
P30 to P37	I/O	Port 3 8-bit I/O port Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	RTP0 to RTP7
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	I/O	Port 5 8-bit I/O port Input/output can be specified in 1-bit units. LEDs can be driven directly. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	—
P51				$\overline{\text{SCK}}$
P52				SI
P53				SO
P54				INTP4/TI000/TO00
P55				INTP5/TI010
P56				INTP6/TI001/TO01
P57				INTP7/TI011
P64	I/O	Port 6 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	$\overline{\text{RD}}$
P65				$\overline{\text{WR}}$
P66				$\overline{\text{WAIT}}$
P67				ASTB

(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	P00/TOFF7
INTP1			Input	P01
INTP2			Input	P02
INTP3			Input	P03/ADTRG
INTP4			Input	P54/TI000/TO00
INTP5			Input	P55/TI010
INTP6			Input	P56/TI001/TO01
INTP7			Input	P57/TI011
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P24/TO50
TI51		External count clock input to 8-bit timer/event counter 51	Input	P25/TO51
TI52		External count clock input to 8-bit timer/event counter 52	Input	P26/TO52
TI000		External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture register (CR000, CR010) in 16-bit timer/event counter 00	Input	P54/INTP4/TO00
TI010		Capture trigger input to capture register (CR000) in 16-bit timer /event counter 00	Input	P55/INTP5
TI001		External count clock input to 16-bit timer/event counter 01 Capture trigger input to capture register (CR001, CR011) in 16-bit timer/event counter 01	Input	P56/INTP6/TO01
TI011		Capture trigger input to capture register (CR001) in 16-bit timer /event counter 01	Input	P57/INTP7
TO50	Output	8-bit timer/event counter 50 output	Input	P24/TI50
TO51		8-bit timer/event counter 51 output	Input	P25/TI51
TO52		8-bit timer/event counter 52 output	Input	P26/TI52
TO00		16-bit timer/event counter 00 output	Input	P54/INTP4/TI000
TO01		16-bit timer/event counter 01 output	Input	P56/INTP6/TI001
RTP0 to RTP7	Output	Real-time output port that outputs pulses in synchronization with trigger signals output from the real-time pulse unit	Input	P30 to P37
TxD00	Output	Asynchronous serial interface serial data output	Input	P21
TxD01			Input	P23
RxD00	Input	Asynchronous serial interface serial data input	Input	P20
RxD01			Input	P22
$\overline{\text{SCK}}$	I/O	Serial interface serial clock input/output	Input	P51
SI	Input	Serial interface serial data input	Input	P52
SO	Output	Serial interface serial data output	Input	P53
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ADTRG	Input	External trigger signal input to the A/D converter	Input	P03/INTP3
TO70 to TO75	Output	Timer output for the 3-phase PWM inverter control	Hi-Z	–
TOFF7	Input	External input to stop timer output (TO70 to TO75)	Input	P00/INTP0
AD0 to AD7	I/O	Address/data bus for when memory is expanded externally	Input	P40 to P47
$\overline{\text{RD}}$	Output	Strobe signal output for external memory read operation	Input	P64
$\overline{\text{WR}}$		Strobe signal output for external memory write operation	Input	P65
$\overline{\text{WAIT}}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output that externally latches address information output to ports 4 and 5 to access external memory	Input	P67

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

Pin Name	I/O	Function	After Reset	Alternate Function
AV_{REF}	Input	A/D converter reference voltage input	–	–
AV_{DD}	–	A/D converter analog power supply	–	–
AV_{SS}	–	A/D converter ground potential	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
X1	Input	Connection of crystal for system clock oscillation	–	–
X2	–		–	–
V_{DD0}	–	Positive power supply for ports	–	–
V_{SS0}	–	Ground potential for ports	–	–
V_{DD1}	–	Positive power supply (except for ports)	–	–
V_{SS1}	–	Ground potential (except for ports)	–	–
TEST	–	Test mode set pin. Connect to V_{SS0} directly	–	–
V_{PP} Note	–	High-voltage application for program write/verify. Directly connect this pin to V_{SS0} in normal operation mode.	–	–

Note μPD78F0988A only

2.2 Description of Pin Functions

2.2.1 P00 to P03 (Port 0)

These pins constitute a 4-bit I/O port, port 0. In addition, these pins are also used to input external interrupt request signals, a timer output stop external signal and an external trigger signal of the A/D converter.

Port 0 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P00 to P03 function as a 4-bit I/O port which can be set to input or output in 1-bit units by using port mode register 0 (PM0). An internal pull-up resistor can be used if so specified by pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, P00 to P03 are used to input external interrupt requests, a timer output stop external signal, and an external trigger signal of the A/D converter.

(a) INTP0 to INTP3

These pins are external interrupt request input pins for which the valid edge can be specified (rising edge, falling edge, and both rising and falling edges). INTP2 also functions as an external trigger signal input pin of the real-time output port when a valid edge is input.

(b) TOFF7

External input pin to stop timer output (TO70 to TO75).

(c) ADTRG

External trigger signal input pin of the A/D converter.

2.2.2 P10 to P17 (Port 1)

These pins constitute an 8-bit input port, port 1. In addition to the general-purpose input port function, these pins also serve as the analog input pins of the A/D converter.

(1) Port mode

In this mode, P10 to P17 function as an 8-bit input port.

(2) Control mode

In this mode, P10 to P17 function as the analog input pins (ANI0 to ANI7) of the A/D converter.

2.2.3 P20 to P26 (Port 2)

These pins constitute a 7-bit I/O port, port 2. In addition, these pins are also used as the serial interface I/O pins, and the timer I/O pins.

Port 2 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P20 to P26 function as a 7-bit I/O port which can be set to input or output in 1-bit units by using port mode register 2 (PM2). An internal pull-up resistor can be used if so specified by pull-up resistor option register 2 (PU2).

(2) Control mode

In this mode, P20 to P26 function as the serial interface I/O pins, and timer I/O pins.

(a) RxD00, RxD01, TxD00, TxD01

Serial data I/O pins of the serial interface.

(b) TI50 to TI52

External count clock input pins of 8-bit timer/event counters 50 to 52.

(c) TO50 to TO52

Output pins of 8-bit timer/event counters 50 to 52.

2.2.4 P30 to P37 (Port 3)

These pins constitute an 8-bit I/O port, port 3. In addition, they also function as a real-time output port.

Port 3 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P30 to P37 function as an 8-bit I/O port which can be set to input or output in 1-bit units by using port mode register 3 (PM3). An internal pull-up resistor can be used if so specified by pull-up resistor option register 3 (PU3).

(2) Control mode

In this mode, P30 to P37 are used as a real-time output port (RTP0 to RTP7) that outputs data in synchronization with a trigger.

2.2.5 P40 to P47 (Port 4)

These pins constitute an 8-bit I/O port, port 4. In addition, they also function as an address/data bus.

Port 4 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P40 to P47 function as an 8-bit I/O port which can be set to input or output in 1-bit units by using port mode register 4 (PM4). An internal pull-up resistor can be used if so specified by pull-up resistor option register 4 (PU4).

(2) Control mode

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

2.2.6 P50 to P57 (Port 5)

These pins constitute an 8-bit I/O port, port 5. In addition, these pins also function as the serial interface clock and I/O, data I/O, timer I/O, and external interrupt request input pins.

These pins can directly drive LEDs.

Port 5 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P50 to P57 function as an 8-bit I/O port which can be set to input or output in 1-bit units by using port mode register 5 (PM5). An internal pull-up resistor can be used if so specified by pull-up resistor option register 5 (PU5).

(2) Control mode

In this mode, P50 to P57 function as the serial interface clock and data I/O, timer I/O and external interrupt request input pins.

(a) $\overline{\text{SCK}}$

Serial interface's serial clock I/O pins.

(b) SI, SO

Serial interface's serial data I/O pins.

(c) TI000, TI001

The pin that inputs the external counter clock to 16-bit timer/event counters 00 and 01 and the pin that inputs the capture trigger signal to the capture register of 16-bit timer/event counters 00 and 01.

(d) TI010 and TI011

The pins that input the capture trigger signal to the capture register of 16-bit timer/event counters 00 and 01.

(e) TO00 and TO01

Output pins of 16-bit timer/event counters 00 and 01.

(f) INTP4 to INTP7

External interrupt request input pins for which valid edges (rising edge, falling edge, and both rising and falling edges) can be specified.

2.2.7 P64 to P67 (Port 6)

These pins constitute a 4-bit I/O port, port 6, which can also be used to output control signals in the external memory expansion mode.

Port 6 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P64 to P67 function as a 4-bit I/O port which can be set to input or output in 1-bit units by using port mode register 6 (PM6). An internal pull-up resistor can be used if so specified by pull-up resistor option register 6 (PU6).

(2) Control mode

In this mode, P64 to P67 function as control signal output pins ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, and ASTB) in the external memory expansion mode. The pins used as control signal output pins are automatically disconnected from internal pull-up resistors.

Caution If the external wait state is not used in the external memory expansion mode, P66 can be used as an I/O port pin.

2.2.8 TO70 to TO75

These are the timer output pins for 3-phase PWM inverter control.

2.2.9 AV$_{REF}$

This pin inputs a reference voltage to the A/D converter.

Connect this pin to V$_{SS0}$ when the A/D converter is not used.

2.2.10 AV$_{DD}$

This is the analog power supply pin of the A/D converter.

Keep this pin at the same voltage as the V$_{DD0}$ pin even when the A/D converter is not used.

2.2.11 AV$_{SS}$

This is the ground pin of the A/D converter.

Keep this pin at the same voltage as the V$_{SS0}$ pin even when the A/D converter is not used.

2.2.12 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.2.13 X1 and X2

These pins are used to connect a crystal resonator for system clock oscillation.

To supply an external clock, input the clock to X1 and input the inverted signal to X2.

2.2.14 V$_{DD0}$ and V$_{DD1}$

V$_{DD0}$ is the positive power supply pin for ports.

V$_{DD1}$ is the positive power supply pin for blocks other than ports.

2.2.15 V$_{SS0}$ and V$_{SS1}$

V$_{SS0}$ is the ground pin for ports.

V$_{SS1}$ is the ground pin for blocks other than ports.

2.2.16 V$_{PP}$ (μPD78F0988A only)

A high voltage should be applied to this pin when the program is written or verified.

Directly connect this pin to V$_{SS0}$ in the normal operation mode.

2.2.17 TEST (Mask ROM version only)

This pin is used for IC testing. Connect directly to V$_{SS0}$.

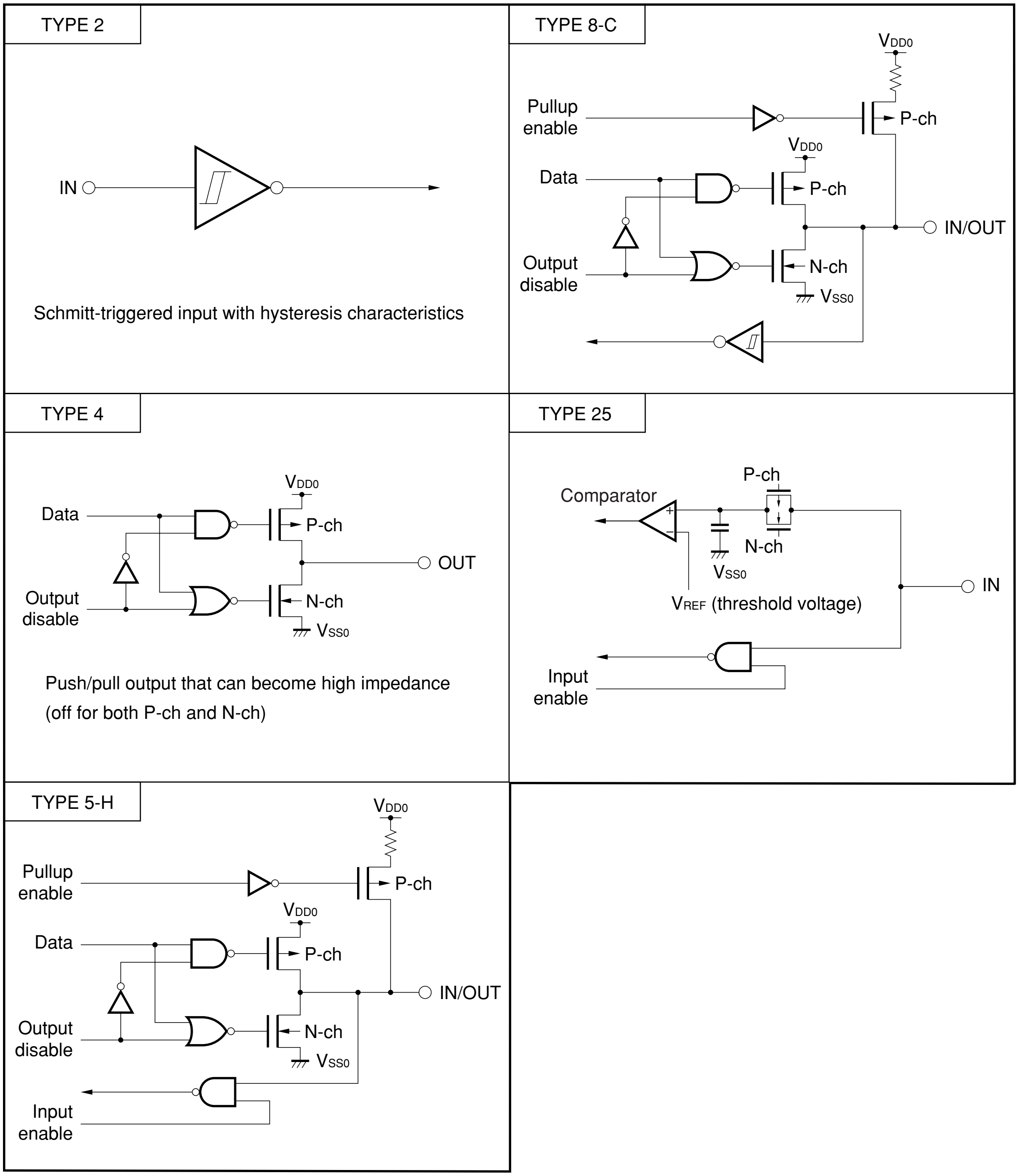
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

The I/O circuit type of each pin and recommended connections of unused pins are shown in Table 2-1.
For each I/O circuit configuration, refer to Figure 2-1.

Table 2-1. Types of Pin I/O Circuits

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP0/TOFF7	8-C	I/O	Input: Independently connect to V_{SS0} via a resistor. Output: Leave open.
P01/INTP1			
P02/INTP2			
P03/INTP3/ADTRG			
P10/ANI0 to P17/ANI7	25	Input	Connect to V_{DD0} or V_{SS0}.
P20/RxD00	8-C	I/O	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P21/TxD00	5-H		
P22/RxD01	8-C		
P23/TxD01	5-H		
P24/TI50/TO50	8-C		
P25/TI51/TO51			
P26/TI52/TO52			
P30/RTP0 to P37/RTP7	5-H		
P40/AD0 to P47/AD7			
P50			
P51/$\overline{\text{SCK}}$	8-C		
P52/SI	5-H		
P53/SO			
P54/INTP4/TI000/TO00			
P55/INTP5/TI010			
P56/INTP6/TI001/TO01			
P57/INTP7/TI011			
P64/$\overline{\text{RD}}$			
P65/$\overline{\text{WR}}$			
P66/$\overline{\text{WAIT}}$			
P67/ASTB			
TO70 to TO75	4	Output	Leave open.
$\overline{\text{RESET}}$	2	—	—
AV_{DD}	—		Connect to V_{DD0}.
AV_{REF}			Connect to V_{SS0}.
AV_{SS}			
TEST (mask ROM version)			Connect directly to V_{SS0}.
V_{PP} (flash memory version)			

Figure 2-1. Pin I/O Circuits



Schmitt-triggered input with hysteresis characteristics

Push/pull output that can become high impedance
(off for both P-ch and N-ch)

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Each product in the μPD780988 Subseries can access a memory space of 64 KB. Figures 3-1 to 3-6 show the memory maps of the respective products.

Cautions 1. The initial value of the internal memory size switching register (IMS) is fixed (to CFH) for all the products in the μPD780988 Subseries, regardless of the capacity of the internal memory. Therefore, set the values shown below for each microcontroller before use.

μPD780982:	**C4H**
μPD780983:	**C6H**
μPD780984:	**C8H**
μPD780986:	**CCH**
μPD780988:	**CFH (No need to change the initial value because the μPD780988 is set to CFH.)**
μPD78F0988A:	**Value corresponding to mask ROM versions**

2. The initial value of the internal expansion RAM size switching register (IXS) is fixed (to 0CH) for all the products in the μPD780988 Subseries, regardless of the capacity of the internal expansion RAM. Therefore, set the values shown below for each microcontroller before use.

μPD780982, 780983, 780984:	**0CH (No need to change the initial value because the μPD780982, 780983, 780984 are set to 0CH).**
μPD780986, 780988:	**0AH**
μPD78F0988A:	**Value corresponding to mask ROM versions**

Figure 3-1. Memory Map (μPD780982)

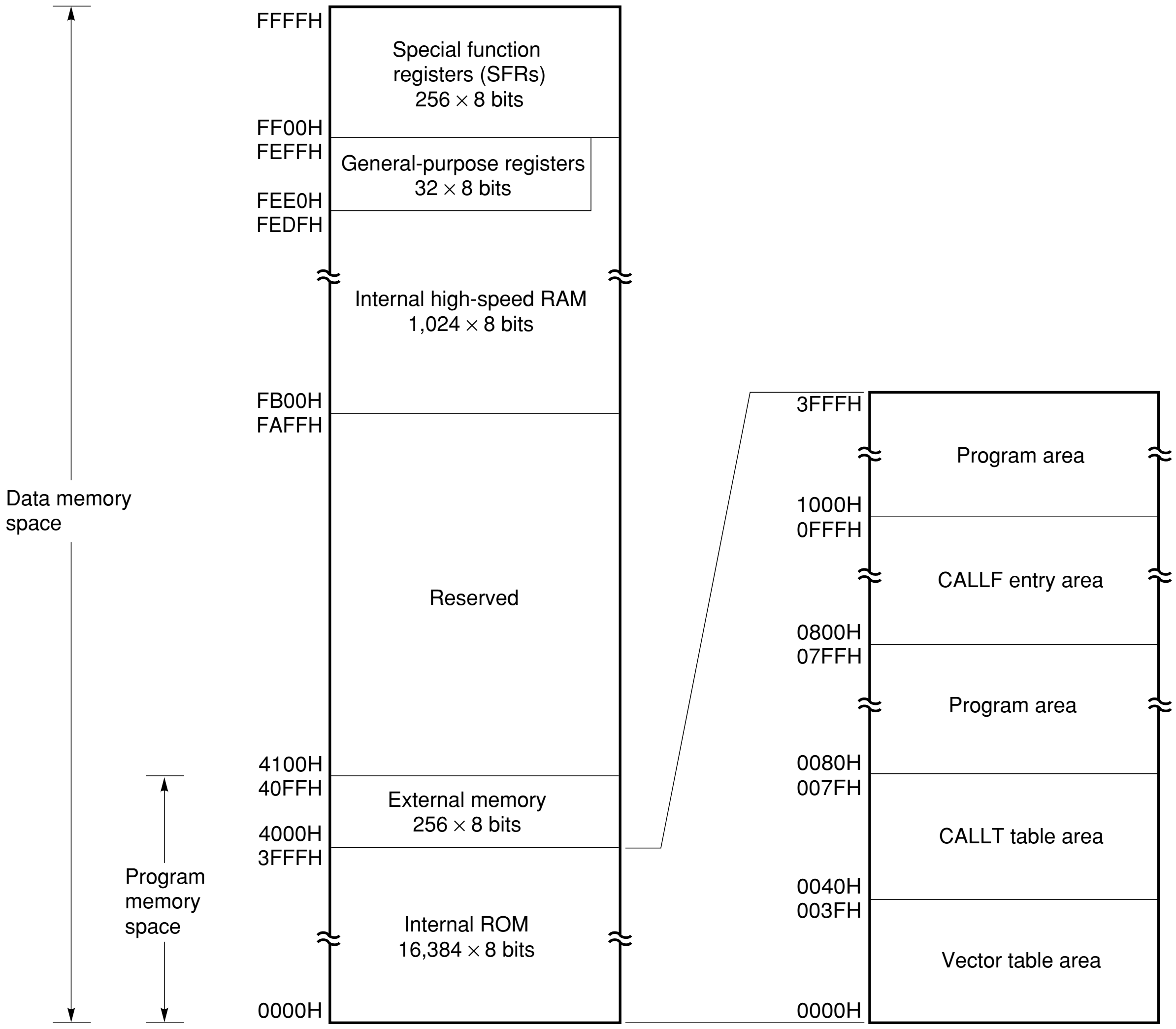


Figure 3-2. Memory Map (μPD780983)

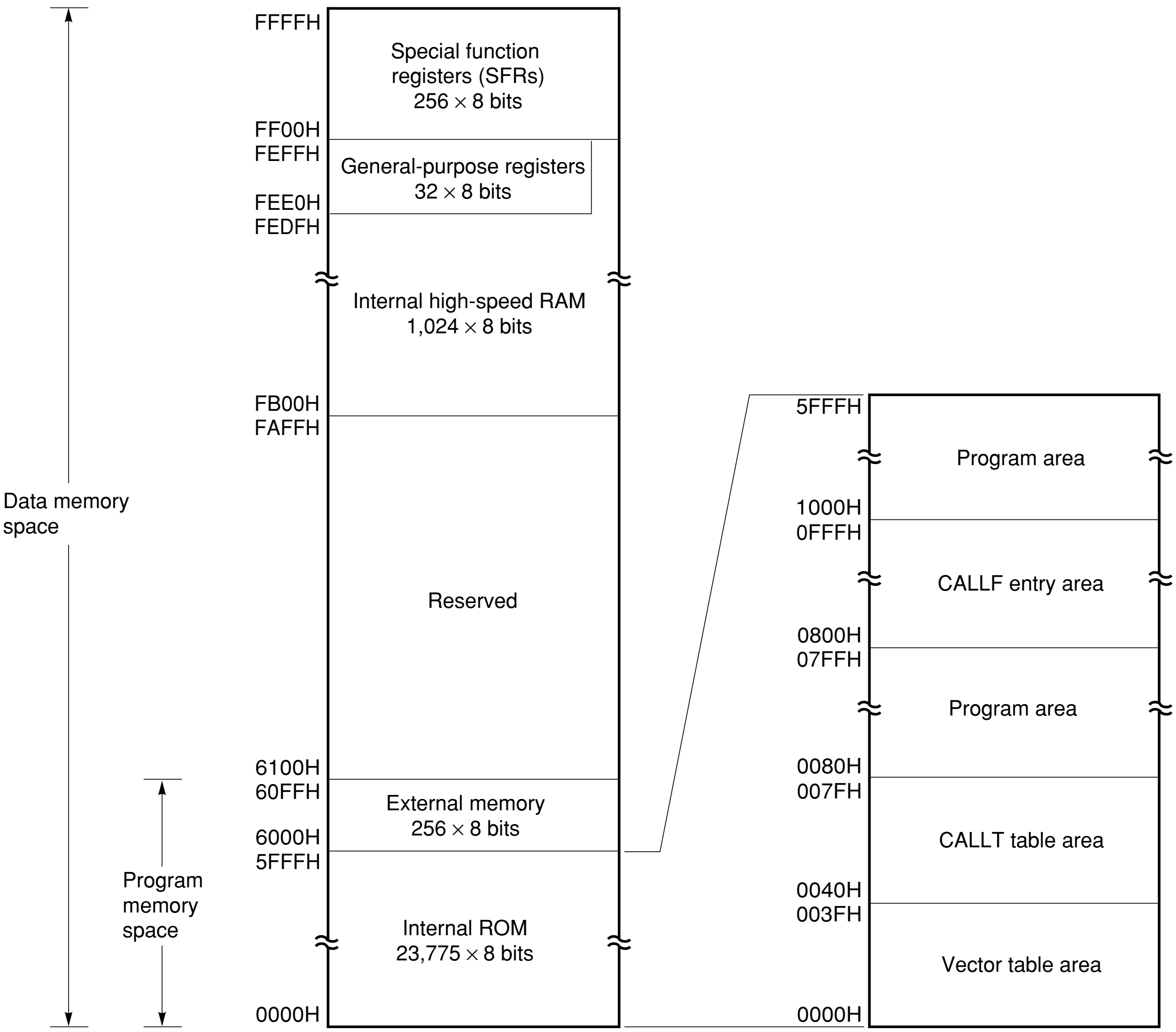


Figure 3-3. Memory Map (μPD780984)

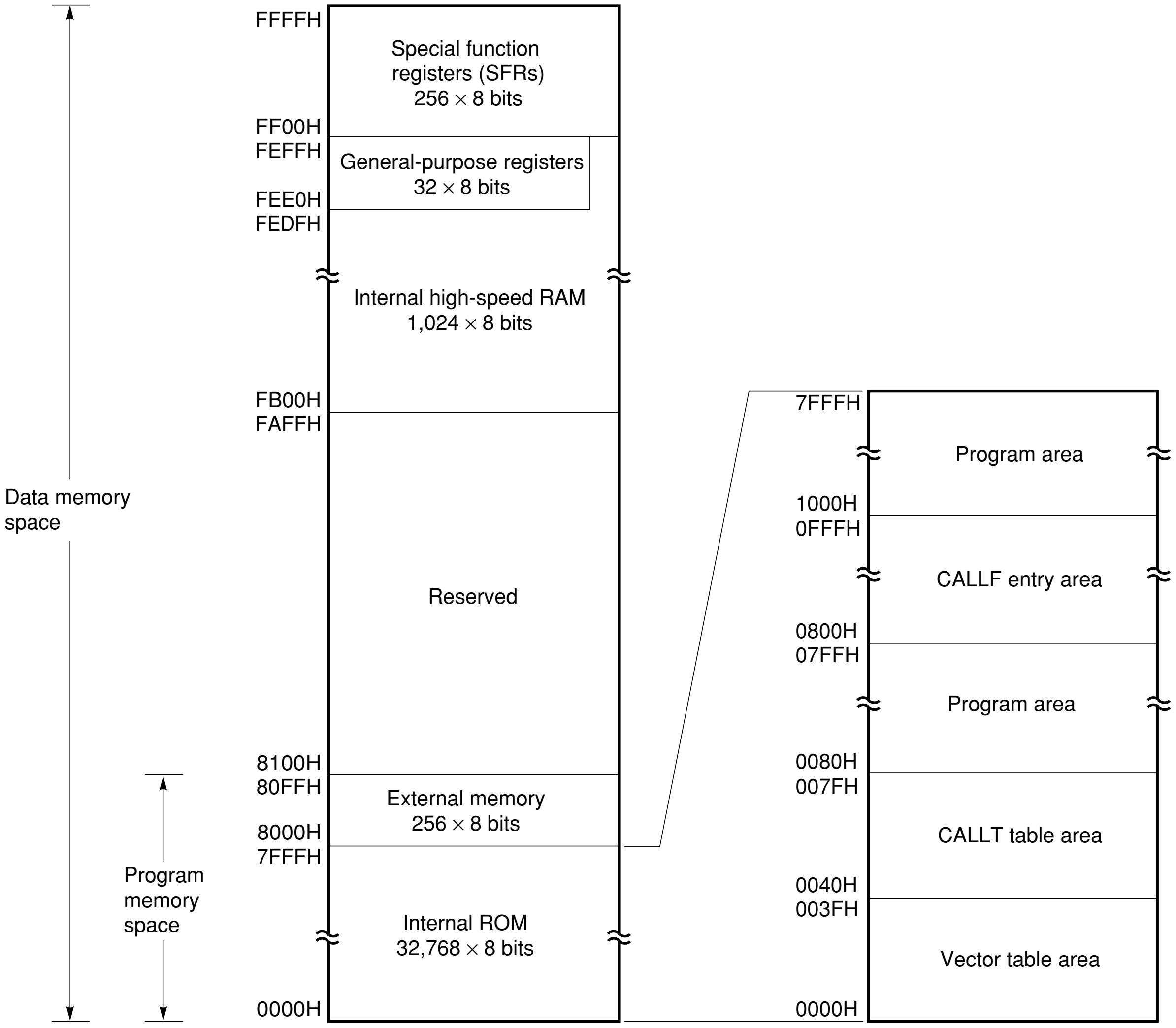


Figure 3-4. Memory Map (μPD780986)

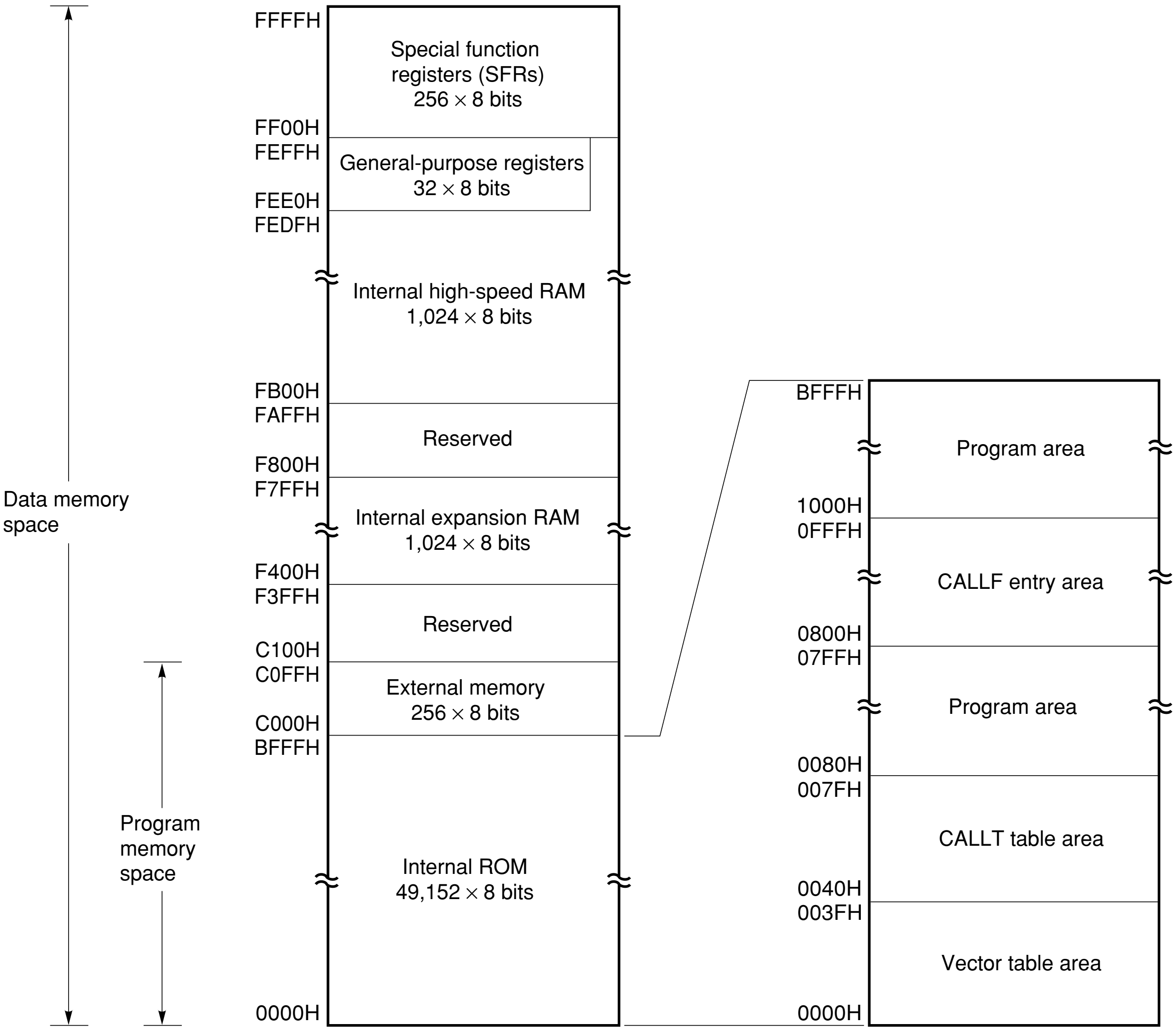


Figure 3-5. Memory Map (μPD780988)

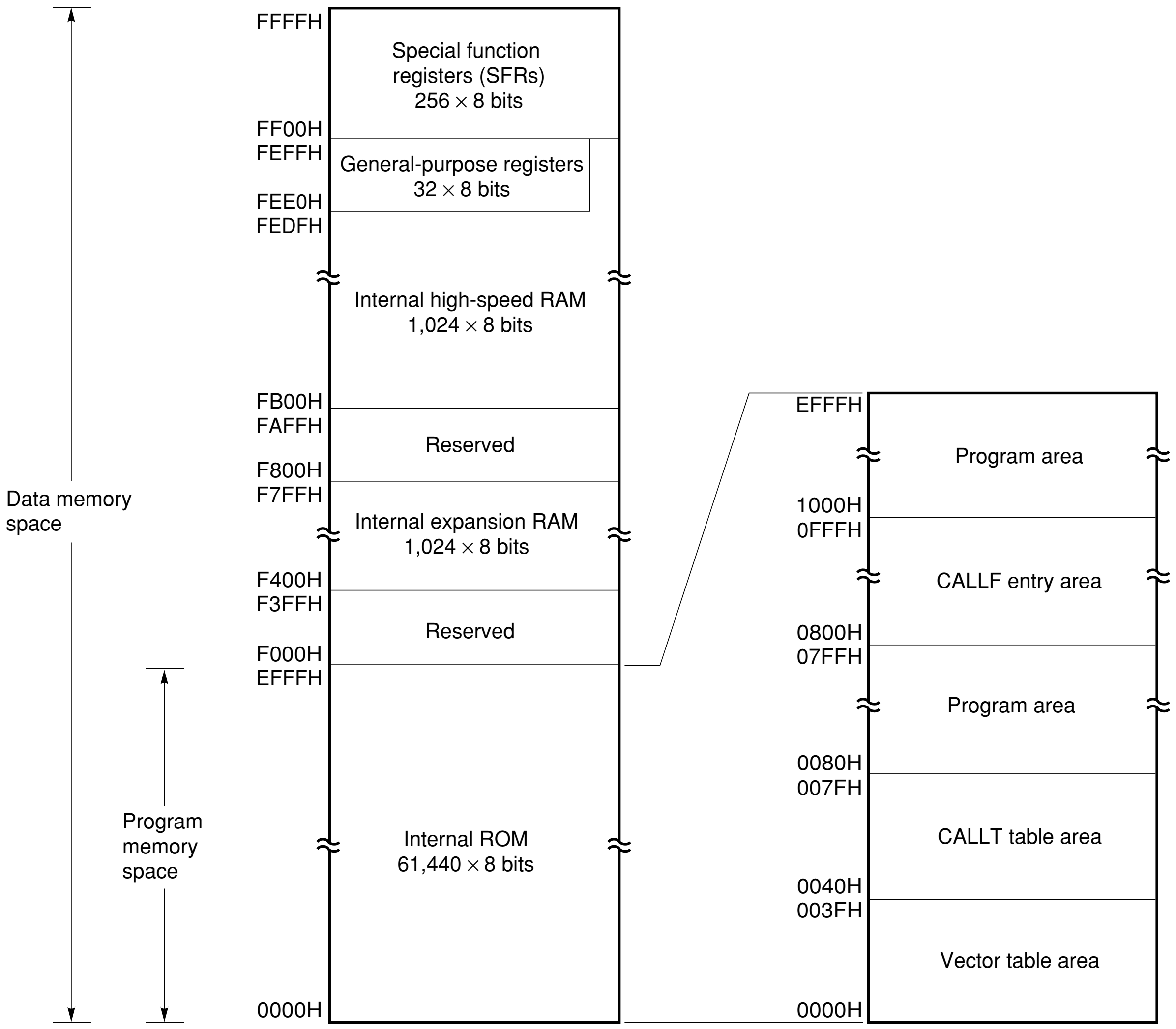
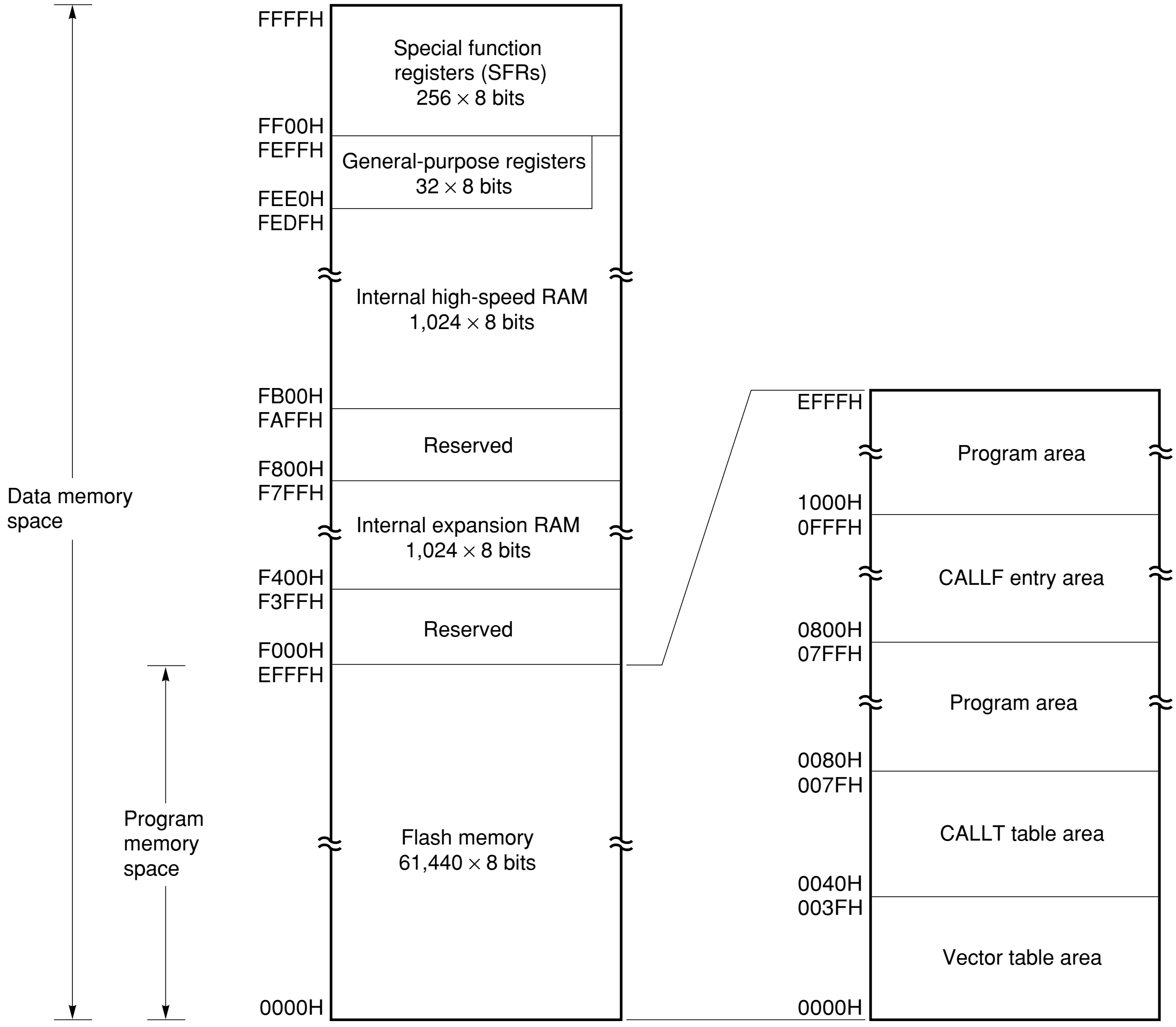


Figure 3-6. Memory Map (μPD78F0988A)



3.1.1 Internal program memory space

The internal program memory space stores programs and table data. This space is usually addressed by the program counter (PC).

Each model in the μPD780988 Subseries is provided with the following internal ROM (or flash memory).

Table 3-1. Internal ROM Capacity

Part Number	Capacity	
	Structure	
μPD780982	Mask ROM	16,384 × 8 bits (0000H to 3FFFH)
μPD780983		23,775 × 8 bits (0000H to 5FFFH)
μPD780984		32,768 × 8 bits (0000H to 7FFFH)
μPD780986		49,152 × 8 bits (0000H to BFFFH)
μPD780988		61,440 × 8 bits (0000H to EFFFH)
μPD78F0988A	Flash memory	61,440 × 8 bits (0000H to EFFFH)

The following areas are allocated to the internal program memory space.

(1) Vector table area

The 64-byte area of addresses 0000H to 003FH is reserved as a vector table area. This area stores program start addresses to which the program branches when the $\overline{\text{RESET}}$ signal is input or when an interrupt request is generated. Of a 16-bit program start address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input	001CH	INTTM001
0004H	INTWDT	001EH	INTTM011
0006H	INTP0	0020H	INTSER0
0008H	INTP1	0022H	INTSR0
000AH	INTP2	0024H	INTST0
000CH	INTP3	0026H	INTSR1
000EH	INTP4	0028H	INTST1
0010H	INTP5	002AH	INTTM50
0012H	INTP6	002CH	INTTM51
0014H	INTP7	002EH	INTTM52
0016H	INTTM7	0030H	INTCSI3
0018H	INTTM000	0032H	INTAD0
001AH	INTTM010	003EH	BRK instruction

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

A subroutine can be directly called from the area of addresses 0800H to 0FFFH by using a 2-byte call instruction (CALLF).

3.1.2 Internal data memory space

The μPD780988 Subseries are provided with the following RAM.

(1) Internal high-speed RAM

The internal expansion RAM is allocated to the 1024-byte area FB00H to FEFFH. Four of these banks of general-purpose registers with eight 8-bit registers per bank are allocated to the 32-byte area FEE0H to FEFFH.

★ 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

In the μPD780986, 780988, and 78F0988A only, the internal expansion RAM is allocated to the 1024-byte area F400H to F7FFH.

★ 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

Special function registers (SFRs) of on-chip peripheral hardware are allocated to the area of FF00H to FFFFH (refer to **Table 3-4 Special Function Register List** in **3.2.3 Special function registers (SFRs)**).

Caution Do not access an address to which no SFR is allocated.

3.1.4 External memory space

This is an external memory space that can be accessed by setting the memory expansion mode register (MEM). This space can store programs and table data, and can be assigned to peripheral devices.

3.1.5 Data memory addressing

The manner of specifying the address of the instruction to be executed next or specifying the address of a register or memory to be manipulated when an instruction is executed is called addressing.

The address of the instruction to be executed next is specified by the program counter (PC) (for details, refer to **3.3 Instruction Address Addressing**).

However, in consideration of operability, the μPD780988 Subseries is equipped with a wide range of addressing modes for memory addresses that are operational objects during instruction execution. Especially, in the areas to which the data memory is assigned (addresses FB00H to FFFFH), the special function registers (SFRs) and general-purpose registers can be addressed in accordance with their function. Figures 3-7 to 3-12 illustrate the addressing of the data memory.

For details of each addressing, refer to **3.4 Operand Address Addressing**.

Figure 3-7. Data Memory Addressing (μPD780982)

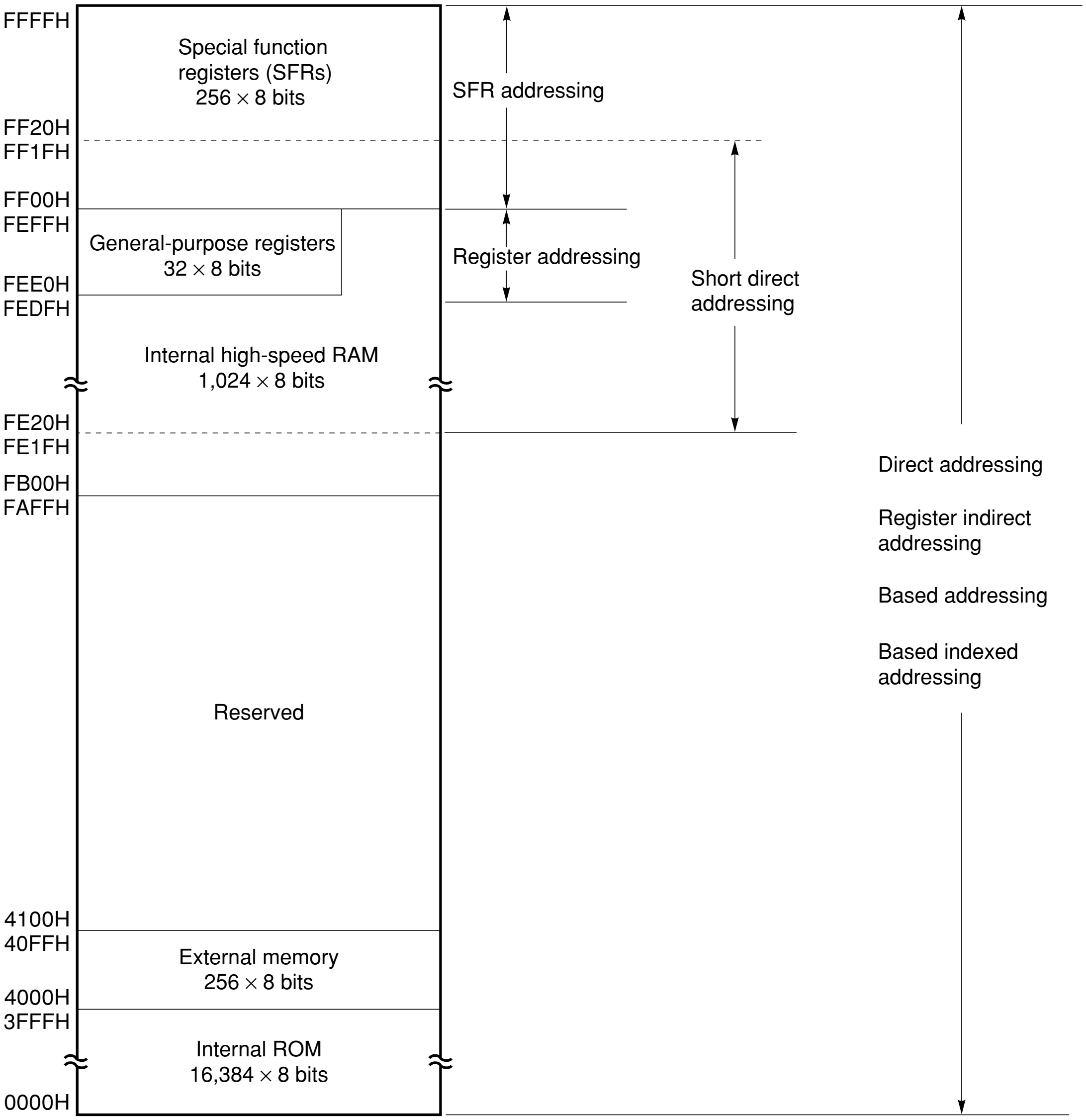

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1,024 × 8 bits
FE20H
FE1FH
FB00H
FAFFH
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing
Reserved
4100H
40FFH
External memory 256 × 8 bits
4000H
3FFFH
Internal ROM 16,384 × 8 bits
0000H

Figure 3-8. Data Memory Addressing (μPD780983)

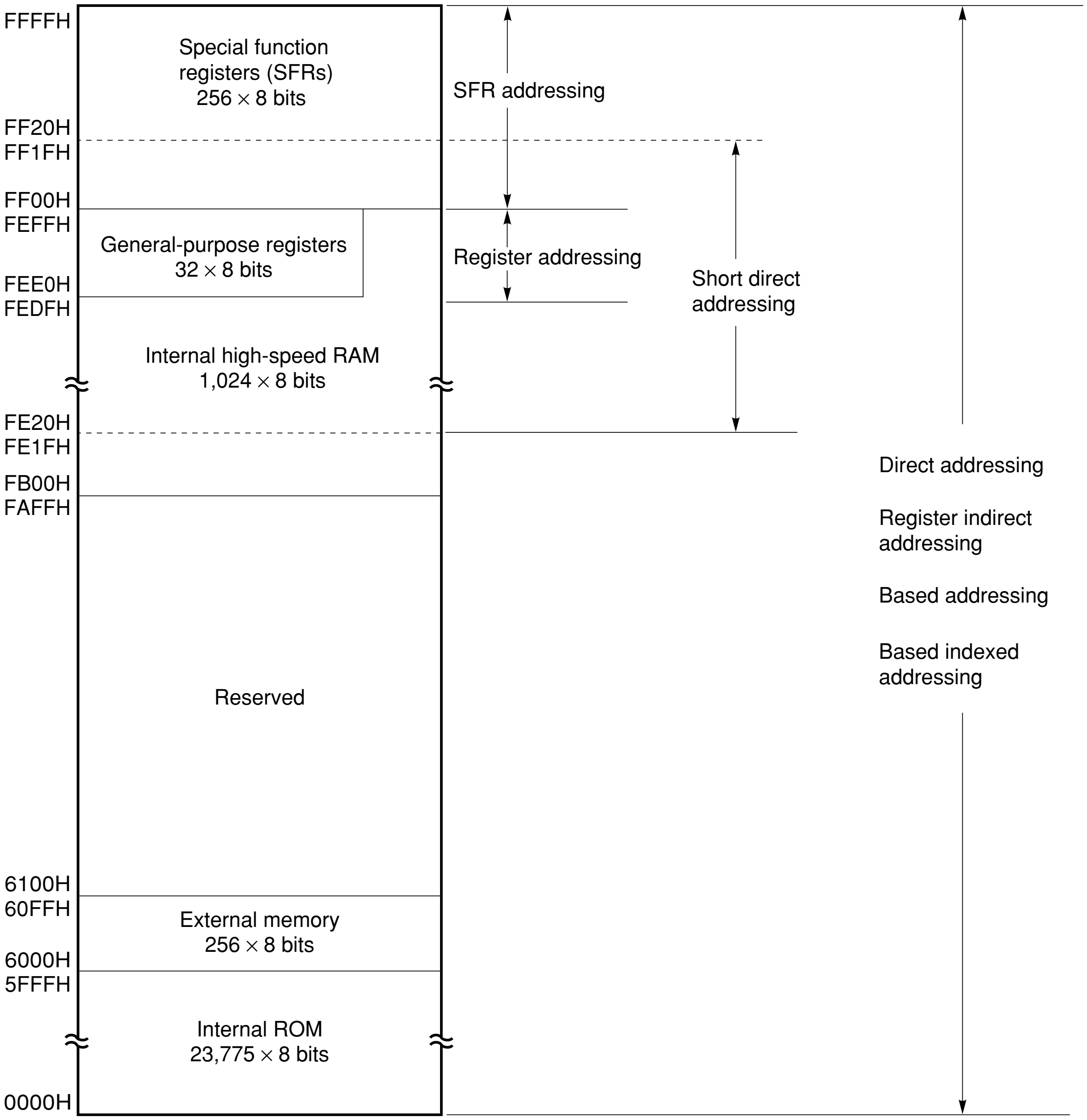

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1,024 × 8 bits
FE20H
FE1FH
FB00H
FAFFH
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing
Reserved
6100H
60FFH
External memory 256 × 8 bits
6000H
5FFFH
Internal ROM 23,775 × 8 bits
0000H

Figure 3-9. Data Memory Addressing (μPD780984)

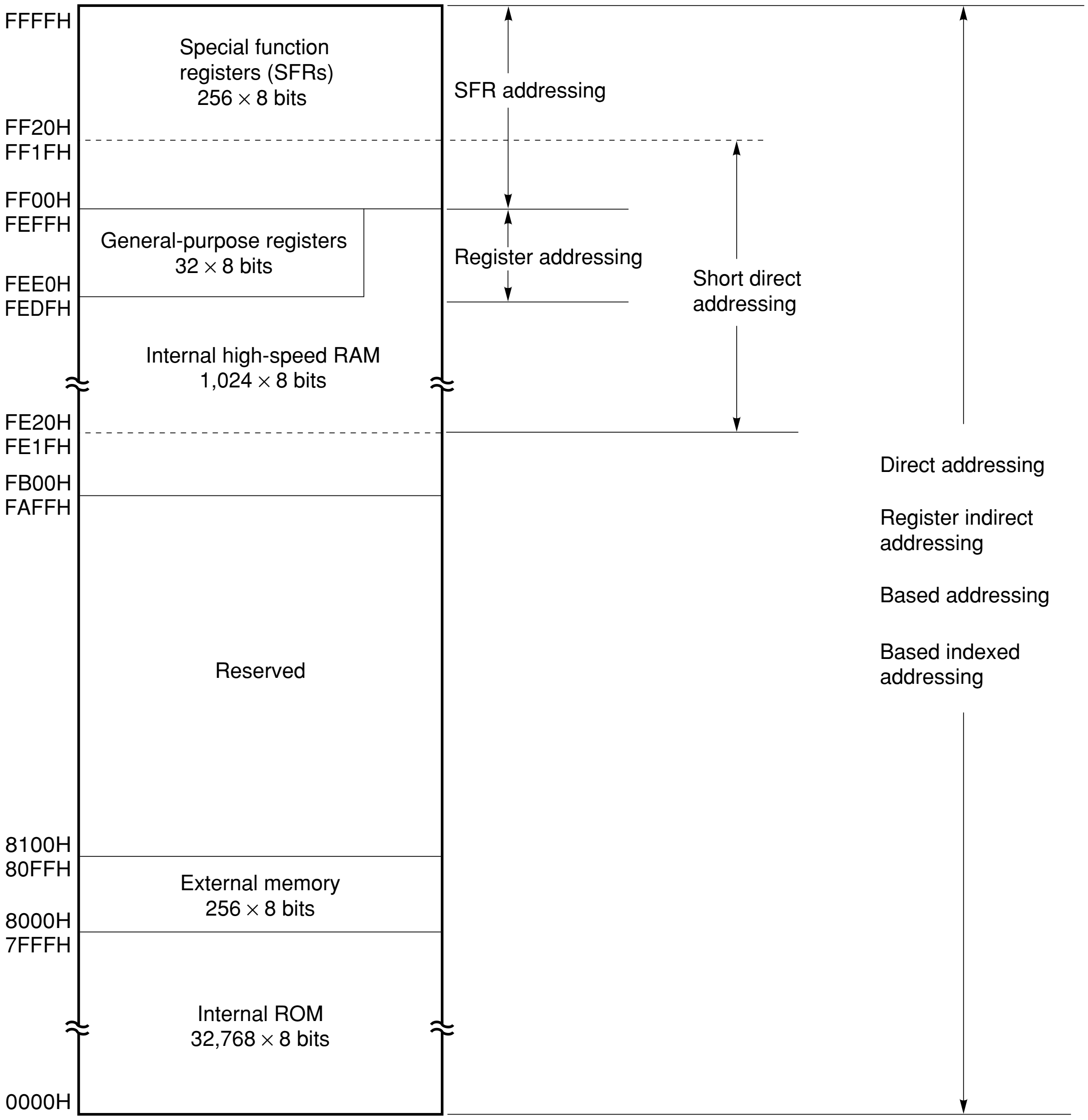

FFFFH
FF20H
FF1FH
FF00H
FEFFH
FEE0H
FEDFH
FE20H
FE1FH
FB00H
FAFFH
8100H
80FFH
8000H
7FFFH
0000H
Special function registers (SFRs) 256 × 8 bits
General-purpose registers 32 × 8 bits
Internal high-speed RAM 1,024 × 8 bits
Reserved
External memory 256 × 8 bits
Internal ROM 32,768 × 8 bits
SFR addressing
Register addressing
Short direct addressing
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing

Figure 3-10. Data Memory Addressing (μPD780986)

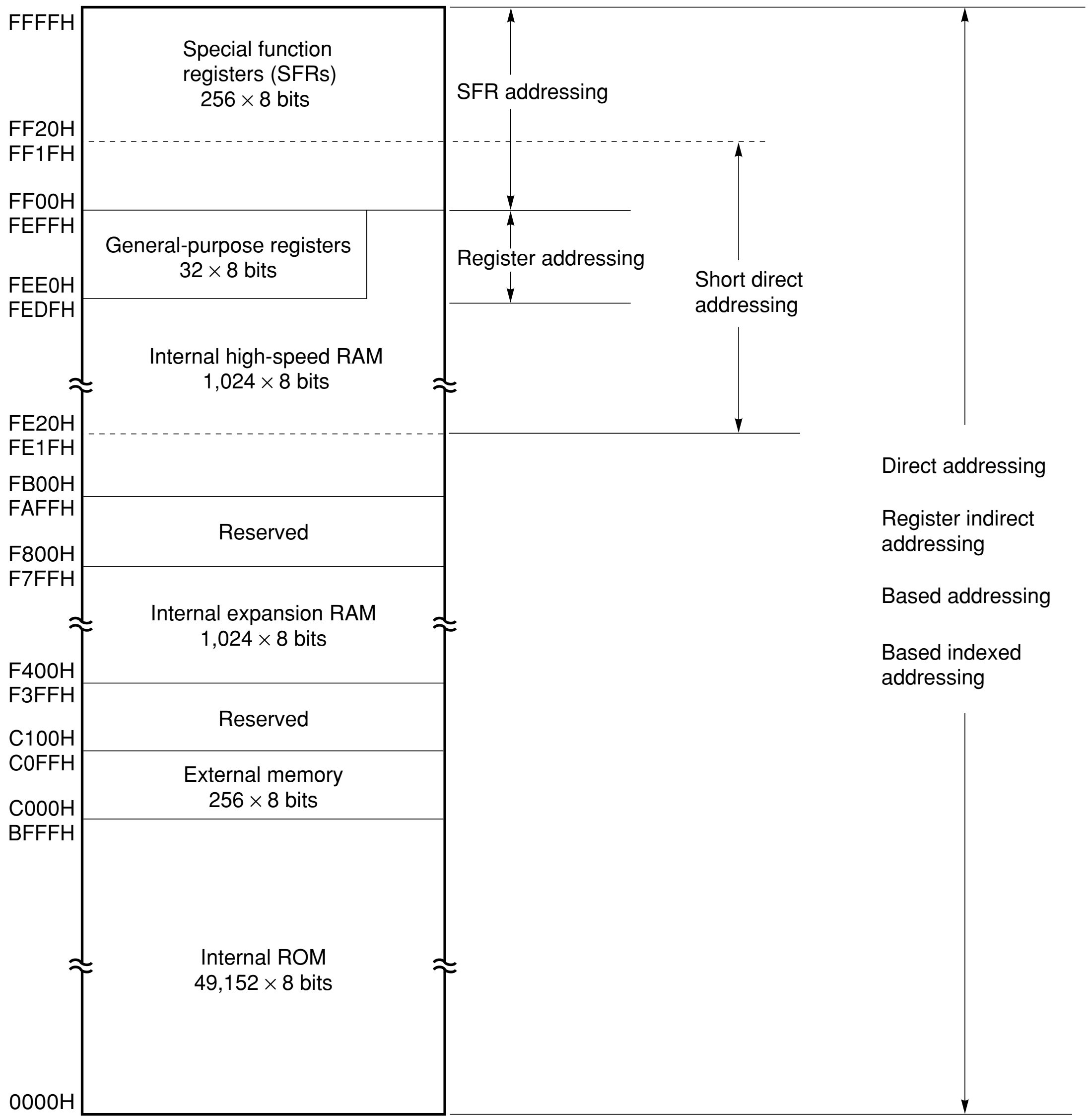

FFFFH
FF20H
FF1FH
FF00H
FEFFH
FEE0H
FEDFH
FE20H
FE1FH
FB00H
FAFFH
F800H
F7FFH
F400H
F3FFH
C100H
C0FFH
C000H
BFFFH
0000H
Special function registers (SFRs) 256 × 8 bits
General-purpose registers 32 × 8 bits
Internal high-speed RAM 1,024 × 8 bits
Reserved
Internal expansion RAM 1,024 × 8 bits
Reserved
External memory 256 × 8 bits
Internal ROM 49,152 × 8 bits
SFR addressing
Register addressing
Short direct addressing
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing

Figure 3-11. Data Memory Addressing (μPD780988)

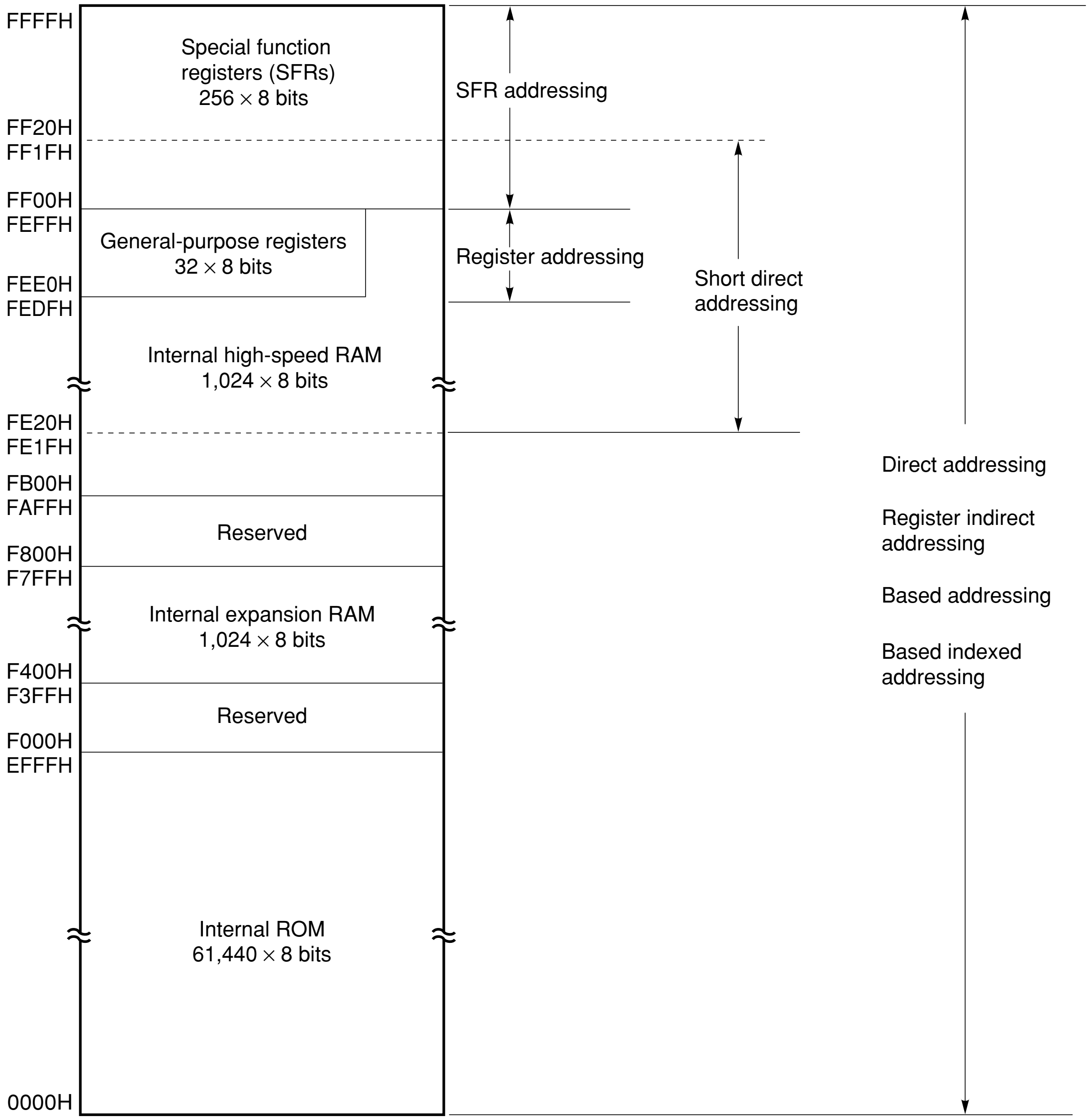
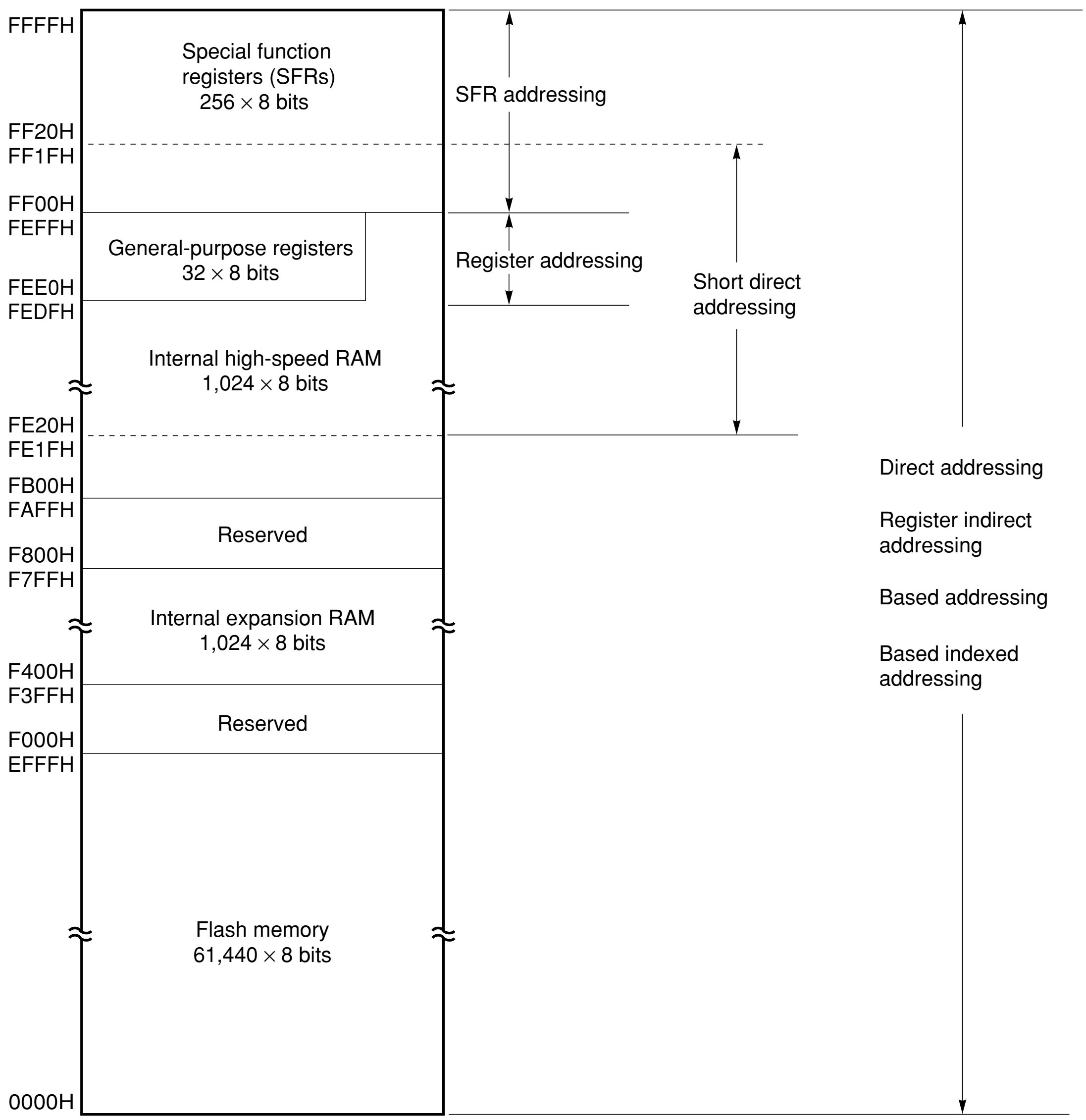


Figure 3-12. Data Memory Addressing (μPD78F0988A)



3.2 Processor Registers

The μPD780988 Subseries is provided with the following processor registers.

3.2.1 Control registers

Each of these registers has a dedicated function such as to control the program sequence, status, and stack memory. The control registers include the program counter (PC), program status word (PSW), and stack pointer (SP).

(1) Program counter (PC)

The program counter is a 16-bit register that holds the address of the program to be executed next.

The contents of this register are automatically incremented according to the number of bytes of the instruction to be fetched when a normal operation is performed. When a branch instruction is executed, immediate data or the contents of a register are set to the program counter.

When the $\overline{\text{RESET}}$ signal is input, the value of the reset vector table at addresses 0000H and 0001H is set to the program counter.

Figure 3-13. Program Counter Configuration

	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 flags that are set or reset as a result of instruction execution.

The contents of the program status word are automatically pushed to the stack when an interrupt request is generated or when the PUSH PSW instruction is executed, and are automatically popped from the stack when the RETB, RETI, or POP PSW instruction is executed.

The contents of the program status word are set to 02H when the $\overline{\text{RESET}}$ signal is input.

Figure 3-14. Program Status Word Configuration

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

This flag controls acknowledgement of an interrupt request by the CPU.

When IE = 0, all interrupt requests except the non-maskable interrupt are disabled (DI status).

When IE = 1, interrupts are enabled (EI status). At this time, acknowledgement of interrupt requests is controlled with an in-service priority flag (ISP), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The interrupt enable flag is reset to 0 when the DI instruction is executed or when an interrupt request is acknowledged, and set to 1 when the EI instruction is executed.

(b) Zero flag (Z)

This flag is set to 1 when the result of an operation performed is 0; otherwise, it is reset to 0.

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

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

2-bit information indicating the register bank selected by execution of the “SEL RBn” instruction is stored in these flags.

(d) Auxiliary carry flag (AC)

This flag is set to 1 when a carry from or a borrow to bit 3 occurs as a result of an operation; otherwise, it is reset to 0.

(e) In-service priority flag (ISP)

This flag controls the priority of maskable vectored interrupts that can be acknowledged. When ISP = 0, the vectored interrupt request whose priority is specified by the priority specification flag registers (PR0L, PR0H, PR1L) (refer to **14.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) to be low is disabled. Whether the interrupt request is actually acknowledged is controlled by the status of the interrupt enable flag (IE).

(f) Carry flag (CY)

This flag records an overflow or underflow that occurs as the result of executing an add or subtract instruction. It also records the value shifted out when a rotate instruction is executed and functions as a bit accumulator when a bit operation instruction is executed.

(3) Stack pointer (SP)

This is a 16-bit register that holds the first address of the stack area in the memory. Only the internal high-speed RAM area (FB00H to FEFFH) can be specified as the stack area.

Figure 3-15. Stack Pointer Configuration

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

The contents of the stack pointer are decremented when data is written (saved) to the stack memory, and incremented when data is read (restored) from the stack memory.

The data saved/restored as a result of each stack operation is as shown in Figures 3-16 and 3-17.

Caution The contents of the SP become undefined when the $\overline{\text{RESET}}$ signal is input. Be sure to initialize the SP before executing an instruction.

Figure 3-16. Data Saved to Stack Memory

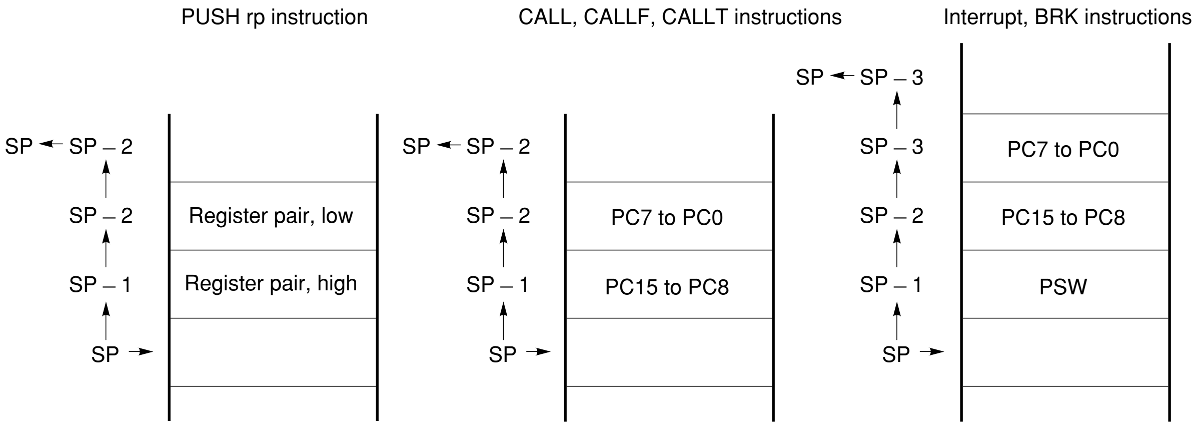
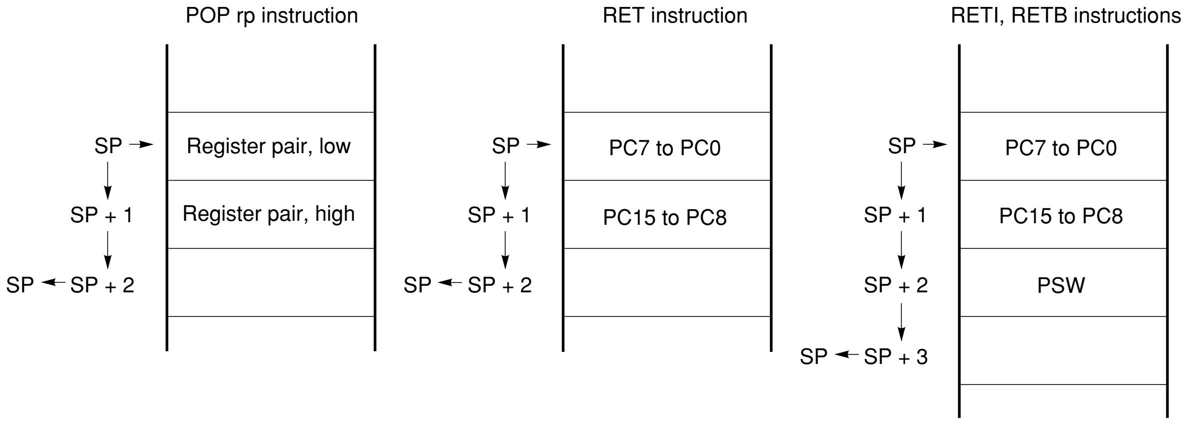


Figure 3-17. Data Restored from Stack Memory



3.2.2 General-purpose registers

General-purpose registers are mapped to specific addresses of the data memory (FEE0H to FEFFH). Four banks of general-purpose registers, each consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H) are available.

Each register can be used as an 8-bit register. Moreover, 8-bit registers can be used in pairs as 16-bit registers (AX, BC, DE, and HL).

Each register can be described not only by a function name (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) but also by an absolute name (R0 to R7, RP0 to RP3).

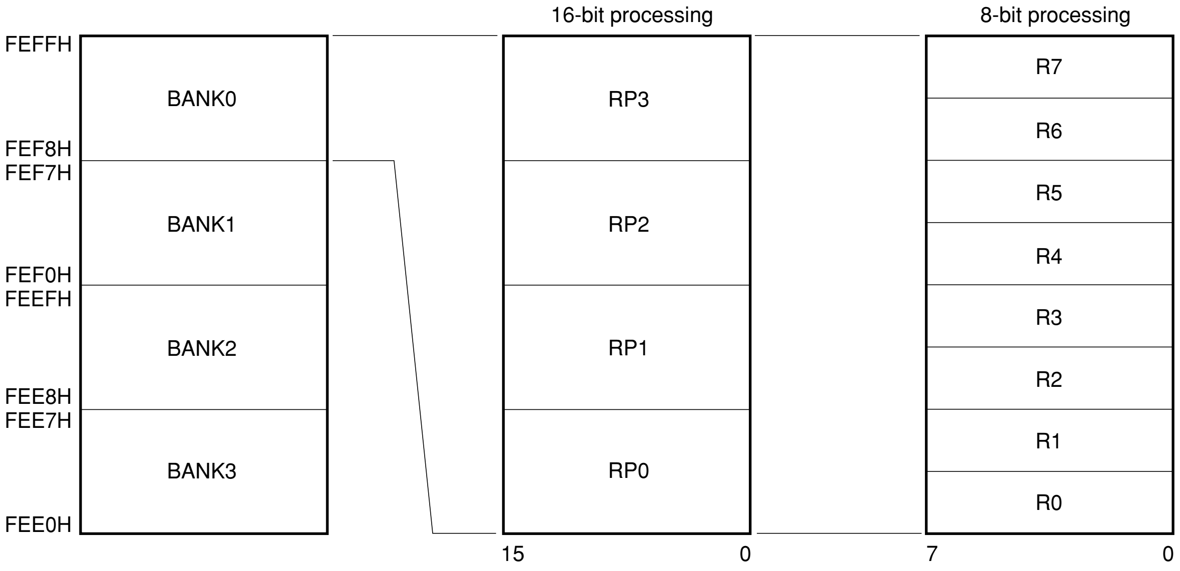
The register bank used for instruction execution is set by the CPU control instruction (SEL RBn). Because four register banks are provided, an efficient program can be developed by using one register bank for ordinary processing and another bank for interrupt servicing.

Table 3-3. Absolute Addresses of General-Purpose Registers

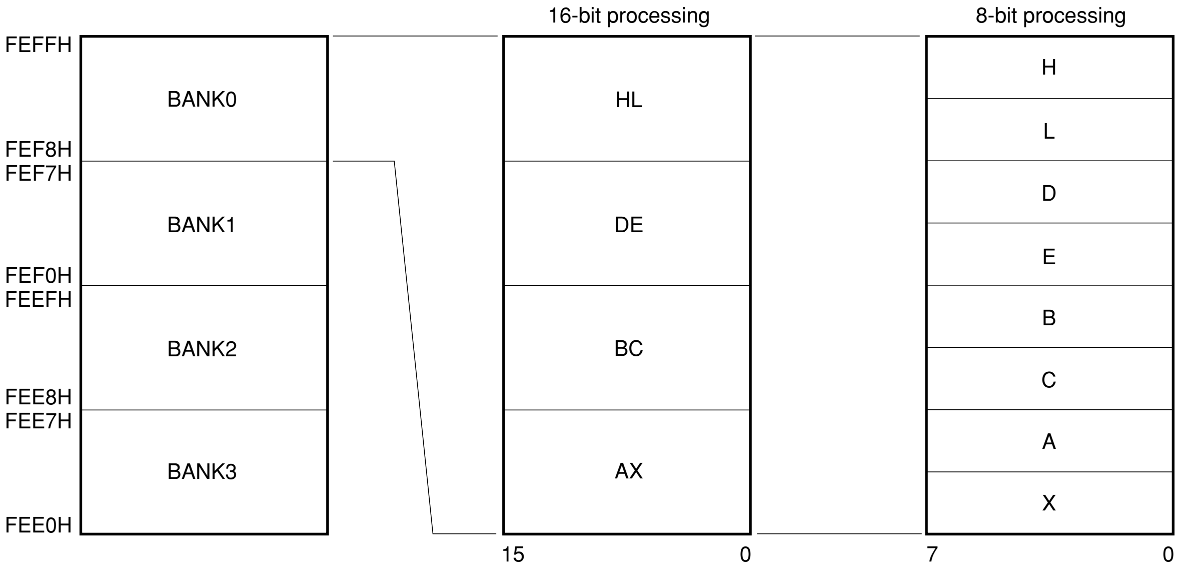
Bank Name	Register		Absolute Address	Bank Name	Register		Absolute Address
	Function Name	Absolute Name			Function Name	Absolute Name	
BANK0	H	R7	FEFFH	BANK2	H	R7	FEEFH
	L	R6	FEFEH		L	R6	FEEEH
	D	R5	FEFDH		D	R5	FEEDH
	E	R4	FEFCH		E	R4	FEECH
	B	R3	FEFBH		B	R3	FEEBH
	C	R2	FEFAH		C	R2	FEEAH
	A	R1	FEF9H		A	R1	FEE9H
	X	R0	FEF8H		X	R0	FEE8H
BANK1	H	R7	FEF7H	BANK3	H	R7	FEE7H
	L	R6	FEF6H		L	R6	FEE6H
	D	R5	FEF5H		D	R5	FEE5H
	E	R4	FEF4H		E	R4	FEE4H
	B	R3	FEF3H		B	R3	FEE3H
	C	R2	FEF2H		C	R2	FEE2H
	A	R1	FEF1H		A	R1	FEE1H
	X	R0	FEF0H		X	R0	FEE0H

Figure 3-18. General-Purpose Register Configuration

(a) Absolute name



(b) Function name



3.2.3 Special function registers (SFRs)

Unlike the general-purpose registers, special function registers have their own functions and are allocated to the area of addresses FF00H to FFFFH.

The special function registers can also be manipulated in the same manner as the general-purpose registers by using operation, transfer, and bit manipulation instructions. The bit units in which one register is to be manipulated (1, 8, or 16 bits) differ from those of another register.

The bit unit for manipulation is specified as follows:

- **1-bit manipulation**
  A symbol reserved by the assembler is described as the operand (sfr.bit) of a 1-bit manipulation instruction. An address can also be specified.
- **8-bit manipulation**
  A symbol reserved by the assembler is described as the operand (sfr) of an 8-bit manipulation instruction. An address can also be specified.
- **16-bit manipulation**
  A symbol reserved by the assembler is described as the operand (sfrp) of a 16-bit manipulation instruction. When specifying an address, describe an even address.

Table 3-4 lists the special function registers. The meanings of the symbols in this table are as follows:

- **Symbol**
  These symbols indicate the addresses of the special function registers.
  They are reserved words for the RA78K0 and defined by header file sfrbit.h for the CC78K0. These symbols can be described as the operands of instructions when the RA78K0, ID78K0-NS, ID78K0, and SM78K0 are used.
- **R/W**
  Indicates whether the special function register in question can be read or written.
  R/W: Read/write
  R: Read only
  W: Write only
- **Bit unit for manipulation**
  "√" indicates the manipulatable bit unit (1, 8, or 16). "—" indicates the bit units for which manipulation is not possible.
- **After reset**
  Indicates the status of the special function register when the $\overline{\text{RESET}}$ signal is input.

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

Address	Special Function Register (SFR) Name	Symbol		R/W		Bit Unit for Manipulation			After Reset
						1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W		√	√	–	00H
FF01H	Port 1	P1		R		√	√	–	
FF02H	Port 2	P2		R/W		√	√	–	
FF03H	Port 3	P3				√	√	–	
FF04H	Port 4	P4				√	√	–	
FF05H	Port 5	P5				√	√	–	
FF06H	Port 6	P6				√	√	–	
FF07H	8-bit timer counter 52	TM52		R		–	√	–	
FF08H	10-bit buffer register 0	BFCM0	BFCM0L	R/W	R/W	–	√	√	0000H
FF09H			–		–		–		
FF0AH	10-bit buffer register 1	BFCM1	BFCM1L	R/W	R/W	–	√	√	
FF0BH			–		–		–		
FF0CH	10-bit buffer register 2	BFCM2	BFCM2L	R/W	R/W	–	√	√	
FF0DH			–		–		–		
FF0EH	10-bit buffer register 3	BFCM3	BFCM3L	R/W	R/W	–	√	√	00FFH
FF0FH			–		–		–		
FF10H	16-bit timer counter 00	TM00		R		–	–	√	0000H
FF11H									
FF12H	16-bit timer counter 01	TM01				–	–	√	
FF13H									
FF14H	8-bit timer counter 50	TM5	TM50			–	√	√	00H
FF15H	8-bit timer counter 51		TM51			–	√		
FF16H	8-bit compare register 50	CR5	CR50	R/W		–	√	√	Undefined
FF17H	8-bit compare register 51		CR51			–	√		
FF18H	A/D conversion result register 0	ADCR0		R		–	–	√	
FF19H									
FF1AH	Transmit shift register 0	TXS00		W		–	√	–	FFH
	Receive buffer register 0	RXB00		R		–	√	–	
FF1BH	Transmit shift register 1	TXS01		W		–	√	–	
	Receive buffer register 1	RXB01		R		–	√	–	
FF1FH	Serial I/O shift register 3	SIO3		R/W		–	√	–	Undefined
FF20H	Port mode register 0	PM0				√	√	–	FFH
FF22H	Port mode register 2	PM2				√	√	–	
FF23H	Port mode register 3	PM3				√	√	–	
FF24H	Port mode register 4	PM4				√	√	–	
FF25H	Port mode register 5	PM5				√	√	–	
FF26H	Port mode register 6	PM6				√	√	–	

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Unit for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
FF32H	Pull-up resistor option register 2	PU2		√	√	–	
FF33H	Pull-up resistor option register 3	PU3		√	√	–	
FF34H	Pull-up resistor option register 4	PU4		√	√	–	
FF35H	Pull-up resistor option register 5	PU5		√	√	–	
FF36H	Pull-up resistor option register 6	PU6		√	√	–	
FF42H	Watchdog timer clock select register	WDCS		–	√	–	
★ FF47H	Memory extension mode register	MEM	W	–	√	–	
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	
FF49H	External interrupt falling edge enable register	EGN		√	√	–	
FF60H	16-bit timer mode control register 00	TMC00		√	√	–	
FF61H	Prescaler mode register 00	PRM00		–	√	–	
FF62H	Capture/compare control register 00	CRC00		√	√	–	
FF63H	Timer output control register 00	TOC00		√	√	–	
FF64H	16-bit capture/compare register 000	CR000		–	–	√	Undefined
FF65H							
FF66H	16-bit capture/compare register 010	CR010		–	–	√	
FF67H							
FF68H	16-bit timer mode control register 01	TMC01		√	√	–	00H
FF69H	Prescaler mode register 01	PRM01		–	√	–	
FF6AH	Capture/compare control register 01	CRC01		√	√	–	
FF6BH	Timer output control register 01	TOC01		√	√	–	
FF6CH	16-bit capture/compare register 001	CR001		–	–	√	Undefined
FF6DH							
FF6EH	16-bit capture/compare register 011	CR011		–	–	√	
FF6FH							
FF70H	8-bit timer mode control register 50	TMC50		√	√	–	00H
FF71H	Timer clock select register 50	TCL50		–	√	–	
FF74H	8-bit timer mode control register 51	TMC51		√	√	–	
FF75H	Timer clock select register 51	TCL51		–	√	–	
FF78H	8-bit timer mode control register 52	TMC52		√	√	–	
FF79H	Timer clock select register 52	TCL52		–	√	–	
FF7AH	8-bit compare register 52	CR52		–	√	–	Undefined
FF7CH	External interrupt rising edge enable register 5	EGP5		√	√	–	00H
FF7DH	External interrupt falling edge enable register 5	EGN5		√	√	–	
FF80H	A/D converter mode register 0	ADM0		√	√	–	
FF81H	Analog input channel specification register 0	ADS0		–	√	–	
FF84H	Real-time output buffer register 0L	RTBL00		√	√	–	
FF85H	Real-time output buffer register 0H	RTBH00		√	√	–	

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

Address	Special Function Register (SFR) Name	Symbol		R/W		Bit Unit for Manipulation			After Reset
						1 Bit	8 Bits	16 Bits	
FF86H	Real-time output port mode register 0	RTPM00		R/W		√	√	–	00H
FF87H	Real-time output port control register 0	RTPC00				√	√	–	
FF89H	Flash programming mode control register	FLPMC				√	√	–	08H[Note 1]
FF90H	Inverter timer control register 7	TMC7				√	√	–	00H
FF91H	Inverter timer mode register 7	TMM7				√	√	–	
FF92H	10-bit compare register 0	CM0		R/W	R/W	–	√	√	0000H
FF93H					–		–		
FF94H	10-bit compare register 1	CM1		R/W	R/W	–	√	√	
FF95H					–		–		
FF96H	10-bit compare register 2	CM2		R/W	R/W	–	√	√	
FF97H					–		–		
FF98H	10-bit compare register 3	CM3		R/W	R/W	–	√	√	00FFH
FF99H					–		–		
FF9AH	Dead time reload register	DTIME		W		–	√	–	FFH
FF9CH	Real-time output buffer register 1L	RTBL01		R/W		√	√	–	00H
FF9DH	Real-time output buffer register 1H	RTBH01				√	√	–	
FF9EH	Real-time output port mode register 1	RTPM01				√	√	–	
FF9FH	Real-time output port control register 1	RTPC01				√	√	–	
FFA0H	Asynchronous serial interface mode register 0	ASIM00				√	√	–	
FFA1H	Asynchronous serial interface status register 0	ASIS00		R		–	√	–	
FFA2H	Baud rate generator control register 0	BRGC00		R/W		–	√	–	
FFA8H	Asynchronous serial interface mode register 1	ASIM01				√	√	–	
FFA9H	Asynchronous serial interface status register 1	ASIS01		R		–	√	–	
FFAAH	Baud rate generator control register 1	BRGC01		R/W		–	√	–	
FFB0H	Serial operation mode register 3	CSIM3				√	√	–	
FFB8H	DC control register 0	DCCTL0				√	√	–	
FFBCH	DC control register 1	DCCTL1				√	√	–	
FFD0H to FFDFH	External access area[Note 2]					√	√	–	Undefined
FFE0H	Interrupt request flag register 0L	IF0L	IF0			√	√	√	00H
FFE1H	Interrupt request flag register 0H	IF0H				√	√		
FFE2H	Interrupt request flag register 1L	IF1L				√	√	–	
FFE4H	Interrupt mask flag register 0L	MK0L	MK0			√	√	√	FFH
FFE5H	Interrupt mask flag register 0H	MK0H				√	√		
FFE6H	Interrupt mask flag register 1L	MK1L				√	√	–	
FFE8H	Priority specification flag register 0L	PR0L	PR0			√	√	√	

Notes 1. Bit 2 changes according to the voltage level of V_{PP}.

2. The external access area cannot be accessed in the SFR addressing mode. Access this area with direct addressing.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Unit for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
FFE9H	Priority specification flag register 0H	PR0H	PR0	R/W	√	√	√	FFH
FFEAH	Priority specification flag register 1L	PR1L			√	√	–	
FFF0H	Memory size switching register	IMS			–	√	–	CFH^{Note 1}
FFF4H	Internal expansion RAM size switching register	IXS			–	√	–	0CH^{Note 2}
FFF8H	Memory extension wait setting register	MM			–	√	–	10H
FFF9H	Watchdog timer mode register	WDTM			√	√	–	00H
FFFAH	Oscillation stabilization time select register	OSTS			–	√	–	04H
FFFBH	Processor clock control register	PCC			√	√	–	

Notes **1.** The initial value is CFH, but set and operate each microcontroller with the values shown below.

µPD780982:	C4H
µPD780983:	C6H
µPD780984:	C8H
µPD780986:	CCH
µPD780988:	CFH (No need to change the initial value because the µPD780988 is set to CFH).
µPD78F0988A:	Value corresponding to those of mask ROM versions

2. The initial value is 0CH, but set and operate each microcontroller with the values shown below.

µPD780982, 780983, 780984:	0CH (No need to change the initial value because the µPD780982, 780983, 780984 are set to 0CH).
µPD780986, 780988:	0AH
µPD78F0988A:	Value corresponding to mask ROM versions

3.3 Instruction Address Addressing

An instruction address is determined by the contents of the program counter (PC). The contents of the PC are usually automatically incremented by the number of bytes of the instruction to be fetched (by 1 per byte) every time an instruction is executed. When an instruction that causes program execution to branch is performed, the address information of the branch destination is set to the PC by means of the following addressing (for details of each instruction, refer to **78K/0 Series User's Manual Instructions (U12326E)**).

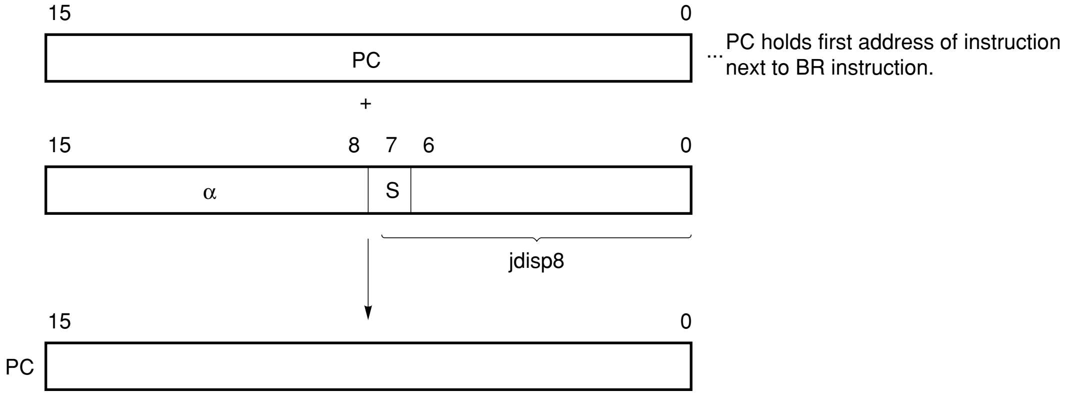
3.3.1 Relative addressing

[Function]

The 8-bit immediate data (displacement value: jdisp8) of the instruction code is added to the first address of the next instruction, the resultant sum is transferred to the program counter (PC), and program execution branches. The displacement value is treated as signed 2's complement data (–128 to +127), and bit 7 serves as a sign bit. In other words, relative addressing consists of relative branching from the first address of the following instruction to the –128 to +127 range.

This addressing is used when the "BR $addr16" instruction or conditional branch instruction is executed.

[Operation]



15
0
PC
... PC holds first address of instruction next to 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

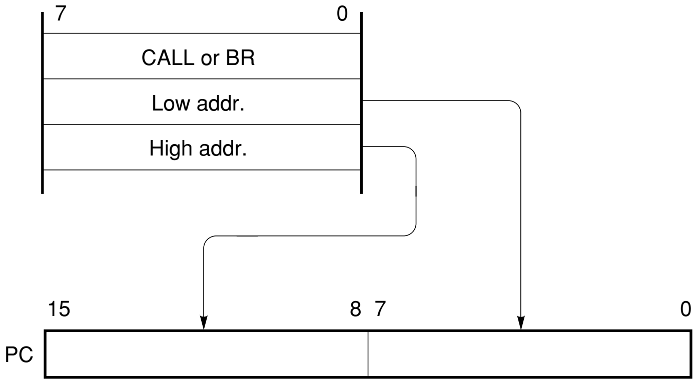
[Function]

The immediate data in an instruction word is transferred to the program counter (PC), and program execution branches.

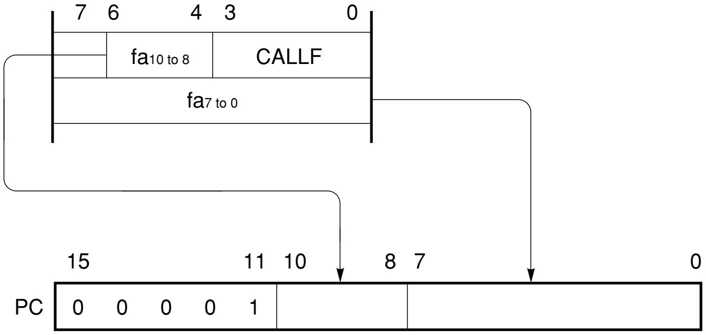
This addressing is used when the “CALL !addr16”, “BR !addr16”, or “CALLF !addr11” instruction is executed. The CALL !addr16 and BR !addr16 instructions allow the program to branch to the entire memory space. The CALLF !addr11 instruction allows the program to branch to the 0800H to 0FFFH area.

[Operation]

When the “CALL !addr16” or “BR !addr16” instruction is executed



When the “CALLF !addr11” instruction is executed



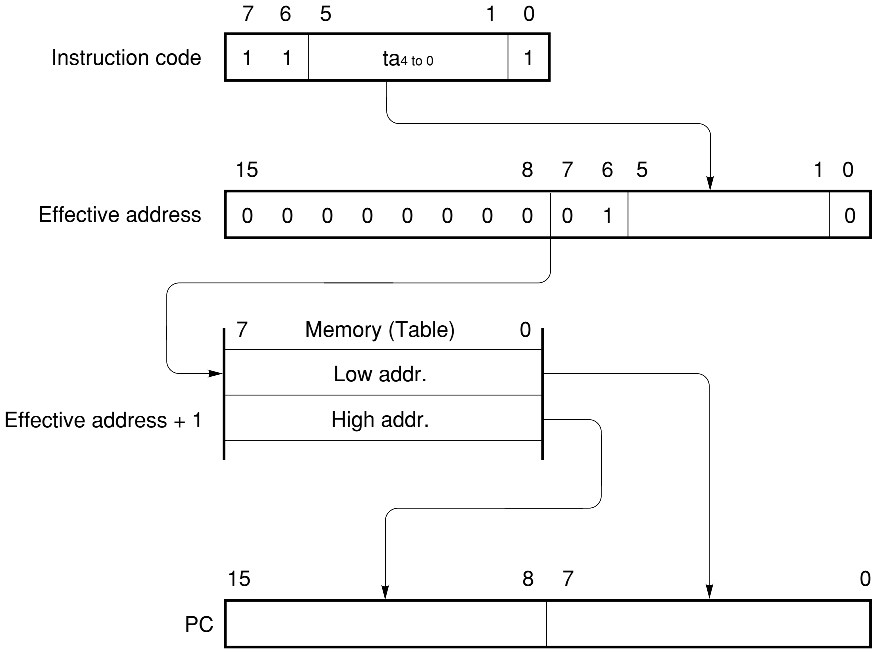
3.3.3 Table indirect addressing

[Function]

The contents of a specific location table (branch destination address) addressed by the immediate data of bits 1 to 5 of an instruction code are transferred to the program counter (PC), and program execution branches.

This addressing is used when the “CALLT [addr5]” instruction is executed. This instruction references the addresses stored in the memory table from 40H to 7FH, and allows the program to branch to the entire memory space.

[Operation]



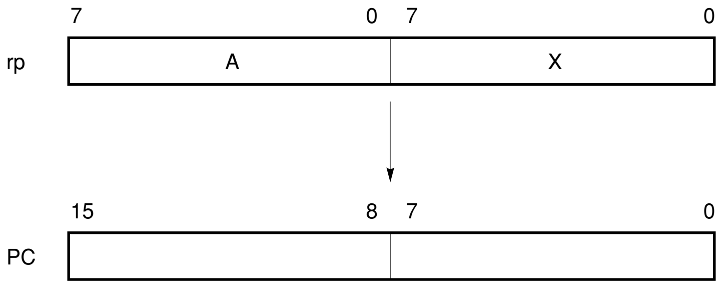
3.3.4 Register addressing

[Function]

The contents of the register pair (AX) specified by an instruction word are transferred to the program counter (PC), and program execution branches.

This addressing is used when the “BR AX” instruction is executed.

[Operation]



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]

This addressing is used to automatically (implicitly) address a register that functions as an accumulator (A or AX) in the general-purpose register area.

The instruction words of the μPD780988 Subseries that use implied addressing are as follows.

Instruction	Register Specified by Implied Addressing
MULU	Register A to store multiplicand and register AX to store product
DIVUW	Register AX to store dividend and quotient
ADJBA/ADJBS	Register A to store numeric value subject to decimal adjustment
ROR4/ROL4	Register A to store digit data subject to digit rotation

[Operand Format]

No specific operand format is used because the operand format is automatically determined by the instruction.

[Example]

MULU X

The product between registers A and X is stored in register AX as a result of executing the multiply instruction of 8 bits × 8 bits. In this operation, registers A and AX are specified by implied addressing.

3.4.2 Register addressing

[Function]

This addressing is used to access a general-purpose register as an operand. The general-purpose register to be accessed is specified by the register bank select flags (RBS0 and RBS1) and with the register specification code (Rn and RPn) in an instruction code.

Register addressing is used when an instruction that has the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified by 3 bits in the instruction code.

[Operand Format]

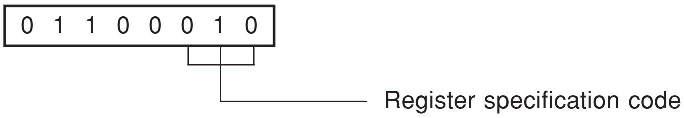
Representation	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

r and rp can be described not only by a function name (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) but also by an absolute name (R0 to R7, RP0 to RP3).

[Example]

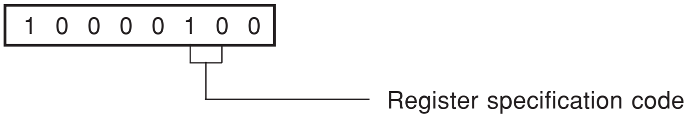
MOV A, C; To select C register as r

Instruction code



INCW DE; To select DE register pair as rp

Instruction code



3.4.3 Direct addressing

[Function]

The addressing is used to directly address the memory indicated by the immediate data in an instruction word.

[Operand Format]

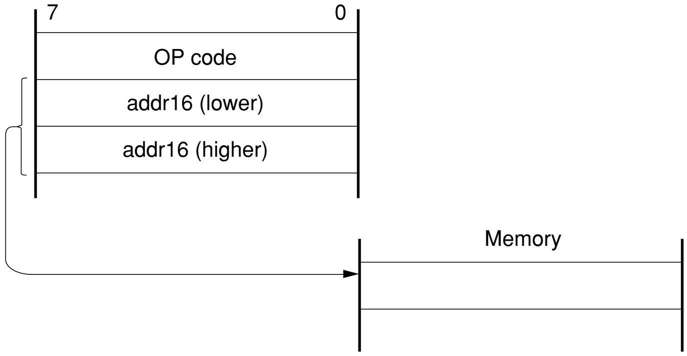
Representation	Description
addr16	Label or 16-bit immediate data

[Example]

MOV A, !0FE00H; To specify FE00H as !addr16

Instruction 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

[Operation]



3.4.4 Short direct addressing

[Function]

This addressing directly addresses a memory area to be manipulated from a fixed space by using the 8-bit data in an instruction word.

This addressing is applicable to the fixed 256-byte space of FE20H to FF1FH. The internal high-speed RAM is mapped to addresses FE20H to FEFFH, and special function registers (SFRs) are mapped to addresses FF00H to FF1FH.

The SFR area (FF00H to FF1FH) to which short direct addressing is applied is a part of the entire SFR area. Ports that are frequently accessed in the program, and compare and capture registers of timer/event counters are mapped to the SFR area. These SFRs can be manipulated with a few bytes and clocks.

Bit 8 of the effective address is 0 if the 8-bit immediate data is in the range of 20H to FFH, and 1 if the data is in the range of 00H to 1FH. Refer to **[Operation]**.

[Operand Format]

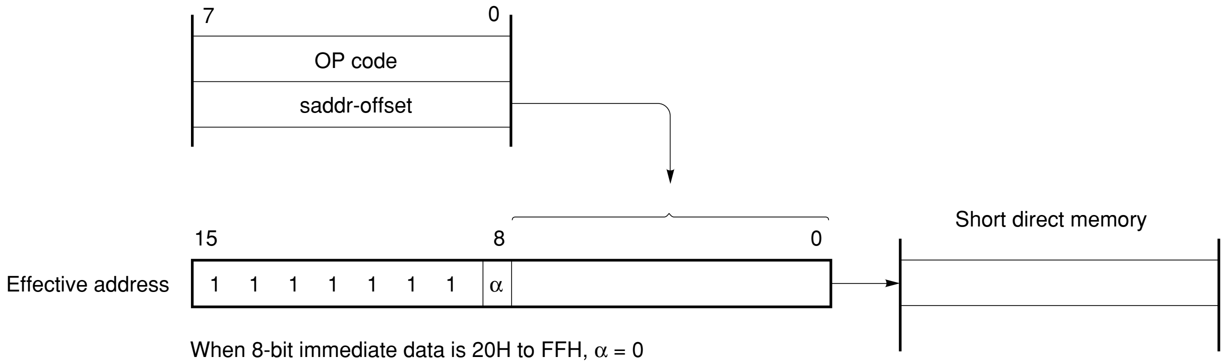
Representation	Description
saddr	Label or immediate data FE20H to FF1FH
saddrp	Label or immediate data FE20H to FF1FH (even address only)

[Example]

MOV 0FE30H, #50H; To specify FE30H as saddr and 50H as immediate data

Instruction 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)

[Operation]



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

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

3.4.5 Special function register (SFR) addressing

[Function]

This addressing is to address special function registers (SFRs) mapped to the memory by using the 8-bit immediate data in an instruction word.

This addressing is applied to the 240-byte space of FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped to the area of FF00H to FF1FH can also be accessed by means of short direct addressing.

[Operand Format]

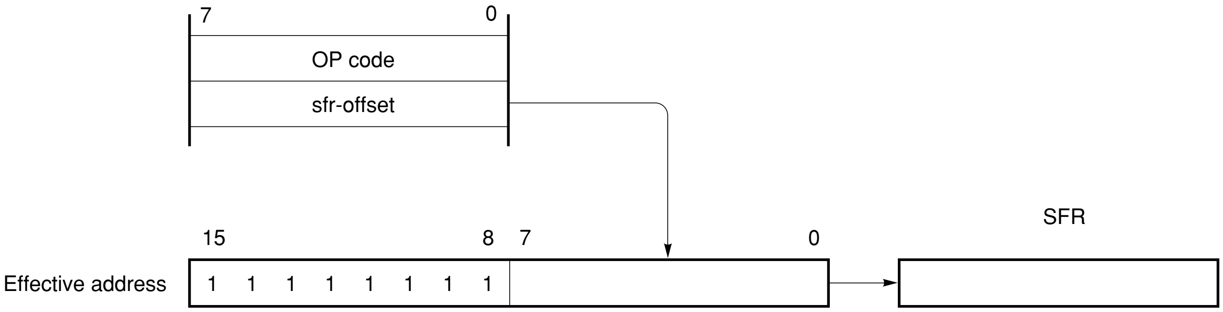
Representation	Description
sfr	Special function register name
sfrp	Name of special function register that can be manipulated in 16-bit units (even address only)

[Example]

MOV PM0, A; To select PM0 (FF20H) as sfr

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

[Operation]



3.4.6 Register indirect addressing

[Function]

This addressing is used to address memory using the contents of a specified register pair as an operand. The register pair to be accessed is specified by the register bank select flags (RBS0 and RBS1) and the register pair specification code in an instruction code. This addressing can address the entire memory space.

[Operand Format]

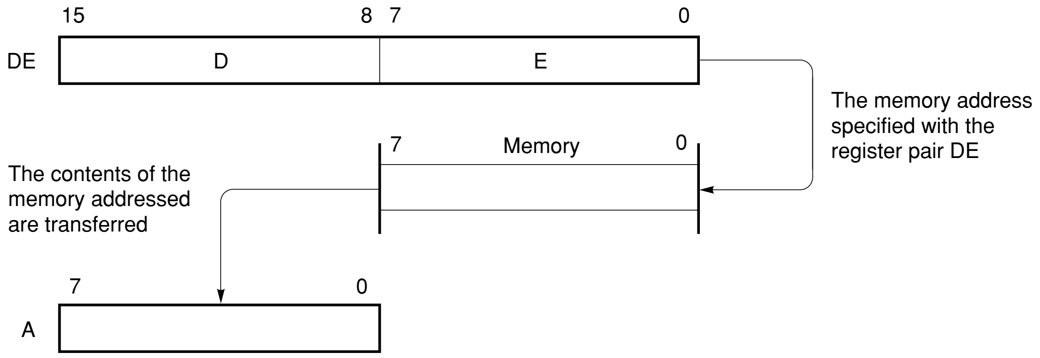
Representation	Description
—	[DE], [HL]

[Example]

MOV A, [DE]; To select [DE] as register pair

Instruction code 1 0 0 0 0 1 0 1

[Operation]


15
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]

This addressing is used to address the memory by using the result of adding 8-bit immediate data to the contents of the HL register pair used as a base register. The HL register pair to be accessed is in the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is executed by extending the offset data to 16 bits as a positive number. A carry from the 16th bit is ignored. This addressing can address the entire memory space.

[Operand Format]

Representation	Description
—	[HL + byte]

[Example]

MOV A, [HL + 10H]; To specify 10H as byte

```
Instruction code  1 0 1 0 1 1 1 0

                  0 0 0 1 0 0 0 0
```

3.4.8 Based indexed addressing

[Function]

This addressing is used to address the memory by using the result of adding the contents of the B or C register specified in the instruction word to the contents of the HL register used as a base register. The HL, B, and C registers accessed are in the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is executed with the contents of the B or C register extended to 16 bits as a positive number. A carry from the 16th bit is ignored.

This addressing can address the entire memory space.

[Operand Format]

Representation	Description
—	[HL + B], [HL + C]

[Example]

When MOV A, [HL + B]

Instruction code `1 0 1 0 1 0 1 1`

3.4.9 Stack addressing

[Function]

This addressing is used to indirectly address the stack area by using the contents of the stack pointer (SP).

This addressing is automatically used to save/restore register contents when the PUSH, POP, subroutine call, or return instruction is executed, or when an interrupt request is generated.

Stack addressing can access the internal high-speed RAM area only.

[Example]

When PUSH DE is executed

Instruction code `1 0 1 1 0 1 0 1`

CHAPTER 4 PORT FUNCTIONS

4.1 Function of Ports

The μPD780988 Subseries is provided with eight input port pins and 39 I/O port pins. Figure 4-1 shows these port pins. Each port can be manipulated in 1-bit or 8-bit units and controlled in various ways. Moreover, some port pins also serve as the I/O pins of the internal hardware.

Figure 4-1. Types of Ports

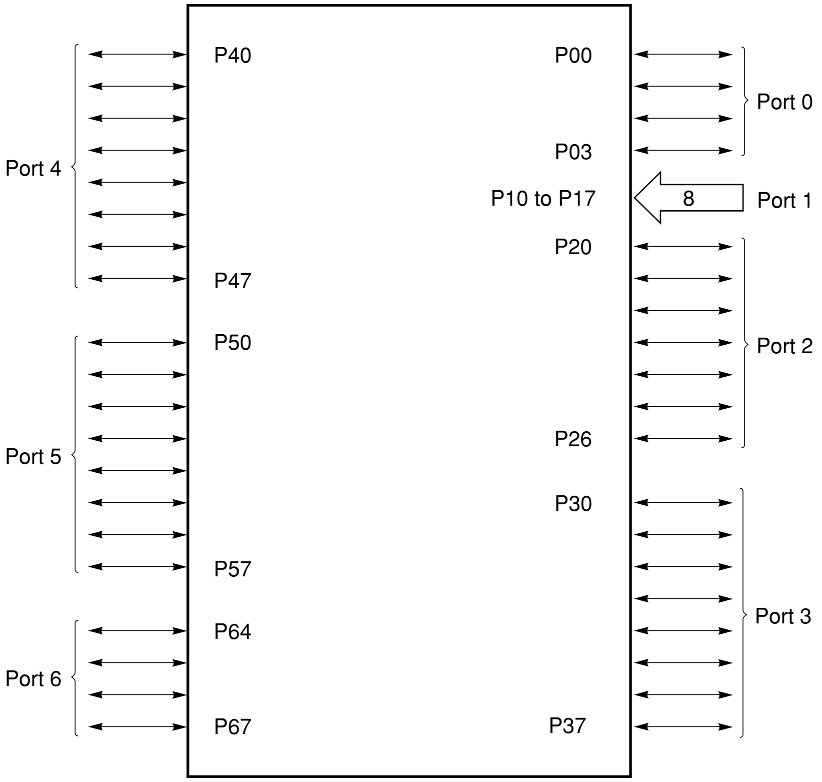


Table 4-1. Port Functions

Pin Name		Function	Alternate Function
Port 0	P00	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.	INTP0/TOFF7
	P01		INTP1
	P02		INTP2
	P03		INTP3/ADTRG
Port 1	P10 to P17	8-bit input-only port	ANI0 to ANI7
Port 2	P20	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.	RxD00
	P21		TxD00
	P22		RxD01
	P23		TxD01
	P24		TI50/TO50
	P25		TI51/TO51
	P26		TI52/TO52
Port 3	P30 to P37	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.	RTP0 to RTP7
Port 4	P40 to P47	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.	AD0 to AD7
Port 5	P50	8-bit I/O port LEDs can be driven directly. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	—
	P51		$\overline{\text{SCK}}$
	P52		SI
	P53		SO
	P54		INTP4/TI000/TO00
	P55		INTP5/TI010
	P56		INTP6/TI001/TO01
	P57		INTP7/TI011
Port 6	P64	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.	$\overline{\text{RD}}$
	P65		$\overline{\text{WR}}$
	P66		$\overline{\text{WAIT}}$
	P67		ASTB

4.2 Configuration of Ports

A port consists of the following hardware.

Table 4-2. Port Configuration

Item		Configuration
Control registers		Port mode register (PM0, PM2 to PM6) Pull-up resistor option register (PU0, PU2 to PU6)
Ports	Total	47
	Input	8
	I/O	39
Pull-up resistors		39 (software control)

4.2.1 Port 0

This is a 4-bit I/O port with output latches. Port 0 can be set in the input or output mode in 1-bit units via port mode register 0 (PM0). When using port 0, internal pull-up resistors can be connected in 1-bit units by using pull-up resistor option register 0 (PU0).

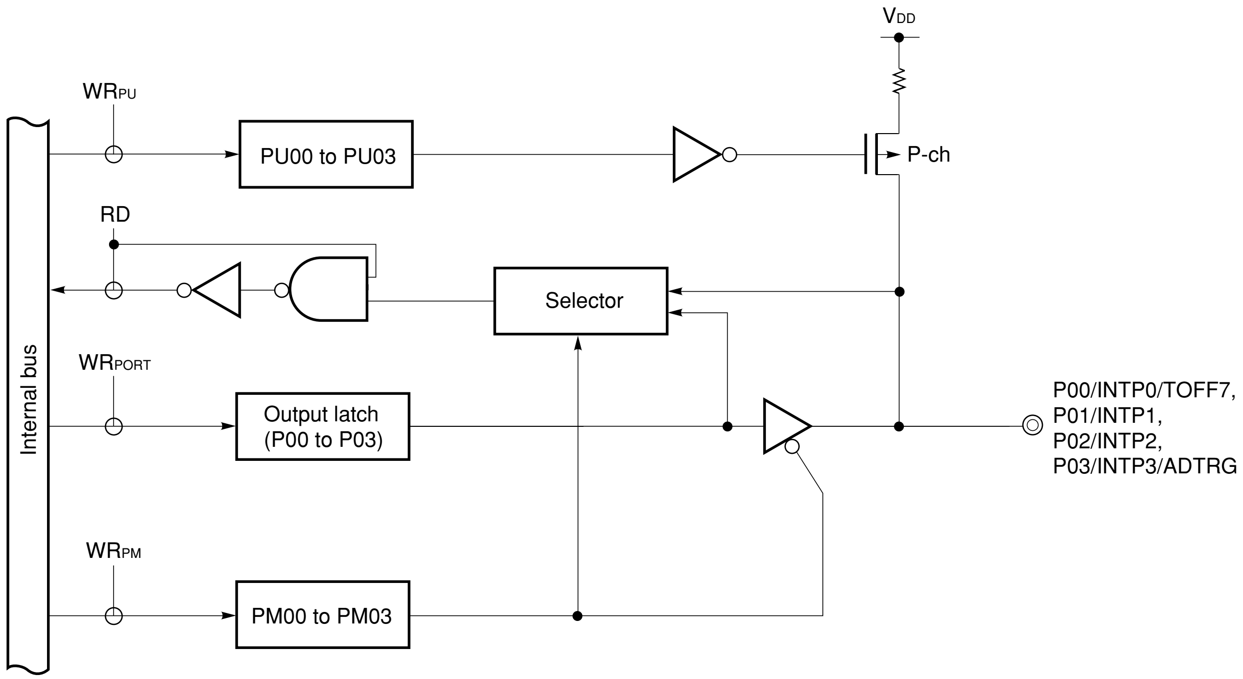
Alternate functions include external interrupt request input, external input to stop timer output, and an external trigger signal for the A/D converter.

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

Figure 4-2 shows the block diagram of port 0.

Caution Because port 0 is also used as an external interrupt request input, an interrupt request flag is set when the port is set in the output mode and its output level is changed. When using port 0 in the output mode, therefore, set the interrupt mask flag to 1.

Figure 4-2. Block Diagram of P00 to P03



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 0

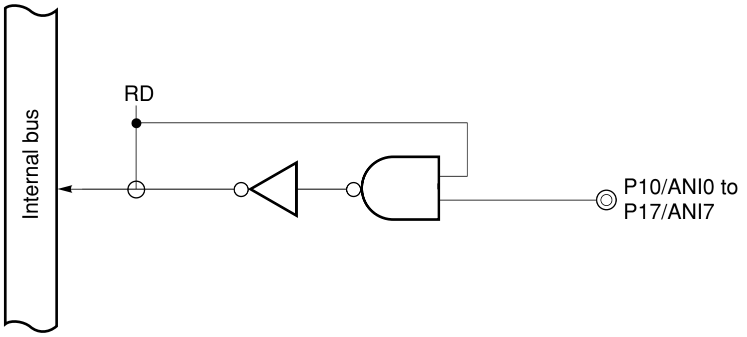
WR: Write signal of port 0

4.2.2 Port 1

This is an 8-bit input port. Alternate functions include A/D converter analog input.

Figure 4-3 shows the block diagram of port 1.

Figure 4-3. Block Diagram of P10 to P17



4.2.3 Port 2

This is a 7-bit I/O port with output latches. Port 2 can be set in the input or output mode in 1-bit units via port mode register 2 (PM2). When using port 2, internal pull-up resistors can be connected in 1-bit units by using pull-up resistor option register 2 (PU2).

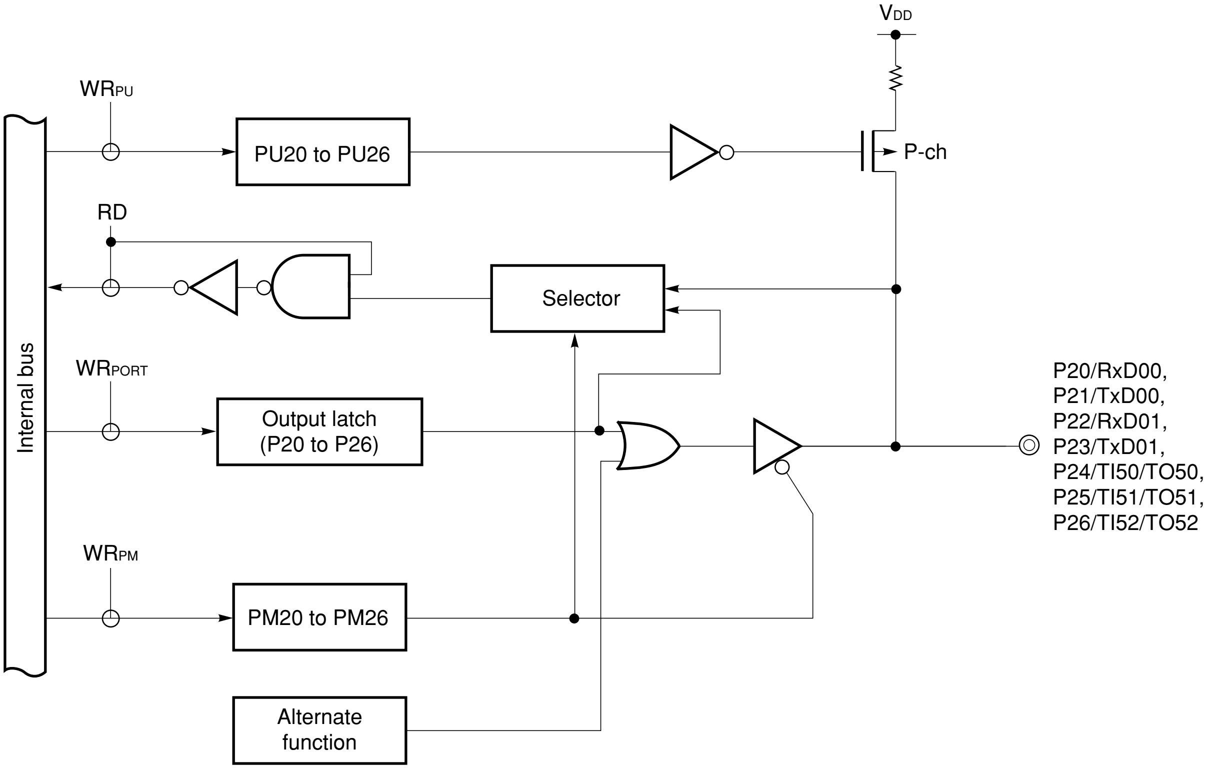
Alternate functions include serial interface data I/O and timer I/O.

$\overline{\text{RESET}}$ input sets port 2 to input mode.

Figure 4-4 shows the block diagram of port 2.

Caution When performing transmission using the serial interface or timer output, set the pins to be used to output mode, and set the output latch to 0.
When performing reception or timer input, set the pins to be used to input mode.

Figure 4-4. Block Diagram of P20 to P26



PU: Pull-up resistor option register
PM: Port mode register
RD: Read signal of port 2
WR: Write signal of port 2

4.2.4 Port 3

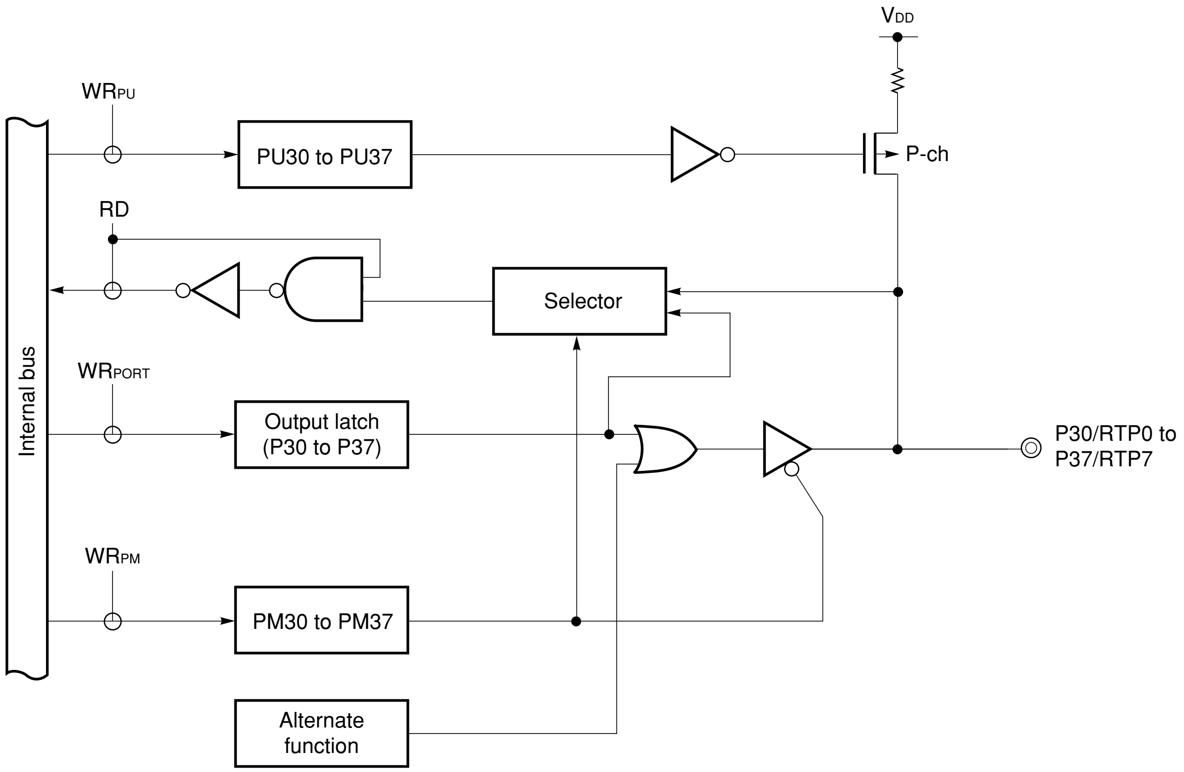
This is an 8-bit I/O port with output latches. Port 3 can be set in the input or output mode in 1-bit units via port mode register 3 (PM3). When using port 3, internal pull-up resistors can be connected in 1-bit units by using pull-up resistor option register 3 (PU3).

Alternate functions include use as a real-time output port.

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

Figure 4-5 shows the block diagram of port 3.

Figure 4-5. Block Diagram of P30 to P37



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 3

WR: Write signal of port 3

4.2.5 Port 4

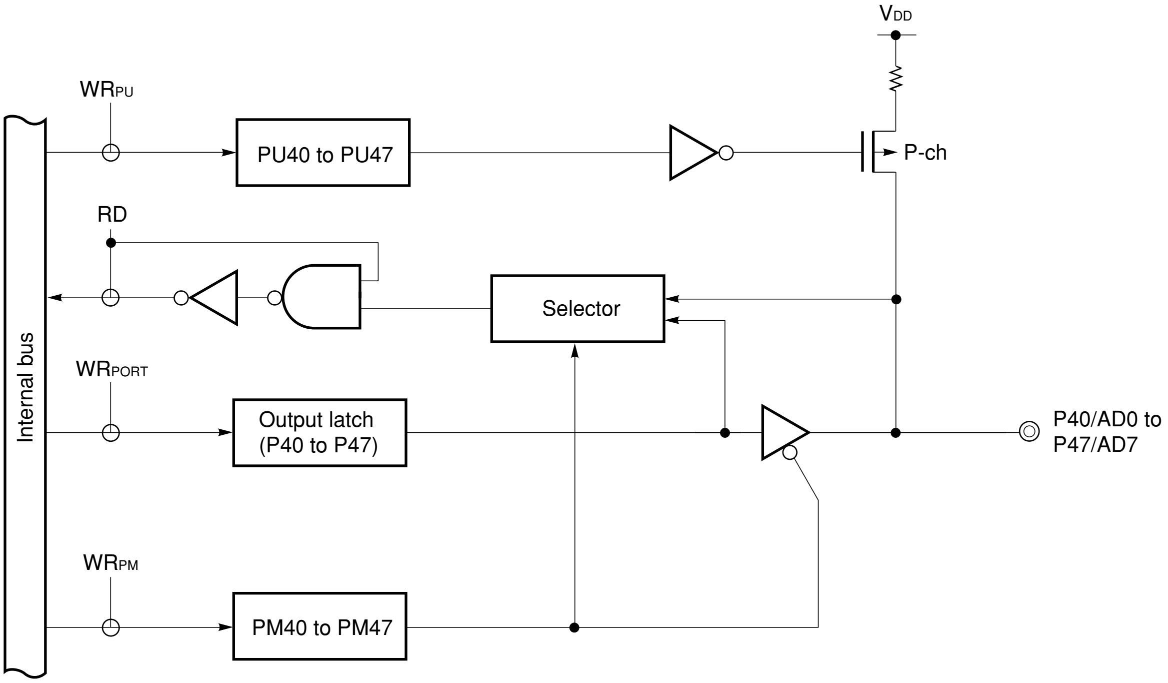
This is an 8-bit I/O port with output latches. Port 4 can be set in the input or output mode in 1-bit units via port mode register 4 (PM4). When using port 4, internal pull-up resistors can be connected in 1-bit units by using pull-up resistor option register 4 (PU4).

Alternate functions include an address/data bus function that is used in the external memory expansion mode.

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

Figure 4-6 shows the block diagram of port 4.

Figure 4-6. Block Diagram of P40 to P47



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 4

WR: Write signal of port 4

4.2.6 Port 5

This is an 8-bit I/O port with output latches. Port 5 can be set in the input or output mode in 1-bit units via port mode register 5 (PM5). When using port 5, internal pull-up resistors can be connected in 1-bit units by using pull-up resistor option register 5 (PU5).

Port 5 can directly drive LEDs.

Alternate functions include serial interface clock and data I/O, timer I/O, and external interrupt request input.

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

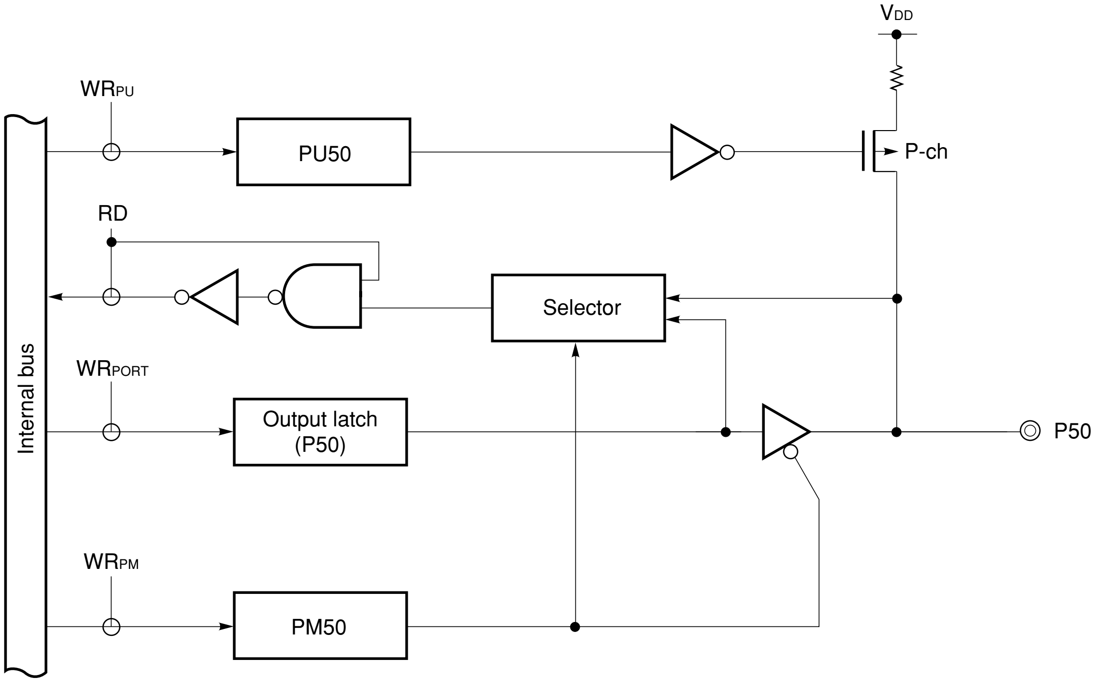
Figures 4-7 and 4-8 show the block diagram of port 5.

Cautions 1. When performing transmission using the serial interface or timer output, set the pins to be used to output mode, and set the output latch to 0.

When performing reception or timer input, set the pins to be used to input mode.

2. Because pins P54 to P57 are also used as external interrupt request input pins, an interrupt request flag is set when the port is set in the output mode and its output level is changed. When using the output mode, therefore, set the interrupt mask flag to 1.

Figure 4-7. Block Diagram of P50



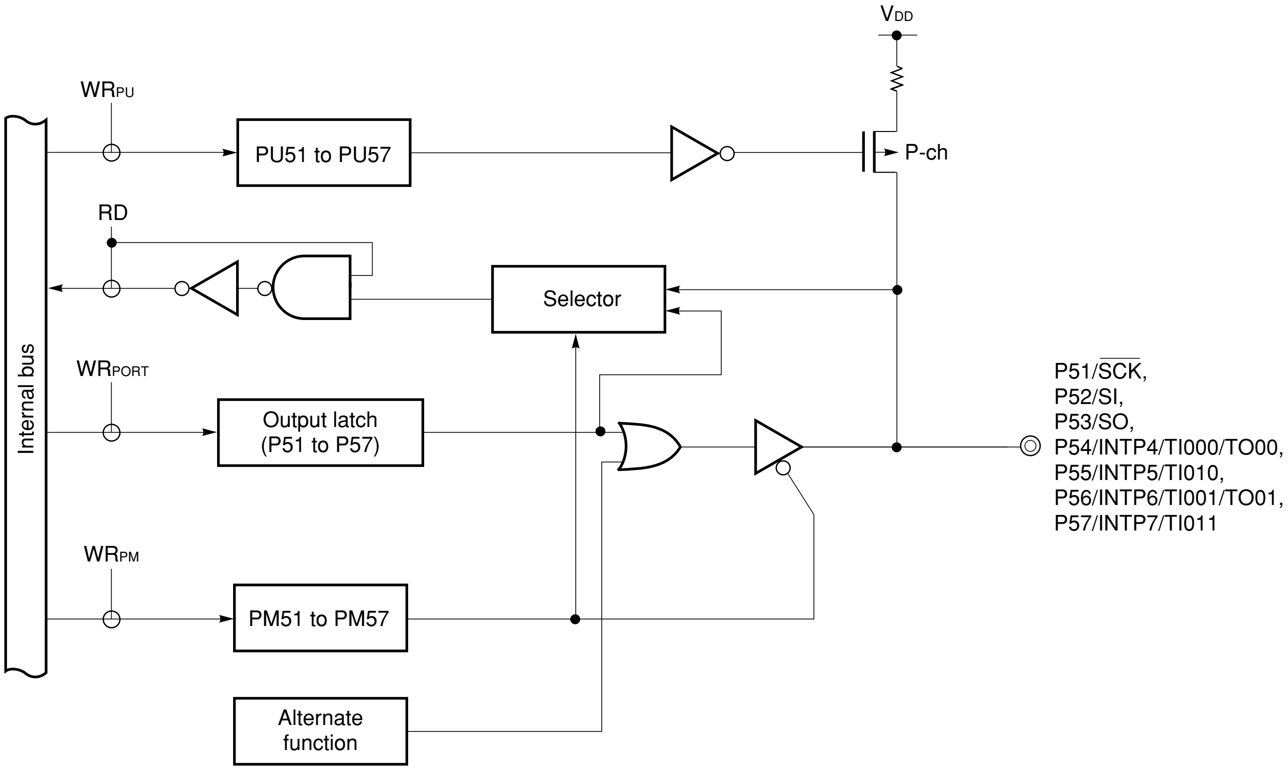
PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 5

WR: Write signal of port 5

Figure 4-8. Block Diagram of P51 to P57



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 5

WR: Write signal of port 5

4.2.7 Port 6

This is a 4-bit I/O port with output latches. Port 6 can be set in the input or output mode in 1-bit units via port mode register 6 (PM6). When using port 6, internal pull-up resistors can be connected in 1-bit units by using pull-up resistor option register 6 (PU6).

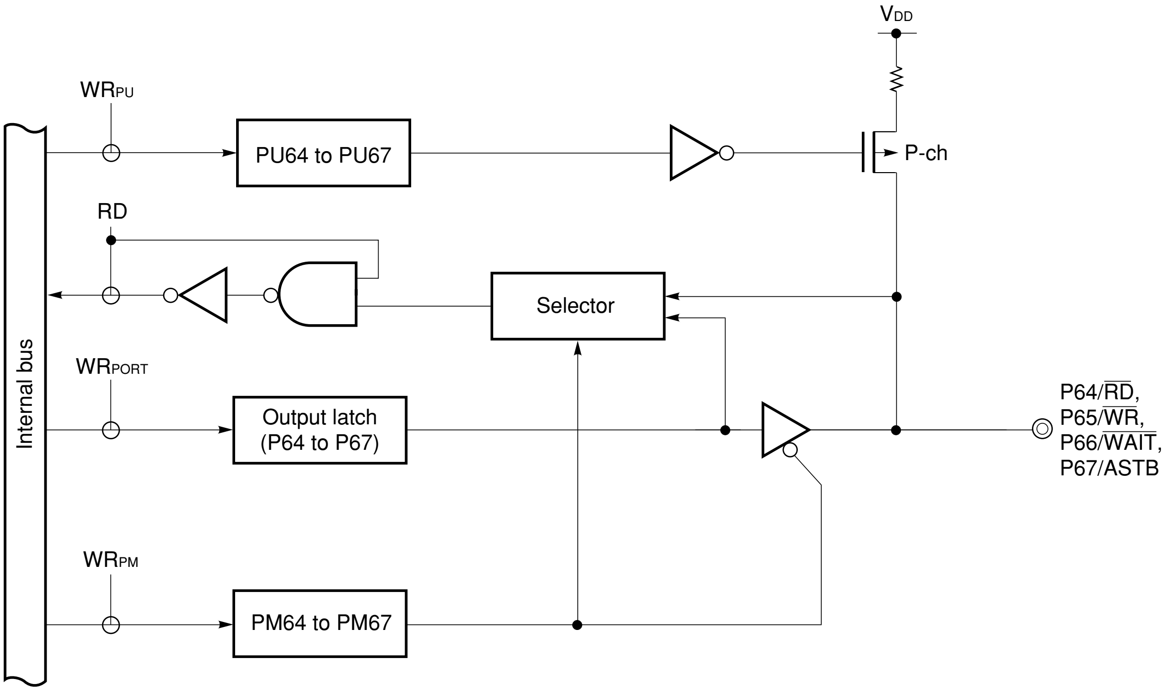
Alternate functions include a control signal output function in the external memory expansion mode.

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

Figure 4-9 shows the block diagram of port 6.

Caution P66 can be used as an I/O port pin when no external wait state is used in the external memory expansion mode.

Figure 4-9. Block Diagram of P64 to P67



VDD
WRPU
PU64 to PU67
P-ch
RD
Selector
Internal bus
WRPORT
Output latch (P64 to P67)
P64/$\overline{\text{RD}}$,
P65/$\overline{\text{WR}}$,
P66/$\overline{\text{WAIT}}$,
P67/ASTB
WRPM
PM64 to PM67

PU: Pull-up resistor option register
PM: Port mode register
RD: Read signal of port 6
WR: Write signal of port 6

4.3 Registers Controlling Port Functions

The following two types of registers control the ports.

- Port mode registers (PM0, PM2, PM3, PM4, PM5, PM6)
- Pull-up resistor option registers (PU0, PU2, PU3, PU4, PU5, PU6)

(1) Port mode registers (PM0, PM2, PM3, PM4, PM5, PM6)

These registers set the corresponding ports in the input or output mode in 1-bit units.

PM0, PM2, PM3, PM4, PM5, and PM6 are manipulated by a 1-bit or 8-bit memory manipulation instruction.

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

Cautions 1. Because port 0 and pins P54 to P57 are also used as external interrupt request input pins, interrupt request flags are set when the output mode of the port function is specified and the output level is changed. To use this port in the output mode, therefore, set the interrupt mask flags to 1 in advance.

2. Since pull-up resistors will not be disconnected even if ports 0 and 2 to 6 are set to output mode, set corresponding pull-up resistor option registers to 0 when those ports are used in output mode.

3. When a port pin that has an alternate function serves as an alternate function output pin, set its output latch to 0.

Figure 4-10. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	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	1	1	1	1	FF26H	FFH	R/W

PMmn	Selects I/O mode of Pmn pin (m = 0: n = 0 to 3) (m = 2: n = 0 to 6) (m = 3, 4, 5: n = 0 to 7) (m = 6: n = 4 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Pull-up resistor option registers (PU0, PU2, PU3, PU4, PU5, PU6)

These registers set whether the internal pull-up resistor is connected to each port. By setting PU0 and PU2 to PU6, on-chip pull-up resistors corresponding to bits in PU0 and PU2 to PU6 can be used.

PU0, PU2 to PU6 are individually set by a 1-bit or 8-bit memory manipulation instruction.

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

Cautions 1. Port 1 is not provided with an on-chip pull-up resistor.

2. When PUm is set to 1, an on-chip pull-up resistor is connected regardless of whether the mode is input/output mode or external expansion mode. Accordingly, when using the port in output or external expansion mode, set the corresponding bit of PUm to 0 (m = 0, 2 to 6).

Figure 4-11. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU2	0	PU26	PU25	PU24	PU23	PU22	PU21	PU20	FF32H	00H	R/W
PU3	PU37	PU36	PU35	PU34	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

PUmn	Selects internal pull-up resistor of Pmn pin (m = 0: n = 0 to 3) (m = 2: n = 0 to 6) (m = 3, 4, 5: n = 0 to 7) (m = 6: n = 4 to 7)
0	On-chip pull-up resistor is not used
1	On-chip pull-up resistor is used

4.4 Operation of Port Functions

The operation of a port differs depending on whether the port is set in the input or output mode, as described below.

4.4.1 Writing to I/O port

(1) In output mode

A value can be written to the output latch of a port by using a transfer instruction. The contents of the output latch can be output from the pins of the port.

Data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. However, the status of the port pin is not changed because the output buffer is off.

Data once written to the output latch is retained until new data is written to the output latch.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set in the input mode and not subject to manipulation become undefined.

4.4.2 Reading from I/O port

(1) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch are not changed.

(2) In input mode

The status of a pin can be read by using a transfer instruction. The contents of the output latch are not changed.

4.4.3 Arithmetic operation of I/O port

(1) In output mode

An arithmetic operation can be performed on the contents of the output latch. The result of the operation is written to the output latch. The contents of the output latch are output from the port pins.

Data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

The contents of the output latch become undefined. However, the status of the pin is not changed because the output buffer is off.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set in the input mode and not subject to manipulation become undefined.

CHAPTER 5 CLOCK GENERATOR

★ 5.1 Function of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware. Oscillation can be stopped by executing the STOP instruction.

- Expanded-specification products
  The system oscillator oscillates a frequency of 1.0 to 12.0 MHz.
- Conventional products
  The system oscillator oscillates a frequency of 1.0 to 8.38 MHz.

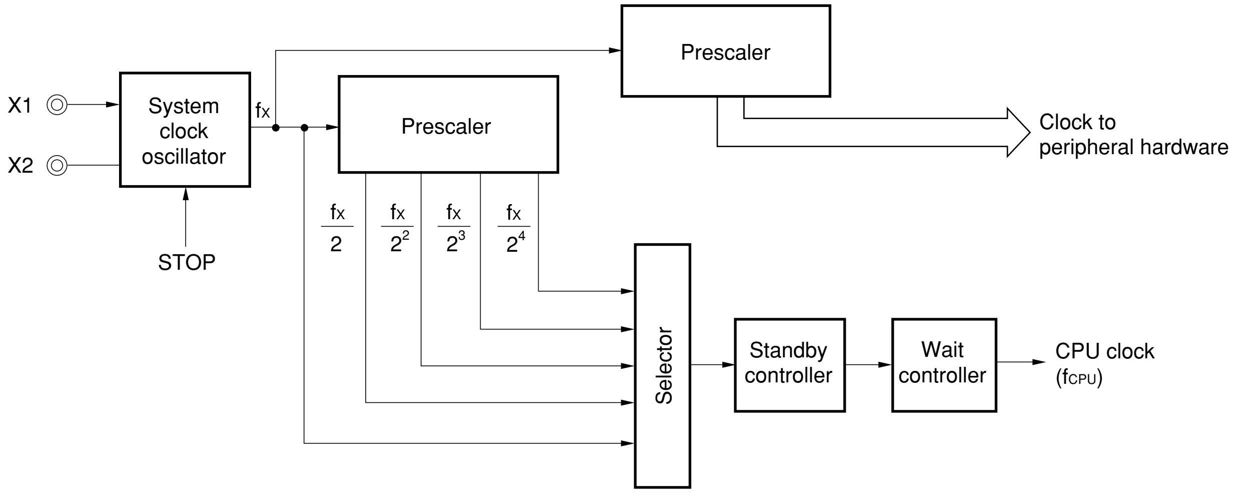
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillator	System clock oscillator

Figure 5-1. Clock Generator Block Diagram



5.3 Register Controlling Clock Generator

The clock generator is controlled by the processor clock control register (PCC). This register selects the CPU clock. PCC is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets this register to 04H.

Figure 5-2. Format of Processor Clock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PCC	0	0	0	0	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W

PCC2	PCC1	PCC0	Selects CPU clock (f_{CPU})
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	$f_X/2^4$
Other than above			Setting prohibited

Caution Be sure to clear bits 3 to 7 to 0.

Remark f_X: System clock oscillation frequency

The fastest instruction of the μPD780988 Subseries is executed in two CPU clocks. Therefore, the relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 5-2.

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

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$	
	At f_X = 12 MHz^{Note}	At f_X = 8.38 MHz
f_X	0.166 μs	0.238 μs
$f_X/2$	0.33 μs	0.48 μs
$f_X/2^2$	0.66 μs	0.96 μs
$f_X/2^3$	1.3 μs	1.9 μs
$f_X/2^4$	2.6 μs	3.8 μs

Note Expanded-specification products only.

Remark f_X: System clock oscillation frequency

5.4 System Clock Oscillators

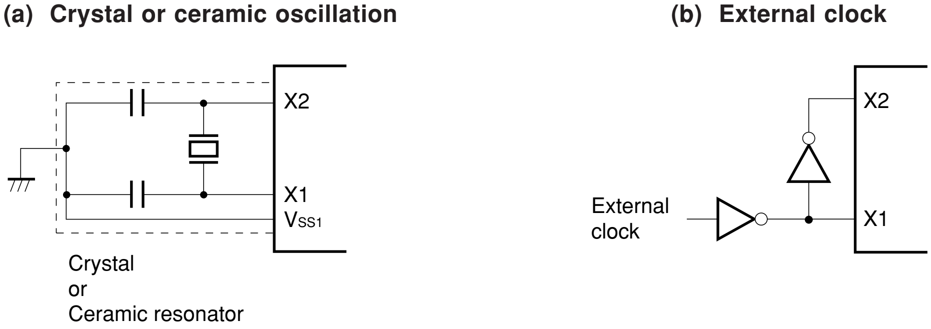
5.4.1 System clock oscillator

★ The system clock oscillator is oscillated by the crystal or ceramic resonator (12 MHz TYP.) connected across the X1 and X2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the X1 pin, and input the inverted signal to the X2 pin.

Figure 5-3 shows the external circuit of the system clock oscillator.

Figure 5-3. External Circuit of System Clock Oscillator



(a) Crystal or ceramic oscillation

(b) External clock

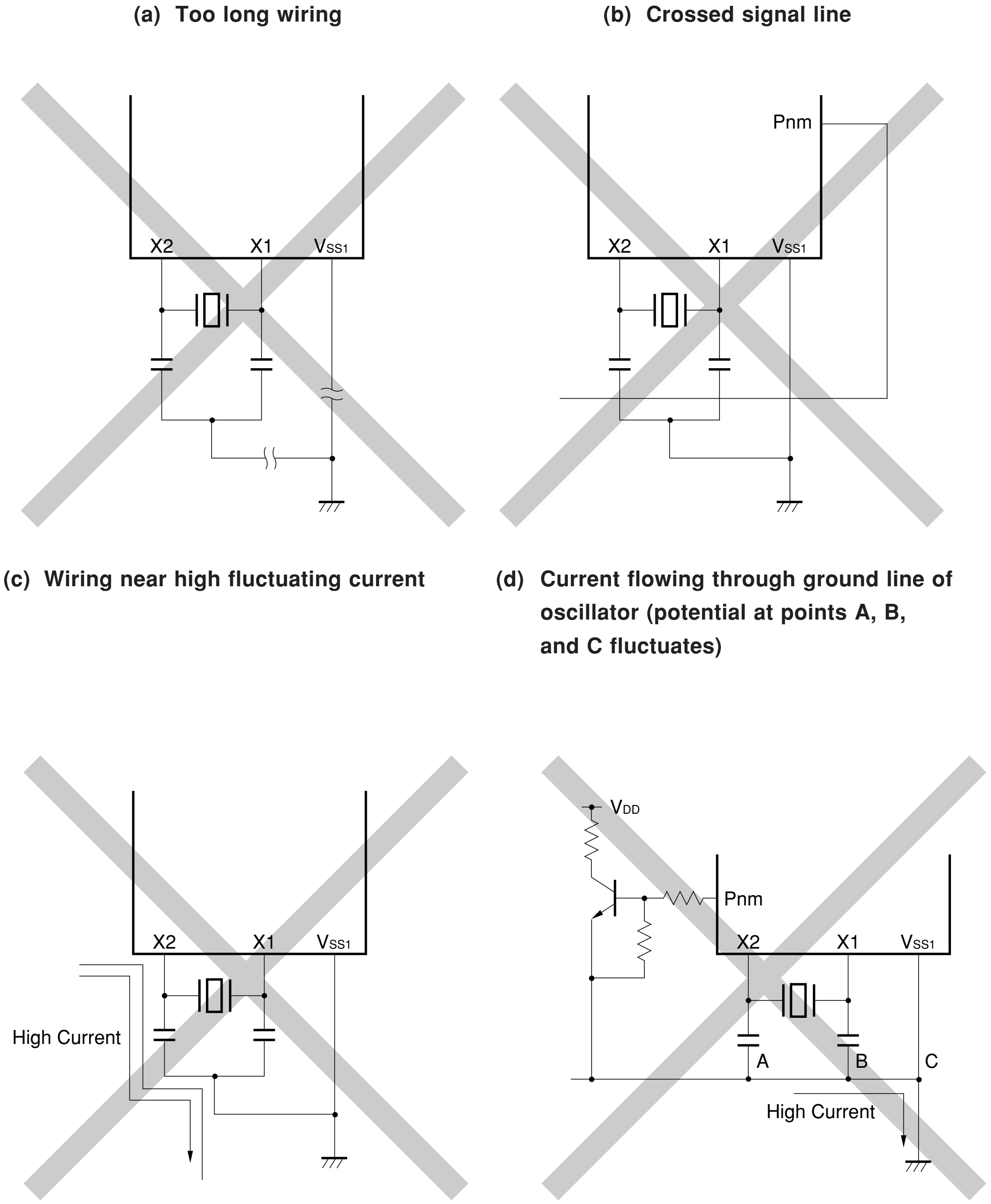
Cautions 1. The STOP instruction cannot be executed when the external clock is input. This is because if the STOP instruction is executed, the system clock operation is stopped, and the X2 pin is pulled up to V_{DD1}.

2. When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS1}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

Figure 5-4 shows examples of incorrect resonator connection.

Figure 5-4. Examples of Incorrect Resonator Connection (1/2)



(a) Too long wiring

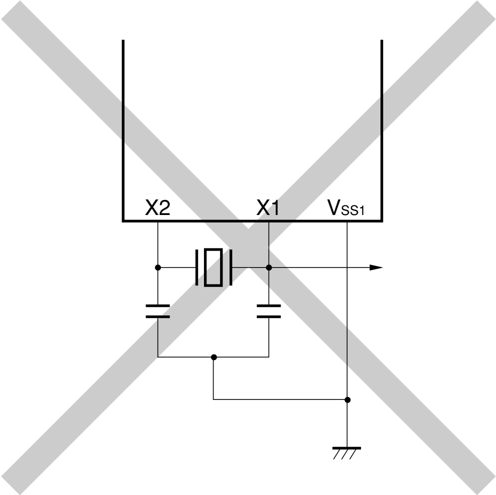
(b) Crossed signal line

(c) Wiring near high fluctuating current

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

Figure 5-4. Examples of Incorrect Resonator Connection (2/2)

(e) Signal is fetched



5.4.2 Divider

The divider divides the output of the system clock oscillator (fx) to generate various clocks.

5.5 Operation of Clock Generator

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

- System clock f_X
- CPU clock f_{CPU}
- Clock to peripheral hardware

The operation of the clock generator is determined by the processor clock control register (PCC), as follows.

★ (a) The slowest mode (2.6 μs @ 12 MHz operation, 3.8 μs @ 8.38 MHz operation) of the system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 04H). While a low level is input to the $\overline{\text{RESET}}$ pin, oscillation of the system clock is stopped.

★ (b) Five types of minimum instruction execution time (0.166 μs, 0.33 μs, 0.66 μs, 1.3 μs, and 2.6 μs @ 12 MHz operation/0.238 μs, 0.48 μs, 0.96 μs, 1.9 μs, and 3.8 μs @ 8.38 MHz operation) can be selected via a PCC setting when the system clock is in the selected state.

(c) Two standby modes, STOP and HALT, can be used.

(d) The clock to the peripheral hardware is supplied by dividing the system clock. Therefore, the other peripheral hardware is stopped when the system clock is stopped (except, however, the external clock input operation).

5.6 Changing Setting of CPU Clock

5.6.1 Time required for switching CPU clock

The CPU clock can be selected by using bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC).

Actually, the specified clock is not selected immediately after the setting of PCC has been changed, and the old clock is used for the duration of several instructions after that (refer to **Table 5-3**).

Table 5-3. Maximum Time Required for Switching CPU Clock

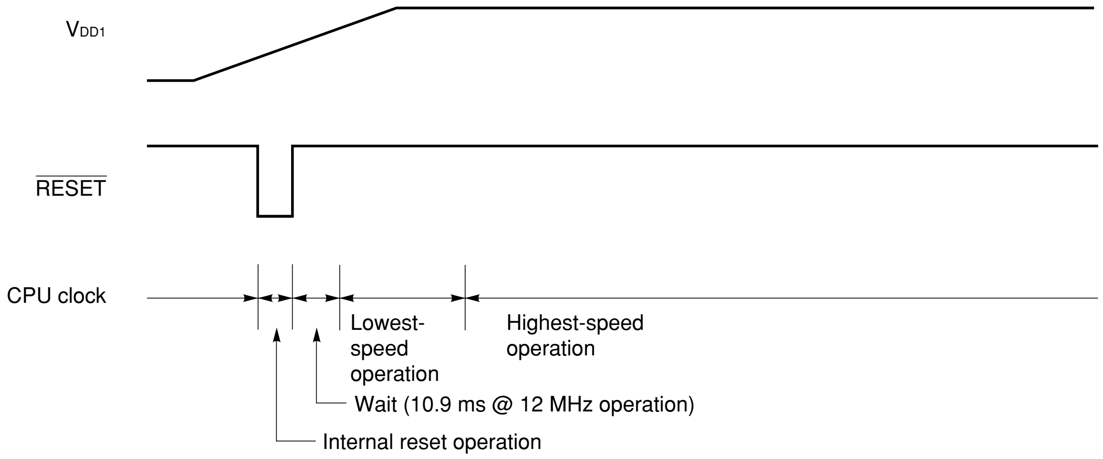
Set Value Before Switching			Set Value After Switching														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 instructions			16 instructions			16 instructions			16 instructions		
0	0	1	8 instructions						8 instructions			8 instructions			8 instructions		
0	1	0	4 instructions			4 instructions						4 instructions			4 instructions		
0	1	1	2 instructions			2 instructions			2 instructions						2 instructions		
1	0	0	1 instruction			1 instruction			1 instruction			1 instruction					

Remark One instruction is the minimum instruction execution time of the CPU clock before switching.

5.6.2 Switching CPU clock

The following figure illustrates how the CPU clock is switched.

Figure 5-5. Switching Between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. The effect of resetting is released when the $\overline{\text{RESET}}$ pin is later made high, and the system clock starts oscillating. At this time, the time during which oscillation stabilizes (2^{17}/fx) is automatically secured.

★ After that, the CPU starts instruction execution at the slowest speed of the system clock (2.6 μs @ 12 MHz operation, 3.8 μs @ 8.38 MHz operation).

<2> After the time during which the V_{DD1} voltage rises to the level at which the CPU can operate at the highest speed has elapsed, processor clock control register (PCC) is rewritten so that the highest speed can be selected.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

6.1 Outline of 16-Bit Timer/Event Counter

A 16-bit timer/event counter can be used as an interval timer, for PPG output, pulse width measurement (infrared remote control receive function), as an external event counter, or for square-wave output of any frequency.

6.2 Function of 16-Bit Timer/Event Counter

The 16-bit timer/event counters have the following functions.

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

(1) Interval timer
TM0n generates interrupt requests at the preset time interval.

(2) PPG output
TM0n can output a square wave whose frequency and output pulse can be set freely.

(3) Pulse width measurement
TM0n can measure the pulse width of an externally input signal.

(4) External event counter
TM0n can measure the number of pulses of an externally input signal.

(5) Square-wave output
TM0n can output a square wave with any selected frequency.

6.3 Configuration of 16-Bit Timer/Event Counter

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

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

Item	Configuration
Timer register	16-bit timer counter 0n (TM0n)
Register	16-bit capture/compare register 00n, 01n (CR00n, CR01n)
Timer output	TO0n
Control register	16-bit timer mode control register 0n (TMC0n) Capture/compare control register 0n (CRC0n) Timer output control register 0n (TOC0n) Prescaler mode register 0n (PRM0n) Port mode register 5 (PM5)Note

Note Refer to **Figure 4-8 Block Diagram of P51 to P57**.

Remark n = 0, 1

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

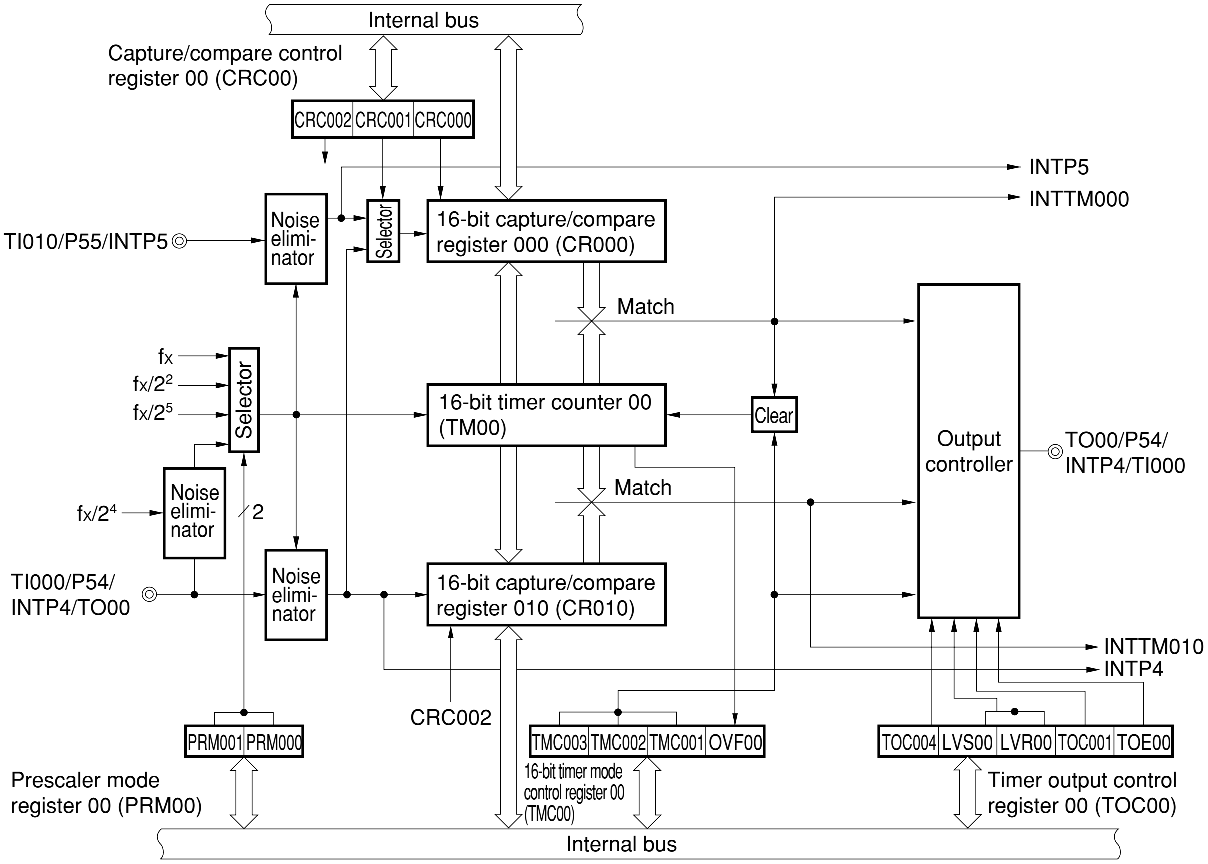
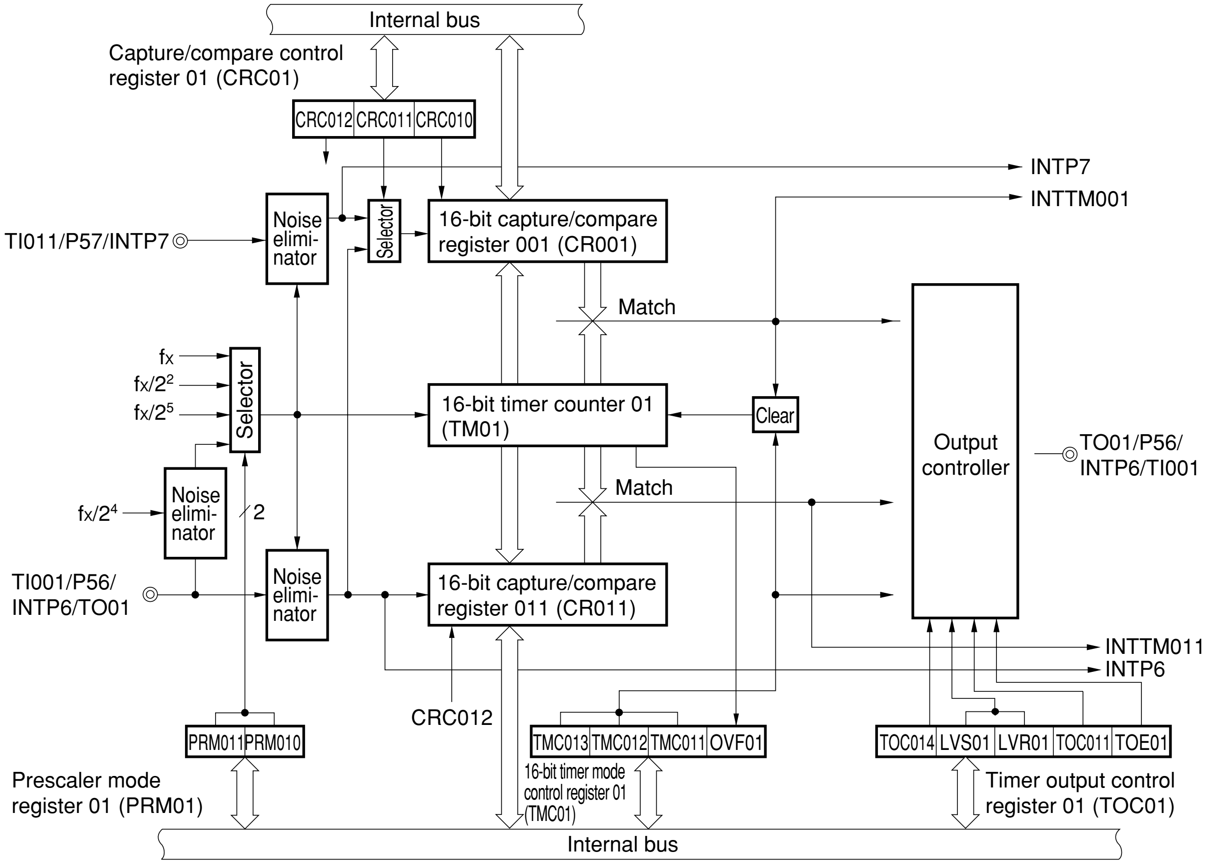


Figure 6-2. Block Diagram of 16-Bit Timer/Event Counter 01



(1) 16-bit timer counter 00, 01 (TM00, TM01)

TM00 and TM01 are 16-bit read-only registers that count the count pulses.

The counter is incremented in synchronization with the rising edge of the input clock. If the count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read. 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 clear & start mode entered by inputting the valid edge of TI00n

<4> If TM0n and CR00n match in the clear & start mode entered on a match between TM0n and CR00n

Remark n = 0, 1

(2) 16-bit capture/compare register 000, 001 (CR000, CR001)

CR000 and CR001 are 16-bit registers that have the functions of both a capture register and a compare register. Whether to be 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. Setting of the TI00n or TI01n valid edge is performed by means of prescaler mode register 0n (PRM0n).

  If CR00n is specified as a capture register and the capture trigger is specified to be the valid edge of the TI00n pin, the situation is as shown in Table 6-2. 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 6-3.

Table 6-2. TI00n Pin Valid Edge and CR00n, CR01n Capture Triggers

ES0n1	ES0n0	TI00n Pin Valid Edge	CR00n Capture Trigger	CR01 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

n = 0, 1

Table 6-3. 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

n = 0, 1

CR00n is set by a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input makes the value of CR00n undefined.

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

2. If the value of CR00n after changing is smaller than the value of 16-bit timer counter 0n (TM0n), TM0n continues counting and overflows, then starts counting again from 0. Also, if the value of CR00n after changing is less than the value before changing, it is necessary to restart the timer after CR00n changes.

3. When P54 (P56) is used as the valid edge of TI000 (TI001), it cannot be used as the timer output (TO00 (TO01)). Also, if it is used as TO00 (TO01), it cannot be used as the valid edge of TI000 (TI001).

Remark n = 0, 1

(3) 16-bit capture/compare register 010, 011 (CR010, CR011)

CR010 and CR011 are 16-bit registers that have the functions of both a capture register and a compare register. Whether to be 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 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 means of prescaler mode register 0n (PRM0n).

CR01n is set by a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input makes the value of CR01n undefined.

Caution **In the clear & start mode entered on a match between TM0n and CR00n, set CR01n to a value other than 0000H. However, in the free-running mode and the clear mode of the valid edge of TI00n, if CR01n is set to 0000H, an interrupt request (INTTM01n) is generated after the overflow (FFFFH).**

Remark n = 0, 1

6.4 Registers Controlling 16-Bit Timer/Event Counter

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

- 16-bit timer mode control register 00, 01 (TMC00, TMC01)
- Capture/compare control register 00, 01 (CRC00, CRC01)
- Timer output control register 00, 01 (TOC00, TOC01)
- Prescaler mode register 00, 01 (PRM00, PRM01)
- Port mode register 5 (PM5)

(1) 16-bit timer mode control register 00, 01 (TMC00, TMC01)

These registers set the 16-bit timer operating mode, 16-bit timer counter 00, 01 (TM00, TM01) clear mode, and output timing, and detect an overflow.
TMC00 and TMC01 are set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TMC00 and TMC01 to 00H.

Caution **16-bit timer counter 0n (TM0n) starts operating the instant that TMC0n2 and TMC0n3 (n = 0, 1) are set to a value other than 0 (operation stop mode). To stop operation, set TMC0n2 and TMC0n3 to 0.**

Figure 6-3. Format of 16-Bit Timer Mode Control Register 00

Symbol	7	6	5	4	3	2	1	⓪	Address	After reset	R/W
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00	FF60H	00H	R/W

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 on TI000 valid edge	—	
1	0	1			
1	1	0	Clear & start 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. Write to a bit other than the OVF00 flag after timer operation stops.

2. Set the valid edge of the TI000/TO00/INTP4/P54 pin with prescaler mode register 00 (PRM00).

3. If clear & start mode entered on a match between TM00 and CR000 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 TO00: 16-bit timer/event counter 00 output pin

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

TM00: 16-bit timer counter 00

CR000: 16-bit capture/compare register 000

CR010: 16-bit capture/compare register 010

Figure 6-4. Format of 16-Bit Timer Mode Control Register 01

Symbol	7	6	5	4	3	2	1	⓪	Address	After reset	R/W
TMC01	0	0	0	0	TMC013	TMC012	TMC011	OVF01	FF68H	00H	R/W

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 on TI001 valid edge	—	
1	0	1			
1	1	0	Clear & start 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. Write to a bit other than the OVF01 flag after timer operation stops.

2. Set the valid edge of the TI001/TO01/INTP6/P56 pin with prescaler mode register 01 (PRM01).

3. If clear & start mode entered on a match between TM01 and CR001 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	
TO01:	16-bit timer/event counter 01 output pin
TI001:	16-bit timer/event counter 01 input pin
TM01:	16-bit timer counter 01
CR001:	16-bit capture/compare register 001
CR011:	16-bit capture/compare register 011

(2) Capture/compare control register 00, 01 (CRC00, CRC01)

These registers control the operation of the 16-bit capture/compare registers (CR000, CR010, CR001, CR011).

CRC00 and CRC01 are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CRC00 and CRC01 to 00H.

Figure 6-5. Format of Capture/Compare Control Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000	FF62H	00H	R/W

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 inverted phase of valid edge of TI000

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 clear & start mode entered 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. If both the rising and falling edges are selected as the valid edges of TI000, capture is not performed.

4. In order to ensure the capture operation, a pulse longer than two clocks of the count clock specified by prescaler mode register 00 (PRM00) is required for a capture trigger.

Figure 6-6. Format of Capture/Compare Control Register 01

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CRC01	0	0	0	0	0	CRC012	CRC011	CRC010	FF6AH	00H	R/W

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 inverted phase of valid edge of TI001

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 clear & start mode 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. If both the rising and falling edges are selected as the valid edges of TI001, capture is not performed.

4. In order to ensure the capture operation, a pulse longer than two clocks of the count clock specified by prescaler mode register 01 (PRM01) is required for a capture trigger.

(3) Timer output control register 00, 01 (TOC00, TOC01)

These registers control the operation of the 16-bit timer/event counter 00, 01 output control circuit, including R-S type flip-flop (LV0) setting/resetting, output inversion enabling/disabling, and 16-bit timer/event counter 00, 01 timer output enabling/disabling.

TOC00 and TOC01 are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TOC00 and TOC01 to 00H.

Figure 6-7. Format of Timer Output Control Register 00

Symbol	7	6	5	4	③	②	1	⓪	Address	After reset	R/W
TOC00	0	0	0	TOC004	LVS00	LVR00	TOC001	TOE00	FF63H	00H	R/W

TOC004	Timer output F/F control by match of CR010 and TM00
0	Inversion operation disabled
1	Inversion operation enabled

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 by match of CR000 and TM00
0	Inversion operation disabled
1	Inversion operation enabled

TOE00	16-bit timer/event counter 00 timer output control
0	Output disabled (Output set to level 0)
1	Output enabled

Cautions 1. Timer operation must be stopped before setting TOC00.

2. Be sure to set bits 5 to 7 of TOC00 to 0.

Remark If LVS00 and LVR00 are read after data is set, they will be 0.

Figure 6-8. Format of Timer Output Control Register 01

Symbol	7	6	5	4	③	②	1	⓪	Address	After reset	R/W
TOC01	0	0	0	TOC014	LVS01	LVR01	TOC011	TOE01	FF6BH	00H	R/W

TOC014	Timer output F/F control by match of CR011 and TM01
0	Inversion operation disabled
1	Inversion operation enabled

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 by match of CR001 and TM01
0	Inversion operation disabled
1	Inversion operation enabled

TOE01	16-bit timer/event counter 01 timer output control
0	Output disabled (Output set to level 0)
1	Output enabled

Cautions 1. Timer operation must be stopped before setting TOC01.

2. Be sure to set bits 5 to 7 of TOC01 to 0.

Remark If LVS01 and LVR01 are read after data is set, they will be 0.

(4) Prescaler mode register 00, 01 (PRM00, PRM01)

This register is used to set the 16-bit timer counter 00, 01 (TM00, TM01) count clock and TI000, TI001 input valid edges.

PRM00 and PRM01 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PRM00 and PRM01 to 00H.

★ **Figure 6-9. Format of Prescaler Mode Register 00**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000	FF61H	00H	R/W

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		
			At fx = 12 MHz[Note 1]	At fx = 8.38 MHz
0	0	fx	12 MHz	8.38 MHz
0	1	$fx/2^2$	3 MHz	2.09 MHz
1	0	$fx/2^5$	375 kHz	262 kHz
1	1	TI000 valid edge[Note 2]		

Notes 1. Expanded-specification products only

2. The external clock requires a pulse longer than two internal clocks ($fx/2^4$).

Cautions 1. If the valid edge of TI000 is set for the count clock, do not set it for the clear & start mode or the capture trigger. Also, the P54/TI000/TO00/INTP4 pin cannot be used as a timer output (TO00).

2. PRM00 should be set only after timer operation has been stopped.

3. If the TI000 pin or TI010 pin is high level immediately after system reset, and the rising edge or both edges are specified as the valid edge of TI000 pin or TI010 pin thus enabling operation of 16-bit timer counter 00 (TM00), the rising edge will be detected immediately. Care is therefore needed if the TI000 pin or TI010 pin is pulled up. When operation is enabled again after being stopped, the rising edge cannot be detected.

Remarks 1. fx: System clock oscillation frequency

2. TI000, TI010: Input pins of 16-bit timer/event counter 00

★

Figure 6-10. Format of Prescaler Mode Register 01

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PRM01	ES111	ES110	ES011	ES010	0	0	PRM011	PRM010	FF61H	00H	R/W

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		
			At f_X = 12 MHz[Note 1]	At f_X = 8.38 MHz
0	0	f_X	12 MHz	8.38 MHz
0	1	$f_X/2^2$	3 MHz	2.09 MHz
1	0	$f_X/2^5$	375 kHz	262 kHz
1	1	TI001 valid edge[Note 2]		

Notes 1. Expanded-specification products only

2. The external clock requires a pulse longer than two internal clocks ($f_X/2^4$).

Cautions 1. If the valid edge of TI001 is set for the count clock, do not set it for the clear and start mode or the capture trigger. Also, the P56/TI001/TO01/INTP6 pin cannot be used as a timer output (TO01).

2. PRM01 should be set only after timer operation has been stopped.

3. If the TI001 pin or TI011 pin is high level immediately after system reset, and the rising edge or both edges are specified as the valid edge of the TI001 pin or TI011 pin thus enabling operation of 16-bit timer counter 01 (TM01), the rising edge will be detected immediately. Care is therefore needed if the TI001 pin or TI011 pin is pulled up. When operation is enabled again after being stopped, the rising edge cannot be detected.

Remarks 1. f_X: System clock oscillation frequency

2. TI001, TI011: Input pins of 16-bit timer/event counter 01

(5) Port mode register 5 (PM5)

This register sets port 5 to input/output in 1-bit units.

When using the P54/TO00/TI000/INTP4 pin or P56/TO01/TI001/INTP6 pin for timer output, set PM54 or PM56 and the output latch of P54 or P56 to 0.

PM5 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM5 to FFH.

Figure 6-11. Format of Port Mode Register 5

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W

PM5n	P5n pin input/output mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.5 Operation of 16-Bit Timer/Event Counter

6.5.1 Interval timer operation

Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 6-12 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value preset in 16-bit capture/compare register 00n (CR00n) as the interval.

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

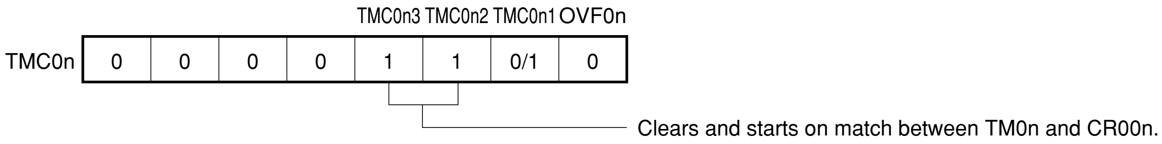
The count clock of TM0n can be selected with bits 0 and 1 (PRM0n0, PRM0n1) of prescaler mode register 0n (PRM0n).

See **6.6 Notes on 16-Bit Timer/Event Counter (3) Operation after compare register change during timer count operation** about the operation when the compare register value is changed during timer count operation.

Remark n = 0, 1

Figure 6-12. Control Register Settings for Interval Timer Operation

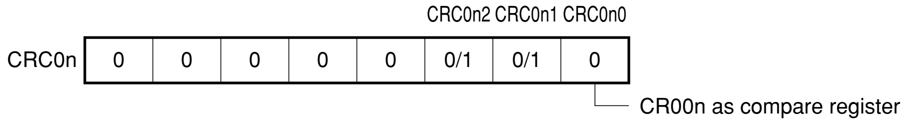
(a) 16-bit timer mode control register 0n (TMC0n)



					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)



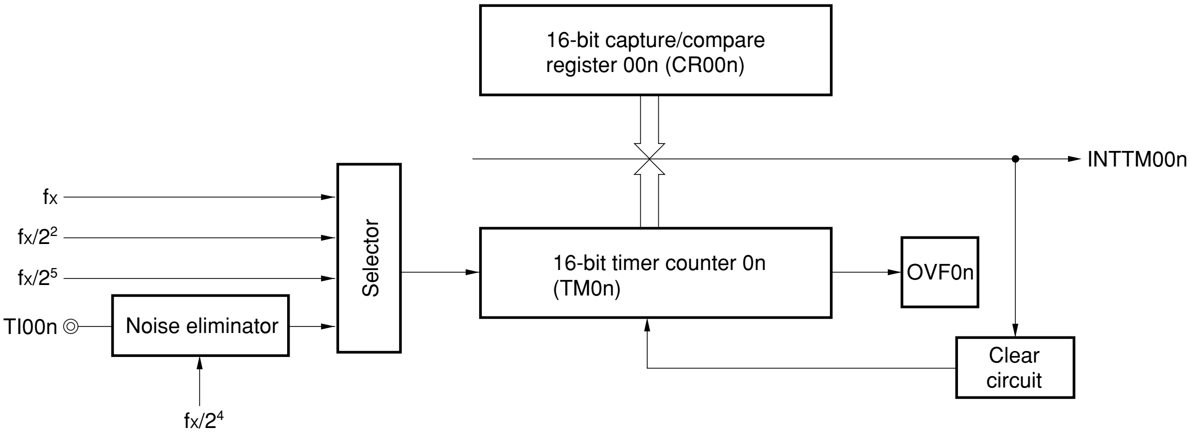
						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0/1	0/1	0

CR00n as compare register

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See Figures 6-3 to 6-6.

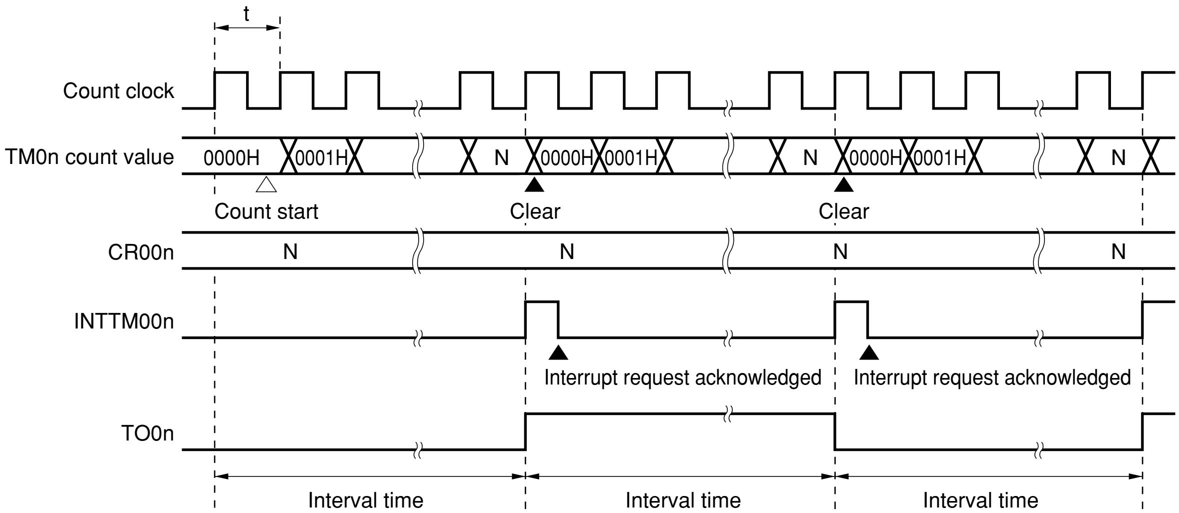
2. n = 0, 1

Figure 6-13. Interval Timer Configuration Diagram


16-bit capture/compare register 00n (CR00n)
INTTM00n
fx
fx/2²
fx/2⁵
TI00n
Noise eliminator
Selector
16-bit timer counter 0n (TM0n)
OVF0n
Clear circuit
fx/2⁴

Remark   n = 0, 1

Figure 6-14. Timing of Interval Timer Operation



Remarks 1. Interval time = (N + 1) × t: N = 0001H to FFFFH

2. n = 0, 1

6.5.2 PPG output operation

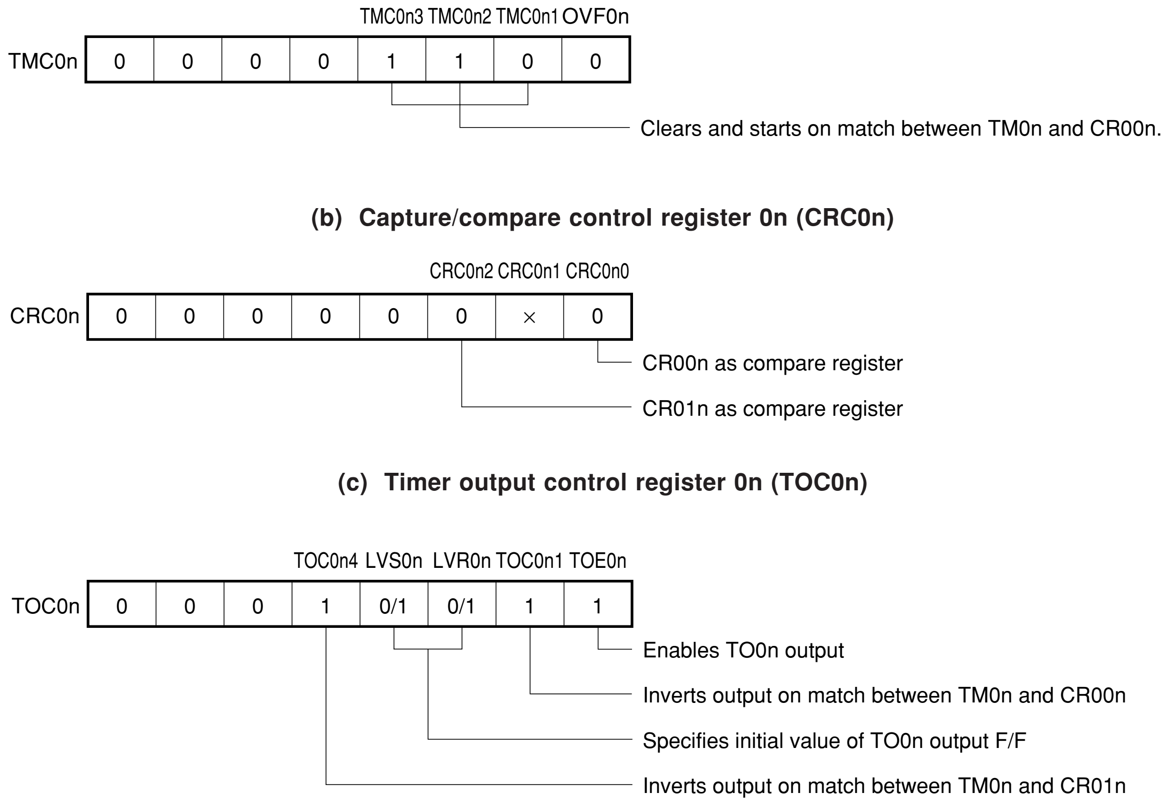
Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 6-15 allows operation as PPG (Programmable Pulse Generator) output.

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

Remark n = 0, 1

Figure 6-15. Control Register Settings for PPG Output Operation

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



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

(c) Timer output control register 0n (TOC0n)

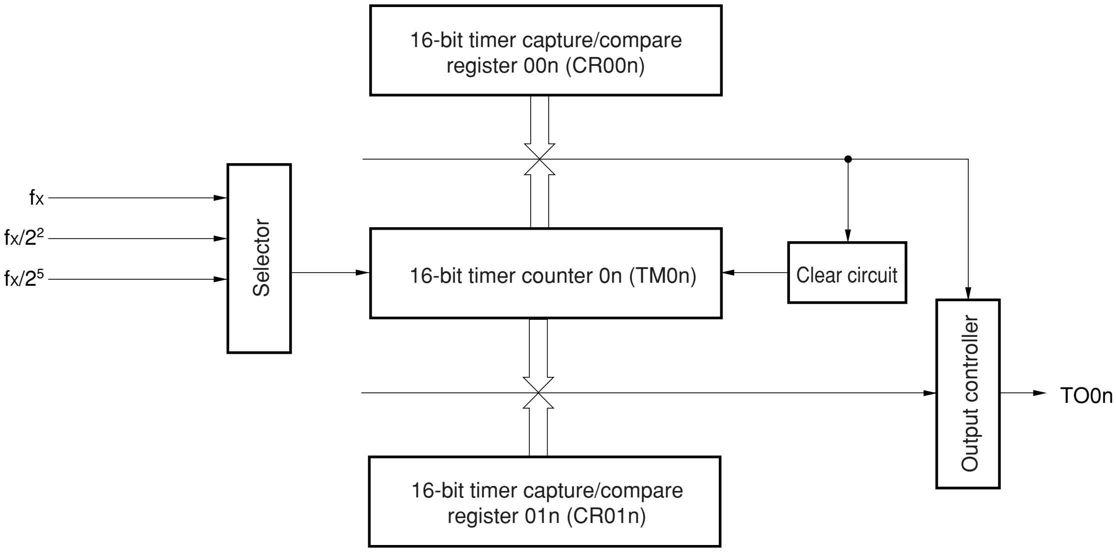
TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 1 1 0 0
Clears and starts on match between TM0n and CR00n.
CRC0n2 CRC0n1 CRC0n0
CRC0n
0 0 0 0 0 0 × 0
CR00n as compare register
CR01n as compare register
TOC0n4 LVS0n LVR0n TOC0n1 TOE0n
TOC0n
0 0 0 1 0/1 0/1 1 1
Enables TO0n output
Inverts output on match between TM0n and CR00n
Specifies initial value of TO0n output F/F
Inverts output on match between TM0n and CR01n

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 by PPG output becomes (CR00n setting + 1), and the duty becomes (CR01n setting + 1)/(CR00n setting + 1).

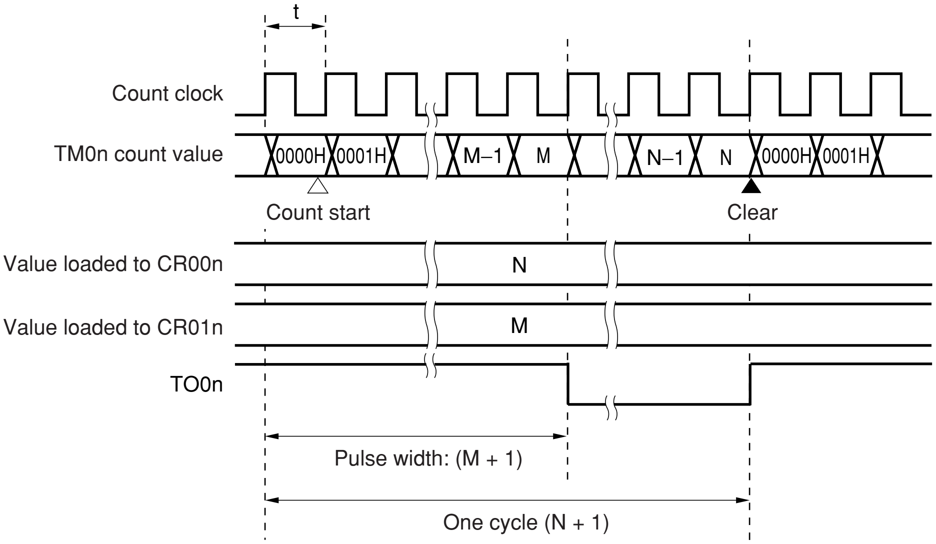
Remark × : don't care
n = 0, 1

★ **Figure 6-16. Configuration Diagram of PPG Output**


16-bit timer capture/compare register 00n (CR00n)
fx
fx/2²
fx/2⁵
Selector
16-bit timer counter 0n (TM0n)
Clear circuit
Output controller
TO0n
16-bit timer capture/compare register 01n (CR01n)

Remark n = 0, 1

★ **Figure 6-17. PPG Output Operation Timing**



Remarks 1. 0000H < M < N ≤ FFFFH

2. n = 0, 1

6.5.3 Pulse width measurement operation

It is possible to measure the pulse width of the signals input to the TI00n and TI01n pins 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

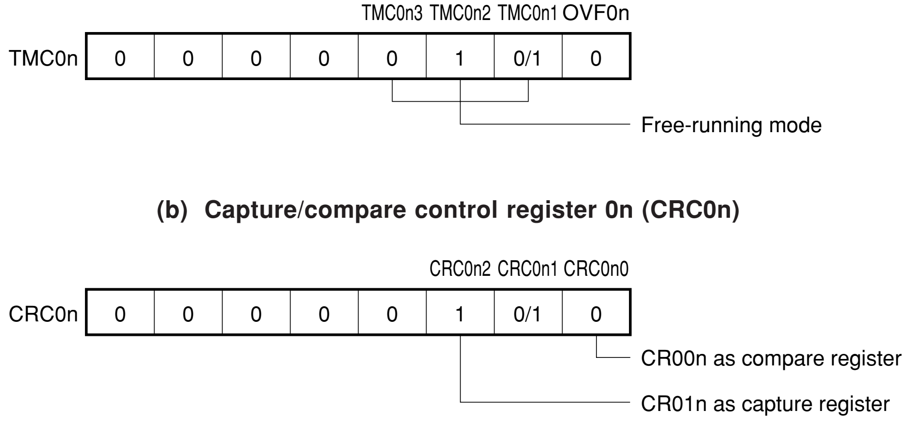
When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 6-18**), and the edge specified by prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

The valid edge of the TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of PRM0n, and the rising edge, falling edge or both edges can be selected.

When sampling is performed at the count clock cycle selected by PRM0n and the valid level of the TI00n pin is detected twice, the first capture operation is performed, resulting in the elimination of short pulse width noise.

Remark n = 0, 1

Figure 6-18. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register



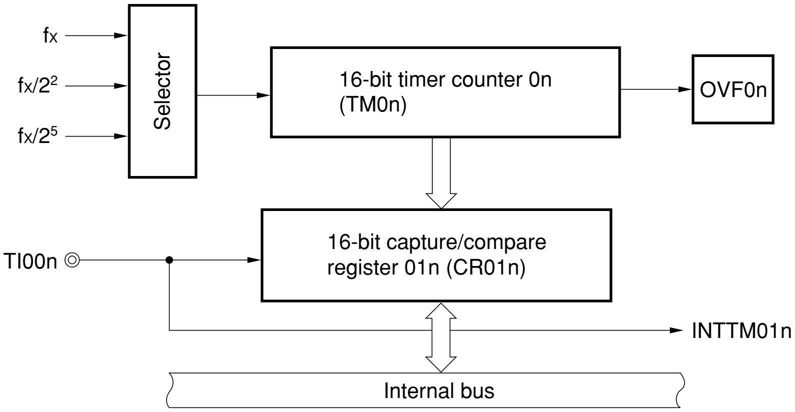
(a) 16-bit timer mode control register 0n (TMC0n)

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

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

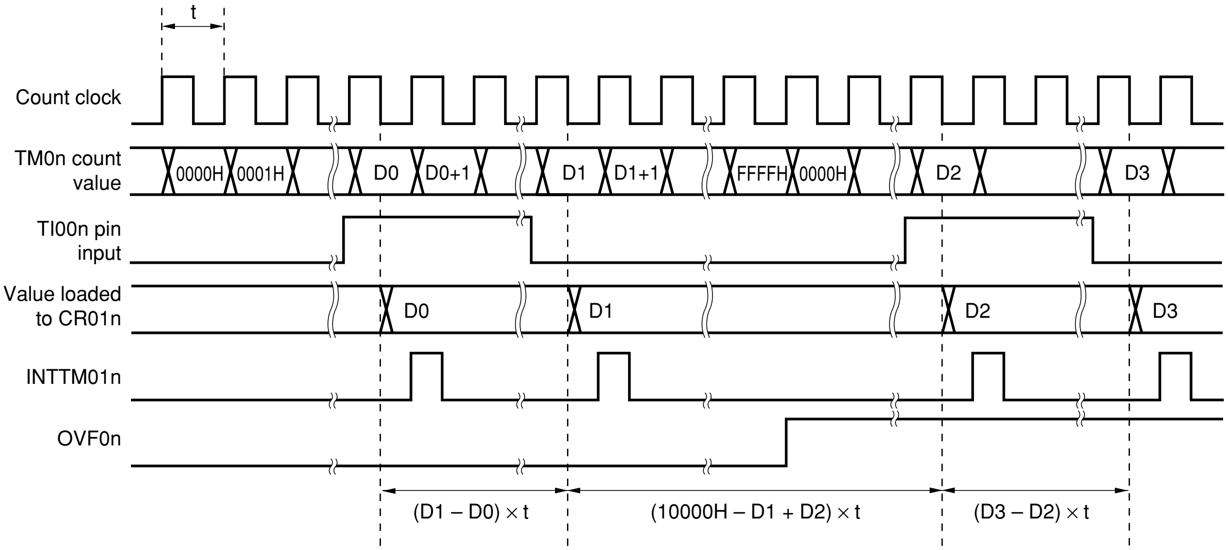
2. n = 0, 1

Figure 6-19. Configuration Diagram for Pulse Width Measurement with Free-Running Counter



Remark n = 0, 1

Figure 6-20. Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)



Remark n = 0, 1

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

When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 6-21**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00n and the TI01n pins.

When the edge specified by bits 4 and 5 (ES0n0, ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

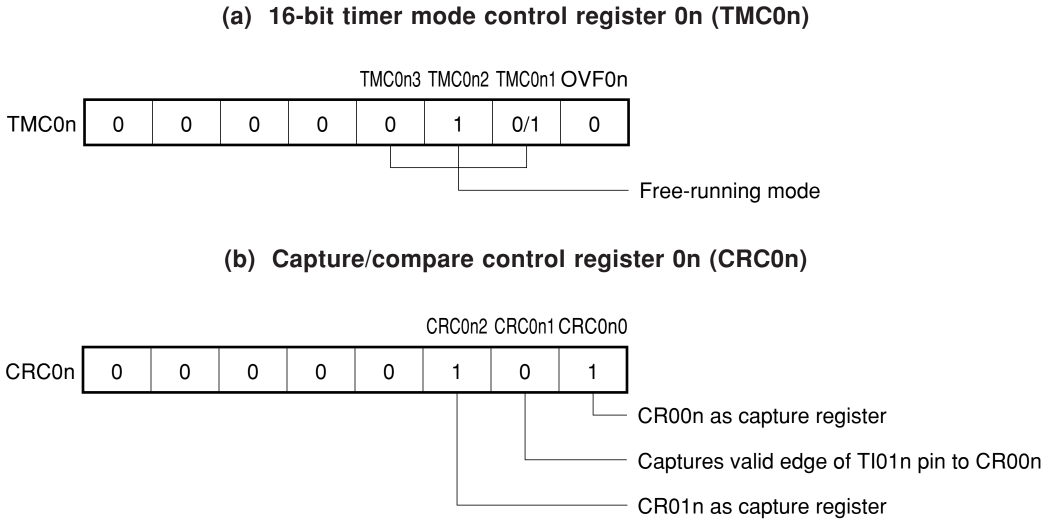
Also, when the edge specified by bits 6 and 7 (ES1n0, ES1n1) of PRM0n is input to the TI01n pin, the value of TM0n is taken into 16-bit capture/compare register 00n (CR00n) and an external interrupt request signal (INTTM00n) is set.

The valid edges of the TI00n and TI01n pins are specified by bits 4 and 5 (ES0n0, ES0n1), and bits 6 and 7 (ES1n0, ES1n1) of PRM0n, respectively. It is possible to select the rising edge, falling edge or both edges as the valid edge.

When sampling is performed at the count clock cycle selected by PRM0n and the valid level of the TI00n or TI01 pin is detected twice, the first capture operation is performed, resulting in the elimination of short pulse width noise.

Remark n = 0, 1

Figure 6-21. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter



(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
0
1
CR00n as capture register
Captures valid edge of TI01n pin to CR00n
CR01n as capture register

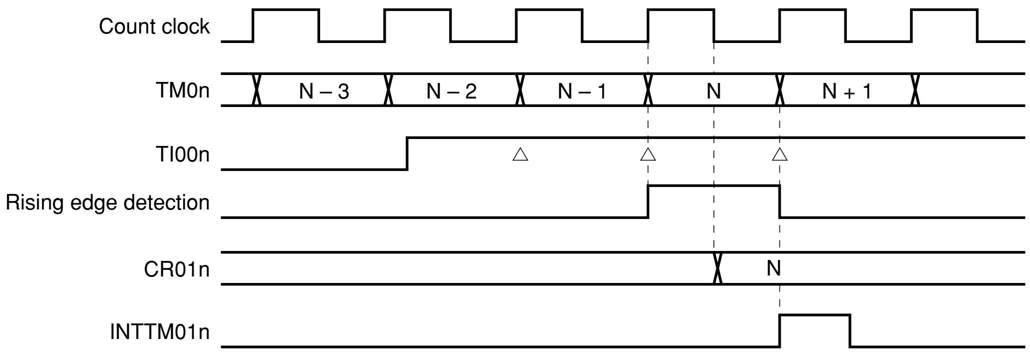
Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 6-3** and **6-4** for details.

2. n = 0, 1

- **Capture operation (Free-running mode)**

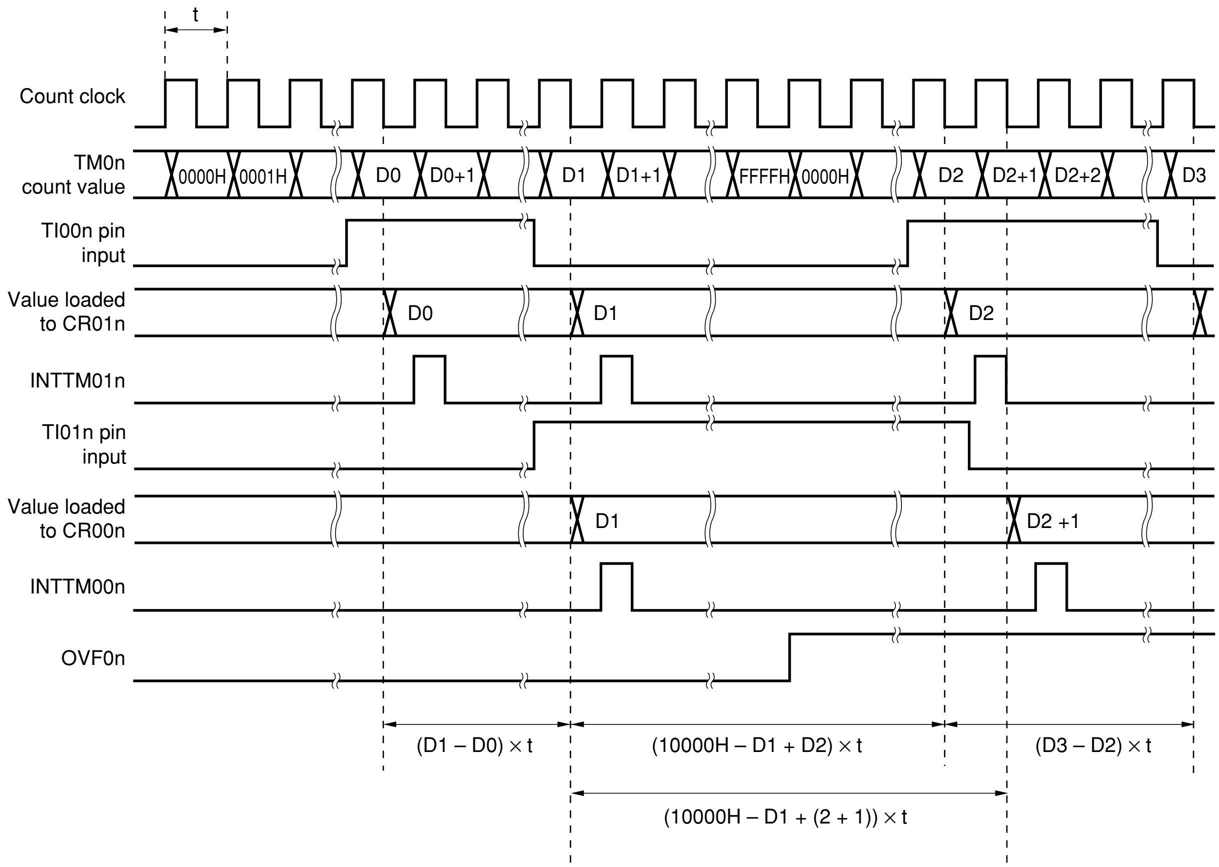
  The capture register operation of when the capture trigger is input is shown.

Figure 6-22. CR01n Capture Operation with Rising Edge Specified



Remark n = 0, 1

Figure 6-23. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



Remark n = 0, 1

(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 register settings in **Figure 6-24**), 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, ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

Also, when the inverse edge to that of the capture operation to CR01n is input, the value of TM0n is taken into 16-bit capture/compare register 00n (CR00n).

The valid edge of the TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of PRM0n, and it is possible to select the rising edge or falling edge.

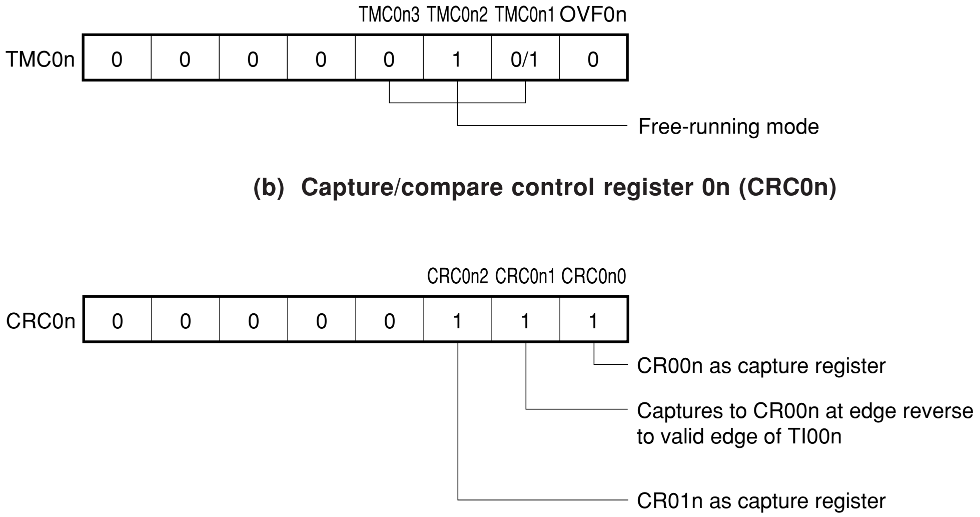
When sampling is performed at the count clock cycle selected by PRM0n and the valid level of the TI00n pin is detected twice, the first capture operation is performed, resulting in the elimination of short pulse width noise.

Caution If the valid edge of the TI00n pin is specified to be both the rising and falling edges, capture/compare register 00n (CR00n) cannot perform the capture operation.

Remark n = 0, 1

Figure 6-24. 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

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

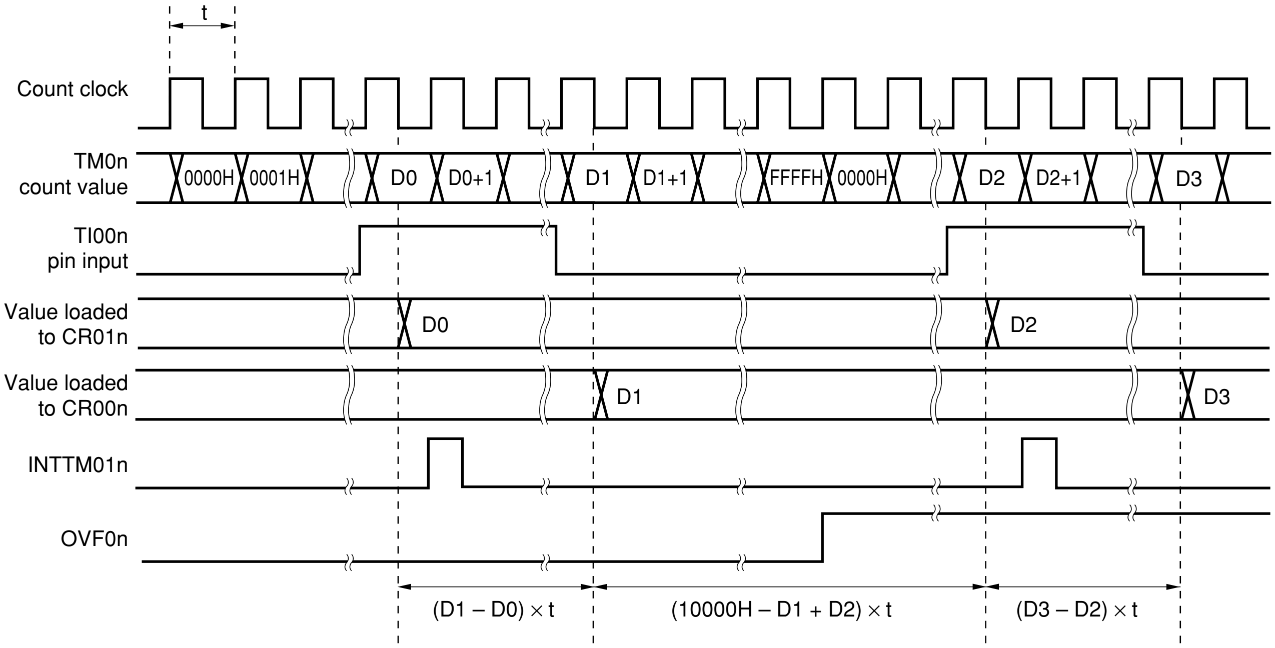
						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	1	1	1

CR00n as capture register
Captures to CR00n at edge reverse to valid edge of TI00n
CR01n as capture register

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 6-3** and **6-4** for details.

2. n = 0, 1

Figure 6-25. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Remark n = 0, 1

(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 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 (see register settings in **Figure 6-26**).

The valid edge of the TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of prescaler mode register 0n (PRM0n), and it is possible to select either the rising edge or falling edge.

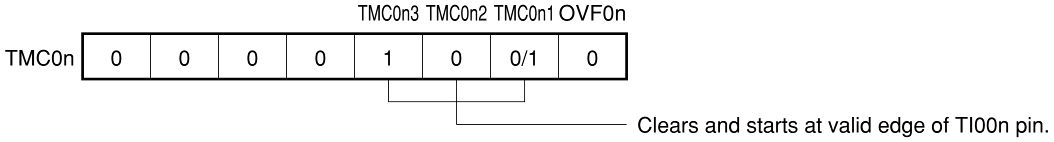
When sampling is performed at the count clock cycle selected by PRM0n and the valid level of the TI00n pin is detected twice, the first capture operation is performed, resulting in the elimination of short pulse width noise.

Caution **If the valid edge of the TI00n pin is specified to be both the rising and falling edges, capture/compare register 00n (CR00n) cannot perform the capture operation.**

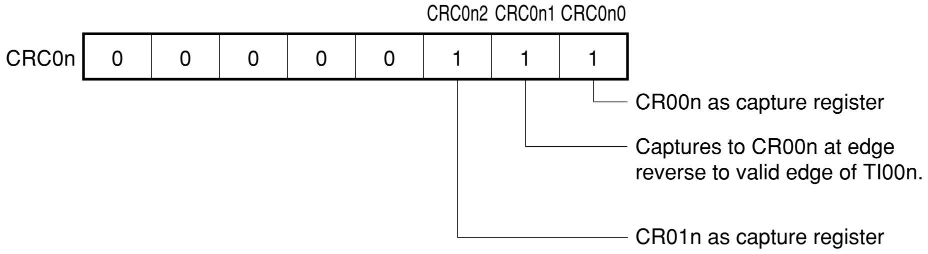
Remark n = 0, 1

Figure 6-26. Control Register Settings for Pulse Width Measurement by Means of Restart

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



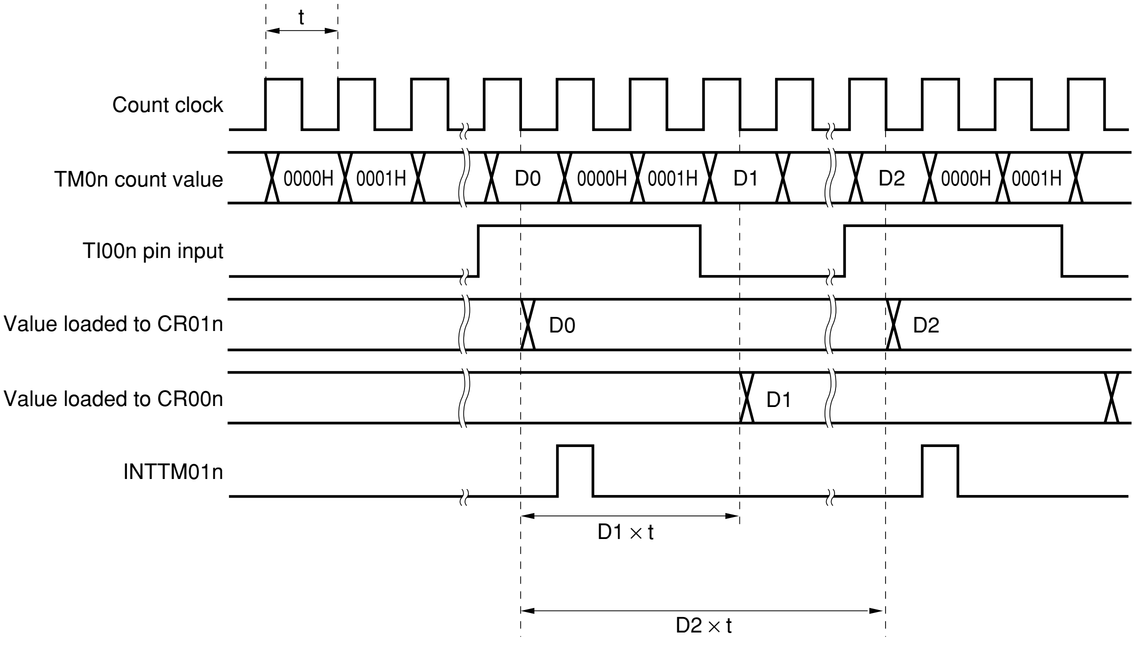
(b) Capture/compare control register 0n (CRC0n)


CRC0n2
CRC0n1
CRC0n0
CRC0n
0
0
0
0
0
1
1
1
CR00n as capture register
Captures to CR00n at edge
reverse to valid edge of TI00n.
CR01n as capture register

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 6-3** and **6-4** for details.

2. n = 0, 1

Figure 6-27. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



Remark   n = 0, 1

6.5.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00n pin by 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 capture/compare register 00n (CR00n) value, TM0n is cleared to 0 and an interrupt request signal (INTTM00n) is generated.

A value other than 0000H should be set for CR00n (a 1-pulse count operation is not possible).

Specify the valid edge of the TI00n pin using bits 4 and 5 (ES0n0, ES0n1) of PRM0n. It is possible to select the rising edge, falling edge or both edges.

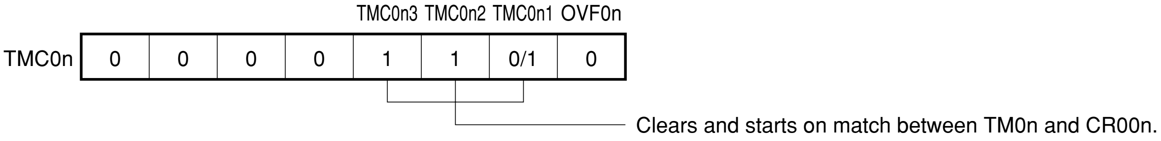
When sampling is performed at the internal clock (fx/2^4) and the valid level of the TI00n pin is detected twice, the first capture operation is performed, resulting in the elimination of short pulse width noise.

Caution When the 16-bit timer/event counter is being used as an external event counter, the P54/TI000/TO00/INTP4 pin (P56/TI001/TO01/INTP6 pin) cannot be used for timer output (TO00 (TO01)).

Remark n = 0, 1

Figure 6-28. Control Register Settings in External Event Counter Mode

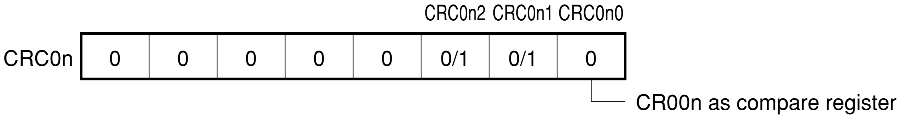
(a) 16-bit timer mode control register 0n (TMC0n)



					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)



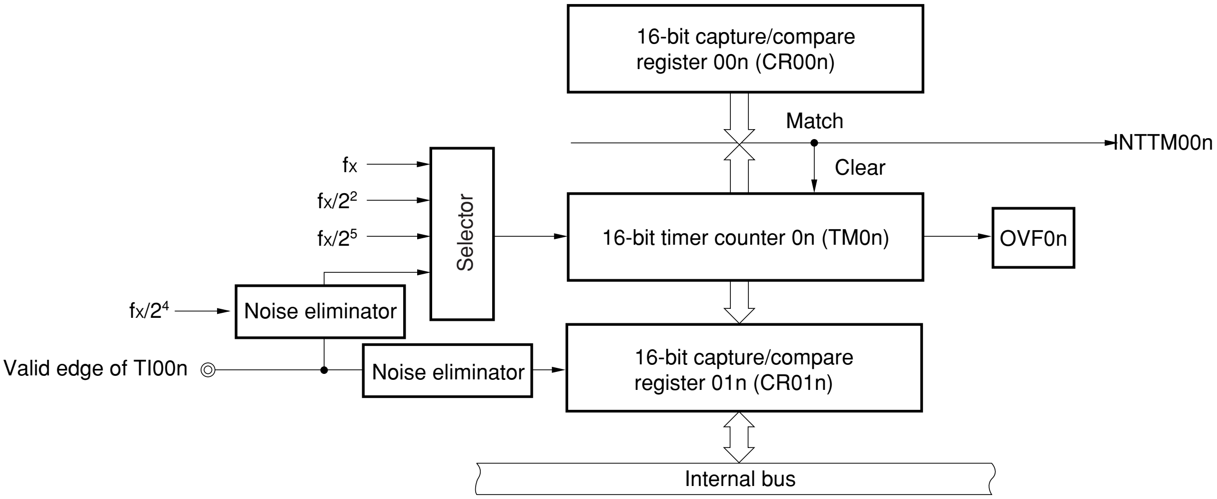
						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0/1	0/1	0

CR00n as compare register

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See **Figures 6-3** to **6-6** for details.

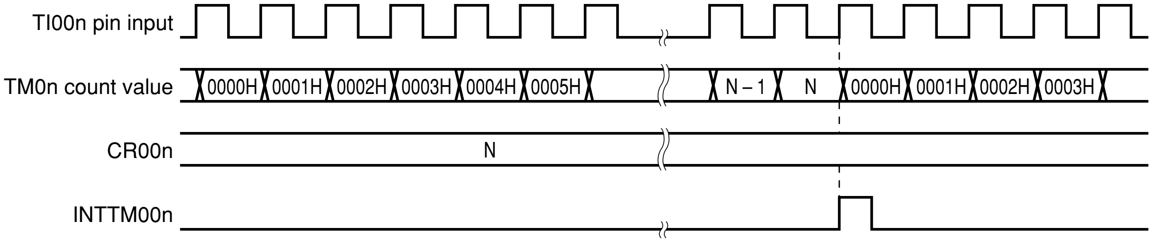
2. n = 0, 1

Figure 6-29. External Event Counter Configuration Diagram


16-bit capture/compare register 00n (CR00n)
Match
INTTM00n
Clear
fx
fx/2²
fx/2⁵
Selector
16-bit timer counter 0n (TM0n)
OVF0n
fx/2⁴
Noise eliminator
Valid edge of TI00n
Noise eliminator
16-bit capture/compare register 01n (CR01n)
Internal bus

Remark n = 0, 1

Figure 6-30. External Event Counter Operation Timings (with Rising Edge Specified)



Caution When reading the external event counter count value, TM0n (n = 0, 1) should be read.

Remark n = 0, 1

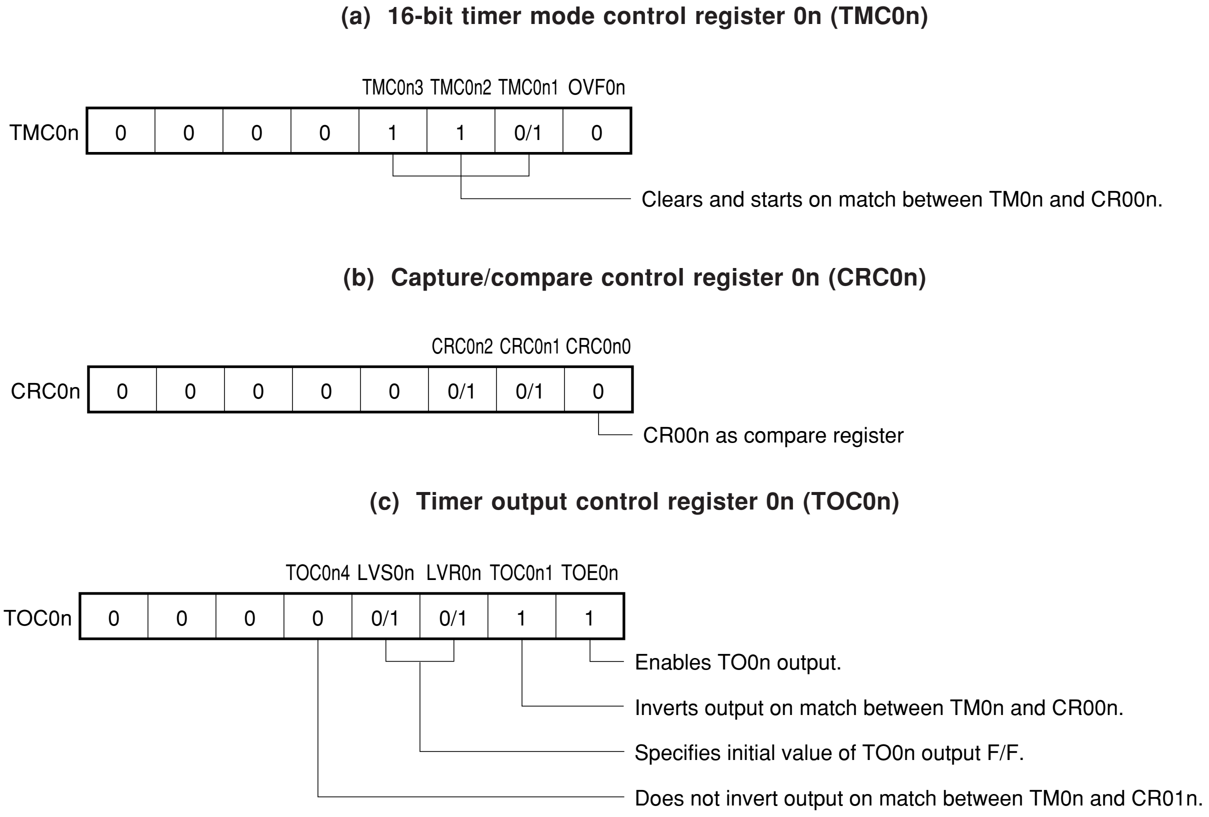
6.5.5 Square-wave output operation

This is an operation whereby a square wave with any selected frequency is output using the count value preset to 16-bit capture/compare register 00n (CR00n) as the interval.

The TO0n pin output status is inverted at the intervals of the count value preset to CR00n by setting bit 0 (TOE0n) and bit 1 (TOC0n1) of timer output control register 0n (TOC0n) to 1. This enables a square wave with any selected frequency to be output.

Remark n = 0, 1

Figure 6-31. 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 0/1 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 as compare register

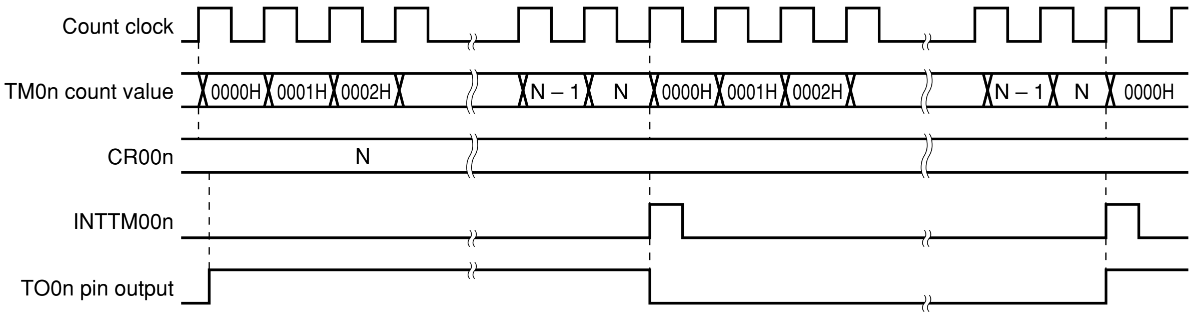
(c) Timer output control register 0n (TOC0n)

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.

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See **Figures 6-3** to **6-8** for details.

2. n = 0, 1

Figure 6-32. Square-Wave Output Operation Timing



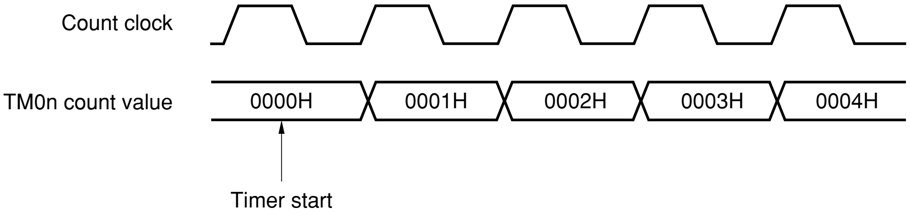
Remark n = 0, 1

6.6 Notes on 16-Bit Timer/Event Counter

(1) Timer start errors

An error of a maximum of one clock may occur during the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 0n (TM0n: n = 0, 1) is started asynchronously to the count clock.

Figure 6-33. 16-Bit Timer Counter Start Timing



Remark n = 0, 1

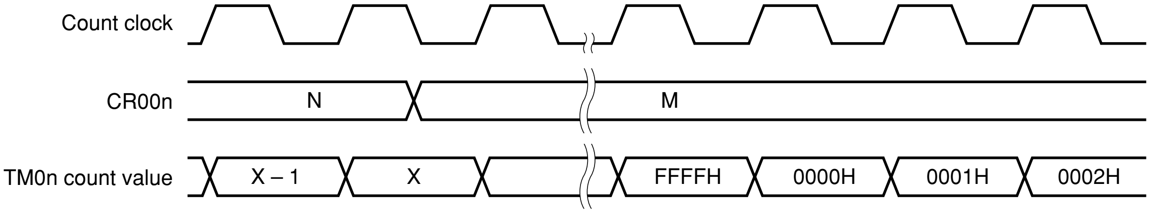
(2) 16-bit compare register setting (clear & start mode entered on a match between TM0n and CR00n)

Set 16-bit capture/compare registers 00n, 01n (CR00n, CR01n: n = 0, 1) to other than 0000H. This means a 1-pulse count operation cannot be performed when they are used as event counters.

(3) Operation after compare register change during timer count operation

If the value after the change of 16-bit capture/compare register 00n (CR00n: n = 0, 1) is smaller than that of 16-bit timer counter 0n (TM0n: n = 0, 1), TM0n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after the CR00n change is smaller than that (N) before the change, it is necessary to reset the timer to restart after CR00n is changed.

Figure 6-34. Timing After Change of Compare Register During Timer Count Operation



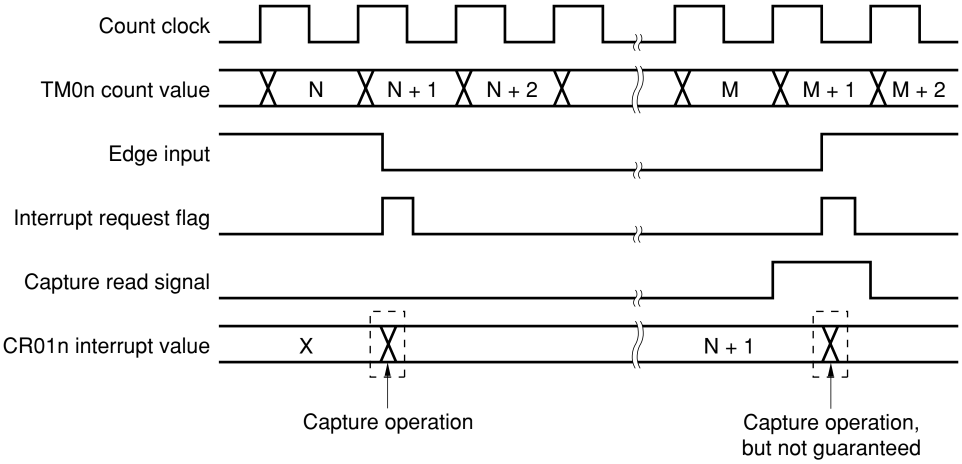
Remark N > X > M
n = 0, 1

(4) Capture register data retention timing

If the valid edge of the TI00n pin is input during 16-bit capture/compare register 01n (CR01n) read, CR01n performs a capture operation, but the capture value at this time is not guaranteed. However, the interrupt request signal (TMIF01n) is set upon detection of the valid edge.

Remark n = 0, 1

Figure 6-35. Capture Register Data Retention Timing



Remark n = 0, 1

(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, and stopping timer operation. The valid edge is set with bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

Remark n = 0, 1

(6) Operation of OVF0n flag

<1> The OVF0n flag (bit 6 of 16-bit timer mode control register 0n (TMC0n)) is set to 1 the next time.

One of clear & start mode entered on match between TM0n and CR00n, clear & start mode entered at the valid edge of TI00n, and free-running mode is selected.

↓

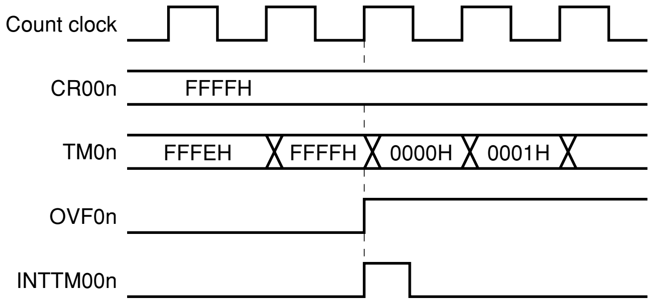
CR00n is set to FFFFH.

↓

When TM0n is counted up from FFFFH to 0000H.

Remark n = 0, 1

Figure 6-36. Operation Timing of OVF0n Flag



Remark n = 0, 1

<2> After TM0n overflows, it is reset and the clear instruction becomes invalid even though the OVF0n flag is cleared before the next count clock (before TM0n becomes 0001H).

Remark n = 0, 1

(7) Conflicting Operations

<1> Conflicting operations between the read time of 16-bit capture/compare register 00n, 01n (CR00n, CR01n) and capture trigger input (CR00n and CR01n used as capture register)

Capture trigger input has priority. The data read from CR00n and CR01n is undefined.

<2> Match timing of conflicting operations between the write period of 16-bit capture/compare register 00n, 01n (CR00n, CR01n) and 16-bit timer counter 0n (TM0n) (CR00n and CR01n used as compare register)

Match judgement is not performed normally. Do not write any data to CR00n and CR01n near the match timing.

Remark n = 0, 1

(8) Timer operation

<1> Even if 16-bit timer counter 0n (TM0n) is read, the value is not captured in 16-bit capture/compare register 01n (CR01n).

<2> Regardless of the operation mode of the CPU, if the timer is stopped, the noise of the external interrupt request input is not removed.

Remark n = 0, 1

(9) Capture operation

<1> When the valid edge of TI00n (n = 0, 1) is specified for the count clock, the capture register that specified TI00n as the trigger cannot perform the capture operation normally.

<2> A capture operation is not performed when both the rising and falling edges are specified for the TI00n valid edge.

<3> In order to ensure the capture operation, a pulse longer than two clocks of the count clock specified by prescaler mode register 0n (PRM0n) is required for a capture trigger.

<4> Capture operations start at the falling edge of the count clock. However, interrupt request input (INTTM00n) starts at the rising edge of the count clock.

Remark n = 0, 1

(10) Compare operation

<1> If values are written to 16-bit capture/compare registers 00n and 01n (CR00n, CR01n) at the timing when the set values of CR00n and CR01n and the count value of 16-bit timer counter 0n (TM0n) match generating INTTM00n and INTTM01n, INTTM00n and INTTM01n may not be generated. Therefore, do not write values to CR00n and CR01n repeatedly even if the values are the same.

<2> CR00n and CR01n set in the compare mode cannot perform a capture operation even if the capture trigger is input.

Remark n = 0, 1

(11) Edge detection

<1> When the TI00n pin or the TI01n pin is high level immediately after system reset, and if the rising edge or both edges are specified as the valid edge of the TI00n pin or the TI01n pin, then the rising edge is detected immediately after operation of 16-bit timer counter 0n (TM0n) is enabled. Be careful when the TI00n pin or the TI01n pin is pulled up. When operation is enabled again after once being stopped, the rising edge cannot be detected.

<2> A different sampling clock for noise elimination is used when the TI00n pin valid edge is used for the count clock and when it is used for capture trigger. In the former case, a count clock of $f_X/2^4$ is used, and in the latter case the count clock specified by prescaler mode register 0n (PRM0n) is used for sampling. A capture operation is only performed when sampling is performed at the above described sampling clock and when a valid level is detected twice, thus eliminating noise with a short-pulse width.

Remark n = 0, 1

CHAPTER 7 8-BIT TIMER/EVENT COUNTER

7.1 Outline of 8-Bit Timer/Event Counter

An 8-bit timer/event counter can be used as an interval timer, external event counter, to output a square wave with any selected frequency, and for PWM output. Two 8-bit timer/event counters can be used as one 16-bit timer/event counter.

7.2 Function of 8-Bit Timer/Event Counter

The 8-bit timer/event counters (50, 51, and 52) have the following two modes.

- Mode in which an 8-bit timer/event counter is used alone (single mode)
- Mode in which two or more 8-bit timer/event counters are connected in cascade (16-bit resolution: cascade mode)

These two modes are explained below.

(1) Mode in which an 8-bit timer/event counter is used alone (single mode)

In this mode, the 8-bit timer/event counter can be used for the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

(2) Mode in which TM50 and TM51, or TM51 and TM52 are connected in cascade (16-bit resolution: cascade mode)

By connecting 8-bit timer/event counters in cascade, they can be used as a 16-bit timer/event counter.

In the cascade mode, the 8-bit timer/event counters can be used for the following functions.

- 16-bit resolution interval timer
- 16-bit resolution external event counter
- 16-bit resolution square-wave output

7.3 Configuration of 8-Bit Timer/Event Counter

An 8-bit timer/event counter includes the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counter

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit compare register 5n (CR5n)
Timer output	TO5n
Control registers	8-bit timer mode control register 5n (TMC5n) Timer clock select register 5n (TCL5n) Port mode register 2 (PM2)**Note**

Note Refer to **Figure 4-4 Block Diagram of P20 to P26**.

Remark n = 0 to 2

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

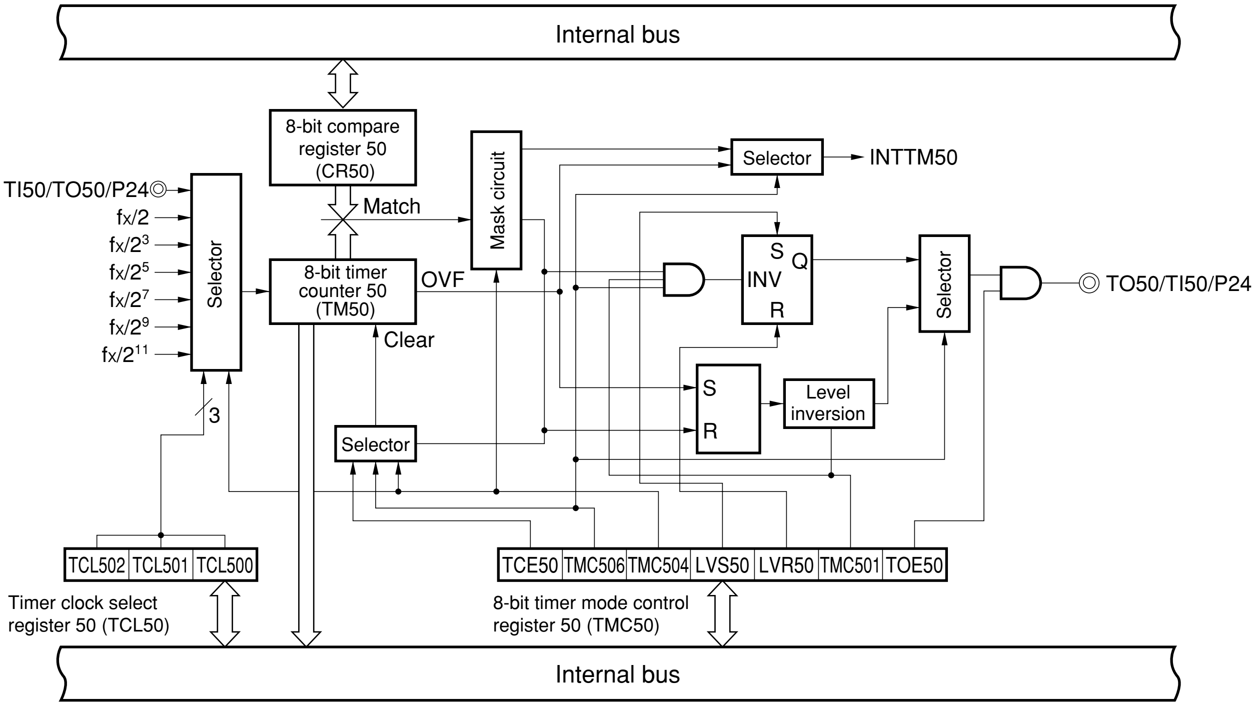


Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51

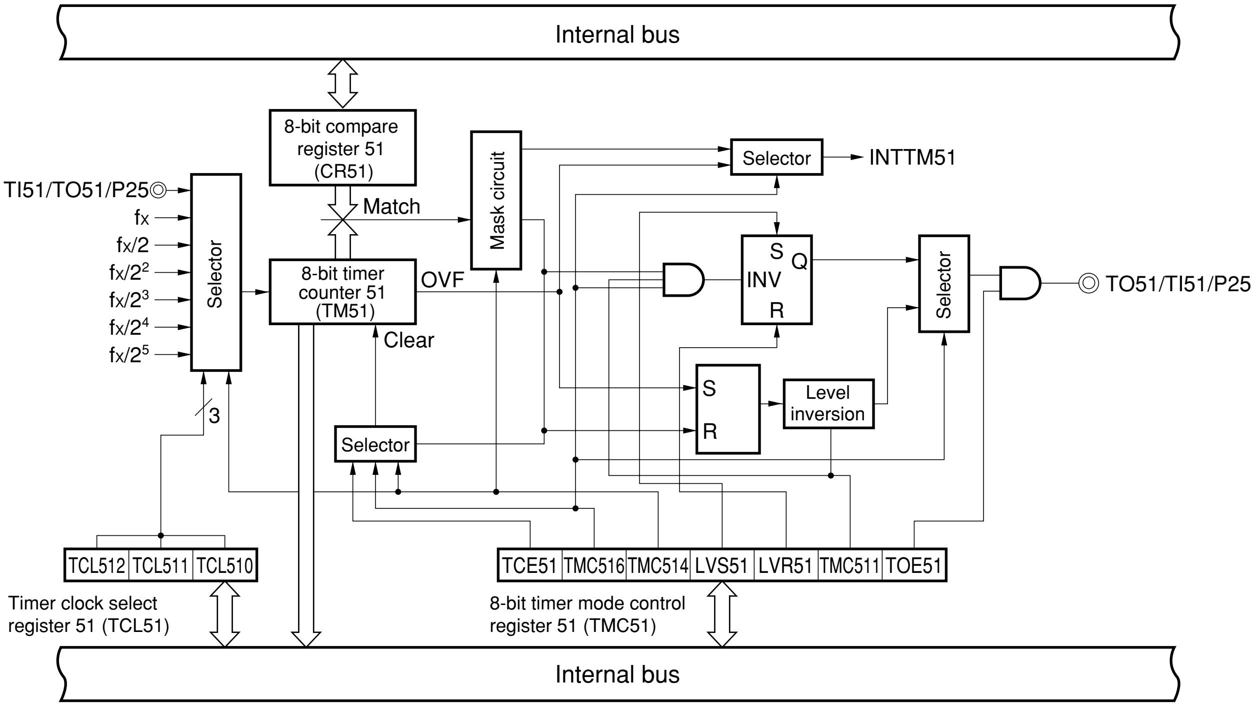
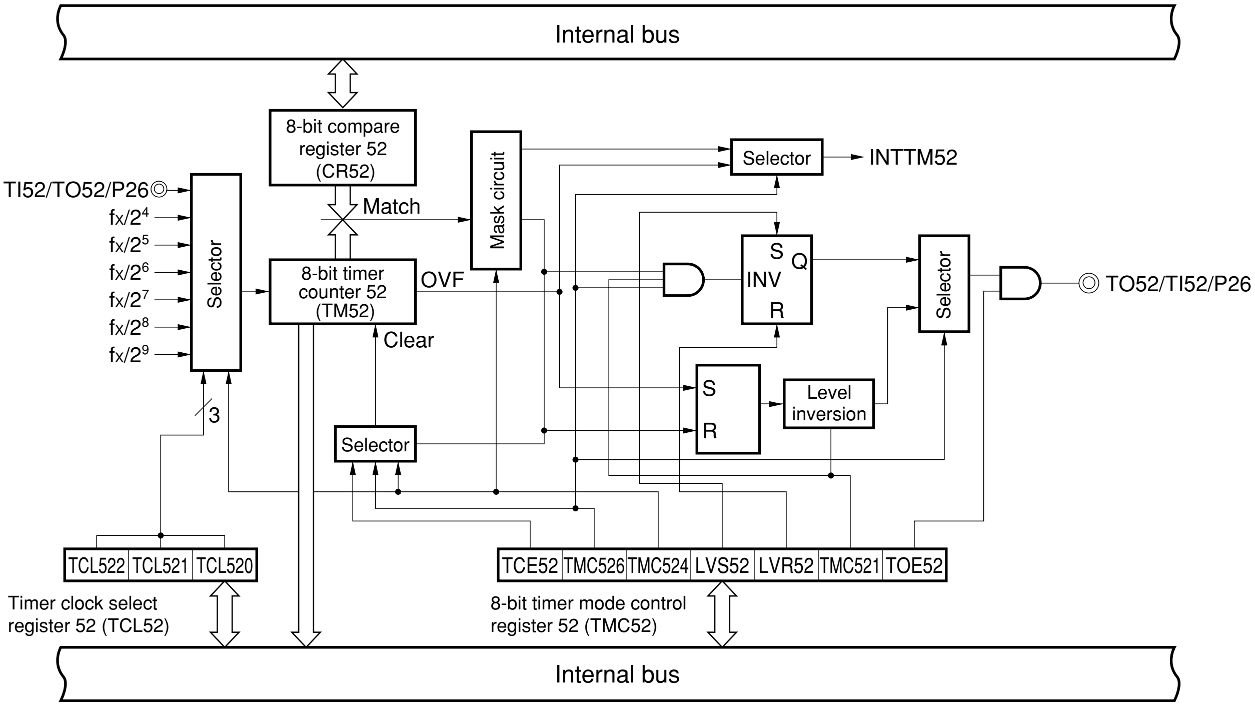


Figure 7-3. Block Diagram of 8-Bit Timer/Event Counter 52



(1) 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52)

TM50, TM51, and TM52 are 8-bit read-only registers that count count pulses.

These counters are incremented in synchronization with the rising edge of the count clock.

TM50 and TM51, or TM51 and TM52 can be connected in cascade and used as a 16-bit timer.

When TM50 and TM51 are connected in cascade and used as a 16-bit timer, the values of these timer counters can be read using a 16-bit manipulation instruction. TM50 and TM51 are connected with an internal 8-bit bus, and are read one at a time. This means that the value of TM50, for example, may change while that of TM51 is read. Therefore, read TM50 and TM51 two times to compare their first and second values for the sake of accuracy.

When TM51 and TM52 are connected in cascade and used as a 16-bit timer, they cannot be read using a 16-bit manipulation instruction. When reading TM51 and TM52, read them separately using an 8-bit manipulation instruction.

If the count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read. The count value is cleared to 00H in the following cases.

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

<2> Clearing TCE5n

<3> Match between TM5n and CR5n in clear & start mode

Caution In a cascade connection, the 16-bit timer is cleared to 00H regardless of whether TCE51 of TM51 or TCE52 of TM52 is cleared.

Remark n = 0 to 2

(2) 8-bit compare registers 50, 51, and 52 (CR50, CR51, and CR52)

The value set to CR5n is always compared with the count value of 8-bit timer counter 5n (TM5n). When the value of the compare register matches the value of the timer counter, an interrupt request (INTTM5n) is generated (in a mode other than the PWM mode).

The value of CR5n can be set in the range of 00H to FFH and can be rewritten during counting.

If TM50 and TM51 are connected in cascade and used as a 16-bit timer, CR50 and CR51 operate as a 16-bit compare register. Therefore, the count value and register value are compared in 16-bit units, and if the two values match, an interrupt request (INTTM50) is generated. At this time, the INTTM51 interrupt request is also generated. When connecting TM50 and TM51 in cascade, therefore, mask the INTTM51 interrupt request.

The same applies when TM51 and TM52 are connected in cascade. If the value of the 16-bit timer matches that of the 16-bit compare register, the INTTM51 interrupt request is generated (so mask the INTTM52 interrupt request).

CR50, CR51, and CR52 are set by an 8-bit memory manipulation instruction.

When CR50 and CR51 are connected in cascade, these registers function as the CR5 register and can be accessed in 16 bits.

$\overline{\text{RESET}}$ input makes these registers undefined.

Caution When changing the setting value of 8-bit compare register 5n (CR5n) in cascade mode, stop each timer operation of 8-bit timer counter 5n (TM5n) connected in cascade.

Remark n = 0 to 2

7.4 Registers Controlling 8-Bit Timer/Event Counter

The following seven registers control 8-bit timer/event counters 50, 51, and 52.

- 8-bit timer mode control registers 50, 51, and 52 (TMC50, TMC51, and TMC52)
- Timer clock select registers 50, 51, and 52 (TCL50, TCL51, and TCL52)
- Port mode register 2 (PM2)

(1) 8-bit timer mode control registers 50, 51, and 52 (TMC50, TMC51, and TMC52)

TMC50, TMC51, and TMC52 perform the following six operations.

<1> Control of count operation of 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52)
<2> Selection of operation mode of 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52)
<3> Selection of single mode or cascade mode
<4> Setting of status of timer output F/F (flip-flop)
<5> Control of timer F/F or selection of active level in PWM (free-running) mode
<6> Control of timer output

TMC50, TMC51, and TMC52 are set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets these registers to 00H.
Figures 7-4 to 7-6 show the formats of TMC50, TMC51, and TMC52.

Figure 7-4. Format of 8-Bit Timer Mode Control Register 50

Symbol	⑦	6	5	4	③	②	1	⓪	Address	After reset	R/W
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50	FF70H	00H	R/W

TCE50	TM50 count operation control
0	Disables count operation after clearing counter to 0 (disables prescaler).
1	Starts counting.

TMC506	TM50 operating mode selection
0	Clears and starts on match between TM50 and CR50.
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting of 8-bit timer/event counter 50
0	0	Not affected
0	1	Resets timer output F/F (to 0).
1	0	Sets timer output F/F (to 1).
1	1	Setting prohibited

TMC501	Other than PWM mode (TMC506 = 0)	PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Disables inverted operation.	High active
1	Enables inverted operation.	Low active

TOE50	Timer output control of 8-bit timer/event counter 50
0	Disables output (port mode).
1	Enables output.

Caution Be sure to set (1) the interrupt mask flag (TMMK50) before clearing (0) TCE50 to avoid generating an interrupt when TCE50 is cleared. The procedure to clear (0) TCE50 is as follows.

```
TMMK50 = 1    ; Mask set
TCE50 = 0     ; Timer clear
TMIF50 = 0    ; Interrupt request flag clear
TMMK50 = 0    ; Mask clear
     :
TCE50 = 1     ; Timer start
     :
```

Remarks **1.** The PWM output is at the inactive level in the PWM mode because TCE50 = 0.

2. If LVS50 and LVR50 are read immediately after data has been set, these bits are 0.

Figure 7-5. Format of 8-Bit Timer Mode Control Register 51

Symbol	⑦	6	5	4	③	②	1	⓪	Address	After reset	R/W
TMC51	TCE51	TMC516	0	TMC514	LVS51	LVR51	TMC511	TOE51	FF74H	00H	R/W

TCE51	TM51 count operation control
0	Disables count operation after clearing counter to 0 (disables prescaler).
1	Starts counting.

TMC516	TM51 operating mode selection
0	Clears and starts on match between TM51 and CR51.
1	PWM (free-running) mode

TMC514	Single mode/cascade mode selection
0	Single mode
1	Cascade mode (connected to TM50)

LVS51	LVR51	Timer output F/F status setting of 8-bit timer/event counter 51
0	0	Not affected
0	1	Resets timer output F/F (to 0).
1	0	Sets timer output F/F (to 1).
1	1	Setting prohibited

TMC511	Other than PWM mode (TMC516 = 0)	PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Disables inverted operation.	High active
1	Enables inverted operation.	Low active

TOE51	Timer output control of 8-bit timer/event counter 51
0	Disables output (port mode).
1	Enables output.

Caution Be sure to set (1) the interrupt mask flag TMMK51 before clearing (0) TCE51 to avoid generating an interrupt when TCE51 is cleared. The procedure to clear (0) TCE51 is as follows.

```
TMMK51 = 1    ; Mask set
TCE51 = 0     ; Timer clear
TMIF51= 0     ; Interrupt request flag clear
TMMK51 = 0    ; Mask clear
      :
TCE51 = 1     ; Timer start
      :
```

Remarks 1. PWM output is at the inactive level in the PWM mode because TCE51 = 0.

2. If LVS51 and LVR51 are read immediately after data has been set, these bits are 0.

Figure 7-6. Format of 8-Bit Timer Mode Control Register 52

Symbol	⑦	6	5	4	③	②	1	⓪	Address	After reset	R/W
TMC52	TCE52	TMC526	0	TMC524	LVS52	LVR52	TMC521	TOE52	FF78H	00H	R/W

TCE52	TM52 count operation control
0	Disables count operation after clearing counter to 0 (disables prescaler).
1	Starts counting.

TMC526	TM52 operating mode selection
0	Clears and starts on match between TM52 and CR52.
1	PWM (free-running) mode

TMC524	Single mode/cascade mode selection
0	Single mode
1	Cascade mode (connected to TM51)

LVS52	LVR52	Timer output F/F status setting of 8-bit timer/event counter 52
0	0	Not affected
0	1	Resets timer output F/F (to 0).
1	0	Sets timer output F/F (to 1).
1	1	Setting prohibited

TMC521	Other than PWM mode (TMC526 = 0)	PWM mode (TMC526 = 1)
	Timer F/F control	Active level selection
0	Disables inverted operation.	High active
1	Enables inverted operation.	Low active

TOE52	Timer output control of 8-bit timer/event counter 52
0	Disables output (port mode).
1	Enables output.

Caution Be sure to set (1) the interrupt mask flag (TMMK52) before clearing (0) TCE52 to avoid generating an interrupt when TCE52 is cleared. The procedure to clear (0) TCE52 is as follows.

```
TMMK52 = 1   ; Mask set
TCE52 = 0    ; Timer clear
TMIF52 = 0   ; Interrupt request flag clear
TMMK52 = 0   ; Mask clear
     :
TCE52 = 1    ; Timer start
     :
```

Remarks 1. PWM output is at the inactive level in the PWM mode because TCE52 = 0.

2. If LVS52 and LVR52 are read immediately after data has been set, these bits are 0.

(2) Timer clock select registers 50, 51, and 52 (TCL50, TCL51, and TCL52)

These registers specify the count clock of 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52) and the valid edges of the TI50, TI51, and TI52 inputs.

TCL50, TCL51, and TCL52 are set by an 8-bit memory manipulation instruction.

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

Figures 7-7 to 7-9 show the formats of TCL50, TCL51, and TCL52.

★ **Figure 7-7. Format of Timer Clock Select Register 50**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500	FF71H	00H	R/W

TCL502	TCL501	TCL500	Count clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	Falling edge of TI50		
0	0	1	Rising edge of TI50		
0	1	0	$f_X/2$	6 MHz	4.19 MHz
0	1	1	$f_X/2^3$	1.5 MHz	1.05 MHz
1	0	0	$f_X/2^5$	375 kHz	262 kHz
1	0	1	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	0	$f_X/2^9$	23.4 kHz	16.4 kHz
1	1	1	$f_X/2^{11}$	5.85 kHz	4.09 kHz

Note Expanded-specification products only.

Cautions 1. Before rewriting the data of TCL50, stop the timer operation once.

2. Be sure to clear bits 3 to 7 of TCL50 to 0.

Remark f_X: System clock oscillation frequency

★

Figure 7-8. Format of Timer Clock Select Register 51

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510	FF75H	00H	R/W

TCL512	TCL511	TCL510	Count clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	Falling edge of TI51		
0	0	1	Rising edge of TI51		
0	1	0	f_X	12 MHz	8.38 MHz
0	1	1	$f_X/2$	6 MHz	4.19 MHz
1	0	0	$f_X/2^2$	3 MHz	2.1 MHz
1	0	1	$f_X/2^3$	1.5 MHz	1.05 MHz
1	1	0	$f_X/2^4$	750 kHz	524 kHz
1	1	1	$f_X/2^5$	375 kHz	262 kHz

Note Expanded-specification products only.

Cautions 1. Before rewriting the data of TCL51, stop the timer operation once.

2. Be sure to clear bits 3 to 7 of TCL51 to 0.

Remarks 1. f_X: System clock oscillation frequency

2. The settings of TCL510 to TCL512 are invalid when TM50 and TM51 are connected in cascade.

★

Figure 7-9. Format of Timer Clock Select Register 52

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520	FF79H	00H	R/W

TCL522	TCL521	TCL520	Count clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	Falling edge of TI52		
0	0	1	Rising edge of TI52		
0	1	0	$f_X/2^4$	750 kHz	524 kHz
0	1	1	$f_X/2^5$	375 kHz	262 kHz
1	0	0	$f_X/2^6$	187 kHz	131 kHz
1	0	1	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	0	$f_X/2^8$	46.8 kHz	32.7 kHz
1	1	1	$f_X/2^9$	23.4 kHz	16.4 kHz

Note Expanded-specification products only.

Cautions 1. Before rewriting the data of TCL52, stop the timer operation once.

2. Be sure to clear bits 3 to 7 of TCL52 to 0.

Remarks 1. f_X: System clock oscillation frequency

2. The settings of TCL520 to TCL522 are invalid when TM51 and TM52 are connected in cascade.

(3) Port mode register 2 (PM2)

This register sets port 2 in the input or output mode in 1-bit units.

When the P24/TI50/TO50 to P26/TI52/TO52 pins are used for timer output, clear PM24 to PM26 and the output latches of P24 to P26 to 0.

PM2 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-10. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM2n	P2n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.5 Operation of 8-Bit Timer/Event Counter

7.5.1 Interval timer (8-bit) operation

The 8-bit timer/event counters operate as interval timers that repeatedly generate an interrupt request at time intervals specified by the count values preset to corresponding 8-bit compare register 5n (CR5n).

When the count values of 8-bit timer counter 5n (TM5n) match the values set to corresponding compare register CR5n, the value of TM5n is cleared to 0, TM5n continues counting, and at the same time, an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected by bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register 5n (TCL5n).

For the operation to be performed when the value of the compare register is changed during timer count operation, refer to **7.6 Notes on 8-Bit Timer/Event Counter (2)**.

<Setting>

<1> Set each register.

- TCL5n: Selects count clock.
- CR5n: Compare value
- TMC5n: Selects clear & start mode entered on match between TM5n and CR5n (TMC5n = 0000×××0B × = don't care).

<2> The count operation is started when TCE5n is set to 1.

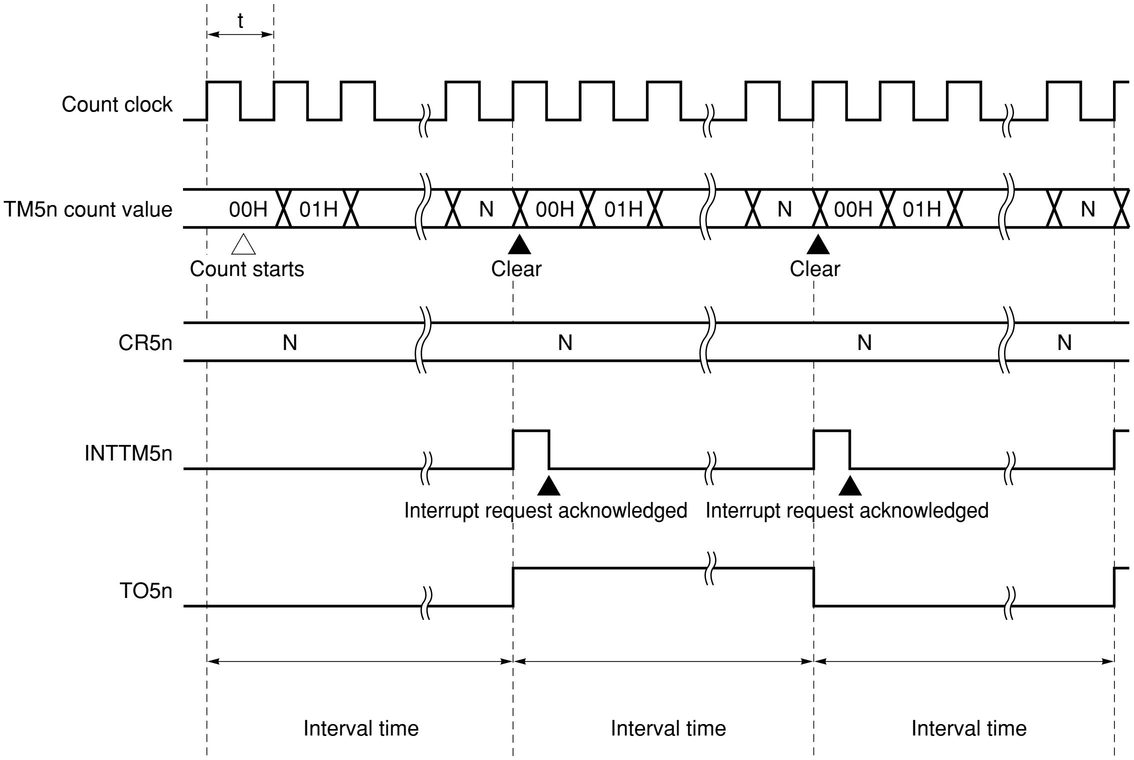
<3> INTTM5n occurs when the values of TM5n and CR5n Match (TM5n is cleared to 00H).

<4> After that, INTTM5n repeatedly occurs at the same interval. To stop the count operation, clear TCE5n to 0.

Remark n = 0 to 2

Figure 7-11. Interval Timer Operation Timing (1/3)

(a) Basic operation

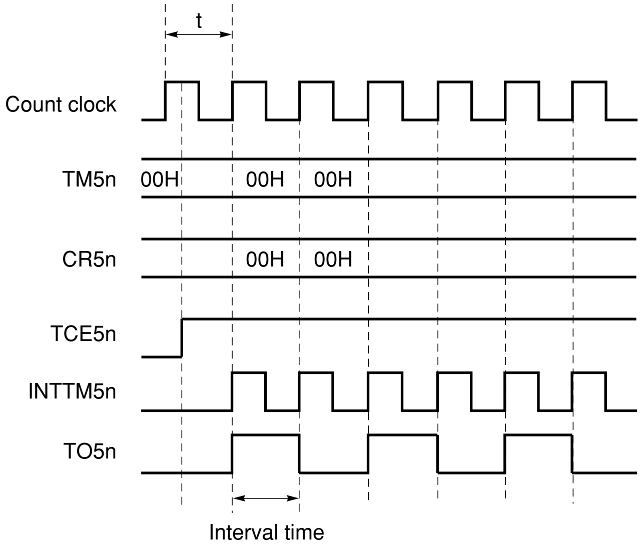


Remarks 1. Interval time = (N + 1) × t: N = 00H to FFH

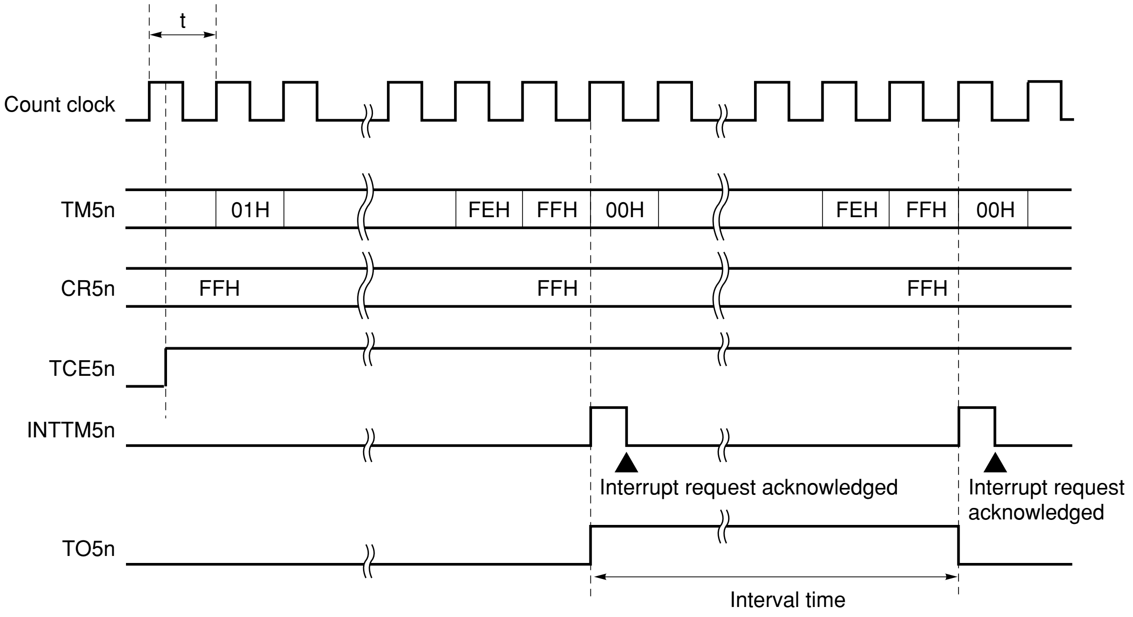
2. n = 0 to 2

Figure 7-11. Interval Timer Operation Timing (2/3)

(b) When CR5n = 00H



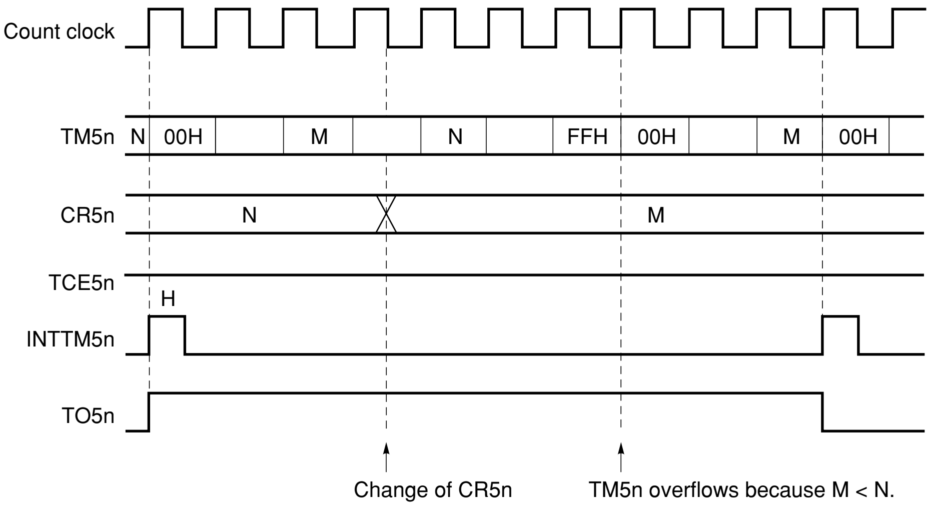
(c) When CR5n = FFH



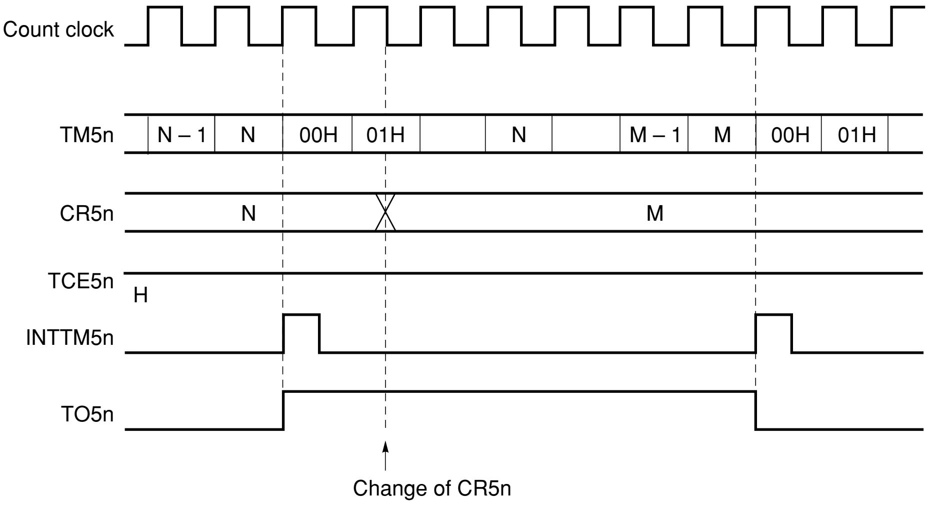
Remark n = 0 to 2

Figure 7-11. Interval Timer Operation Timing (3/3)

(d) Operation when CR5n is changed (M < N)



(e) Operation when CR5n is changed (M > N)



Remark n = 0 to 2

7.5.2 External event counter operation

The external event counter counts the number of clock pulses externally input to the TI50/P24 to TI52/P26 pins by using 8-bit timer counter 5n (TM5n).

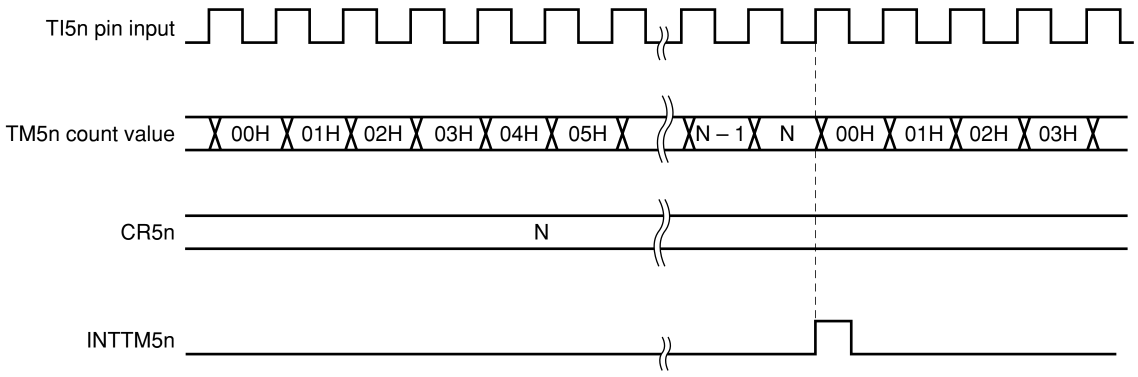
Each time the valid edge specified by timer clock select register 5n (TCL5n) is input, the value of TM5n is incremented. Either the rising edge or falling edge can be specified as the valid edge.

When the count value of TM5n matches the values of corresponding 8-bit 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.

Remark n = 0 to 2

Figure 7-12. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0 to 2

7.5.3 Square-wave output (8-bit resolution) operation

The 8-bit timer/event counters operate as a square wave output at the interval preset to 8-bit compare register 5n (CR5n).

When bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) is set to 1, the output status of TO5n is inverted at the interval time specified by the count value preset to CR5n. In this way, a square-wave of any frequency (duty factor = 50%) can be output.

<Setting>

<1> Set each register.

- Clear the port latch and port mode register 2 (PM2) to 0.
- TCL5n: Selects count clock
- CR5n: Compare value
- TMC5n: Clear & start mode entered on match between TM5n and CR5n

LVS5n	LVR5n	Timer output F/F status setting
1	0	High-level output
0	1	Low-level output

Enables inverting timer output F/F
Timer output enabled → TOE5n = 1

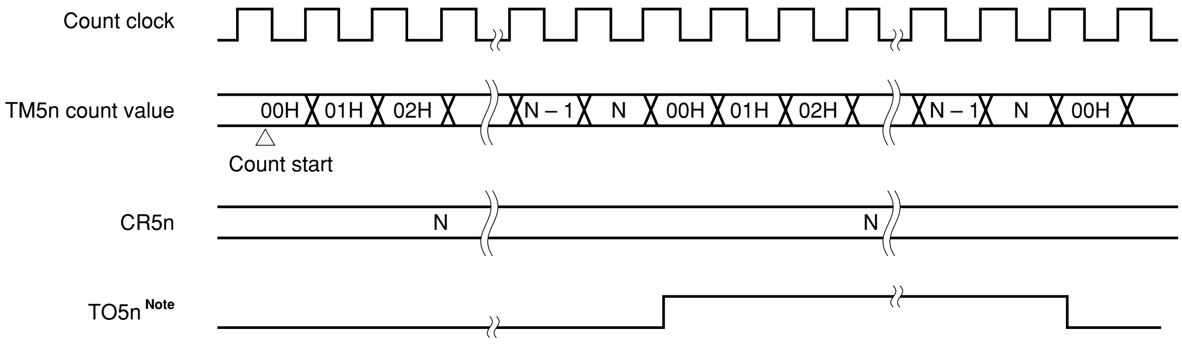
<2> The count operation is started if TCE5n is set to 1.

<3> The timer output F/F is inverted if the values of TM5n and CR5n match.
INTTM5n occurs and TM5n is cleared to 00H.

<4> After that, the timer output F/F is inverted at the same interval, and a square-wave is output from TO5n.

Remark n = 0 to 2

Figure 7-13. Square-Wave Output Operation Timing



Note The initial value of TO5n output can be set with bits 2 and 3 (LVR5n and LVS5n) of 8-bit timer mode control register 5n (TMC5n).

Remark n = 0 to 2

7.5.4 8-bit PWM output operation

The PWM output operation is performed when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

A pulse with a duty factor determined by the value set to 8-bit compare register 5n (CR5n) is output from TO5n.

Set the width of the active level of the PWM pulse to CR5n. The active level can be selected using bit 1 (TMC5n1) of TMC5n.

The count clock can be selected by bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register 5n (TCL5n).

PWM output can be enabled or disabled by bit 0 (TOE5n) of TMC5n.

(1) Basic operation of PWM output

<Setting>

<1> Clear the port latch and port mode register 2 (PM2) to 0.

<2> Set an active level width with 8-bit compare register 5n (CR5n).

<3> Select a count clock with timer clock select register 5n (TCL5n).

<4> Set an active level using bit 1 (TMC5n1) of TMC5n.

<5> The count operation is started when bit 7 (TCE5n) of TMC5n is set to 1.
To stop the count operation, clear TCE5n to 0.

<PWM output operation>

<1> When the count operation is started, an inactive level is output as the PWM output (output from TO5n) until an overflow occurs.

<2> When an overflow occurs, the active level set in step <1> above is output. This active level is continuously output until the value of CR5n matches the count value of 8-bit timer counter 5n (TM5n).

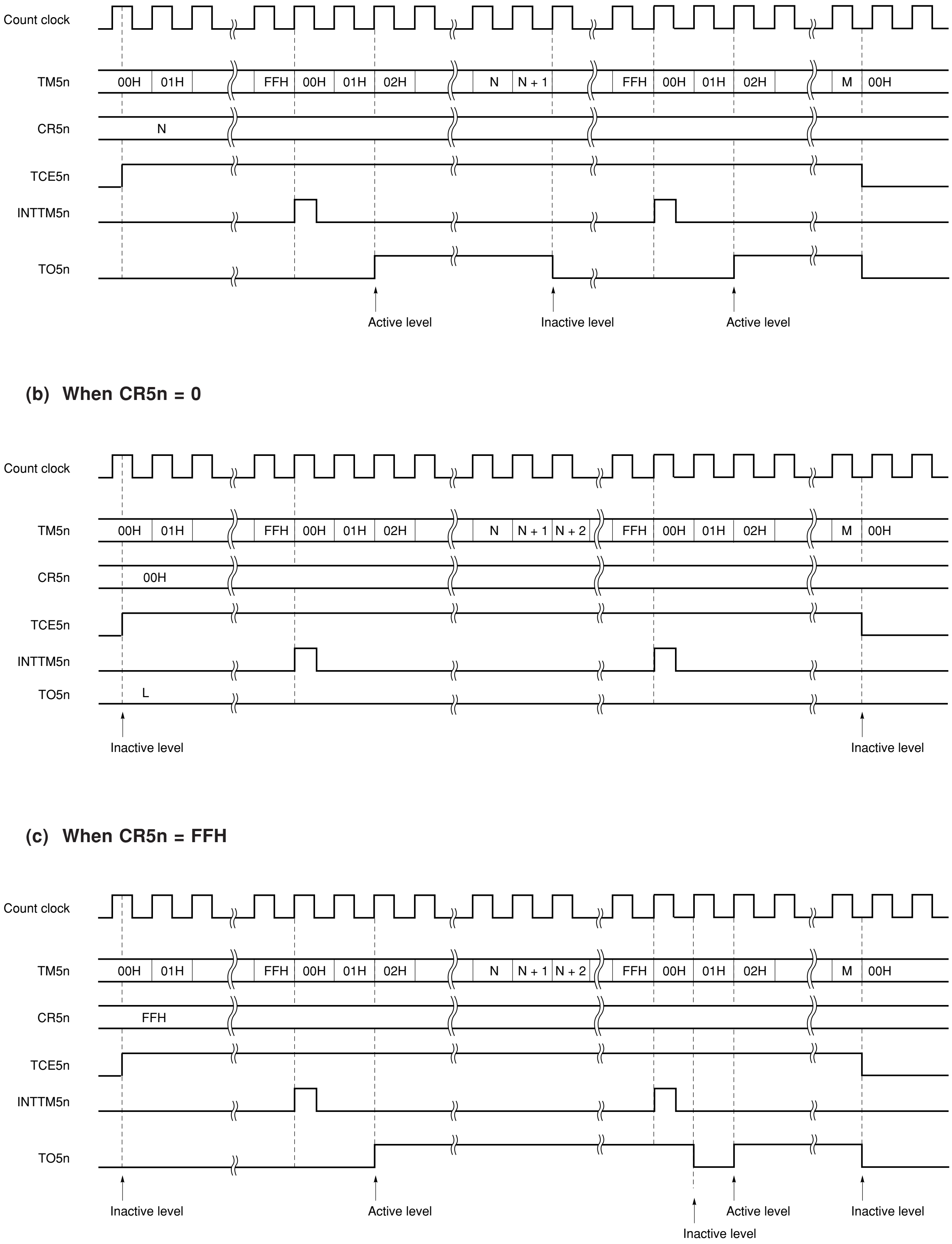
<3> An inactive level is output as the PWM output after the value of CR5n has matched the count value of TM5n, until an overflow occurs again.

<4> After that, <2> and <3> are repeated until the count operation is stopped.

<5> When the count operation is stopped by clearing TCE5n to 0, the PWM output becomes inactive.

Remark n = 0 to 2

Figure 7-14. PWM Output Operation Timing



(a) Basic operation (when active level = H)

(b) When CR5n = 0

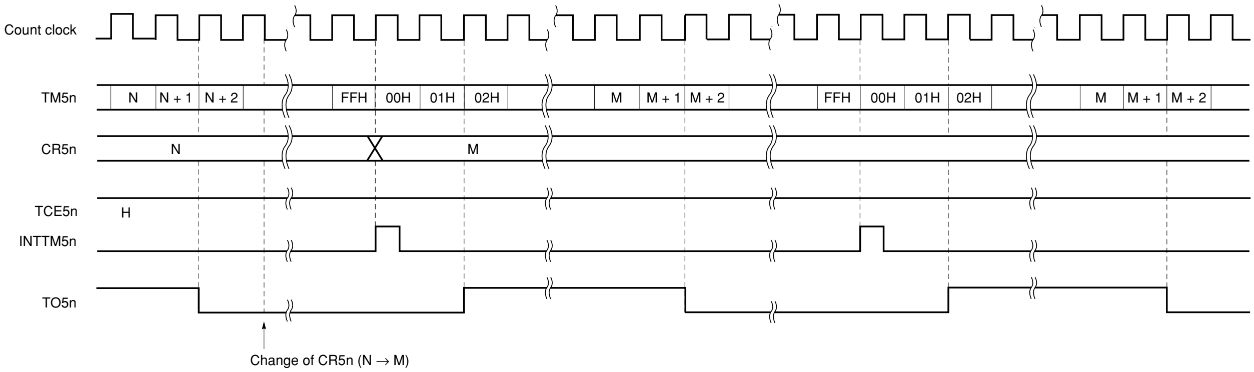
(c) When CR5n = FFH

Remark n = 0 to 2

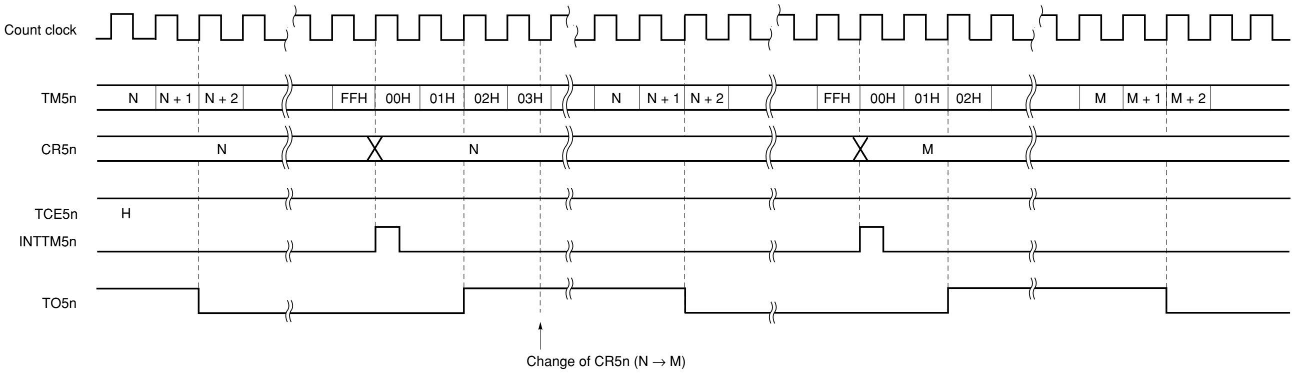
(2) Operation when CR5n is changed

Figure 7-15. Operation Timing When CR5n Is Changed

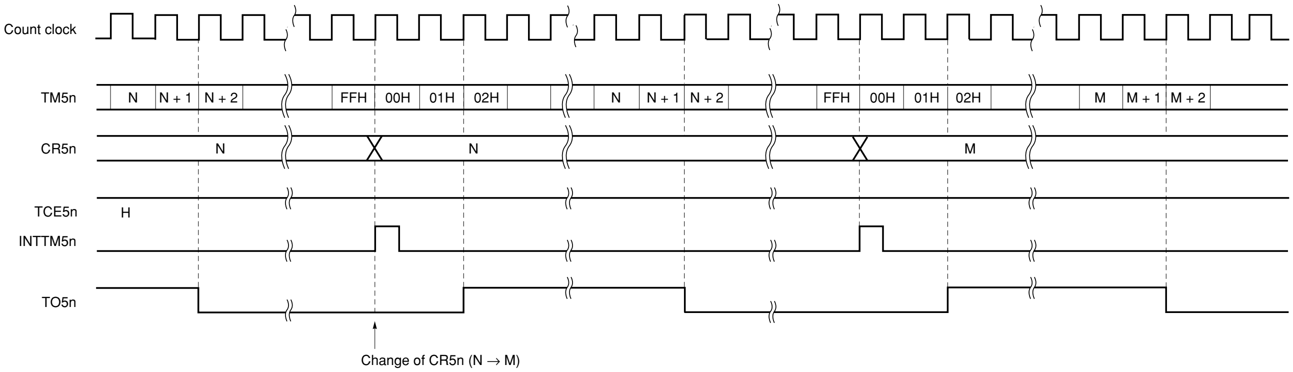
(a) If value of CR5n is changed from N to M before TM5n overflows



(b) If value of CR5n is changed from N to M after TM5n overflows



(c) If value of CR5n is changed from N to M within 2 clocks (00H, 01H) immediately after TM5n overflows



Remark n = 0 to 2

7.5.5 Interval timer (16-bit) operation

(1) Cascade (16-bit timer) mode (TM50 and TM51)

The 16-bit resolution timer/event counter mode is set by setting bit 4 (TMC514) of 8-bit timer mode control register 51 (TMC51) to 1.

In this mode, TM50 and TM51 operate as a 16-bit interval timer that repeatedly generates an interrupt request at intervals specified by the count value preset to 8-bit compare registers 50 and 51 (CR50 and CR51).

<Setting>

<1> Set each register.

- TCL50: TM50 selects a count clock.
  TM51, which is connected in cascade, does not have to be set.
- CR50 and CR51: Compare values (each compare value can be set in the range of 00H to FFH).
- TMC50 and TMC51: Select the mode that clears and starts the timer on a match between TM50 and CR50 (TM51 and CR51).

  TM50 → TMC50 = 0000×××0B ×: don't care
  TM51 → TMC51 = 0001×××0B ×: don't care

<2> By setting TCE51 to 1 for TMC51 first, and then setting TCE50 to 1 for TMC50, the count operation is started.

<3> When the value of TM50 connected in cascade matches the value of CR50, TM50 generates INTTM50 (TM50 and TM51 are cleared to 00H).

<4> After that, INTTM50 is repeatedly generated at the same interval.

Cautions 1. Be sure to set the compare registers (CR50 and CR51) after stopping the timer operation.

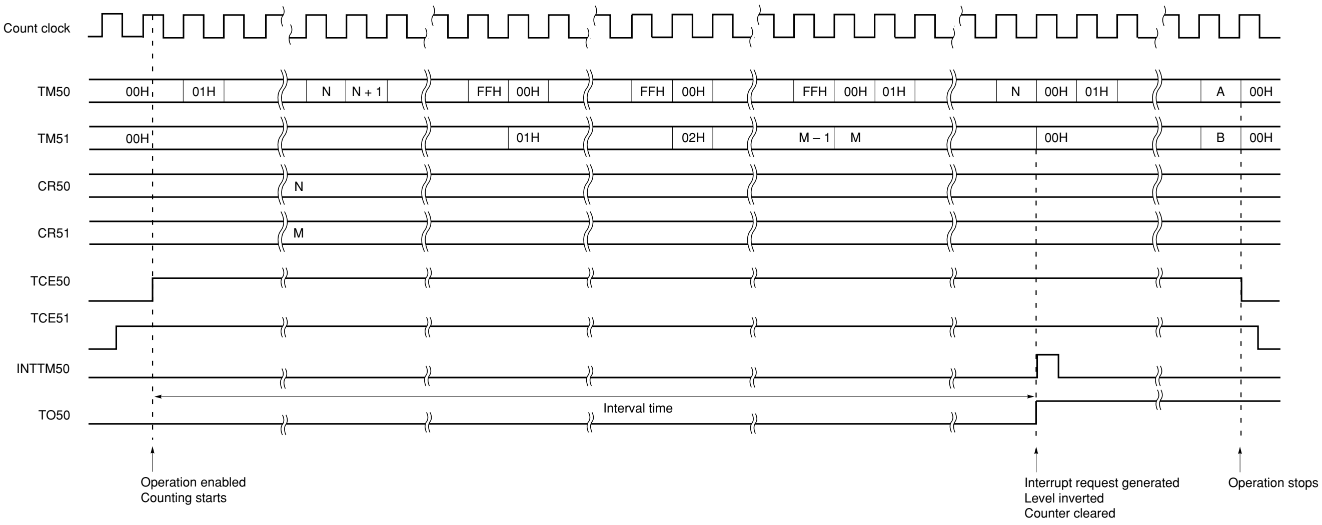
2. Even if the timers are connected in cascade, TM51 generates INTTM51 when the count value of TM51 matches the value of CR51. Be sure to mask TM51 to disable it from generating an interrupt.

3. Set TCE50 and TCE51 in the order of TM51, then TM50.

4. Counting can be started or stopped by setting (1) or clearing (0) only TCE50 of TM50.

Figure 7-16 shows an example of the timing in the 16-bit resolution cascade mode.

Figure 7-16. 16-Bit Resolution Cascade Mode (with TM50 and TM51)



(2) Cascade (16-bit timer) mode (TM51 and TM52)

The 16-bit resolution timer/event counter mode is set by setting bit 4 (TMC524) of 8-bit timer mode control register 52 (TMC52) to 1.

In this mode, TM51 and TM52 operate as a 16-bit interval timer that repeatedly generates an interrupt request at intervals specified by the count value preset to 8-bit compare registers 51 and 52 (CR51 and CR52).

<Setting>

<1> Set each register.

- TCL51: TM51 selects a count clock.
  TM52, which is connected in cascade, does not have to be set.
- CR51 and CR52: Compare values (each compare value can be set in the range of 00H to FFH).
- TMC51 and TMC52: Select the mode that clears and starts the timer on a match between TM51 and CR51 (TM52 and CR52).

TM51 → TMC51 = 0000×××0B ×: don’t care
TM52 → TMC52 = 0001×××0B ×: don’t care

<2> By setting TCE52 to 1 for TMC52 first, and then setting TCE51 to 1 for TMC51, the count operation is started.

<3> When the value of TM51 connected in cascade matches the value of CR51, TM51 generates INTTM51 (TM51 and TM52 are cleared to 00H).

<4> After that, INTTM51 is repeatedly generated at the same interval.

Cautions 1. Be sure to set the compare registers (CR51 and CR52) after stopping the timer operation.

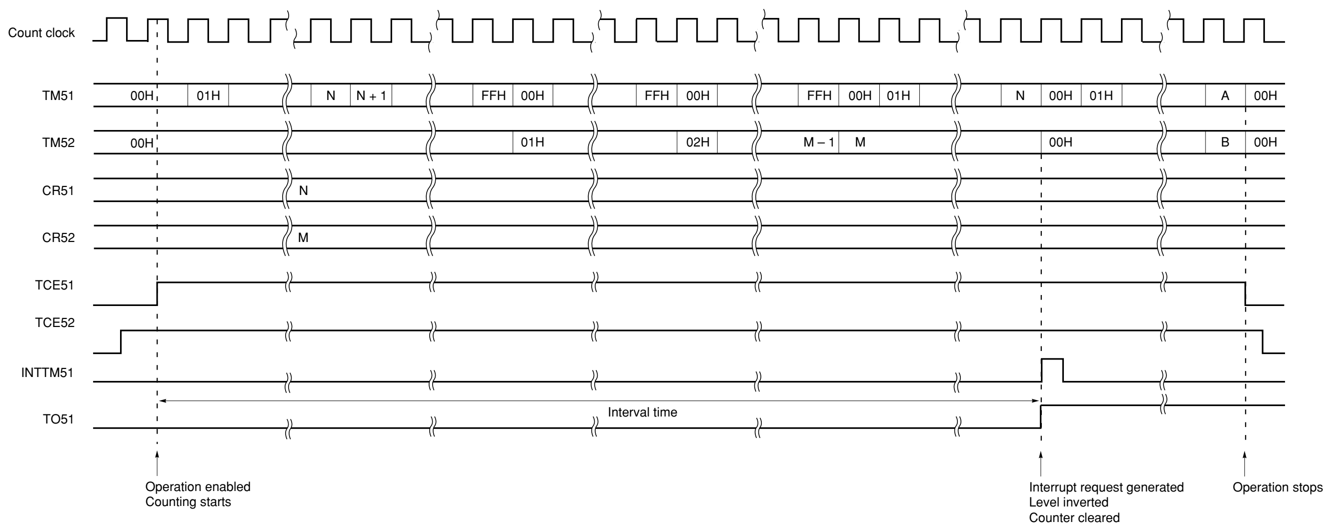
2. Even if the timers are connected in cascade, TM52 generates INTTM52 when the count value of TM52 matches the value of CR52. Be sure to mask TM52 to disable it from generating an interrupt.

3. Set TCE51 and TCE52 in the order of TM52, then TM51.

4. Counting can be started or stopped by setting (1) or clearing (0) only TCE51 of TM51.

Figure 7-17 shows an example of timing in the 16-bit resolution cascade mode.

Figure 7-17. 16-Bit Resolution Cascade Mode (with TM51 and TM52)

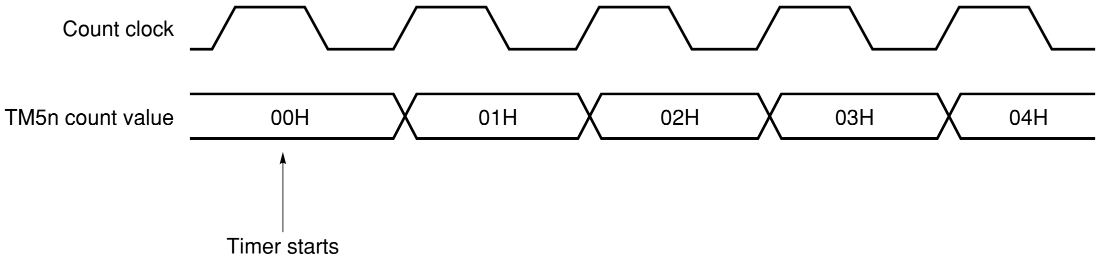


7.6 Notes on 8-Bit Timer/Event Counter

(1) Error on starting timer

An error of up to 1 clock occurs after the timer has been started until a match signal is generated. This is because 8-bit timer counter 5n (TM5n: n = 0 to 2) is started asynchronously to the count clock.

Figure 7-18. Start Timing of 8-Bit Timer Counter

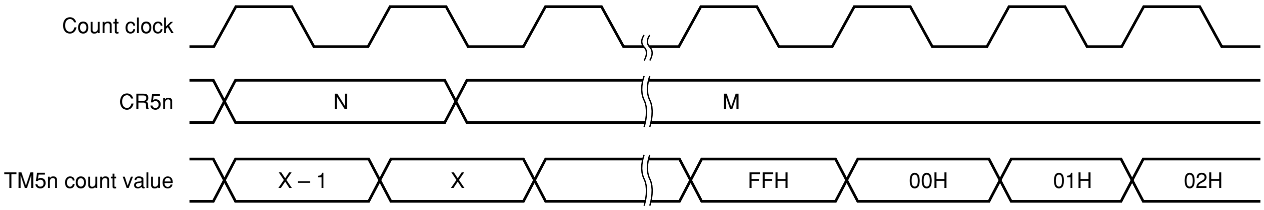


Remark n = 0 to 2

(2) Operation after changing value of compare register during timer count operation

If the new value of 8-bit compare register 5n (CR5n: n = 0 to 2) is less than the value of corresponding 8-bit timer counter 5n (TM5n: n = 0 to 2), TM5n continues counting, overflows, and restarts counting from 0. Therefore, if the new value of CR5n (M) is less than the old value (N), it is necessary to restart the timer after changing the value of CR5n.

Figure 7-19. Timing After Changing Values of Compare Registers During Timer Count Operation



Caution Except when TI5n input is selected, be sure to clear TCE5n to 0 before setting the STOP mode.

Remark N > X > M
n = 0 to 2

(3) Reading TM5n during timer operation

Because the count clock is stopped when TM5n is read during operation, select a count clock with a waveform whose high-/low-level is longer than two CPU clock cycles. For example, in the case of a CPU clock (f_{CPU}) equal to f_X, TM5n can be read as long as the selected count clock is $f_X/4$ or lower.

Remark n = 0 to 2
f_X: System clock oscillation frequency

CHAPTER 8 10-BIT INVERTER CONTROL TIMER

8.1 Outline of 10-Bit Inverter Control Timer

The 10-bit inverter control timer makes inverter control possible. It consists of an 8-bit dead-time generation timer, and allows non-overlapping active-level output.

8.2 Function of 10-Bit Inverter Control Timer

The 10-bit inverter control timer realizes inverter control. It incorporates an 8-bit timer for dead time generation and can output waveforms that do not overlap active levels. A total of six positive phase and negative phase channels are output. In addition, an active level change function and output off function by external input (TOFF7) or watchdog timer interrupt request input are provided.

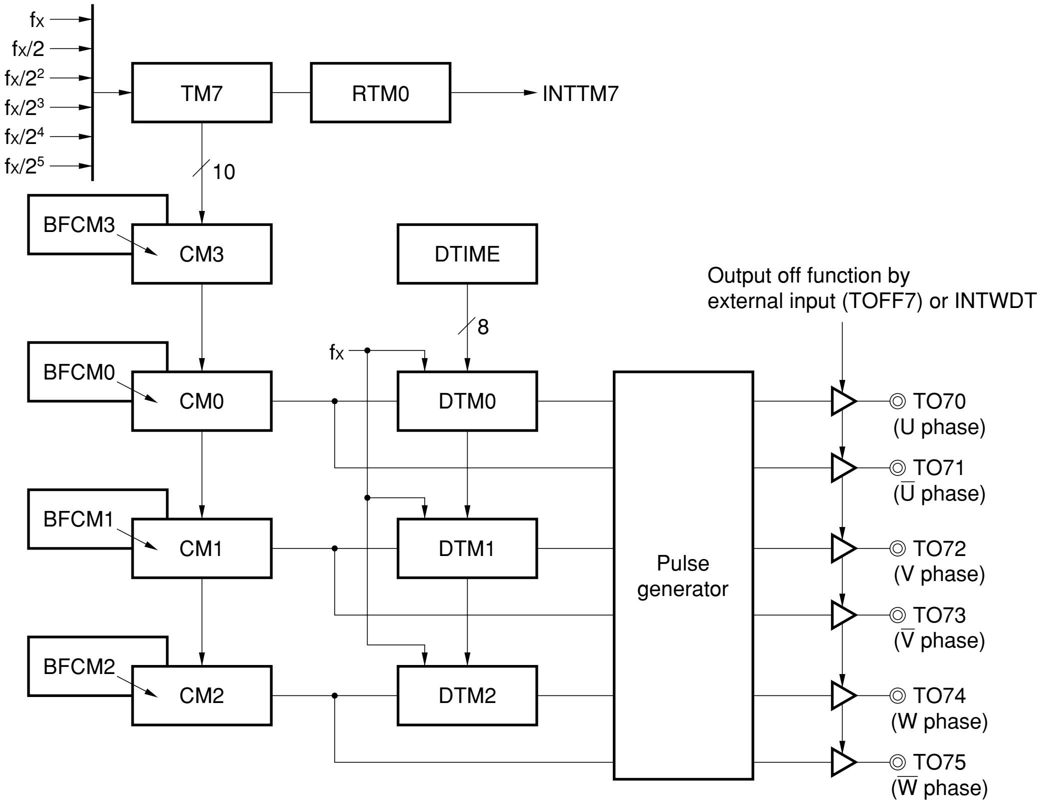
8.3 Configuration of 10-Bit Inverter Control Timer

The 10-bit inverter control timer includes the following hardware.

Table 8-1. Configuration of 10-Bit Inverter Control Timer

Item	Function
Timer counter	10-bit up/down counter × 1 (TM7) Dead-time timers × 3 (DTM0, DTM1, DTM2) Buffer transfer control timer × 1 (RTM0)
Registers	10-bit compare registers × 4 (CM0, CM1, CM2, CM3) 10-bit buffer registers × 4 (BFCM0, BFCM1, BFCM2, BFCM3) Dead-time reload register × 1 (DTIME)
Timer outputs	6 (TO70, TO71, TO72, TO73, TO74, TO75)
Control registers	Inverter timer control register 7 (TMC7) Inverter timer mode register 7 (TMM7)

Figure 8-1. Block Diagram of 10-Bit Inverter Control Timer



(1) **10-bit up/down counter (TM7)**

TM7 is a 10-bit up/down counter that counts count pulses in synchronization with the rising edge of the count clock. When the timer starts, the number of count pulse count is incremented from 0, and when the value preset to compare register 3 (CM3) and TM7 count value match, it is switched to the count down operation.

An underflow signal is generated if the value becomes 000H during the count down operation and interrupt request signal INTTM7 is generated. When an underflow occurs, it is switched from the count down operation to the count up operation. INTTM7 is normally generated at every underflow but the number of occurrences can be divided by the IDEV0 to IDEV2 bits of inverter timer control register 7 (TMC7).

TM7 cannot be read/written.

The cycle of TM7 is controlled by CM3.

The count clock can be selected from 6 types: fx, fx/2, fx/4, fx/8, fx/16, fx/32.

$\overline{\text{RESET}}$ input or clearing the CE7 bit of TMC7 sets TM7 to 000H.

(2) **10-bit compare registers 0 to 2 (CM0 to CM2)**

CM0 to CM2 are 10-bit compare registers that always compare their own value with that of TM7, and if they match, the contents of the flip-flops are changed.

Each of CM0 to CM2 are provided with a buffer register (BFCM0 to BFCM2), so that the contents of the buffer can be transferred to CM0 to CM2 at the timing of interrupt request signal INTTM7 generation.

A write operation to CM0 to CM2 is possible only while TM7 is stopped.

To set the output timing, write data to BFCM0 to BFCM3.

$\overline{\text{RESET}}$ input or clearing the CE7 bit of TMC7 sets these registers to 000H.

(3) 10-bit compare register 3 (CM3)

CM3 is a 10-bit compare register that controls the high limit value of TM7. If the count value of TM7 matches the value of CM3 or 0, count up/down is switched at the next count clock.

CM3 provides a buffer register (BFCM3) whose contents are transferred to CM3 at the timing of interrupt request signal INTTM7 generation.

CM3 can be written to only while TM7 is stopped.

To set the cycle to TM7, write data to BFCM3.

$\overline{\text{RESET}}$ input sets CM3 to 0FFH.

Do not set CM3 to 000H.

(4) 10-bit buffer registers 0 to 3 (BFCM0 to BFCM3)

BFCM0 to BFCM3 are 10-bit registers. They transfer data to the compare register (CM0 to CM3) corresponding to each buffer register at the timing of interrupt request signal INTTM7 generation.

BFCM0 to BFCM3 can be read/written irrespective of whether TM7 count is stopped or operating.

$\overline{\text{RESET}}$ input sets BFCM0 to BFCM2 to 000H, and BFCM3 to 0FFH.

These registers can be read/written in word and byte units. For read/write operations of less than 8 bits, BFCM0L to BFCM3L are used.

(5) Dead-time reload register (DTIME)

DTIME is an 8-bit register to set dead time and is common to three dead-time timers (DTM0 to DTM2). However, the data load timing from DTIME to DTM0, DTM1 and DTM2 is independent.

DTIME can be written only while TM7 counting is stopped. Data does not change even if an instruction to rewrite DTIME is executed during timer operation.

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

Even if DTIME is set to 00H, an output with the dead time of f_X is performed.

(6) Dead-time timers 0 to 2 (DTM0 to DTM2)

DTM0 to DTM2 are 8-bit down counters that generate dead time.

Count down is performed after the value of the dead-time reload register (DTIME) is reloaded with the timing of a compare match between CM0 to CM2 and TM7. DTM0 to DTM2 generate an underflow signal when 00H changes to FFH and stop with FFH.

The count clock is f_X.

DTM0 to DTM2 cannot be read/written.

$\overline{\text{RESET}}$ input or clearing the CE7 bit of TMC7 sets these registers to FFH.

(7) Buffer transfer control timer (RTM0)

RTM0 is a 3-bit up counter. It has the function of dividing interrupt request signal INTTM7.

Incrementing is performed with the TM7 underflow signal and INTTM7 is generated when the value matches the number of divisions set with bits IDEV0 to IDEV2 of TMC7.

RTM0 cannot be read/written.

$\overline{\text{RESET}}$ input sets RTM0 to 7H. Generating INTTM7 and clearing the CE7 bit of TMC7 also sets RTM0 to 7H.

8.4 Registers Controlling 10-Bit Inverter Control Timer

The following two registers control the 10-bit inverter control timer.

- Inverter timer control register 7 (TMC7)
- Inverter timer mode register 7 (TMM7)

(1) Inverter timer control register 7 (TMC7)

TMC7 controls the operation of TM7, dead-time timers 0 to 2 (DTM0 to DTM2), and the buffer transfer control timer (RTM0), specifies the count clock of TM7, and selects the compare register transfer cycle.
TMC7 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TMC7 to 00H.

★

Figure 8-2. Format of Inverter Timer Control Register 7

Symbol	⑦	6	5	4	3	2	1	0	Address	After reset	R/W
TMC7	CE7	0	TCL72	TCL71	TCL70	IDEV2	IDEV1	IDEV0	FF90H	00H	R/W

CE7	TM7, DTM0 to DTM2, RTM0 operation control
0	Clear and stop (TO70 to TO75 are Hi-Z)
1	Count enable

TCL72	TCL71	TCL70	Count clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	f_X	12 MHz	8.38 MHz
0	0	1	$f_X/2$	6 MHz	4.19 MHz
0	1	0	$f_X/2^2$	3 MHz	2.1 MHz
0	1	1	$f_X/2^3$	1.5 MHz	1.05 MHz
1	0	0	$f_X/2^4$	750 kHz	524 kHz
1	0	1	$f_X/2^5$	375 kHz	262 kHz
Other than above			Setting prohibited		

IDEV2	IDEV1	IDEV0	INTTM7 occurrence frequency selection
0	0	0	Occurs once every TM7 underflow.
0	0	1	Occurs once every two TM7 underflows.
0	1	0	Occurs once every three TM7 underflows.
0	1	1	Occurs once every four TM7 underflows.
1	0	0	Occurs once every five TM7 underflows.
1	0	1	Occurs once every six TM7 underflows.
1	1	0	Occurs once every seven TM7 underflows.
1	1	1	Occurs once every eight TM7 underflows.

Note Expanded-specification products only.

Remark f_X: System clock oscillation frequency

(2) Inverter timer mode register 7 (TMM7)

TMM7 controls the operation of and specifies the active level of the TO70 to TO75 outputs, and sets the valid edge of TOFF7.

TMM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMM7 to 00H.

Figure 8-3. Format of Inverter Timer Mode Register 7

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TMM7	0	0	0	PNOFFB[Note]	ALV	TOEDG	TOSPP	TOSPW	FF91H	00H	R/W

PNOFFB[Note]	Control status flag of TM7 output to TO70 to TO75
0	TM7 output disabled status (TO70 to TO75 are Hi-Z)
1	TM7 output enabled status

ALV	TO70 to TO75 output active level specification
0	Low level
1	High level

TOEDG	TOFF7 valid edge specification
0	Falling edge
1	Rising edge

TOSPP	TO70 to TO75 output stop control by valid edge of TOFF7
0	Output not stopped.
1	Output stopped (TO70 to TO75 are Hi-Z).

TOSPW	TO70 to TO75 output stop control by INTWDT
0	Output not stopped.
1	Output stopped (TO70 to TO75 are Hi-Z).

Note The PNOFFB bit is a read-only flag. This bit cannot be set or reset by software. PNOFFB is reset when an output stop is generated by TOFF7 and INTWDT while TM7 is stopped (CE7 = 0) or operating (CE7 = 1).

Caution Always set bits 5 to 7 of TMM7 to 0.

Remarks 1. TO70 to TO75 become Hi-Z state in the following cases. However, the TM7, DTM0 to DTM2, and RTM0 timers do not stop if CE7 = 1 is set.

- A valid edge is input to the TOFF7 pin while TOSPP = 1.
- A specified interrupt request is generated while TOSPW = 1.

To restore the output of TO70 to TO75, perform the procedure below.

<1> Write 0 to CE7 and stop the timer.
<2> Write 0 to the output stop function flag that is used.
<3> Reset the registers to their default values.

2. PNOFFB, ALV, CE7, and TO70 to TO75 are related as follows.

PNOFFB	ALV	CE7	TO70, TO72, TO74	TO71, TO73, TO75
0	0	0	Hi-Z	Hi-Z
0	1	0	Hi-Z	Hi-Z
0	0/1	1	Hi-Z	Hi-Z
1	0/1	1	PWM wave output	PWM wave output

8.5 Operation of 10-Bit Inverter Control Timer

(1) Setting procedure

(a) The TM7 count clock is set with the TCL70 to TCL72 bits of inverter timer control register 7 (TMC7) and the occurrence frequency of interrupt request signal INTTM7 is set with the IDEV0 to IDEV2 bits.

(b) The active level of the TO70 to TO75 pins is set with the ALV bit of inverter timer mode register 7 (TMM7).

(c) Set the half width of the first PWM cycle to 10-bit compare register 3 (CM3).
• PWM cycle = CM3 value × 2 × TM7 clock rate
(The clock rate of TM7 is set with the TMC7)

(d) Set the half width of the second PWM cycle to 10-bit buffer register 3 (BFCM3).

(e) Set the dead time width to the dead time reload register (DTIME).
• Dead time width = (DTIME + 1) × f_X
f_X: Internal system clock

(f) Set the F/F set/reset timing that is used during the first cycle to 10-bit compare registers 0 to 2 (CM0 to CM2).

(g) Set the F/F set/reset timing that is used during the second cycle to BFCM3.

(h) After the CE7 bit of TMC7 is set (1), the operation of TM7, dead-time timers 0 to 2 (DTM0 to DTM2), and buffer transfer control timer (RTM0) is enabled.

Caution Always use a bit manipulation instruction to set the CE7 bit.

(i) Set the F/F set/reset timing that is used for the next cycle to BFCM0 to BFCM3 during TM7 operation.

(j) To stop the TM7 operation, set the CE7 bit of the TMC7 to 0.

Caution Another bit cannot be rewritten at the same time that the CE7 bit is being rewritten.

(2) Output waveform widths corresponding to set values

- PWM cycle = CM3 × 2 × T_{TM7}
- Dead-time width = T_{DTM} = (DTIME + 1) × f_X
- Active width of positive phase (TO70, TO72, TO74 pin)
  = $\{(CM3 - CM_{up}) + (CM3 - CM_{down})\} \times T_{TM7} - T_{DTM}$
- Active width of negative phase (TO71, TO73, TO75 pin)
  = $(CM_{down} + CM_{up}) \times T_{TM7} - T_{DTM}$

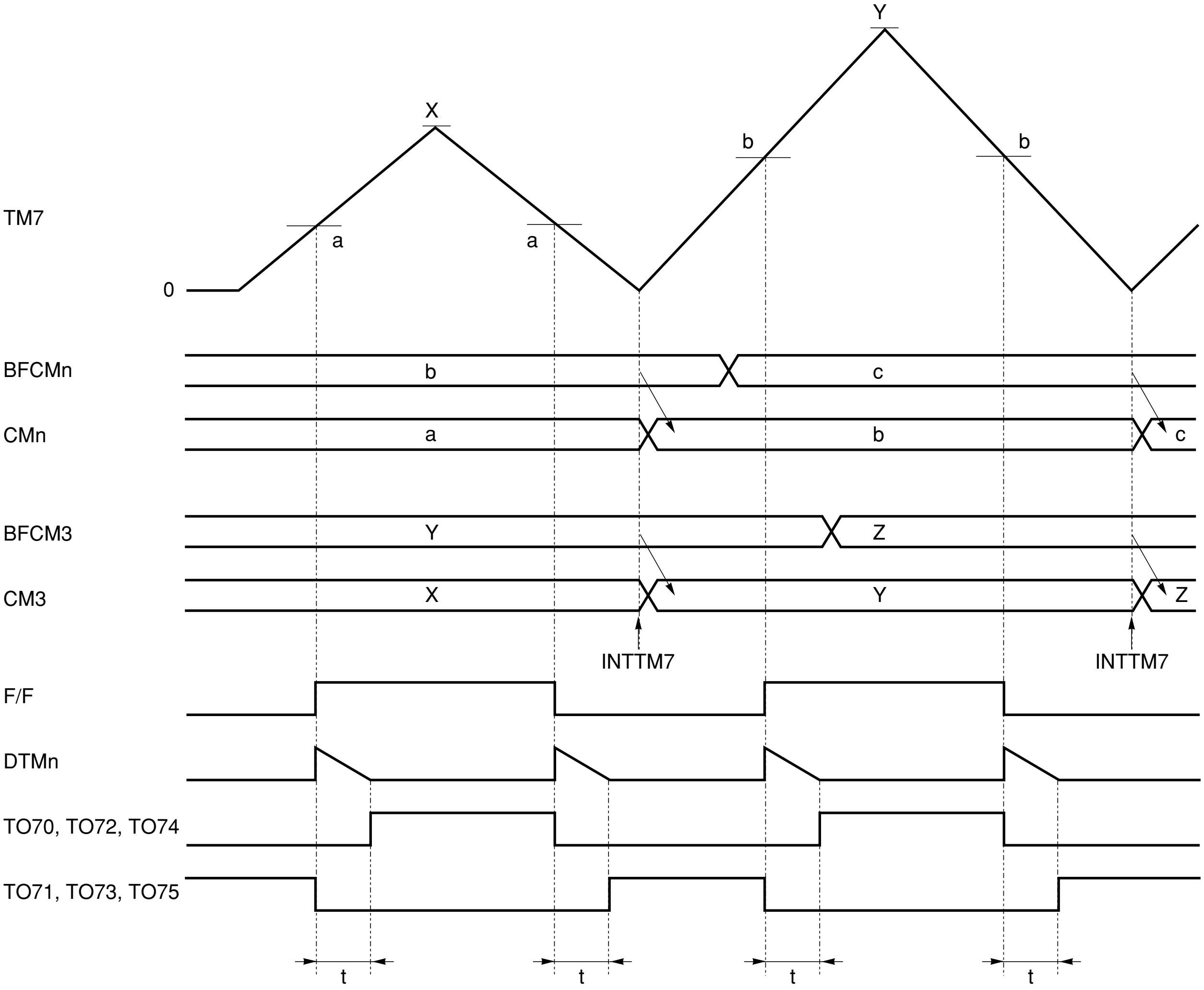
f_X:	System clock oscillation frequency
T_{TM7}:	TM7 count clock
CM_{up}:	Set value of CM0 to CM2 during TM7 count up
CM_{down}:	Set value of CM0 to CM2 during TM7 count down

Caution If a value whose active width in the positive phase or negative phase becomes 0 or negative via the above calculation, TO70 to TO75 output a waveform fixed at the inactive level with an active width of 0 (refer to Figure 8-5).

However, if CMn = 0 and BFCMn ≥ CM3 are set, TO70 to TO75 output a waveform at the active level.

(3) Operation timing

Figure 8-4. TM7 Operation Timing (Basic Operation)

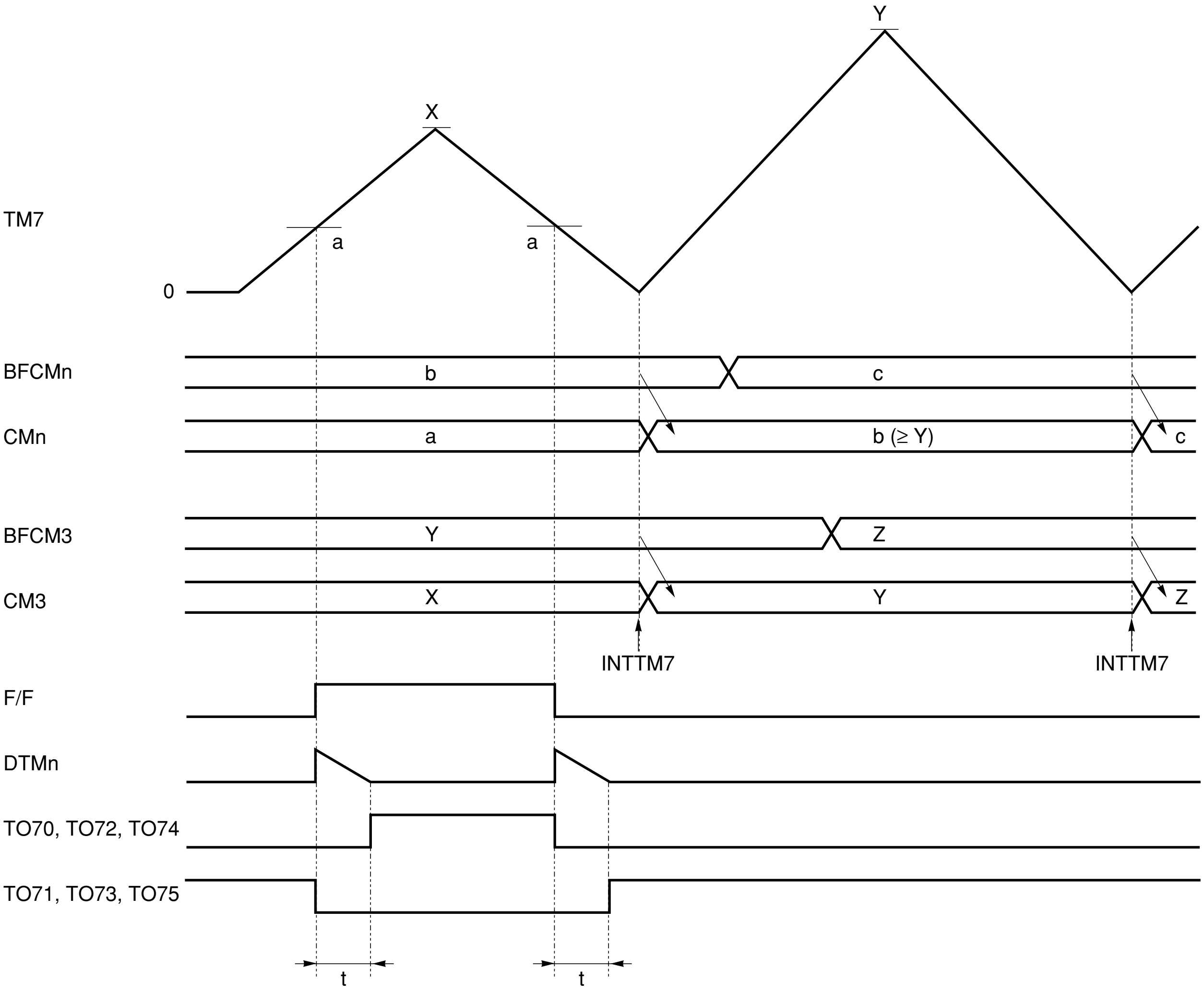


Remarks **1.** n = 0 to 2

2. t: Dead time = (DTIME + 1) × fx
(fx: System clock oscillation frequency)

3. The above figure assumes an active high and undivided INTTM7 occurrence.

Figure 8-5. TM7 Operation Timing (CMn (BFCMn) ≥ CM3 (BFCM3))



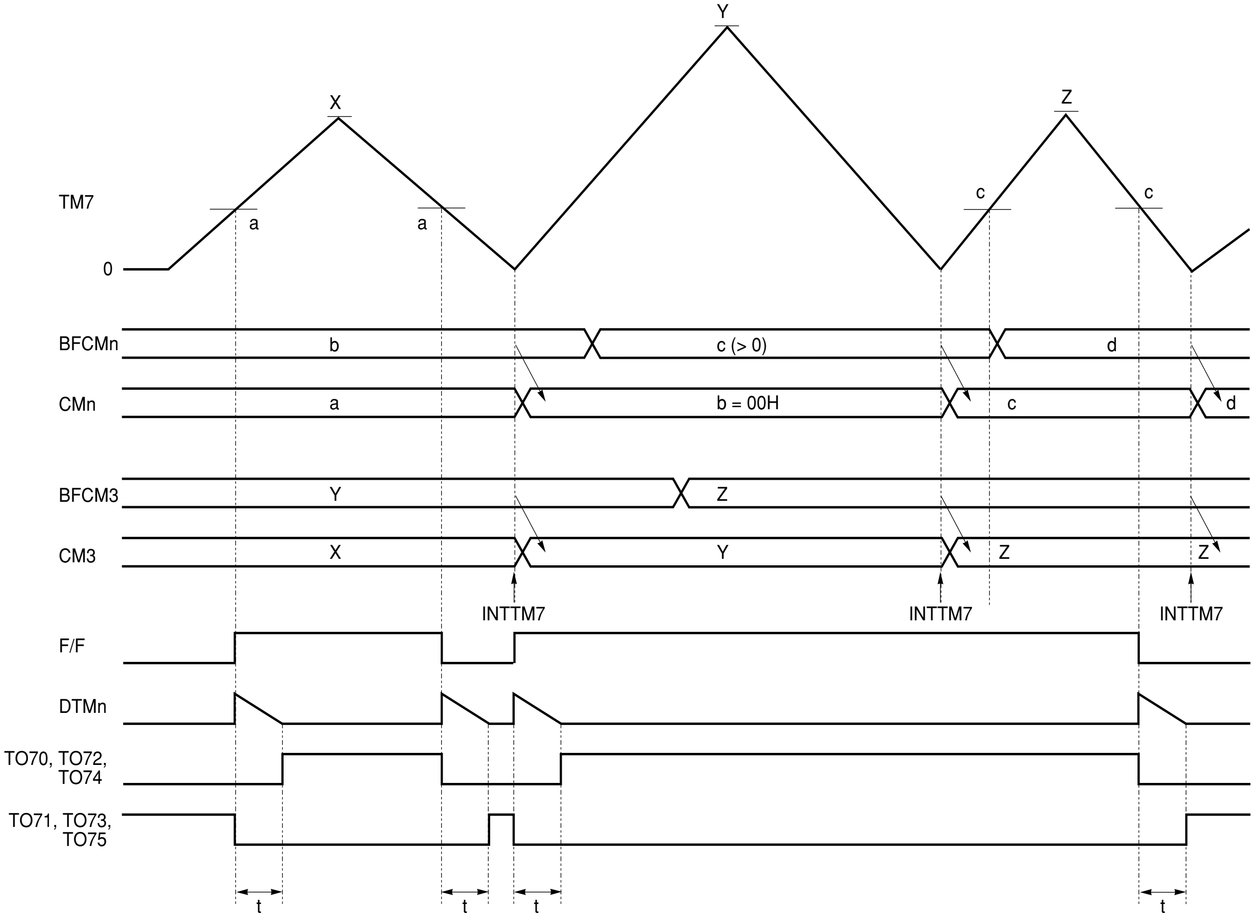
Remarks 1. n = 0 to 2

2. t: Dead time = (DTIME + 1) × fx
(fx: System clock oscillation frequency)

3. The above figure assumes an active high and undivided INTTM7 occurrence.

If a value higher than CM3 is set to BFCMn, low-level output in the positive phases (TO70, TO72, TO74 pins), and high-level output in the negative phases (TO71, TO73, TO75 pins) are continued. This setting is effective to output signals whose low and high widths are longer than the PWM cycle when controlling an inverter, etc.

Figure 8-6. TM7 Operation Timing (CMn (BFCMn) = 000H)

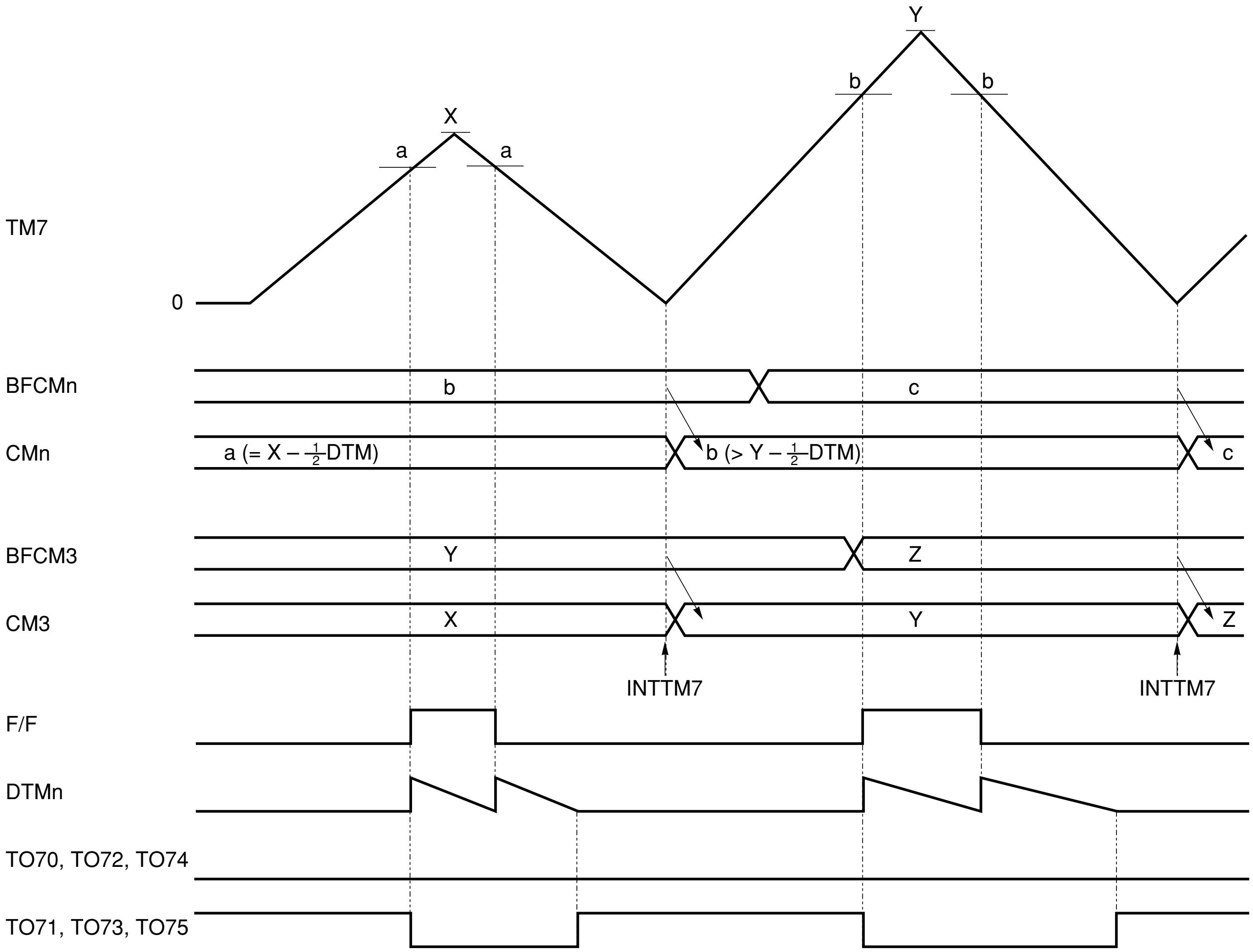


Remarks 1. n = 0 to 2

2. t: Dead time = (DTIME + 1) × fx
(fx: System clock oscillation frequency)

3. The above figure assumes an active high and undivided INTTM7 occurrence.

Figure 8-7. TM7 Operation Timing (CMn (BFCMn) = CM3 – 1/2DTM, CMn (BFCMn) > CM3 – 1/2DTM)



Remarks 1. n = 0 to 2

2. The above figure assumes an active high and undivided INTTM7 occurrence.

CHAPTER 9 WATCHDOG TIMER

9.1 Outline of Watchdog Timer

The watchdog timer can also be used to generate a non-maskable interrupt request, maskable interrupt request, or $\overline{\text{RESET}}$ signal at preset time intervals.

9.2 Function of Watchdog Timer

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Oscillation stabilization time specification

Caution Select the watchdog timer mode or interval timer mode by using the watchdog timer mode register (WDTM). (The watchdog timer and interval timer cannot be used simultaneously.)

(1) Watchdog timer mode

The watchdog timer is used to detect an inadvertent program loop. When a loop is detected, a non-maskable interrupt request or the $\overline{\text{RESET}}$ signal can be generated.

★ **Table 9-1. Loop Detection Time of Watchdog Timer**

Loop Detection Time	At f_X = 12 MHz[Note]	At f_X = 8.38 MHz	Loop Detection Time	At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
$2^{12} \times 1/f_X$	341.3 μs	488.8 μs	$2^{16} \times 1/f_X$	5.46 ms	7.82 ms
$2^{13} \times 1/f_X$	682.6 μs	977.6 μs	$2^{17} \times 1/f_X$	10.9 ms	15.6 ms
$2^{14} \times 1/f_X$	1.36 ms	1.96 ms	$2^{18} \times 1/f_X$	21.8 ms	31.3 ms
$2^{15} \times 1/f_X$	2.73 ms	3.91 ms	$2^{20} \times 1/f_X$	87.3 ms	125.1 ms

Note Expanded-specification products only.

Remark f_X: System clock oscillation frequency

(2) Interval timer mode

When the watchdog timer is used as an interval timer, it generates an interrupt request at preset time intervals.

★ **Table 9-2. Interval Time**

Interval Time	At fX = 12 MHz[Note]	At fX = 8.38 MHz	Interval Time	At fX = 12 MHz[Note]	At fX = 8.38 MHz
$2^{12} \times 1/f_X$	341.3 μs	488.8 μs	$2^{16} \times 1/f_X$	5.46 ms	7.82 ms
$2^{13} \times 1/f_X$	682.6 μs	977.6 μs	$2^{17} \times 1/f_X$	10.9 ms	15.6 ms
$2^{14} \times 1/f_X$	1.36 ms	1.96 ms	$2^{18} \times 1/f_X$	21.8 ms	31.3 ms
$2^{15} \times 1/f_X$	2.73 ms	3.91 ms	$2^{20} \times 1/f_X$	87.3 ms	125.1 ms

Note Expanded-specification products only.

Remark fX: System clock oscillation frequency

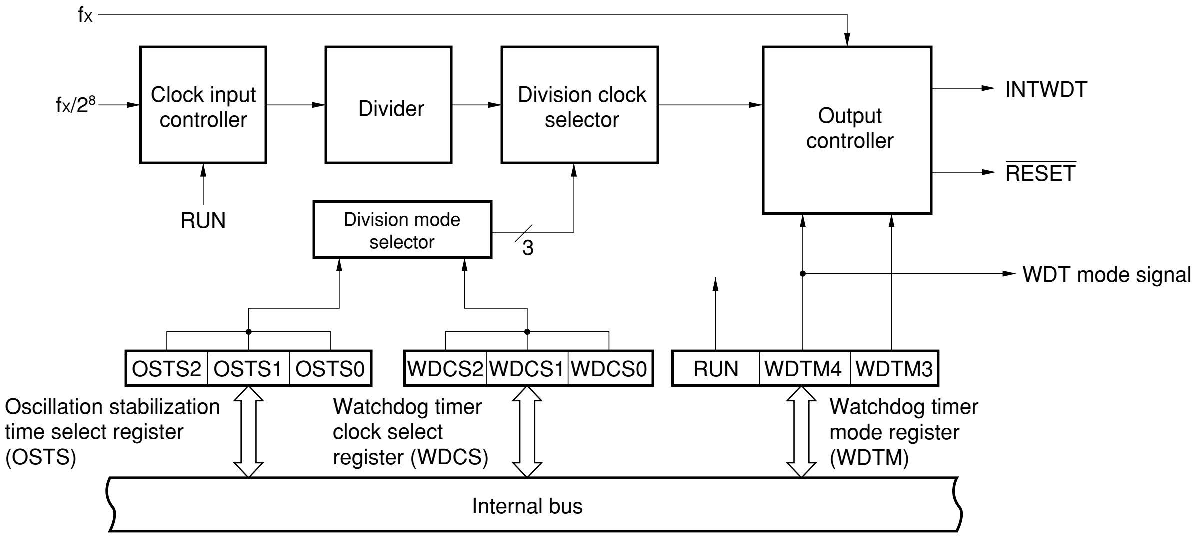
9.3 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time select register (OSTS)

Figure 9-1. Watchdog Timer Block Diagram



9.4 Registers Controlling Watchdog Timer

The following three registers control the watchdog timer.

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)
- Oscillation stabilization time select register (OSTS)

(1) Watchdog timer clock select register (WDCS) (refer to Figure 9-2)

This register sets the overflow time of watchdog timer and interval timer.
WDCS is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

★ **Figure 9-2. Format of Watchdog Timer Clock Select Register**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0	FF42H	00H	R/W

WDCS2	WDCS1	WDCS0	Watchdog timer/interval timer overflow time selection		
				At fx = 12 MHz[Note]	At fx = 8.38 MHz
0	0	0	2^{12}/fx	341.3 μs	488.8 μs
0	0	1	2^{13}/fx	682.6 μs	977.6 μs
0	1	0	2^{14}/fx	1.36 ms	1.96 ms
0	1	1	2^{15}/fx	2.73 ms	3.91 ms
1	0	0	2^{16}/fx	5.46 ms	7.82 ms
1	0	1	2^{17}/fx	10.9 ms	15.6 ms
1	1	0	2^{18}/fx	21.8 ms	31.3 ms
1	1	1	2^{20}/fx	87.3 ms	125.1 ms

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

(2) Watchdog timer mode register (WDTM)

This register sets the operation mode of the watchdog timer, and enables/disables counting of the watchdog timer.

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

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

Figure 9-3. Format of Watchdog Timer Mode Register

Symbol	⑦	6	5	4	3	2	1	0	Address	After reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Selection of operation mode of watchdog timer Note 1 and control of reset by watchdog timer and timer interrupt
0	×	Interval timer mode Note 2 (overflow and maskable interrupt request occur)/PWM output off function of TM7 by INTWDT can be used.
1	0	Watchdog timer mode 1 (overflow and non-maskable interrupt request occur)/PWM output off function of TM7 by INTWDT can be used.
1	1	Watchdog timer mode 2 (overflow occurs and reset operation started)

RUN	Selection of watchdog timer operation Note 3
0	Stops counting.
1	Clears counter and starts counting.

Notes **1.** Once WDTM3 and WDTM4 have been set to 1, they cannot be cleared to 0 by software.

2. The watchdog timer starts operating as an interval timer as soon as the RUN bit has been set to 1.

3. Once RUN has been set to 1, it cannot be cleared to 0 by software. Therefore, when counting is started, it cannot be stopped by any means other than $\overline{\text{RESET}}$ input.

Caution When the watchdog timer is cleared by setting RUN to 1, the actual overflow time is up to
★ **2^8/fx seconds shorter than the time set by the watchdog timer clock select register (WDCS).**

Remark ×: don't care

(3) Oscillation stabilization time select register (OSTS)

This register selects the oscillation stabilization time that elapses after the $\overline{\text{RESET}}$ signal is applied or the STOP mode is released, until oscillation is stabilized.

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 04H. Therefore, to release the STOP mode by inputting the $\overline{\text{RESET}}$ signal, the time required to release the mode is 2^{17}/fx.

★ **Figure 9-4. Format of Oscillation Stabilization Time Select Register**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time when STOP mode is released		
				At fx = 12 MHz[Note]	At fx = 8.38 MHz
0	0	0	2^{12}/fx	341.3 µs	488.8 µs
0	0	1	2^{14}/fx	1.36 ms	1.96 ms
0	1	0	2^{15}/fx	2.73 ms	3.91 ms
0	1	1	2^{16}/fx	5.46 ms	7.82 ms
1	0	0	2^{17}/fx	10.9 ms	15.6 ms
Other than above			Setting prohibited		

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

9.5 Operation of Watchdog Timer

9.5.1 Operation as watchdog timer

The watchdog timer detects an inadvertent program loop when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1.

The loop detection time interval of the watchdog timer can be selected by bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set loop detection time interval after the watchdog timer has been started. By setting RUN to 1, the watchdog timer can be cleared and made to start counting. If RUN is not set to 1 and the loop detection time is exceeded, the system is reset or a non-maskable interrupt request is generated by the value of bit 3 (WDTM3) of WDTM.

The watchdog timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set RUN to 1 before entering the STOP mode to clear the watchdog timer, and then execute the STOP instruction.

★ **Cautions 1. The actual loop detection time may be up to 2^8/fx seconds shorter than the set time.**

2. The count operation of the watchdog timer is stopped when the subsystem clock is selected as the CPU clock.

★ **Table 9-4. Loop Detection Time of Watchdog Timer**

WDCS22	WDCS21	WDCS20	Loop Detection Time	At fx = 12 MHz[Note]	At fx = 8.38 MHz
0	0	0	$2^{12} \times 1/f_X$	341.3 μs	488.8 μs
0	0	1	$2^{13} \times 1/f_X$	682.6 μs	977.6 μs
0	1	0	$2^{14} \times 1/f_X$	1.36 ms	1.96 ms
0	1	1	$2^{15} \times 1/f_X$	2.73 ms	3.91 ms
1	0	0	$2^{16} \times 1/f_X$	5.46 ms	7.82 ms
1	0	1	$2^{17} \times 1/f_X$	10.9 ms	15.6 ms
1	1	0	$2^{18} \times 1/f_X$	21.8 ms	31.3 ms
1	1	1	$2^{20} \times 1/f_X$	87.3 ms	125.1 ms

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

9.5.2 Operation as interval timer

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0, the watchdog timer operates as an interval timer that repeatedly generates an interrupt request at time intervals specified by a preset count value.

Bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS) can be used to select the interval time of interval timer. When bit 7 (RUN) of WDTM is set to 1, the watchdog timer starts operating as an interval timer.

In the interval timer mode, the interrupt mask flag (WDTMK) and priority specification flag (WDTPR) are valid, and a maskable interrupt request (INTWDT) can be generated. The default priority of INTWDT is set as the highest of all the maskable interrupt requests.

The interval timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set RUN to 1 before entering the STOP mode to clear the interval timer, and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM has been set to 1 (when the watchdog timer mode is selected), the interval timer mode is not set, unless the $\overline{\text{RESET}}$ signal is input.

★ **2. The interval time immediately after it has been set by WDTM may be up to 2^8/fx seconds shorter than the set time.**

★ **Table 9-5. Interval Time of Interval Timer**

WDCS2	WDCS1	WDCS0	Interval Time	At fx = 12 MHz[Note]	At fx = 8.38 MHz
0	0	0	$2^{12} \times 1/f_X$	341.3 μs	488.8 μs
0	0	1	$2^{13} \times 1/f_X$	682.6 μs	977.6 μs
0	1	0	$2^{14} \times 1/f_X$	1.36 ms	1.96 ms
0	1	1	$2^{15} \times 1/f_X$	2.73 ms	3.91 ms
1	0	0	$2^{16} \times 1/f_X$	5.46 ms	7.82 ms
1	0	1	$2^{17} \times 1/f_X$	10.9 ms	15.6 ms
1	1	0	$2^{18} \times 1/f_X$	21.8 ms	31.3 ms
1	1	1	$2^{20} \times 1/f_X$	87.3 ms	125.1 ms

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

CHAPTER 10 REAL-TIME OUTPUT PORT

10.1 Function of Real-Time Output Port

Data set previously in the real-time output buffer register can be transferred to the output latch by hardware concurrently with timer interrupts or external interrupt request generation, then output externally. This is called the real-time output function. The pins that output data externally are called real-time output ports.

By using the real-time output port, it is possible to output a signal with no jitter. Therefore, this is most suitable for applications where an arbitrary pattern is output at an arbitrary interval (open-loop control of a stepper motor, etc.).

Also, it is possible to perform PWM modulation at a specified pin for the output pattern.

The μPD780988 Subseries has the following 2 channels of real-time output ports on chip. It is possible to specify the real-time output port in 1-bit units.

- 8 bits × 1, or 4 bits × 2 … Real-time output port 0
- 6 bits × 1, or 4 bits × 1 … Real-time output port 1

10.2 Configuration of Real-Time Output Port

A real-time output port includes the following hardware.

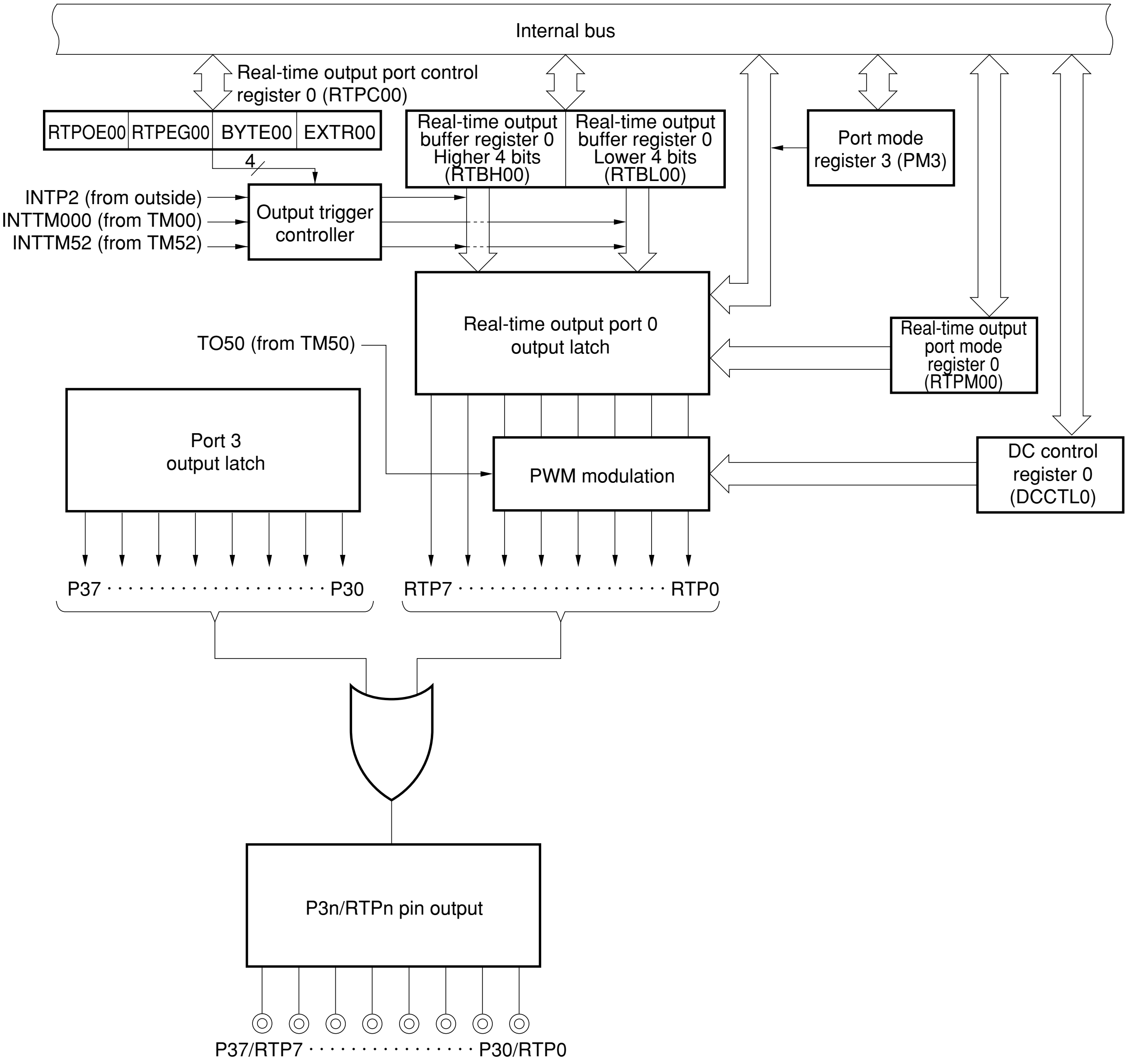
Table 10-1. Configuration of Real-Time Output Port

Item	Configuration
Register	Real-time output buffer register n (RTBL0n, RTBH0n)
Control registers	Port mode register 3 (PM3) Real-time output port mode register n (RTPM0n) Real-time output port control register n (RTPC0n) DC control register n (DCCTLn)

n = 0, 1

Figure 10-1. Block Diagram of Real-Time Output Port (1/2)

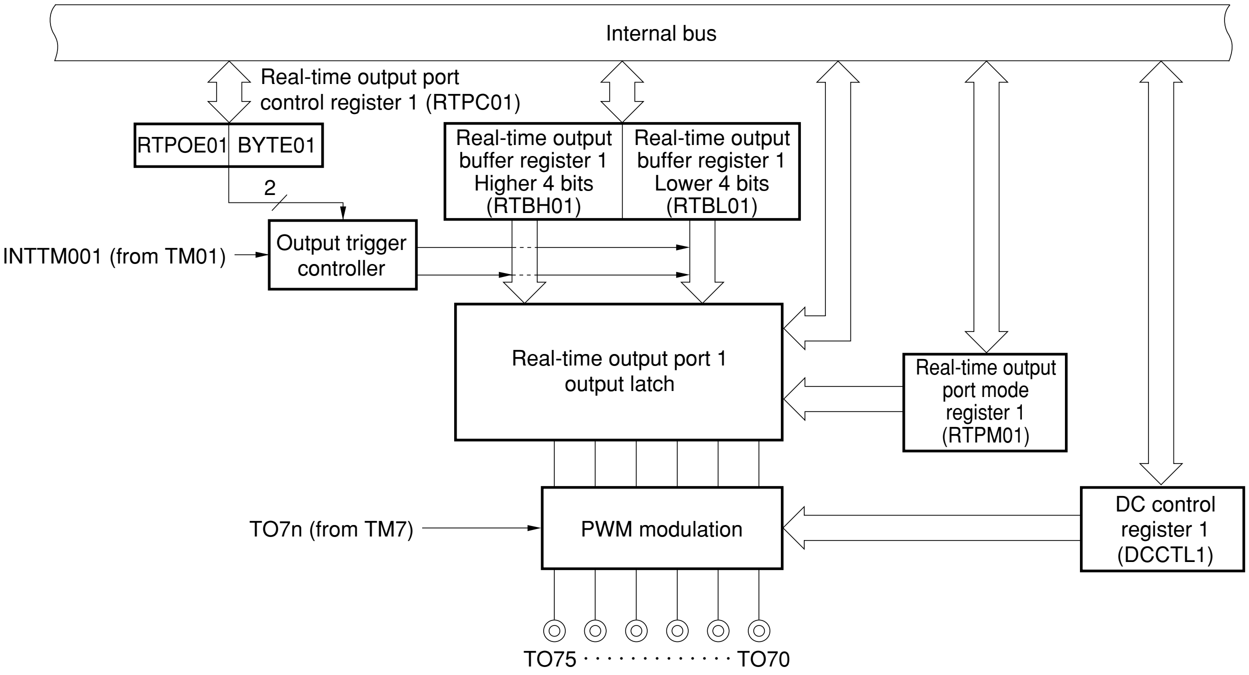
(a) Real-time output port 0 (8 bits × 1, or 4 bits × 2)


Internal bus
Real-time output port control register 0 (RTPC00)
RTPOE00
RTPEG00
BYTE00
EXTR00
4
Real-time output buffer register 0 Higher 4 bits (RTBH00)
Real-time output buffer register 0 Lower 4 bits (RTBL00)
Port mode register 3 (PM3)
INTP2 (from outside)
INTTM000 (from TM00)
INTTM52 (from TM52)
Output trigger controller
Real-time output port 0 output latch
TO50 (from TM50)
Real-time output port mode register 0 (RTPM00)
Port 3 output latch
PWM modulation
DC control register 0 (DCCTL0)
P37 · P30
RTP7 · RTP0
P3n/RTPn pin output
P37/RTP7 · · · · · · · · · · · · · · · · P30/RTP0

Remark n = 0 to 7

Figure 10-1. Block Diagram of Real-Time Output Port (2/2)

(b) Real-time output port 1 (6 bits × 1, or 4 bits × 1)


Internal bus
Real-time output port control register 1 (RTPC01)
RTPOE01
BYTE01
2
Real-time output buffer register 1 Higher 4 bits (RTBH01)
Real-time output buffer register 1 Lower 4 bits (RTBL01)
INTTM001 (from TM01)
Output trigger controller
Real-time output port 1 output latch
Real-time output port mode register 1 (RTPM01)
TO7n (from TM7)
PWM modulation
DC control register 1 (DCCTL1)
TO75 · · · · · · · · · · · · · TO70

Remark n = 0 to 5

(1) Real-time output buffer register 0 (RTBL00, RTBH00)

This register consists of two 4-bit registers that hold output data in advance.

The addresses of RTBL00 and RTBH00 are mapped individually in the special function register (SFR) area as shown in Figure 10-2.

When specifying 4 bits × 2 channels as the operation mode, data is set individually in RTBL00 and RTBH00. The data of both RTBL00 and RTBH00 can be read all at once regardless of which address is specified.

When specifying 8 bits × 1 channel as the operation mode, data is set to both RTBL00 and RTBH00 by writing 8-bit data to either RTBL00 or RTBH00. The data of both RTBL00 and RTBH00 can be read all at once regardless of which address is specified.

Figure 10-2 shows the configuration of RTBL00 and RTBH00, and Table 10-2 shows operations during manipulation of RTBL00 and RTBH00.

Figure 10-2. Configuration of Real-Time Output Buffer Register 0

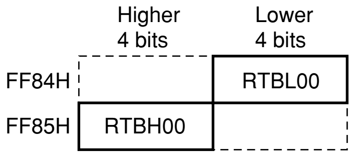


Table 10-2. Operation During Manipulation of Real-Time Output Buffer Register 0

Operating Mode	Register to Be Manipulated	Reading[Note 1]		Writing[Note 2]	
		Higher 4 Bits	Lower 4 Bits	Higher 4 Bits	Lower 4 Bits
4 bits × 2 channels	RTBL00	RTBH00	RTBL00	Invalid	RTBL00
	RTBH00	RTBH00	RTBL00	RTBH00	Invalid
8 bits × 1 channel	RTBL00	RTBH00	RTBL00	RTBH00	RTBL00
	RTBH00	RTBH00	RTBL00	RTBH00	RTBL00

Notes 1. Only the bits set in the real-time output port mode can be read. When a bit set in the port mode is read, 0 is read.

2. After setting data in the real-time output port, output data should be set in RTBL00 and RTBH00 by the time a real-time output trigger is generated.

(2) Real-time output buffer register 1 (RTBL01, RTBH01)

This register consists of two 4-bit[Note] registers that hold output data in advance.

The addresses of RTBL01 and RTBH01 are mapped individually in the special function register (SFR) area as shown in Figure 10-3.

When specifying 4 bits × 1 channel as the operation mode, data is set in RTBL01.

When specifying 6 bits × 1 channel as the operation mode, data is set to both RTBL01 and RTBH01 by writing 6-bit data to either RTBL01 or RTBH01. The data of both RTBL01 and RTBH01 can be read all at once regardless of which address is specified.

Figure 10-3 shows the configuration of RTBL01 and RTBH01, and Table 10-3 shows operations during manipulation of RTBL01 and RTBH01.

Note For RTBH01, only 2 of the 4 bits are valid.

Figure 10-3. Configuration of Real-Time Output Buffer Register 1

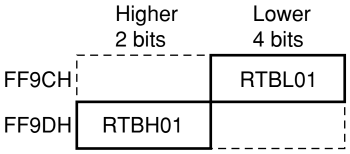


Table 10-3. Operation During Manipulation of Real-Time Output Buffer Register 1

Operating Mode	Register to Be Manipulated	Reading[Note 1]		Writing[Note 2]	
		Higher 2 Bits	Lower 4 Bits	Higher 2 Bits	Lower 4 Bits
4 bits × 1 channel	RTBL01	Invalid	RTBL01	Invalid	RTBL01
6 bits × 1 channel	RTBL01	RTBH01	RTBL01	RTBH01	RTBL01
	RTBH01	RTBH01	RTBL01	RTBH01	RTBL01

Notes 1. Only the bits set in the real-time output port mode can be read. When the bit specified as RTPM01n = 0 (RTPM01n: bit n (n = 0 to 5) of real-time output port mode register 1 (RTPM01)) is read, 0 is read.

2. After setting data in the real-time output port, output data should be set in RTBL01 and RTBH01 by the time a real-time output trigger is generated.

10.3 Registers Controlling Real-Time Output Port

The following seven types of registers control the real-time output ports.

- Port mode register 3 (PM3)
- Real-time output port mode register 0, 1 (RTPM00, RTPM01)
- Real-time output port control register 0, 1 (RTPC00, RTPC01)
- DC control register 0, 1 (DCCTL0, DCCTL1)

(1) Port mode register 3 (PM3)

This register sets the input/output mode of port 3 pins (P30 to P37) that function alternately as real-time output pins (RTP0 to RTP7). To use port 3 as a real-time output port, the input/output mode of the port pins used as real-time output port pins must be set in the output mode (PM3n = 0: n = 0 to 7).

PM3 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-4. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Real-time output port mode register 0 (RTPM00)

This register sets the real-time output port mode or port mode in 1-bit units.

The output is RTP0 to RTP7.

RTPM00 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-5. Format of Real-Time Output Port Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RTPM00	RTPM007	RTPM006	RTPM005	RTPM004	RTPM003	RTPM002	RTPM001	RTPM000	FF86H	00H	R/W

RTPM00n	Real-time output port selection (n = 0 to 7)
0	Port mode
1	Real-time output port mode

Caution When using a port as a real-time output port, set the port in the output mode (by clearing the corresponding bit of port mode register 3 (PM3) to 0).

(3) Real-time output port mode register 1 (RTPM01)

This register sets the real-time output port mode in 1-bit units.

The output is TO70 to TO75.

RTPM01 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-6. Format of Real-Time Output Port Mode Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RTPM01	0	0	RTPM015	RTPM014	RTPM013	RTPM012	RTPM011	RTPM010	FF9EH	00H	R/W

RTPM01n	Real-time output port selection (n = 0 to 5)
0	“0” output
1	Real-time output port mode

Caution Be sure to set bit 6 and 7 of RTPM01 to 0.

Remark When using as a real-time output port, TO70 to TO75 become the output.

(4) Real-time output port control register 0 (RTPC00)

This register is used to set the operation mode, output trigger and operation enable/disable of the real-time output port.

The output is RTP0 to RTP7.

The relationship between the operation mode of the real-time output port and output trigger is as shown in Table 10-4.

RTPC00 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-7. Format of Real-Time Output Port Control Register 0

Symbol	⑦	⑥	⑤	④	3	2	1	0	Address	After reset	R/W
RTPC00	RTPOE00	RTPEG00	BYTE00	EXTR00	0	0	0	0	FF87H	00H	R/W

RTPOE00	Real-time output port operation control
0	Disables operation[Note]
1	Enables operation

RTPEG00	INTP2 valid edge specification
0	Falling edge
1	Rising edge

BYTE00	Real-time output port operation mode
0	4 bits × 2 channels
1	8 bits × 1 channel

EXTR00	Real-time output control by INTP2
0	INTP2 not used as real-time output trigger.
1	INTP2 used as real-time output trigger.

Note When RTPM00n (bit n (n = 0 to 7) of real-time output port mode register 0 (RTPM00)) is 1, INV0 (bit 4 of DC control register 0 (DCCTL0)) is 0, and real-time output operation is disabled (RTPOE00 = 0), RTP0 to RTP7 output “0”.

Table 10-4. Real-Time Output Port Operation Mode and Output Trigger

BYTE00	EXTR00	Operation Mode	RTBH00 → Port Output	RTBL00 → Port Output
0	0	4 bits × 2 channels	INTTM52	INTTM000
0	1		INTTM000	INTP2
1	0	8 bits × 1 channel	INTTM000	
1	1		INTP2	

(5) Real-time output port control register 1 (RTPC01)

This register is used to set the operation mode, and enabling or disabling operation of the real-time output port.

The output is TO70 to TO75.

The relationship between the operation mode of the real-time output port and output trigger is as shown in Table 10-5.

RTPC01 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-8. Format of Real-Time Output Port Control Register 1

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After reset	R/W
RTPC01	RTPOE01	0	BYTE01	0	0	0	0	0	FF9FH	00H	R/W

RTPOE01	Real-time output port operation control
0	Disables operation[Note]
1	Enables operation

BYTE01	Real-time output port operation mode
0	4 bits × 1 channel
1	6 bits × 1 channel

Note When RTPM01n (bit n (n = 0 to 5) of real-time output port mode register 1 (RTPM01)) is 1, INV1 (bit 4 of DC control register 1 (DCCTL1)) is 0, and real-time output operation is disabled (RTPOE01 = 0), TO70 to TO75 output "0".

Table 10-5. Real-Time Output Port Operation Mode and Output Trigger

BYTE01	Operation Mode	RTBH01 → Port Output	RTBL01 → Port Output
0	4 bits × 1 channel	—	INTTM001
1	6 bits × 1 channel	INTTM001	

(6) DC control register 0 (DCCTL0)

This register is used to enable/disable PWM modulation, and enable/disable inversion of the output waveform of the real-time output port.

The output is RTP0 to RTP7.

DCCTL0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-9. Format of DC Control Register 0

Symbol	⑦	⑥	⑤	④	3	2	1	0	Address	After reset	R/W
DCCTL0	DCEN0	PWMCH0	PWMCL0	INV0	0	0	0	0	FFB8H	00H	R/W

DCEN0	Output operation specification
0	RTP output
1	PWM modulated RTP output[Note]

PWMCH0	PWM modulation specification (RTP0, RTP2, RTP4 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

PWMCL0	PWM modulation specification (RTP1, RTP3, RTP5 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

INV0	Output waveform specification
0	Inversion disabled
1	Inversion enabled

Note The PWM signal uses the TO50 output.

Remarks **1.** The output is RTP0 to RTP7.

2. The PWMCH0, PWMCL0, and INV0 settings are valid only when DCEN0 = 1.

(7) DC control register 1 (DCCTL1)

This register is used to enable/disable PWM modulation, and enable/disable inversion of the output waveform of the real-time output port.

The output is TO70 to TO75.

DCCTL1 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-10. Format of DC Control Register 1

Symbol	⑦	⑥	⑤	④	3	2	1	0	Address	After reset	R/W
DCCTL1	DCEN1	PWMCH1	PWMCL1	INV1	0	0	0	0	FFBCH	00H	R/W

DCEN1	Output operation specification
0	Inverter timer output (TO70 to TO75)
1	PWM modulated RTP output[Note]

PWMCH1	PWM modulation specification (TO70, TO72, TO74 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

PWMCL1	PWM modulation specification (TO71, TO73, TO75 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

INV1	Output waveform specification
0	Inversion disabled
1	Inversion enabled

Note The PWM signal uses the inverter timer output (TO70 to TO75).

Remarks **1.** The output is TO70 to TO75.

2. The PWMCH1, PWMCL1, and INV1 settings are valid only when DCEN1 = 1.

10.4 Operation of Real-Time Output Port

(1) Using RTP0 to RTP7 as the real-time output port ..... Real-time output port 0 (8 bits × 1, or 4 bits × 2)

When bit 7 (RTPOE00) of real-time output port control register 0 (RTPC00) is 1, and real-time output operation is enabled, the data in real-time output buffer register 0 (RTBH00, RTBL00) is transferred to the output latch in synchronization with the generation of the selected transfer trigger (set by EXTR00 and BYTE00). Of the transferred data, only the data of the bit specified for the real-time output port by setting real-time output port mode register 0 (RTPM00) is output from bits RTP0 to RTP7. The ports specified as port mode by RTPM00 can be used as general-purpose input/output ports.

The operation mode can be selected as 8 bits × 1, or 4 bits × 2, by setting EXTR00 and BYTE00. By setting INV0, it is possible to invert the output waveform. Also, by setting PWMCL0 and PWMCH0, it is possible to perform PWM modulation of the output pattern.

If real-time output was disabled (RTPOE00 = 0) when RTPM00n = 1 and INV0 = 0, then RTP0 to RTP7 output 0.

The relationship between the settings for each bit of the control register and the real-time output is shown in Table 10-6, and an example of the operation timing is shown in Figure 10-11.

Remark	EXTR00:	Bit 4 of real-time output port control register 0 (RTPC00)
	BYTE00:	Bit 5 of real-time output port control register 0 (RTPC00)
	INV0:	Bit 4 of DC control register 0 (DCCTL0)
	PWMCL0, PWMCH0:	Bits 5 and 6 of DC control register 0 (DCCTL0)
	RTPM00n:	Bit n (n = 0 to 7) of real-time output port mode register 0 (RTPM00)

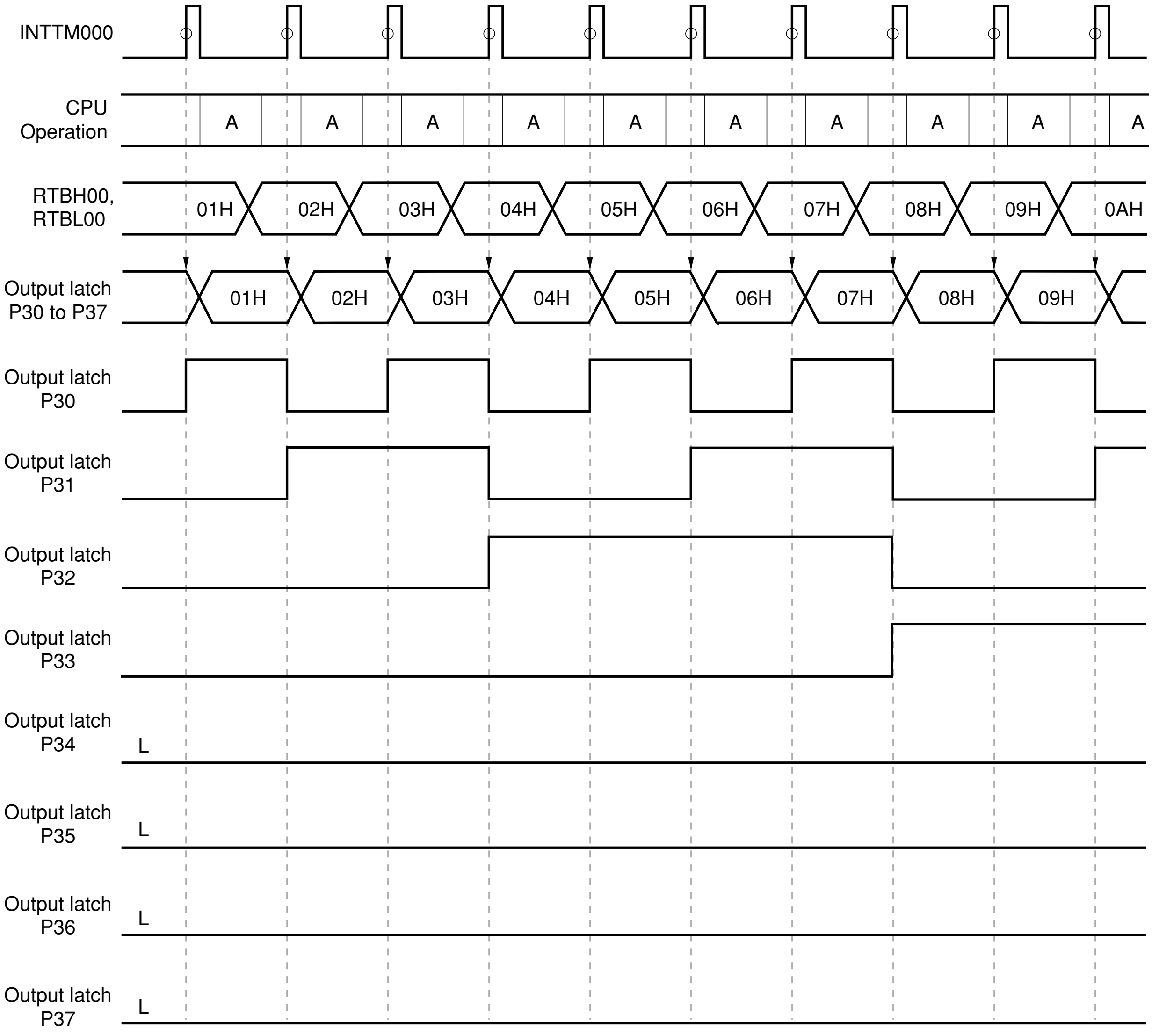
Table 10-6. Relationship Between Settings of Each Bit of Control Register and Real-Time Output

PM3n	P3n	DCEN0	INV0	PWMCH0/ PWMCL0	RTPOE00	RTPM00n	RTBH00m/ RTBL00m	Pin P3n Status
1	×	×	×	×	×	×	×	Input port
0	1	×	×	×	×	×	×	“high” output
	0	0	×	×	0	×	×	“low” output
					1	0	×	“low” output
						1	0	“low” output
							1	“high” output
		1	0	0	0	×	×	“low” output
					1	0	×	“low” output
						1	0	“low” output
							1	“high” output
				1	0	×	×	“TO50” output
					1	0	×	“TO50” output
						1	0	“TO50” output
							1	“high” output
			1	0	0	×	×	“high” output
					1	0	×	“high” output
						1	0	“high” output
							1	“low” output
				1	0	×	×	“$\overline{\text{TO50}}$” output
					1	0	×	“$\overline{\text{TO50}}$” output
						1	0	“$\overline{\text{TO50}}$” output
							1	“low” output

PM3n: Bit n of port mode register 3 (PM3)
P3n: Bit n of port 3 (P3)
DCEN0: Bit 7 of DC control register 0 (DCCTL0)
INV0: Bit 4 of DCCTL0
PWMCH0: Bit 6 of DCCTL0
PWMCL0: Bit 5 of DCCTL0
RTPOE00: Bit 7 of real-time output port control register 0 (RTPC00)
RTPM00n: Bit n of real-time output port mode register 0 (RTPM00)
RTBH00m: Bit m of real-time output buffer register 0H (RTBH00)
RTBL00m: Bit m of real-time output buffer register 0L (RTBL00)
n = 0 to 7
m = 0 to 3
×: don’t care

Figure 10-11. Real-Time Output Port Operation Timing Example (8 Bits × 1) (1/3)

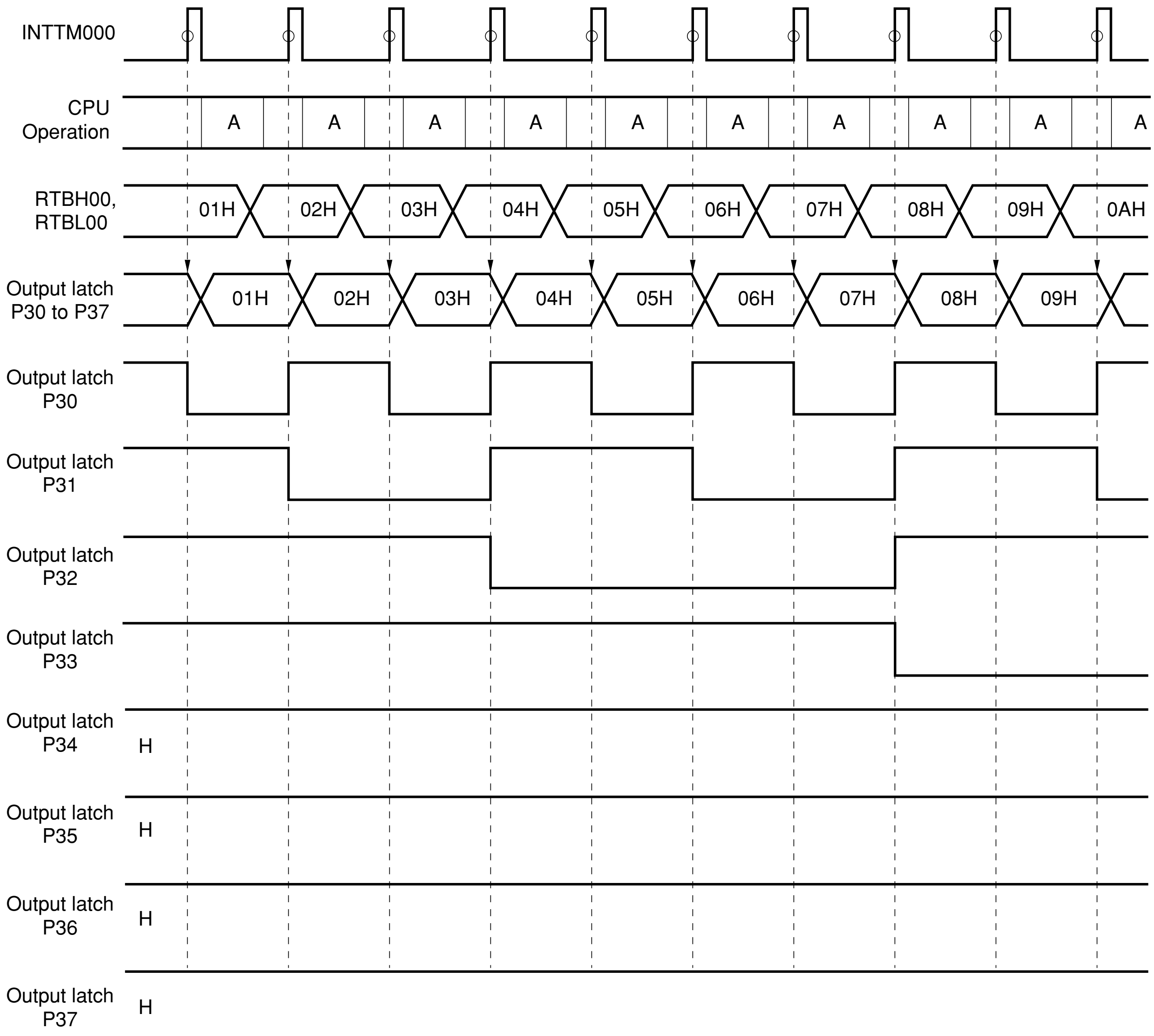
(a) 8 bits × 1 channel, inverted output disabled, no PWM modulation (EXTR00 = 0, BYTE00 = 1, INV0 = 0, PWMCH0 = 0, PWMCL0 = 0)



A: INTTM000 software processing (RTBH00, RTBL00 write)

Figure 10-11. Real-Time Output Port Operation Timing Example (8 Bits × 1) (2/3)

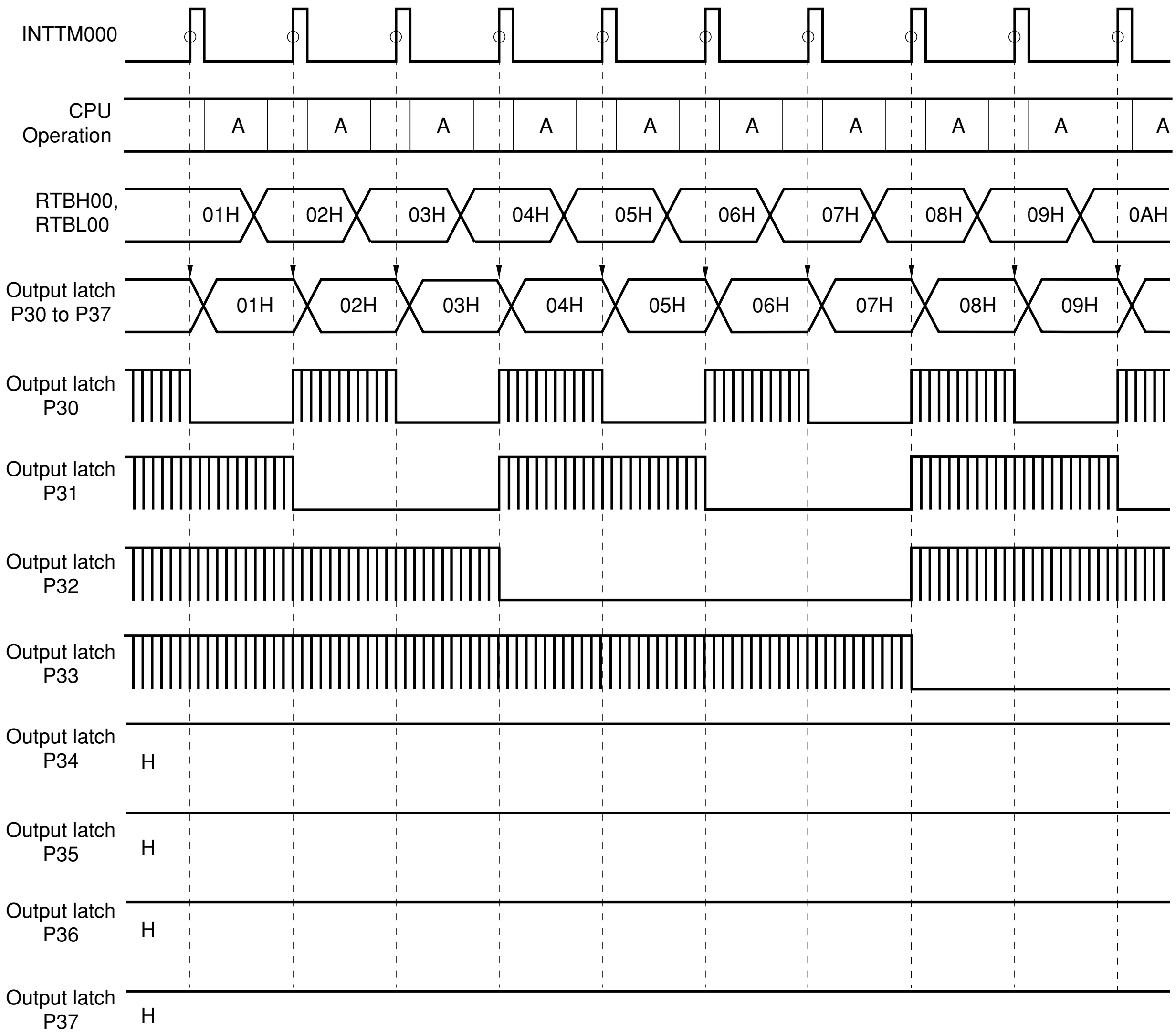
(b) 8 bits × 1 channel, inverted output enabled, no PWM modulation
(EXTR00 = 0, BYTE00 = 1, INV0 = 1, PWMCH0 = 0, PWMCL0 = 0)



A: INTTM000 software processing (RTBH00, RTBL00 write)

Figure 10-11. Real-Time Output Port Operation Timing Example (8 Bits × 1) (3/3)

(c) 8 bits × 1 channel, inverted output enabled, PWM modulation (EXTR00 = 0, BYTE00 = 1, INV0 = 1, PWMCH0 = 1, PWMCL0 = 1)



A: INTTM000 software processing (RTBH00, RTBL00 write)

(2) Using TO70 to TO75 as a real-time output port ..... Real-time output port 1 (6 bits × 1, or 4 bits × 1)

If real-time output is enabled when bit 7 (RTPOE01) of real-time output port control register 1 (RTPC01) is 1, the data of real-time output buffer register 1 (RTBH01, RTBL01) is transferred to the output latch in synchronization with the generation of INTTM001. Of the transferred data, only the data of the bit specified as the real-time output port by setting real-time output port mode register 1 (RTPM01) is output from bits TO70 to TO75. It is possible to use TO70 to TO75 as inverter timer output when inverter timer output is specified by DCEN1.

The operation mode can be selected as 6 bits × 1, or 4 bits × 1, by setting BYTE01.

By setting INV1, it is possible to invert the output waveform. Also, by setting PWMCL1 and PWMCH1, it is possible to perform PWM modulation of the output pattern.

If real-time output was disabled (RTPOE01 = 0) when RTPM01n = 1 and INV1 = 0, then TO70 to TO75 output 0.

The relationship between the settings for each bit of the control register and the real-time output is shown in Table 10-7, and an example of the operation timing is shown in Figure 10-12.

Remark

BYTE01:	Bit 5 of real-time output port control register 1 (RTPC01)
DCEN1:	Bit 7 of DC control register 1 (DCCTL1)
INV1:	Bit 4 of DC control register 1 (DCCTL1)
PWMCL1, PWMCH1:	Bits 5 and 6 of DC control register 1 (DCCTL1)
RTPM01n:	Bit n (n = 0 to 5) of real-time output port mode register 1 (RTPM01)

Table 10-7. Relationship Between Settings of Each Bit of Control Register and Real-Time Output

CE7	DCEN1	INV1	PWMCH1/ PWMCL1	RTPOE01	RTPM01n	RTBH01m/ RTBL01m	Pin TO7n Status
0	×	×	×	×	×	×	Hi-Z
1	0	×	×	×	×	×	TO7n
	1	0	0	0	×	×	“low” output
				1	0	×	“low” output
					1	0	“low” output
						1	“high” output
			1	0	×	×	TO7n
				1	0	×	TO7n
					1	0	TO7n
						1	“high” output
		1	0	0	×	×	“high” output
				1	0	×	“high” output
					1	0	“high” output
						1	“low” output
			1	0	×	×	$\overline{\text{TO7n}}$
				1	0	×	$\overline{\text{TO7n}}$
					1	0	$\overline{\text{TO7n}}$
						1	“low” output

CE7: Bit 7 of inverter timer control register 7 (TMC7)

DCEN1: Bit 7 of DC control register (DCCTL1)

INV1: Bit 4 of DCCTL1

PWMCH1: Bit 6 of DCCTL1

PWMCL1: Bit 5 of DCCTL1

RTPOE01: Bit 7 of real-time output port control register 1 (RTPC01)

RTPM01n: Bit n of real-time output port mode register 1 (RTPM01)

RTBH01m: Bit m of real-time output buffer register 1H (RTBH01)

RTBL01m: Bit m of real-time output buffer register 1L (RTBL01)

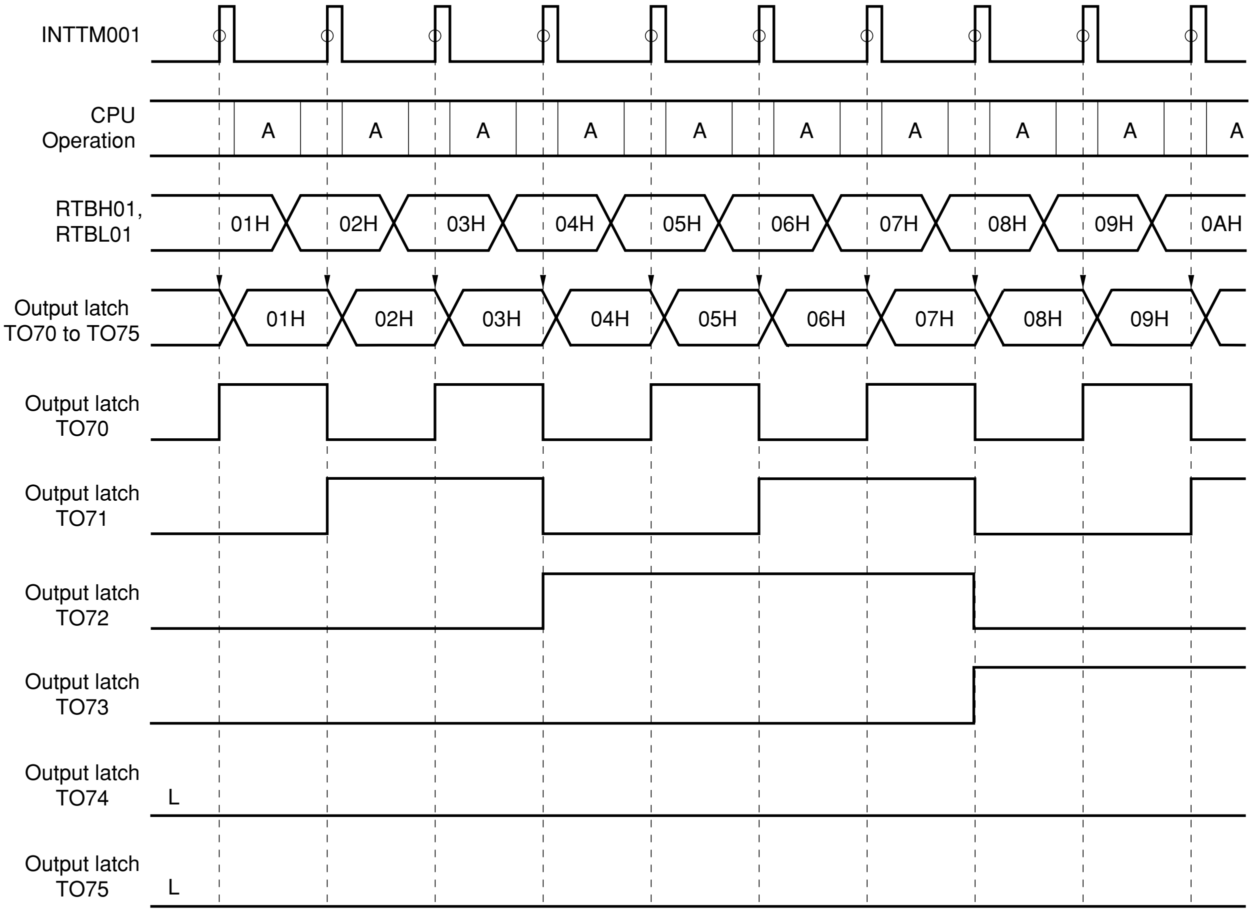
n = 0 to 5

m = 0 to 3

×: don’t care

Figure 10-12. Real-Time Output Port Operation Timing Example (6 Bits × 1) (1/3)

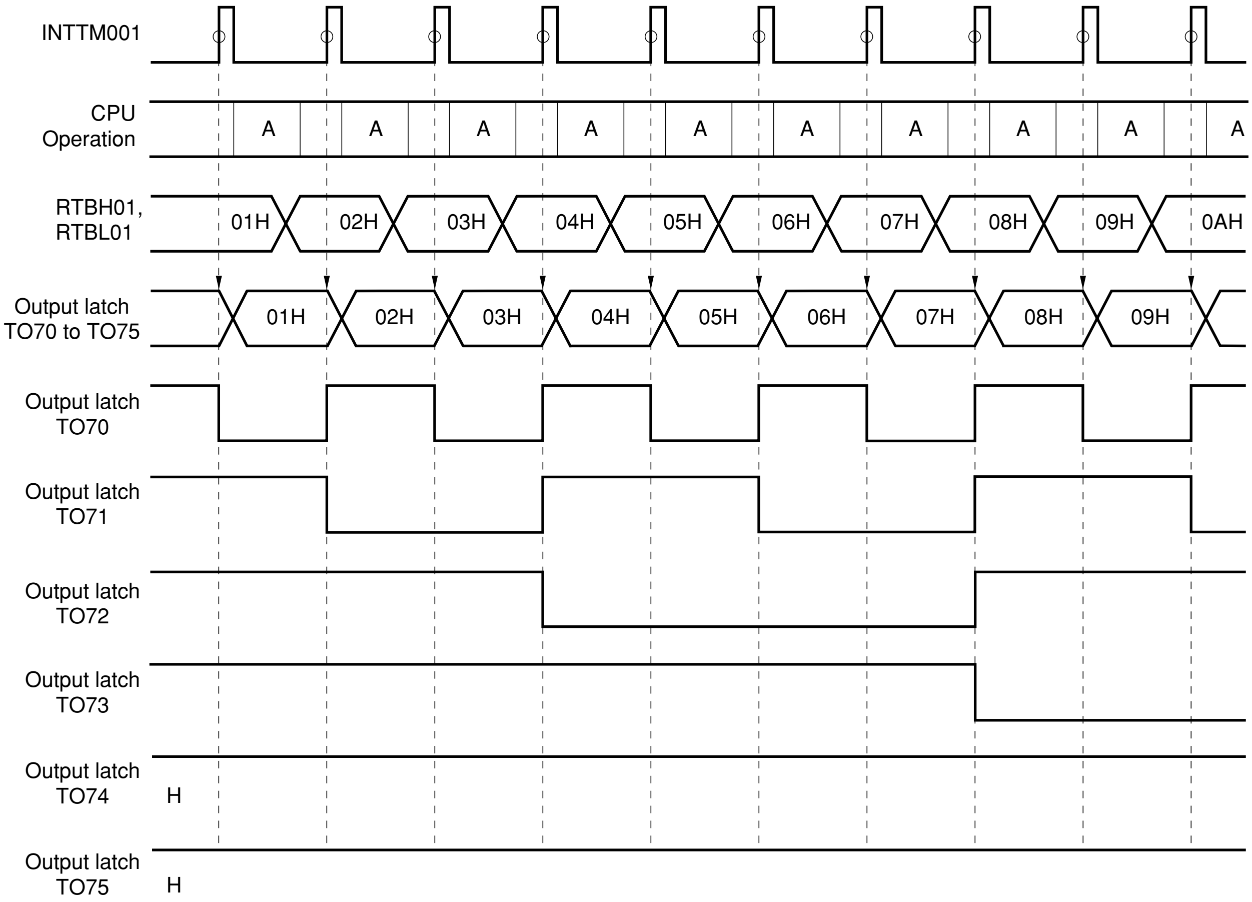
(a) 6 bits × 1 channel, inverted output disabled, no PWM modulation (BYTE01 = 1, INV1 = 0, PWMCH1 = 0, PWMCL1 = 0)



A: INTTM001 software processing (RTBH01, RTBL01 write)

Figure 10-12. Real-Time Output Port Operation Timing Example (6 Bits × 1) (2/3)

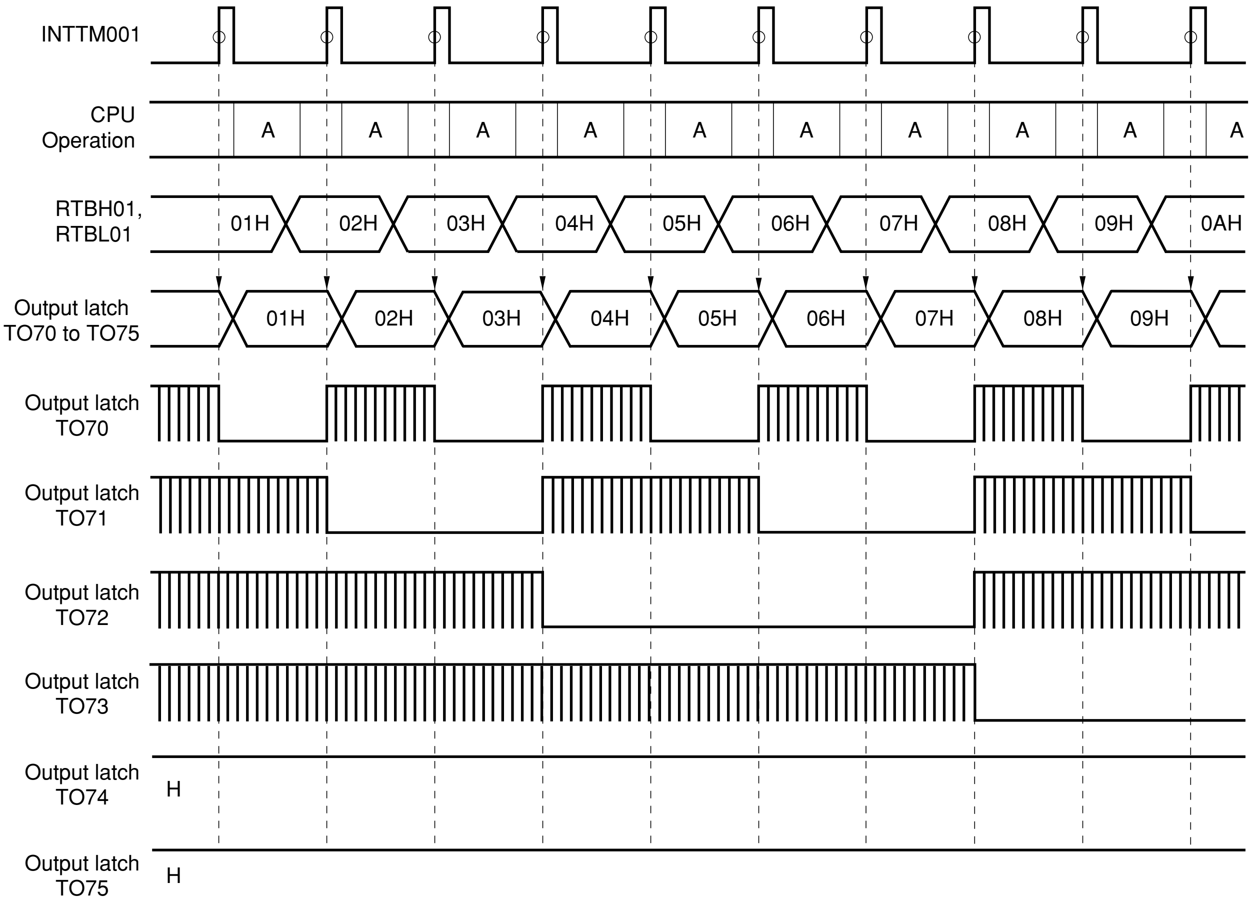
(b) 6 bits × 1 channel, inverted output enabled, no PWM modulation
(BYTE01 = 1, INV1 = 1, PWMCH1 = 0, PWMCL1 = 0)



A: INTTM001 software processing (RTBH01, RTBL01 write)

Figure 10-12. Real-Time Output Port Operation Timing Example (6 Bits × 1) (3/3)

(c) 6 bits × 1 channel, inverted output enabled, PWM modulation (BYTE01 = 1, INV1 = 1, PWMCH1 = 1, PWMCL1 = 1)



A: INTTM001 software processing (RTBH01, RTBL01 write)

10.5 Using Real-Time Output Port

When using the real-time output port, perform the following steps.

(1) Disable real-time output operation.
Clear bit 7 (RTPOE0n) of real-time output port control register n (RTPC0n) to 0.

(2) Initial setting

- Set the initial value to the port output latch (real-time output port 0 only).
- Specify the real-time output port mode in 1-bit units.
  Set real-time output port mode register n (RTPM0n).
- Select the operation mode (trigger and a valid edge).
  Set bits 4, 5, and 6 (EXTR00, BYTE00, and RTPEG00) of RTPC00 or set bit 5 (BYTE01) of RTPC01.
- For real-time output port 0, set an initial value equal to the port output latch in real-time output buffer register 0 (RTBH00, RTBL00).
  For real-time output port 1, set an initial value in real-time output buffer register 1 (RTBH01, RTBL01).
- Set DC control register n (DCCTLn).

(3) Enable the real-time output operation.
RTPOE0n = 1

(4) Set the port output latch to 0 (only for real-time output port 0).

Remark For real-time output port 0, the value output by the real-time output operation is the ORed value of the output latch of the port and real-time output (see **Figure 10-1 (a)**). Therefore, when real-time output port 0 is used, the port output latch should be set to 0 after the real-time output operation is enabled (RTPOE00 = 0 → 1) until the first transfer trigger is generated.

(5) Set the next output to RTBH0n and RTBL0n before the selected transfer trigger is generated.

(6) Sequentially set the next real-time output value to RTBH0n and RTBL0n by using the interrupt servicing corresponding to the selected trigger.

Remark n = 0, 1

10.6 Notes on Real-Time Output Port

(1) Before performing the initial setting, disable the real-time output operation by clearing bit 7 (RTPOE0n) of real-time output port control register n (RTPC0n) to 0 (n = 0, 1).

(2) Once the real-time output operation has been disabled (RTPOE0n = 0), be sure to set the same initial value as the output latch to real-time output buffer register n (RTBH0n and RTBL0n) before enabling the real-time output operation (RTPOE0n = 0 → 1) (n = 0, 1).

CHAPTER 11 A/D CONVERTER

11.1 Function of A/D Converter

The A/D converter converts analog input signals into digital values, and consists of eight channels (ANI0 to ANI7) with a resolution of 10 bits.

This A/D converter is of successive approximation type, and the result of conversion is held by 10-bit A/D conversion result register 0 (ADCR0).

A/D conversion can be started in the following two ways.

(1) Hardware start

Conversion is started by trigger input (ADTRG; rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

Conversion is started by setting A/D converter mode register 0 (ADM0).

One analog input channel is selected from ANI0 to ANI7 and A/D conversion is executed. A/D conversion is stopped, if it was started by means of hardware, after the conversion is complete, and an interrupt request (INTAD0) is generated. When A/D conversion is started by software, conversion is repeatedly performed. Each time conversion is completed once, INTAD0 is generated.

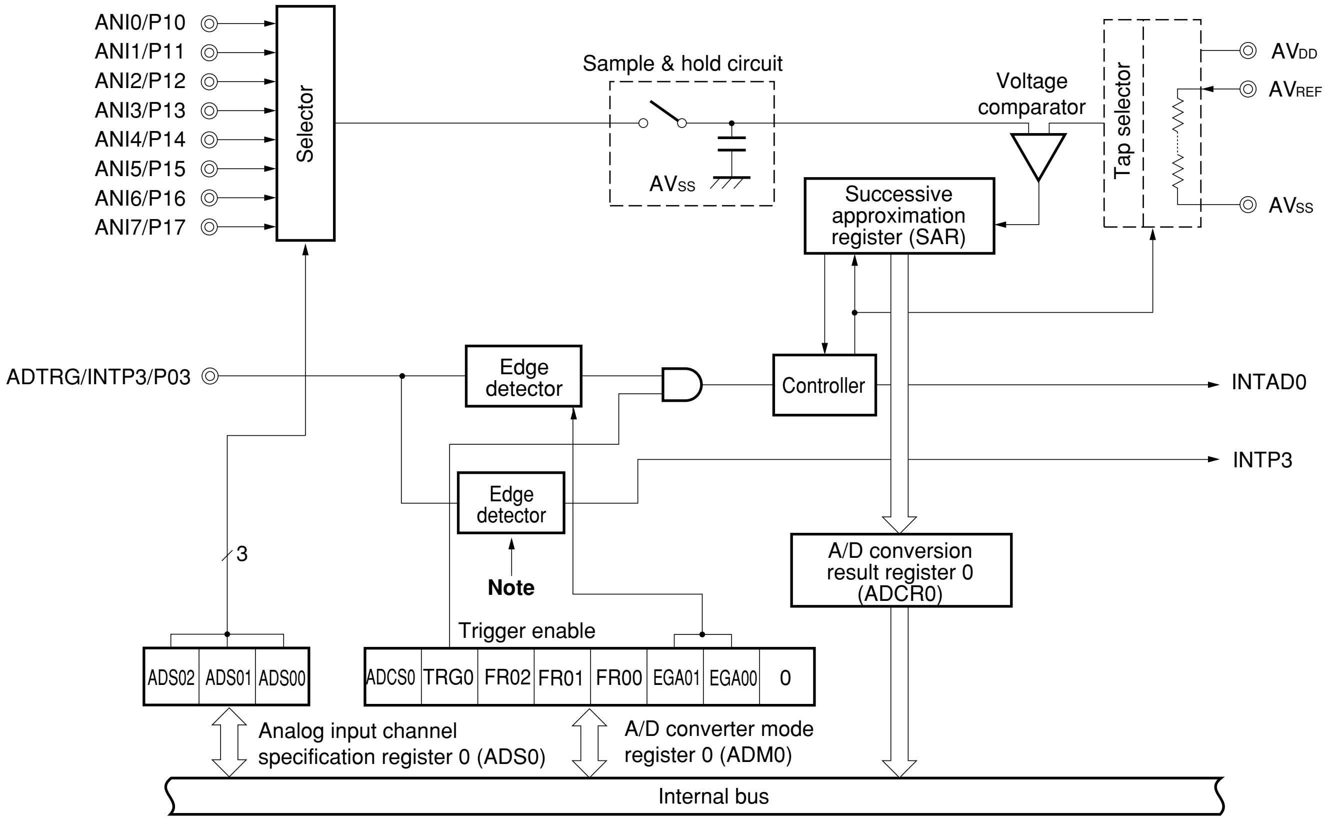
11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

Table 11-1. Configuration of A/D Converter

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Control registers	A/D converter mode register 0 (ADM0) Analog input channel specification register 0 (ADS0)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)

Figure 11-1. A/D Converter Block Diagram



Note Specify the valid edge by using bit 3 (EGP3, EGN3) of the external interrupt rising/falling edge enable registers (EGP, EGN) (refer to **Figure 14-5 Format of External Interrupt Rising Edge Enable Register and External Interrupt Falling Edge Enable Register**).

(1) Successive approximation register (SAR)

This register compares the voltage value of an analog input with the value of a voltage tap (compare voltage) from the series resistor string, and holds the result of the comparison starting from the most significant bit (MSB).

When the result is held down to the least significant bit (LSB) (end of A/D conversion), the contents of SAR are transferred to A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

This is a 16-bit register that stores the results of A/D conversion. The lower 6 bits are fixed to 0. Every time an A/D conversion is complete, the conversion results are loaded from the successive approximation register (SAR). The loaded data is stored in ADCR0 in order from the most significant bit (MSB).

ADCR0 is read with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes the contents of this register undefined.

★ Symbol	FF19H								FF18H								Address	After reset	R/W
ADCR0											0	0	0	0	0	0	FF18H, FF19H	Undefined	R

Caution When a write operation is performed on A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion results after the conversion operation is complete and before the write operation to ADM0 and ADS0. Correct conversion results may not read out at a timing other than the above.

(3) Sample & hold circuit

The sample & hold circuit samples analog input signals sequentially sent from the input circuit on a one-by-one basis, and sends the sampled signals 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 output voltage of the series resistor string.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS} and generates the voltage to be compared with the analog input.

(6) ANI0 to ANI7 pins

These are the eight analog input pin channels of the A/D converter. They input the analog signals that are converted to digital values.

Cautions 1. Observe the rated input voltage range of ANI0 to ANI7. If a voltage of AV_{REF} or higher, or AV_{SS} or lower (even within the range of absolute maximum ratings) is applied to a channel, the converted value of that channel becomes undefined, or the converted value of the other channels may be affected.

2. The analog input pins (ANI0 to ANI7) are also used as input port pins (P10 to P17). When A/D conversion is performed with any of ANI0 to ANI7 selected, do not execute the input instruction to port 1 while conversion is in progress; otherwise, the conversion resolution may be degraded.

If a digital pulse is applied to the pins adjacent to the pin currently being 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 adjacent pins to the pin under going A/D conversion.

(7) AVREF pin

This pin inputs a reference voltage to the A/D converter.

Based on the voltage applied between AV_{REF} and AV_{SS}, the signal input to ANI0 to ANI7 is converted into a digital signal.

Caution A series resistor string of several 10 kΩ is connected between the AV_{REF} and AV_{SS} pins. If the output impedance of the reference voltage source is high, therefore, the error of the reference voltage increases by connecting the impedance in series with the series resistor string between the AV_{REF} and AV_{SS} pins.

(8) AV_{SS} pin

This is the ground pin of the A/D converter. Always make this pin the same potential as the V_{SS0} pin even when the A/D converter is not used.

(9) AV_{DD} pin

This is the analog power supply pin of the A/D converter. Always make this pin the same potential as the V_{DD0} pin even when the A/D converter is not used.

11.3 Registers Controlling A/D Converter

The following two registers control the A/D converter.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register 0 (ADS0)

(1) A/D converter mode register 0 (ADM0)

This register sets conversion time of an analog input to be converted into a digital value, starts/stops the conversion operation, and sets an external trigger.

ADM0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

★

Figure 11-2. Format of A/D Converter Mode Register 0

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After reset	R/W
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	0	FF80H	00H	R/W

ADCS0	A/D conversion operation control
0	Stops operation
1	Enables operation

TRG0	Software start/hardware start selection
0	Software start
1	Hardware start

FR02	FR01	FR00	A/D conversion time selection[Note 1]				
				4.5 V ≤ AV_{DD} ≤ 5.5 V		4.0 V ≤ AV_{DD} < 4.5 V	3.0 V ≤ AV_{DD} < 4.0 V[Note 2]
				At f_X = 12 MHz[Note 2]	At f_X = 8.38 MHz	At f_X = 8.38 MHz	At f_X = 8.38 MHz
0	0	0	144/f_X	12 μs	17.1 μs	17.1 μs	17.1 μs
0	0	1	120/f_X	Setting prohibited[Note 3]	14.3 μs	14.3 μs	Setting prohibited[Note 3]
0	1	0	96/f_X	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]
1	0	0	72/f_X	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]
1	0	1	60/f_X	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]
1	1	0	48/f_X	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]	Setting prohibited[Note 3]
Other than above			Setting prohibited				

EGA01	EGA00	External trigger signal valid edge specification
0	0	No edge is detected
0	1	Detects falling edge
1	0	Detects rising edge
1	1	Detects both rising and falling edges

Notes **1.** Set the A/D conversion time so that it satisfies the following ratings.

<Expanded-specification products>

4.5 V ≤ AV_{DD} ≤ 5.5 V: 12 μs or higher
4.0 V ≤ AV_{DD} < 4.5 V: 14 μs or higher
3.0 V ≤ AV_{DD} < 4.0 V: 17 μs or higher

<Conventional products>

4.0 V ≤ AV_{DD} ≤ 5.5 V: 14 μs or higher

2. Expanded-specification products only

3. Setting prohibited because the A/D conversion time cannot satisfy the ratings in **Note 1** during operation under these conditions.

Caution When rewriting other than the same data to FR00 to FR02, temporarily stop A/D conversion and then rewrite.

Remark f_X: System clock oscillation frequency

(2) Analog input channel specification register 0 (ADS0)

This register sets the input port of the analog voltage to be converted into a digital value.

ADS0 is set by an 8-bit memory manipulation instruction.

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

Figure 11-3. Format of Analog Input Channel Specification Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00	FF81H	00H	R/W

ADS02	ADS01	ADS00	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

11.4 Operation of A/D Converter

11.4.1 Basic operation of A/D converter

<1> Select one channel for A/D conversion using analog input channel specification register 0 (ADS0).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When the voltage has been sampled for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until A/D conversion is completed.

<4> Bit 9 of the successive approximation register (SAR) is set. The voltage tap of the series resistor string is set to (1/2) AV_{REF} by the tap selector.

<5> The voltage differential between the voltage tap of the series resistor string and the analog input is compared by the voltage comparator. If the analog input is higher than (1/2) AV_{REF}, the MSB of the SAR remains set. If it is less than (1/2) AV_{REF}, the MSB is reset.

<6> Next, bit 8 of the SAR is automatically set, and the next voltage differential is compared. Here the voltage tap of the series resistor string is selected as follows, according to the value of bit 9 to which the result of the first comparison has already been set.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

This voltage tap and analog input voltage are compared, and bit 8 of the SAR is manipulated as follows, according to the result of the comparison.

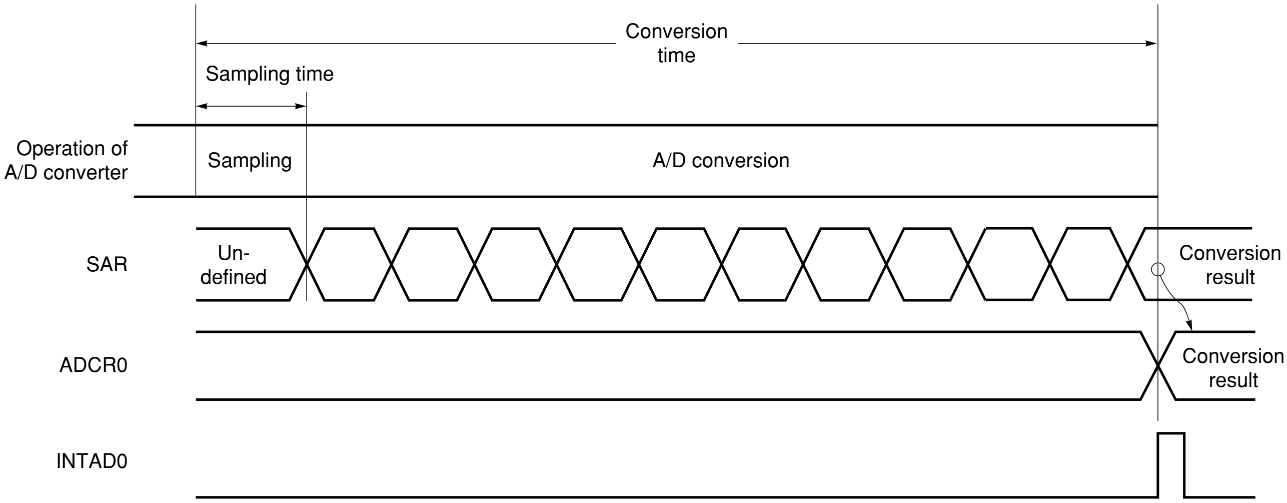
- Analog input voltage ≥ voltage tap: Bit 8 = 1
- Analog input voltage < voltage tap: Bit 8 = 0

<7> In this way, all the bits of the SAR, including bit 0, are compared.

<8> When all the 10 bits of the SAR have been compared, the SAR holds the valid digital results, which are then transferred and latched to A/D conversion result register 0 (ADCR0).
At the same time, an A/D conversion end interrupt request (INTAD0) can be generated.

Caution The first A/D conversion value immediately after starting the A/D conversion operation may not satisfy ratings.

Figure 11-4. Basic Operation of A/D Converter



A/D conversion is performed continuously, until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset to 0 by software.

If the data of ADM0 or analog input channel specification register 0 (ADS0) is rewritten during A/D conversion, the conversion is initialized. If the ADCS0 bit is set to 1 at this time, conversion is performed again from the start.

$\overline{\text{RESET}}$ input makes the contents of A/D conversion result register 0 (ADCR0) undefined.

11.4.2 Input voltage and conversion result

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI7) and A/D conversion result (value stored in A/D conversion result register 0 (ADCR0)) is as follows.

$$\text{ADCR0} = \text{INT}\left(\frac{V_{IN}}{AV_{REF}} \times 1{,}024 + 0.5\right)$$

or,

$$(\text{ADCR0} - 0.5) \times \frac{AV_{REF}}{1{,}024} \leq V_{IN} < (\text{ADCR0} + 0.5) \times \frac{AV_{REF}}{1{,}024}$$

Remark INT( ): Function returning integer of value in parentheses

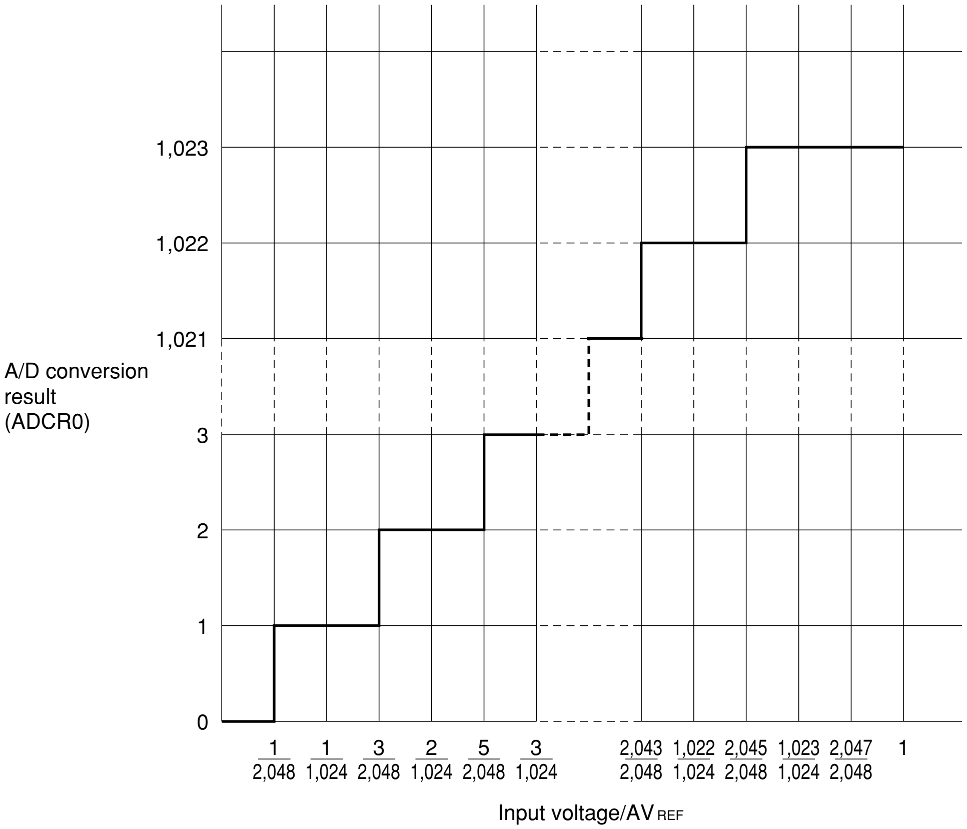
V_{IN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

ADCR0: Value of A/D conversion result register 0 (ADCR0)

Figure 11-5 shows the relationship between the analog input voltage and A/D conversion result.

Figure 11-5. Relationship Between Analog Input Voltage and A/D Conversion Result



11.4.3 Operation mode of A/D converter

One analog input channel is selected from ANI0 to ANI7 for A/D conversion using analog input channel specification register 0 (ADS0).

A/D conversion can be started in the following two ways.

- Hardware start: Conversion is started by trigger input (ADTRG; rising edge/falling edge, or both rising and falling edges can be specified).
- Software start: Conversion is started by setting A/D converter mode register 0 (ADM0).

The result of the A/D conversion is stored in A/D conversion result register 0 (ADCR0), and at the same time, an interrupt request signal (INTAD0) is generated.

(1) A/D conversion operation by hardware start

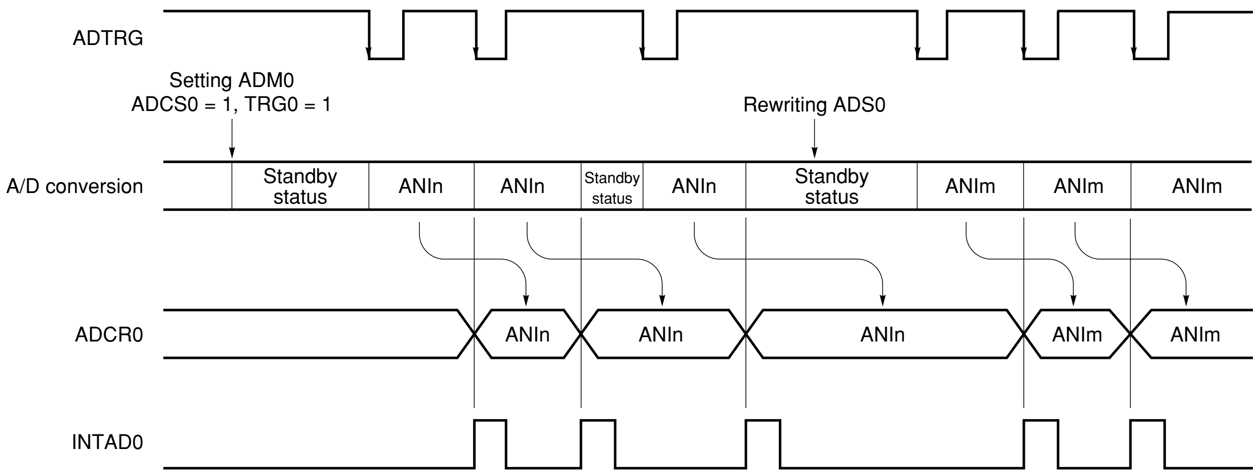
The A/D conversion operation is in standby when both bits 6 (TRG0) and 7 (ADCS0) of A/D converter mode register 0 (ADM0) are set to 1. When an external trigger signal (ADTRG) is input, the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0) is converted into a digital value. When A/D conversion is complete, the result of the conversion is stored in A/D conversion result register 0 (ADCR0), and an interrupt request signal (INTAD0) is generated. Once A/D conversion is started and when one A/D conversion is complete, the next A/D conversion is not started unless a new external trigger signal is input.

If ADS0 is rewritten during A/D conversion, the AD conversion under execution is stopped, and stands by until a new external trigger signal is input. When the external trigger signal is input, A/D conversion is performed again from the start. If ADS0 is rewritten while the A/D converter is standing by, the new A/D conversion operation will be started when the next external trigger signal is input.

When 0 is written to the ADCS0 bit of ADM0 during A/D conversion, the conversion is immediately stopped.

Caution When P03/INTP3/ADTRG is used as an external trigger input (ADTRG), specify the valid edge by using bits 1 and 2 (EGA00 and EGA01) of A/D converter mode register 0 (ADM0) and set the interrupt mask flag (PMK3) to 1.

Figure 11-6. A/D Conversion by Hardware Start (with Falling Edge Specified)



Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

(2) A/D conversion by software start

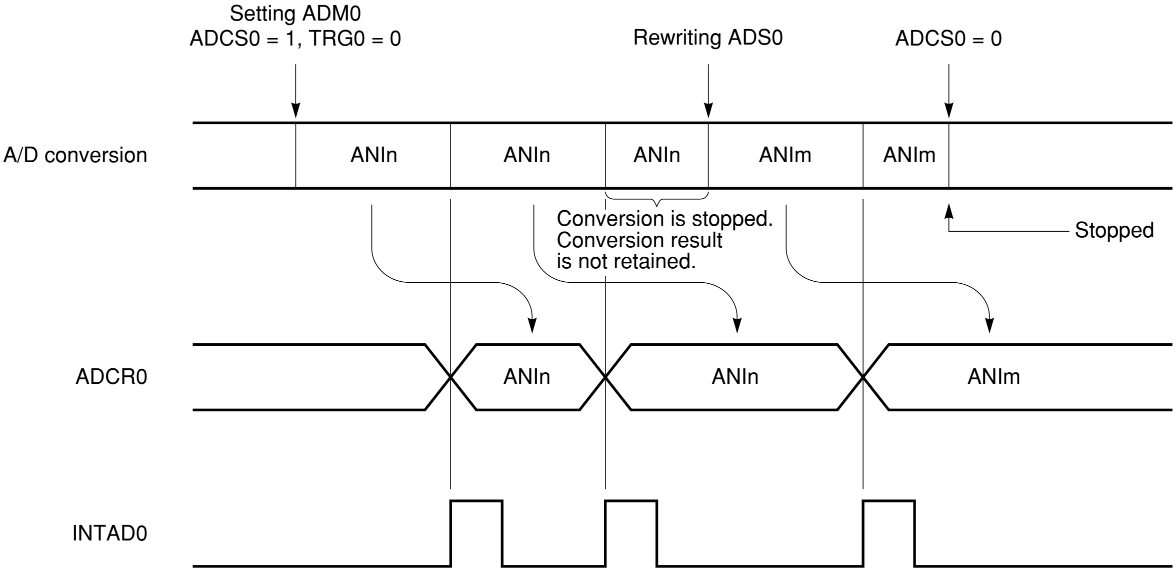
By setting bit 6 (TRG0) of A/D converter mode register 0 (ADM0) to 0 and setting bit 7 (ADCS0) to 1, the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0) is converted into a digital value.

When A/D conversion is complete, the result of the conversion is stored in A/D conversion result register 0 (ADCR0), and an interrupt request signal (INTAD0) is generated. When A/D conversion is started once, and one A/D conversion is complete, the next A/D conversion is immediately started. In this way, A/D conversion is repeatedly executed until new data is written to ADS0.

If ADS0 is rewritten during A/D conversion, the conversion under execution is stopped, and A/D conversion of the newly selected analog input channel is started.

If data whose ADCS0 is 0 is written to ADM0 during A/D conversion, the conversion is immediately stopped.

Figure 11-7. A/D Conversion by Software Start



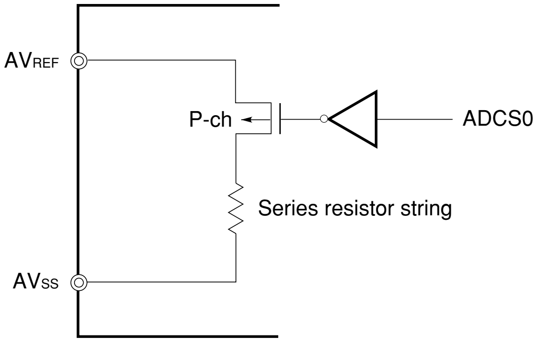
Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

11.5 Notes on 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 clearing bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 0). An example of reducing the current consumption in standby mode is shown in Figure 11-8.

Figure 11-8. Example of Reducing Current Consumption in Standby Mode



(2) ANI0 to ANI7 input range

Observe the rated range of the ANI0 to ANI7 input voltage. If a voltage of AV_{REF} or higher, or AV_{SS} or lower (even 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.

(3) Conflict

<1> Conflict between writing A/D conversion result register 0 (ADCR0) on completion of conversion and reading ADCR0 by instruction

Reading ADCR0 has priority. After it has been read, a new conversion result is written to ADCR0.

<2> Conflict between writing ADCR0 on completion of conversion and external trigger signal input

The external trigger signal is not acknowledged during A/D conversion. Therefore, the external trigger signal is not acknowledged while ADCR0 is being written.

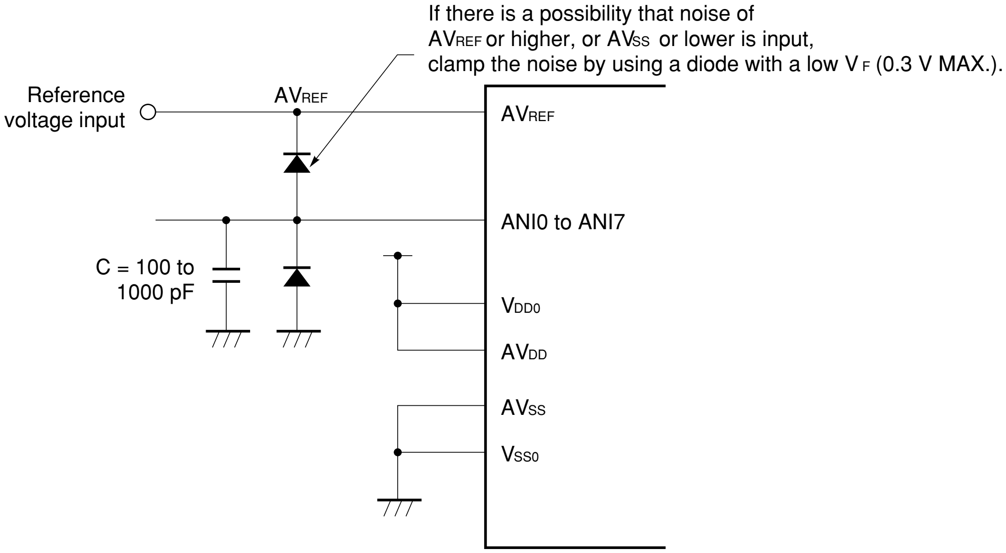
<3> Conflict between writing ADCR0 on completion of conversion and writing A/D converter mode register 0 (ADM0) or writing analog input channel specification register 0 (ADS0)

Writing ADM0 or ADS0 has priority. ADCR0 is not written. The conversion end interrupt request signal (INTAD0) is not generated.

(4) Countermeasures against noise

To keep the resolution of 10 bits, noise superimposed on the AV_{REF} and ANI0 to ANI7 pins must be suppressed as much as possible. The higher the output impedance of the analog input source, the greater the effect. To suppress noise, connecting an external capacitor as shown in Figure 11-9 is recommended.

Figure 11-9. Processing Analog Input Pin


If there is a possibility that noise of
AVREF or higher, or AVSS or lower is input,
clamp the noise by using a diode with a low VF (0.3 V MAX.).
Reference
voltage input
AVREF
C = 100 to
1000 pF
AVREF
ANI0 to ANI7
VDD0
AVDD
AVSS
VSS0

(5) ANI0/P10 to ANI7/P17

The analog input pins (ANI0 to ANI7) are also used as input port pins (P10 to P17).

When A/D conversion is performed with any of ANI0 to ANI7 selected, do not execute the input instruction to port 1 while conversion is in progress; otherwise, the conversion resolution may be degraded.

If a digital pulse is applied to the pins adjacent to the pin currently being 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 under going A/D conversion.

★ ### (6) Input impedance of ANI0 to ANI7 pins

This A/D converter charges the internal sampling capacitor for about 1/10 of the conversion time, and performs sampling.

Therefore at times other than sampling, only the leak current is output. During sampling, the current for charging the capacitor is also output, so the input impedance fluctuates and has no meaning.

However, to ensure adequate sampling, it is recommended that the output impedance of the analog input source be set to below 10 kΩ, or a 100 pF capacitor be connected to the ANI0 to ANI7 pins (see **Figure 11-9**).

(7) Input impedance to AV_{REF} pin

A series resistor string of several 10 kΩ is connected between the AV_{REF} and AV_{SS} pins.

If the output impedance of the reference voltage source is high, therefore, the reference voltage error increases when connecting the impedance in series with the series resistor string between the AV_{REF} and AV_{SS} pins.

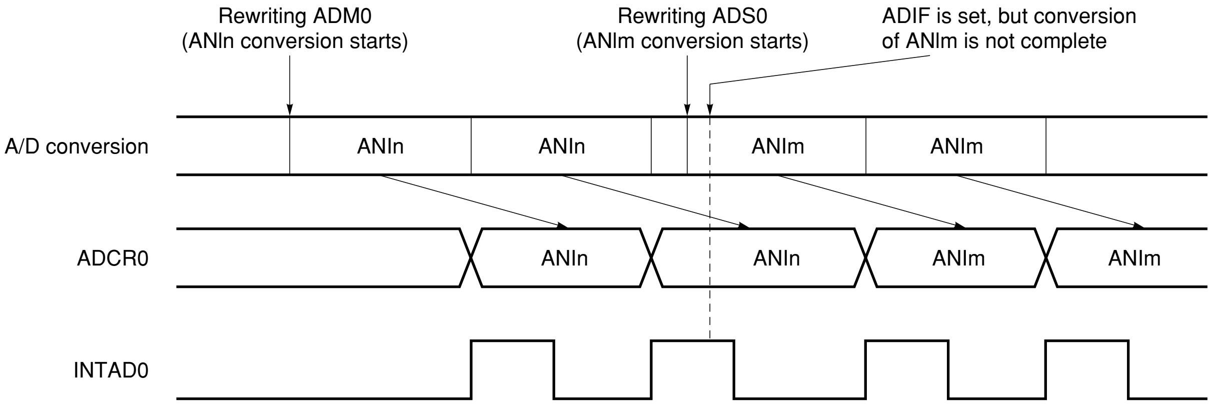
(8) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even when the contents of analog input channel specification register 0 (ADS0) are changed.

When the analog input pin is changed during A/D conversion, therefore, the chances are that the A/D conversion result of the old analog input and interrupt request flags was set immediately before the contents of ADS0 was rewritten. Consequently, ADIF may be set even if A/D conversion for the newly specified analog input pin has not yet been completed when ADIF is read immediately after ADS0 has been rewritten (refer to **Figure 11-10**).

To resume A/D conversion once it has been stopped, clear ADIF first.

Figure 11-10. A/D Conversion End Interrupt Request Generation Timing



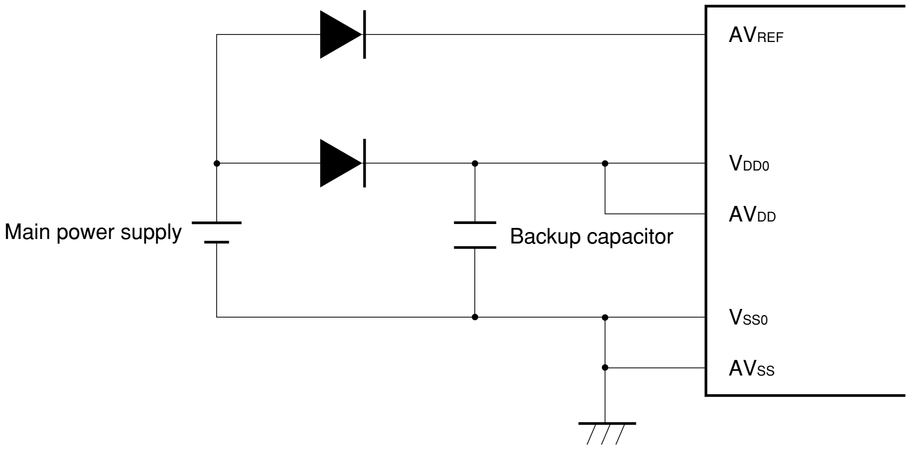
Remark n = 0, 1, ..., 7
n = 0, 1, ..., 7

(9) AV$_{DD}$ pin

The AV$_{DD}$ pin is the power supply pin to the analog circuit and supplies power to the input circuit of ANI0/P10 to ANI7/P17.

Therefore, even in applications that can be switched over to a backup power source, be sure to apply the same voltage as V$_{DD0}$ as shown in Figure 11-11.

Figure 11-11. Processing of AV$_{DD}$ Pin



(10) Conversion result immediately after start of A/D conversion

The first A/D conversion value immediately after starting the A/D conversion operation may not satisfy the ratings.

Poll the A/D conversion end interrupt request (INTAD0) and discard the first conversion result.

(11) Reading A/D conversion result register 0 (ADCR0)

When performing a write operation to A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion results after the conversion operation is complete and before the write operation to ADM0 and ADS0. Correct conversion results may not be read out at a timing other than the above.

(12) Timing at which A/D conversion result is undefined

The A/D conversion value may be undefined if the timing of completion of A/D conversion and the timing of stopping the A/D conversion conflict. Therefore, read the A/D conversion result during the A/D conversion operation. To read the conversion result after stopping the A/D conversion operation, be sure to stop the A/ D conversion before the next conversion ends.

Figures 11-12 and 11-13 show the timing of reading the conversion result.

Figure 11-12. Timing of Reading Conversion Result (When Conversion Result Is Undefined)

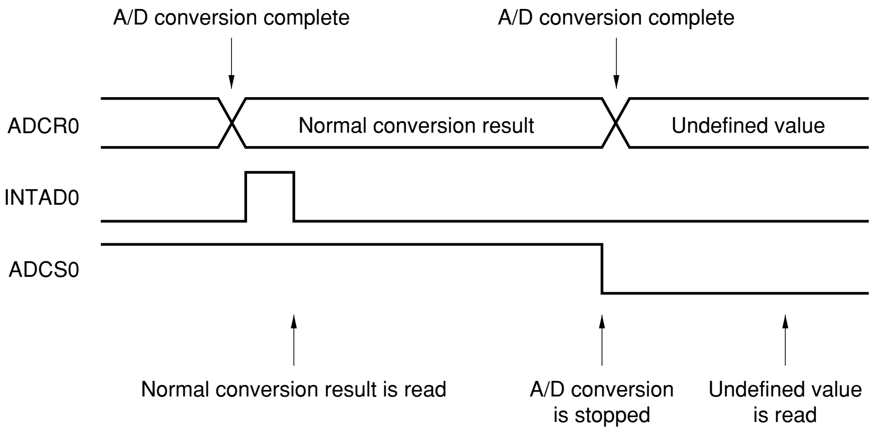
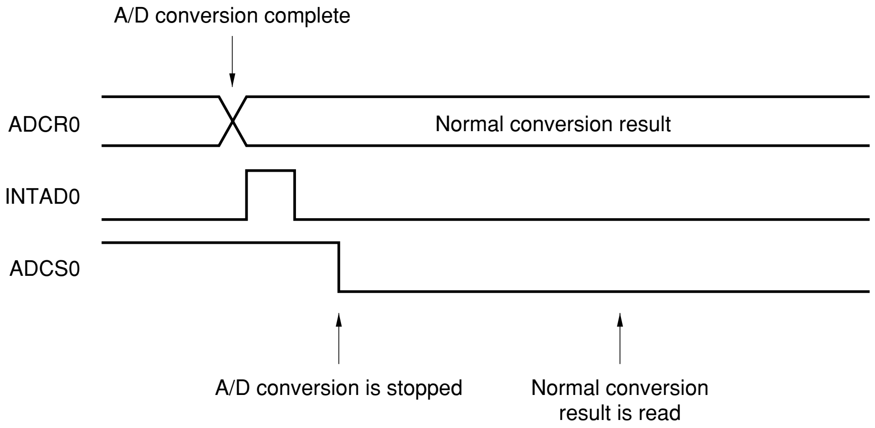


Figure 11-13. Timing of Reading Conversion Result (When Conversion Result Is Normal)



(13) Notes on board design

Locate analog circuits as far away from digital circuits as possible on the board because the analog circuits may be affected by the noise of the digital circuits. In particular, do not cross an analog signal line with a digital signal line, or wire an analog signal line in the vicinity of a digital signal line. Otherwise, the A/D conversion characteristics may be affected by the noise of the digital line.

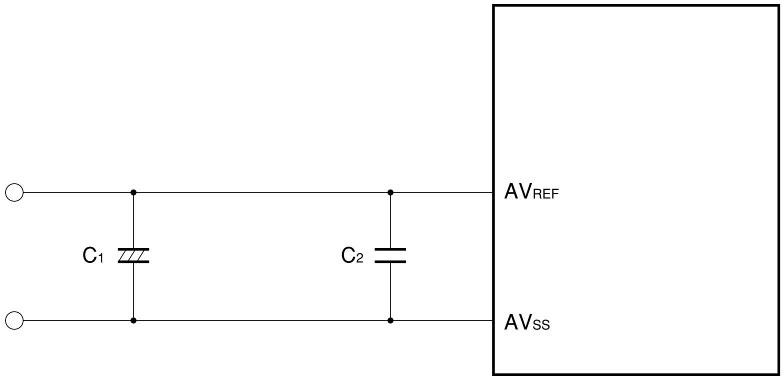
Connect AV_{SS0} and V_{SS0} at one location on the board where the voltages are stable.

(14) AV_{REF} pin

Connect a capacitor to the AV_{REF} pin to minimize conversion errors due to noise. If an A/D conversion operation has been stopped and is then started, the voltage applied to the AV_{REF} pin becomes unstable, causing the accuracy of the A/D conversion to drop. To prevent this, also connect a capacitor to the AV_{REF} pin.

Figure 11-14 shows an example of connecting a capacitor.

Figure 11-14. Example of Connecting Capacitor to AV_{REF} Pin



Remark C1: 4.7 μF to 10 μF (reference value)
C2: 0.01 μF to 0.1 μF (reference value)
Connect C2 as close to the pin as possible.

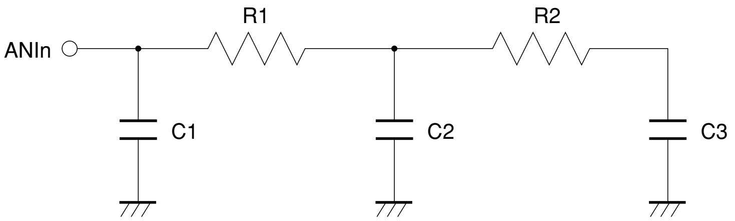
(15) Internal equivalent circuit of ANI0 to ANI7 pins and permissible signal source impedance

To complete sampling within the sampling time with sufficient A/D conversion accuracy, the impedance of the sensor or other signal source must be sufficiently low. Figure 11-15 shows the internal equivalent circuit of the ANI0 to ANI7 pins.

If the impedance of the signal source is high, connect capacitors with a high capacitance to pins ANI0 to ANI7. An example of this is shown in Figure 11-16. In this case, however, the microcontroller cannot follow an analog signal with a high differential coefficient because a lowpass filter is created.

To convert a high-speed analog signal or to convert an analog signal in the scan mode, insert a low-impedance buffer.

Figure 11-15. Internal Equivalent Circuit of Pins ANI0 to ANI7



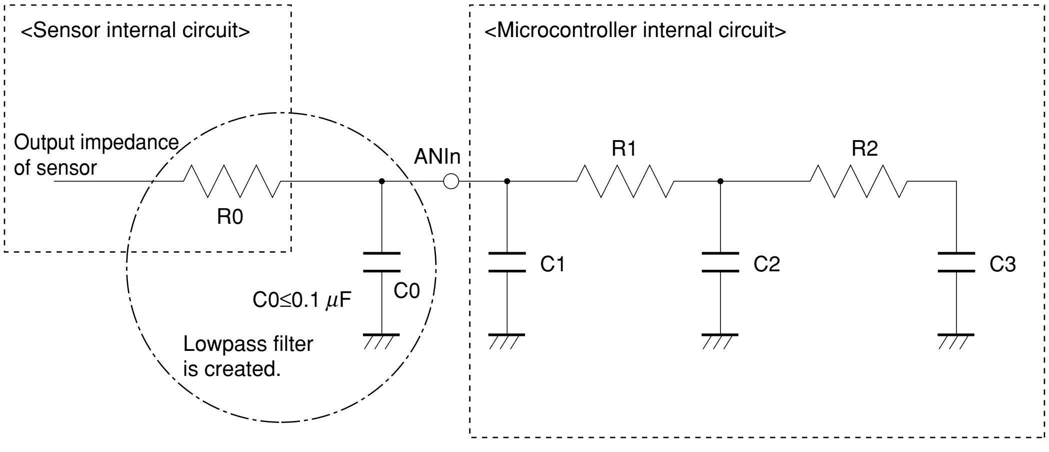
Remark n = 0 to 7

Table 11-2. Resistances and Capacitances of Equivalent Circuit (Reference Values)

AVREF	R1	R2	C1	C2	C3
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF

Caution The resistances and capacitances in Table 11-2 are not guaranteed values.

Figure 11-16. Example of Connection if Signal Source Impedance Is High


<Sensor internal circuit>
<Microcontroller internal circuit>
Output impedance of sensor
R0
ANIn
R1
R2
C0≤0.1 μF
C0
C1
C2
C3
Lowpass filter is created.

Remark n = 0 to 7

11.6 How to Read A/D Converter Characteristics Tables

This section describes the technical terms peculiar to the A/D converter.

(1) Resolution

This is the minimum identifiable analog input voltage. The ratio of 1 digital output bit to an analog input voltage is said to be 1 LSB (Least Significant Bit). The ratio of the full scale to 1 LSB is expressed in %FSR (Full Scale Range).

Where the resolution is 10 bits,

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

The accuracy does not depend on the resolution and is determined by the overall error.

(2) Overall error

This is the maximum difference between actually measured and theoretical values.
The overall error indicates a zero-scale error and full-scale error, an integral linearity error, differential linearity error, and a combination of these errors.
Note that the overall error specified in the characteristics table does not include the quantization error.

(3) Quantization error

This is an error of ±1/2 LSB that inevitably occurs when an analog value is converted into a digital value. Since the A/D converter converts an analog input voltage in the range of ±1/2 LSB into the same digital code, a quantization error is unavoidable.
This error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error specified in the characteristics table.

Figure 11-17. Overall Error

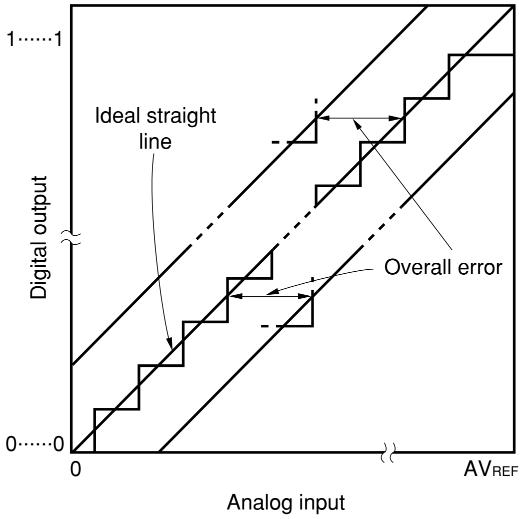
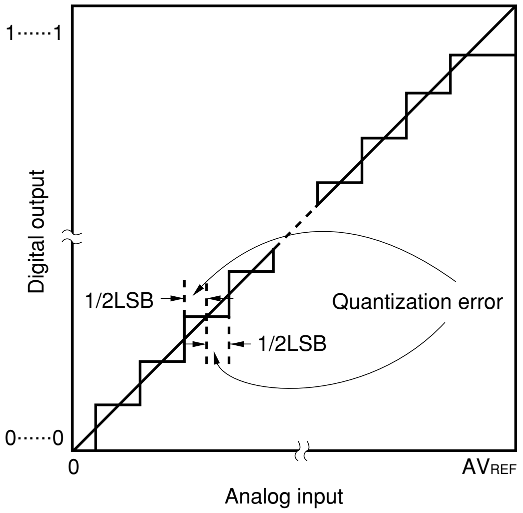


Figure 11-18. Quantization Error



(4) Zero-scale error

This is the difference between the actually measured value and the theoretical value (1/2 LSB) of an analog input voltage when the digital output changes from 0...000 to 0...001. If the measured value is greater than the theoretical value, it is the difference between the actually measured value and the theoretical value (3/2 LSB) of the analog input voltage when the digital output changes from 0...001 to 0...010.

(5) Full-scale error

This is the difference between the actually measured value and the theoretical value (full scale -3/2 LSB) of an analog input voltage when the digital output changes from 1...110 to 1...111.

(6) Integral linearity error

This is the degree to which the conversion characteristics shift from the ideal straight line. It indicates the maximum difference between the measured value and the ideal straight line where the zero-scale error and full-scale error are 0.

(7) Differential linearity error

This is the difference between the actually measured value and the theoretical value of an input voltage when the conversion result changes from a certain value by 1. The differential linearity error indicates the degree of dispersion (relative drift) of input voltage variation required when changing from each conversion value in comparison to the integral linearity error that indicates the absolute value of the drift from the theoretical value.

Figure 11-19. Zero-Scale Error

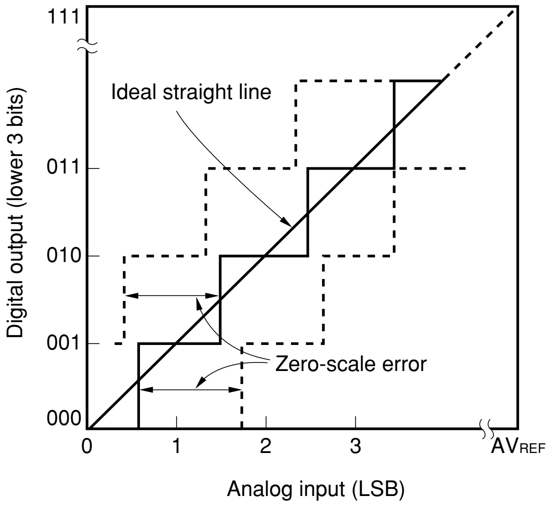


Figure 11-20. Full-Scale Error

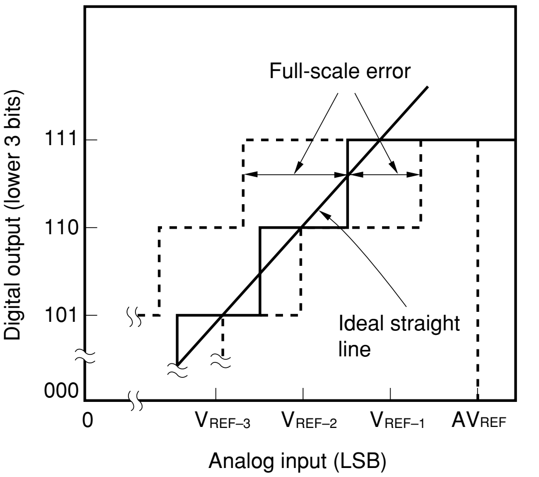


Figure 11-21. Integral Linearity Error

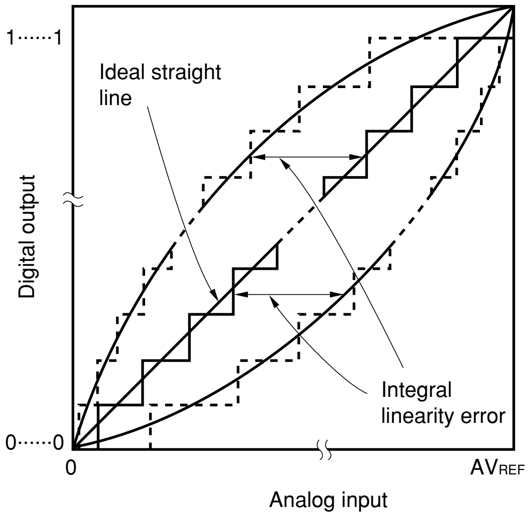
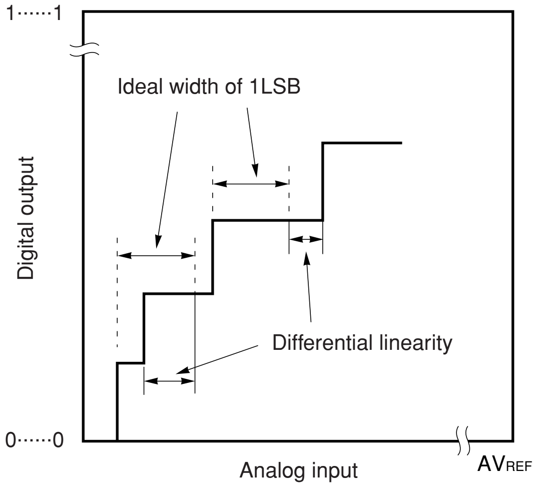


Figure 11-22. Differential Linearity Error



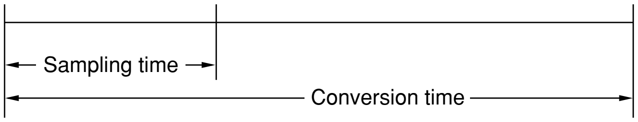
(8) Conversion time

Time required from when an analog input voltage is given until the digital output is obtained.

Sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

Time during which an analog switch is on to load an analog voltage to the sample & hold circuit.



CHAPTER 12 SERIAL INTERFACES UART00 AND UART01

12.1 Function of Serial Interfaces

Serial interfaces UART00 and UART01 have the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- Infrared data transfer mode (UART00 only)

(1) Operation stop mode

This mode is used when serial transfer is not carried out, and is to reduce power consumption.

(2) Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates.

★ Programming baud rate generator control registers 0 and 1 (BRGC00 and BRGC01) allows a baud rate selection of 600 bps to 115.2 kbps (@ f_X = 8.38 MHz operation) or 1200 bps to 153.6 kbps (@ f_X = 12 MHz operation) for UART00 and 300 bps to 38.4 kbps (@ f_X = 8.38 MHz operation) or 600 bps to 76.8 kbps (@ f_X = 12 MHz operation) for UART01.

(3) Infrared data transfer mode (UART00 only)

This mode allows communication at a baud rate of 115.2 kbps (@ f_X = 7.3728 MHz operation).

12.2 Configuration of Serial Interfaces

Serial interfaces UART00 and UART01 includes the following hardware.

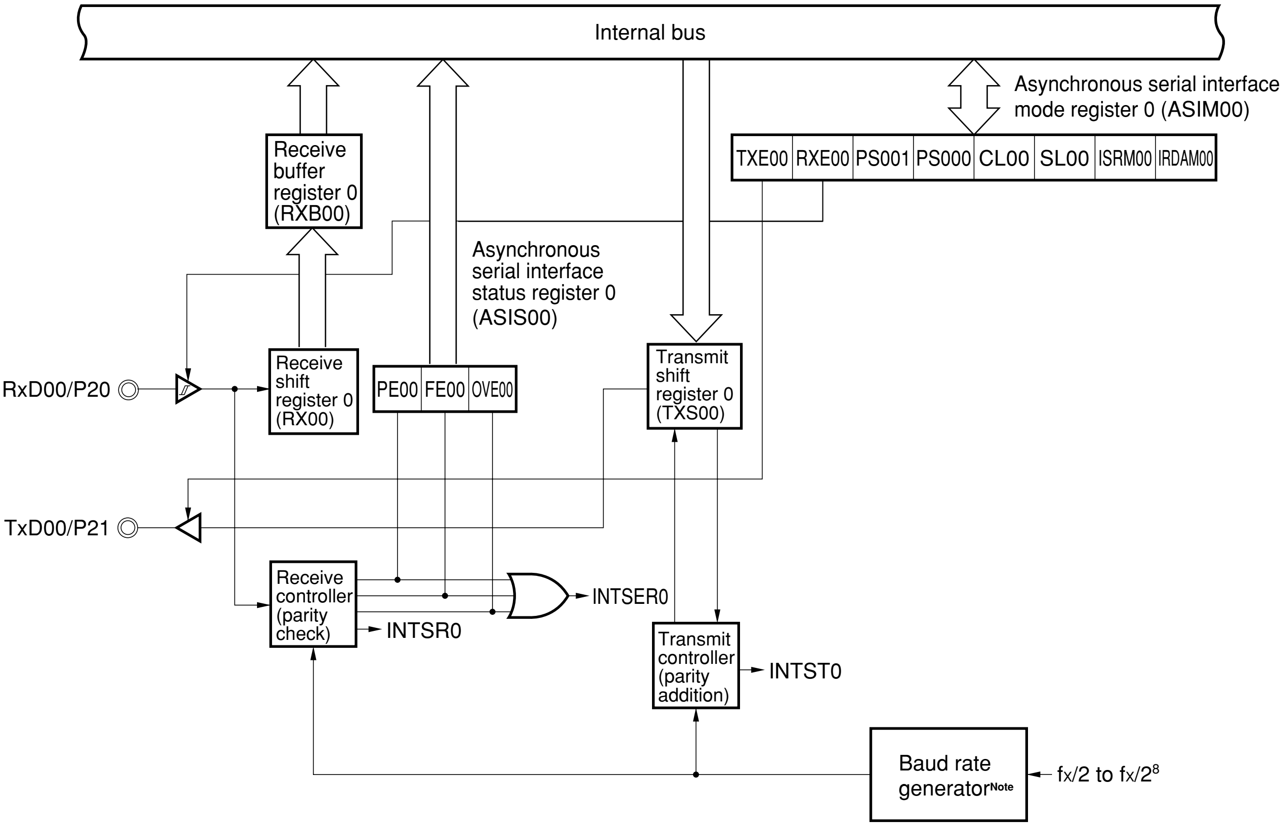
Table 12-1. Configuration of Serial Interfaces

Item	Configuration
Registers	Transmit shift register n (TXS0n) Receive shift register n (RX0n) Receive buffer register n (RXB0n)
Control registers	Asynchronous serial interface mode register n (ASIM0n) Asynchronous serial interface status register n (ASIS0n) Baud rate generator control register n (BRGC0n) Port mode register 2 (PM2)[Note]

Note Refer to **Figure 4-4 Block Diagram of P20 to P26**.

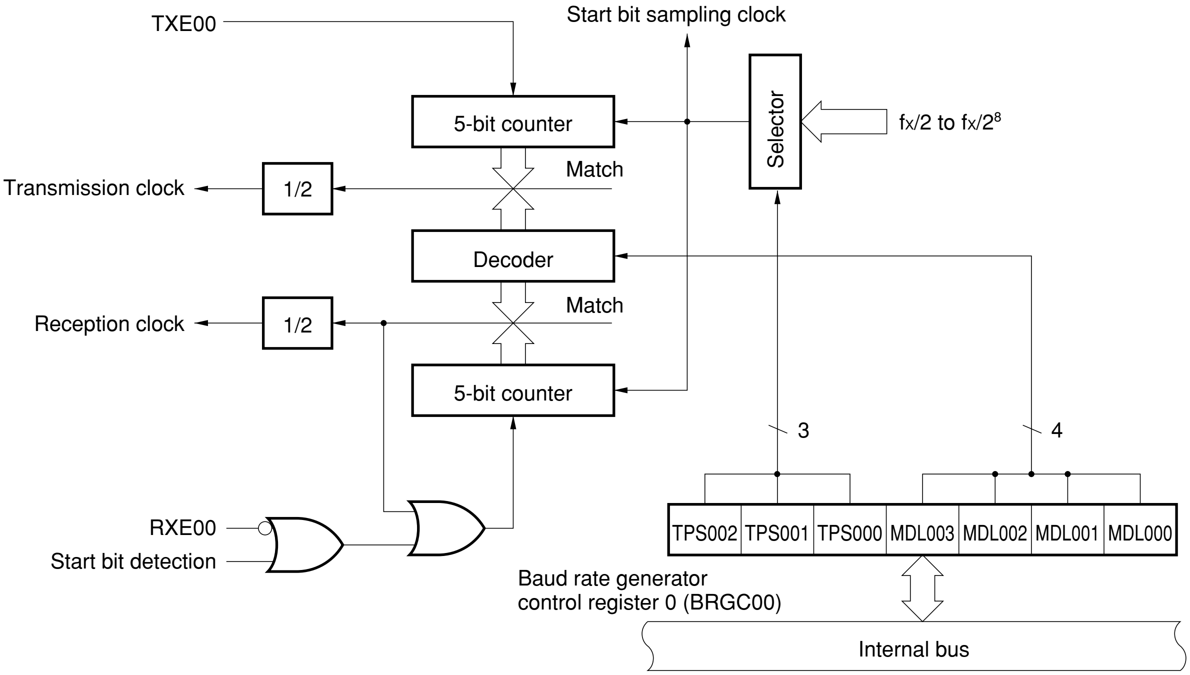
Remark n = 0, 1

Figure 12-1. Block Diagram of Serial Interface UART00


Internal bus
Asynchronous serial interface mode register 0 (ASIM00)
TXE00
RXE00
PS001
PS000
CL00
SL00
ISRM00
IRDAM00
Receive buffer register 0 (RXB00)
Asynchronous serial interface status register 0 (ASIS00)
RxD00/P20
Receive shift register 0 (RX00)
PE00
FE00
OVE00
Transmit shift register 0 (TXS00)
TxD00/P21
Receive controller (parity check)
INTSER0
INTSR0
Transmit controller (parity addition)
INTST0
Baud rate generatorNote
fx/2 to fx/2^8

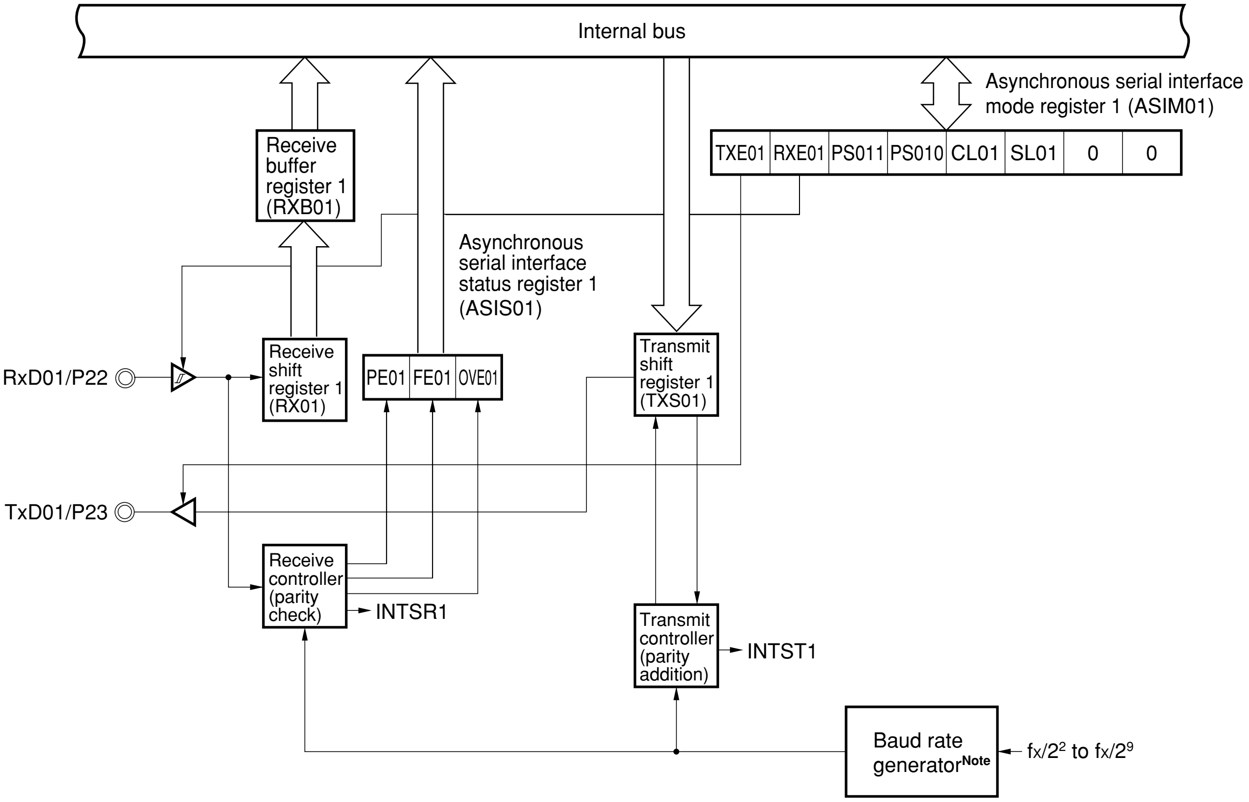
Note Refer to **Figure 12-2** for the baud rate generator configuration.

Figure 12-2. Block Diagram of UART00 Baud Rate Generator



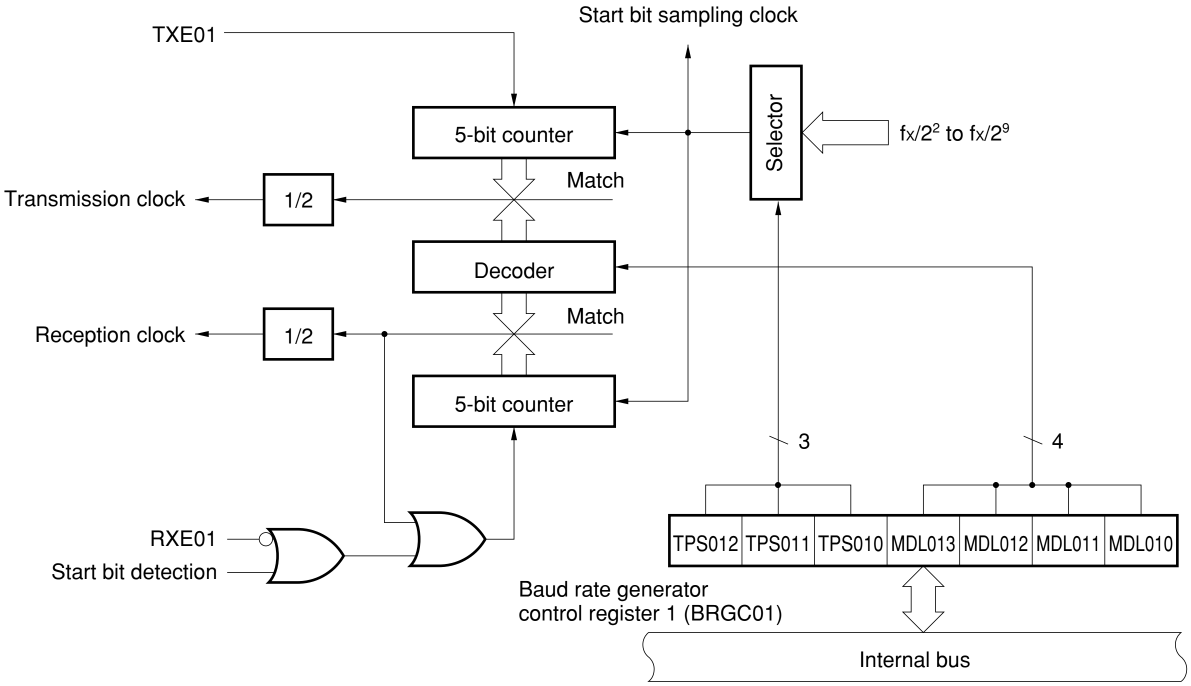
Remark TXE00: Bit 7 of asynchronous serial interface mode register 0 (ASIM00)
RXE00: Bit 6 of asynchronous serial interface mode register 0 (ASIM00)

Figure 12-3. Block Diagram of Serial Interface UART01


Internal bus
Asynchronous serial interface mode register 1 (ASIM01)
TXE01
RXE01
PS011
PS010
CL01
SL01
0
0
Receive buffer register 1 (RXB01)
Asynchronous serial interface status register 1 (ASIS01)
RxD01/P22
Receive shift register 1 (RX01)
PE01
FE01
OVE01
Transmit shift register 1 (TXS01)
TxD01/P23
Receive controller (parity check)
INTSR1
Transmit controller (parity addition)
INTST1
Baud rate generatorNote
fx/2^2 to fx/2^9

Note Refer to **Figure 12-4** for the baud rate generator configuration.

Figure 12-4. Block Diagram of UART01 Baud Rate Generator



Remark TXE01: Bit 7 of asynchronous serial interface mode register 1 (ASIM01)
RXE01: Bit 6 of asynchronous serial interface mode register 1 (ASIM01)

(1) Transmit shift register n (TXS0n)

This register is used to set the transmit data. The data written in TXS0n is transmitted as serial data.

If the data length is specified as 7 bits, bits 0 to 6 of the data written in TXS0n are transferred as transmit data.

Writing data to TXS0n starts the transmit operation.

TXS0n is written to with an 8-bit memory manipulation instruction. It cannot be read.

$\overline{\text{RESET}}$ input sets TXS0n to FFH.

Caution TXS0n must not be written to during a transmit operation. TXS0n and receive buffer register n (RXB0n) are allocated to the same address, and when a read is performed, the value of RXB0n is read.

(2) Receive shift register n (RX0n)

This register is used to convert serial data input to the RxD0n pin to parallel data. When one byte of data is received, the receive data is transferred to receive buffer register n (RXB0n).

RX0n cannot be directly manipulated by a program.

(3) Receive buffer register n (RXB0n)

This register holds receive data. Each time one byte of data is received, new receive data is transferred from receive shift register n (RX0n).

If the data length is specified as 7 bits, the receive data is transferred to bits 0 to 6 of RXB0n, and the MSB of RXB0n is always set to 0.

RXB0n is read with an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input sets RXB0n to FFH.

Caution RXB0n and transmit shift register n (TXS0n) are allocated to the same address, and when a write is performed, the value is written to TXS0n.

(4) Transmission control circuit

This circuit performs transmit operation control such as the addition of a start bit, parity bit and stop bit to data written in transmit shift register n (TXS0n) in accordance with the contents set in asynchronous serial interface mode register n (ASIM0n).

(5) Reception control circuit

This circuit controls receive operations in accordance with the contents set in asynchronous serial interface mode register n (ASIM0n). It performs error checks for parity errors, etc., during a receive operation, and if an error is detected, sets a value in asynchronous serial interface status register n (ASIS0n) in accordance with the error contents.

Remark n = 0, 1

12.3 Registers Controlling Serial Interfaces

The following six registers control the serial interfaces UART00 and UART01.

- Asynchronous serial interface mode registers 0, 1 (ASIM00, ASIM01)
- Asynchronous serial interface status registers 0, 1 (ASIS00, ASIS01)
- Baud rate generator control registers 0, 1 (BRGC00, BRGC01)

(1) Asynchronous serial interface mode registers 0, 1 (ASIM00, ASIM01)

ASIM00 and ASIM01 are 8-bit registers that control the serial transfer operation of the asynchronous serial interface.

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

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

Caution Set the port mode register (PM2x) in the UART mode, as shown below. Set each output latch to 0.

- **For reception**
  Set P20 (RxD00) and P22 (RxD01) to input mode (PM20 = 1, PM22 = 1)
- **For transmission**
  Set P21 (TxD00) and P23 (TxD01) to output mode (PM21 = 0, PM23 = 0)
- **For transmission and reception**
  Set P20 and P22 to input mode and P21 and P23 to output mode.

Figure 12-5. Format of Asynchronous Serial Interface Mode Register 0

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	ISRM00	IRDAM00	FFA0H	00H	R/W

TXE00	RXE00	Operation mode	Function of RxD00/P20 pin	Function of TxD00/P21 pin
0	0	Operation stopped	Port function (P20)	Port function (P21)
0	1	UART mode (reception only)	Serial function (RxD00)	
1	0	UART mode (transmission only)	Port function (P20)	Serial function (TxD00)
1	1	UART mode (transmission/reception)	Serial function (RxD00)	

PS001	PS000	Parity bit specification
0	0	No parity
0	1	Transmission = Always 0 parity addition Reception = Parity not checked (parity error not generated)
1	0	Odd parity
1	1	Even parity

CL00	Character length specification
0	7 bits
1	8 bits

SL00	Transmit data stop bit length specification
0	1 bit
1	2 bits

ISRM00	Reception end interrupt control on occurrence of error
0	Reception end interrupt request generated on occurrence of error.
1	Reception end interrupt request not generated on occurrence of error.

IRDAM00	Infrared data transfer mode operation specification[Note 1]
0	UART (transmission/reception) mode
1	Infrared data transfer (transmission/reception) mode[Note 2]

Notes 1. The UART or infrared data transfer mode is specified by TXE00 and RXE00.

2. When using the infrared data transfer mode, be sure to clear baud rate generator control register 0 (BRGC00) to 00H.

Caution Before changing the operation mode, be sure to stop the serial transmission/reception.

Figure 12-6. Format of Asynchronous Serial Interface Mode Register 1

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After reset	R/W
ASIM01	TXE01	RXE01	PS011	PS010	CL01	SL01	0	0	FFA8H	00H	R/W

TXE01	RXE01	Operation mode	Function of RxD01/P22 pin	Function of TxD01/P23 pin
0	0	Operation stopped	Port function (P22)	Port function (P23)
0	1	UART mode (reception only)	Serial function (RxD01)	
1	0	UART mode (transmission only)	Port function (P22)	Serial function (TxD01)
1	1	UART mode (transmission/reception)	Serial function (RxD01)	

PS011	PS010	Parity bit specification
0	0	No parity
0	1	Transmission = Always 0 parity addition Reception = Parity not checked (parity error not generated)
1	0	Odd parity
1	1	Even parity

CL01	Character length specification
0	7 bits
1	8 bits

SL01	Transmit data stop bit length specification
0	1 bit
1	2 bits

Caution   Before changing the operation mode, be sure to stop the serial transmission/reception.

(2) Asynchronous serial interface status registers 0, 1 (ASIS00, ASIS01)

ASIS00 and ASIS01 are the registers that indicate the error contents when a receive error occurs.

These registers can be read with an 8-bit memory manipulation instruction.

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

Figure 12-7. Format of Asynchronous Serial Interface Status Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS00	0	0	0	0	0	PE00	FE00	OVE00	FFA1H	00H	R

PE00	Parity error flag
0	Parity error does not occur.
1	Parity error occurs (transmit data parity specification and receive data parity do not match).

FE00	Framing error flag
0	Framing error does not occur.
1	Framing error occurs$^{\text{Note 1}}$ (stop bit not detected).

OVE00	Overrun error flag
0	Overrun error does not occur.
1	Overrun error occurs$^{\text{Note 2}}$ (next receive completed before data is read from receive buffer register).

Notes 1. Even if the stop bit length is set to 2 bits using bit 2 (SL00) of asynchronous serial interface mode register 0 (ASIM00), only 1 stop bit is detected during reception.

2. If an overrun error occurs, be sure to read receive buffer register 0 (RXB00). Until RXB00 is read, an overrun error persistently occurs each time data is received.

Figure 12-8. Format of Asynchronous Serial Interface Status Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS01	0	0	0	0	0	PE01	FE01	OVE01	FFA9H	00H	R

PE01	Parity error flag
0	Parity error does not occur.
1	Parity error occurs (transmit data parity specification and receive data parity do not match).

FE01	Framing error flag
0	Framing error does not occur.
1	Framing error occurs[Note 1] (stop bit not detected).

OVE01	Overrun error flag
0	Overrun error does not occur.
1	Overrun error occurs[Note 2] (Next receive completed before data is read from receive buffer register).

Notes 1. Even if the stop bit length is set to 2 bits using bit 2 (SL01) of asynchronous serial interface mode register 1 (ASIM01), only 1 stop bit is detected during reception.

2. If an overrun error occurs, be sure to read receive buffer register 1 (RXB01). Until RXB01 is read, an overrun error persistently occurs each time data is received.

(3) Baud rate generator control registers 0, 1 (BRGC00, BRGC01)

BRGC00 and BRGC01 are the registers that set the serial clock of the asynchronous serial interface.

These registers are set by an 8-bit memory manipulation instruction.

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

★

Figure 12-9. Format of Baud Rate Generator Control Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC00	0	TPS002	TPS001	TPS000	MDL003	MDL002	MDL001	MDL000	FFA2H	00H	R/W

TPS002	TPS001	TPS000	5-bit counter source clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	$f_X/2$	6 MHz	4.19 MHz
0	0	1	$f_X/2^2$	3 MHz	2.1 MHz
0	1	0	$f_X/2^3$	1.5 MHz	1.05 MHz
0	1	1	$f_X/2^4$	750 kHz	524 kHz
1	0	0	$f_X/2^5$	375 kHz	262 kHz
1	0	1	$f_X/2^6$	187 kHz	131 kHz
1	1	0	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	1	$f_X/2^8$	46.8 kHz	32.7 kHz

MDL003	MDL002	MDL001	MDL000	Baud rate generator input clock selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	–

Note Expanded-specification products only.

Cautions 1. If a write to BRGC00 is performed during communication, the output of the baud rate generator may be disrupted, preventing normal communication from continuing. BRGC00 should therefore not be written to during communication.

2. Set BRGC00 to 00H in the infrared data transfer mode.

Remarks 1. f_X: System clock oscillation frequency

2. f_{SCK}: 5-bit counter source clock

3. k: Value set in bits MDL000 to MDL003 ($0 \leq k \leq 14$)

★

Figure 12-10. Format of Baud Rate Generator Control Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC01	0	TPS012	TPS011	TPS010	MDL013	MDL012	MDL011	MDL010	FFAAH	00H	R/W

TPS012	TPS011	TPS010	5-bit counter source clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	$f_X/2^2$	3 MHz	2 MHz
0	0	1	$f_X/2^3$	1.5 MHz	1 MHz
0	1	0	$f_X/2^4$	750 kHz	524 kHz
0	1	1	$f_X/2^5$	375 kHz	262 kHz
1	0	0	$f_X/2^6$	187 kHz	131 kHz
1	0	1	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	0	$f_X/2^8$	46.8 kHz	32.7 kHz
1	1	1	$f_X/2^9$	23.4 kHz	16.4 kHz

MDL013	MDL012	MDL011	MDL010	Baud rate generator input clock selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	–

Note Expanded-specification products only.

Caution If a write to BRGC01 is performed during communication, the output of the baud rate generator may be disrupted, preventing normal communication from continuing. BRGC01 should therefore not be written to during communication.

Remarks 1. f_X: System clock oscillation frequency

2. f_{SCK}: 5-bit counter source clock

3. k: Value set in bits MDL010 to MDL013 ($0 \leq k \leq 14$)

12.4 Operation of Serial Interfaces

The following three operating modes are available for the serial interfaces UART00 and UART01.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- Infrared data transfer mode (UART00 only)

12.4.1 Operation stop mode

Serial transfer is not executed in this mode. Consequently, the power consumption can be reduced.
In the operation stop mode, the pins can be used as ordinary port pins.

(1) Register setting

The operation stop mode is set using asynchronous serial interface mode registers 0 and 1 (ASIM00 and ASIM01).
ASIM00 and ASIM01 are set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears these registers to 00H.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After reset	R/W
ASIM0n	TXE0n	RXE0n	PS0n1	PS0n0	CL0n	SL0n	ISRM0n	IRDAM0n	FFA0H, FFA4H	00H	R/W

TXE0n	RXE0n	Operation mode	Function of RxD00/P20, RxD01/P22 pins	Function of TxD00/P21, TxD01/P23 pins
0	0	Operation stopped	Port function	Port function
0	1	UART mode (reception only)	Serial function	
1	0	UART mode (transmission only)	Port function	Serial function
1	1	UART mode (transmission/reception)	Serial function	

Caution Before changing the operation mode, be sure to stop the serial transmission/reception.

Remark n = 0, 1

12.4.2 Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following a start bit, and full-duplex operation is possible.

A baud rate generator is incorporated, enabling communication to be performed at any of a wide range of baud rates.

Programming baud rate generator control registers 0 and 1 (BRGC00 and BRGC01) allows a baud rate selection ★ of 600 bps to 115.2 kbps (@ f_X = 8.38 MHz operation) or 1200 bps to 153.6 kbps (@ f_X = 12 MHz operation) for UART00 and 300 bps to 38.4 kbps (@ f_X = 8.38 MHz operation) or 600 bps to 76.8 kbps (@ f_X = 12 MHz operation) for UART01.

(1) Register setting

The UART mode is set using asynchronous serial interface mode registers 0 and 1 (ASIM00 and ASIM01), asynchronous serial interface status registers 0 and 1 (ASIS00 and ASIS01), and baud rate generator control registers 0 and 1 (BRGC00 and BRGC01).

(a) Asynchronous serial interface mode registers 0, 1 (ASIM00, ASIM01)

ASIM00 and ASIM01 are set by a 1-bit or 8-bit memory manipulation instruction.

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

Caution Set the port mode register (PM2x) in the UART mode, as shown below. Set each output latch to 0.

- **For reception**
  Set P20 (RxD00) and P22 (RxD01) to input mode (PM20 = 1, PM22 = 1)
- **For transmission**
  Set P21 (TxD00) and P23 (TxD01) to output mode (PM21 = 0, PM23 = 0)
- **For transmission and reception**
  Set P20 and P22 to input mode and P21 and P23 to output mode.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	ISRM00	IRDAM00	FFA0H	00H	R/W

TXE00	RXE00	Operation mode	Function of RxD00/P20 pin	Function of TxD00/P21 pin
0	0	Operation stopped	Port function (P20)	Port function (P21)
0	1	UART mode (reception only)	Serial function (RxD00)	
1	0	UART mode (transmission only)	Port function (P20)	Serial function (TxD00)
1	1	UART mode (transmission/reception)	Serial function (RxD00)	

PS001	PS000	Parity bit specification
0	0	No parity
0	1	Transmission = Always 0 parity addition Reception = Parity not checked (parity error not generated)
1	0	Odd parity
1	1	Even parity

CL00	Character length specification
0	7 bits
1	8 bits

SL00	Transmit data stop bit length specification
0	1 bit
1	2 bits

ISRM00	Reception end interrupt control on occurrence of error
0	Reception end interrupt request generated on occurrence of error.
1	Reception end interrupt request not generated on occurrence of error.

IRDAM00	Infrared data transfer mode operation specification[Note 1]
0	UART (transmission/reception) mode
1	Infrared data transfer (transmission/reception) mode[Note 2]

Notes 1. The UART or infrared data transfer mode is specified by TXE00 and RXE00.

2. When using the infrared data transfer mode, be sure to clear baud rate generator control register 0 (BRGC00) to 00H.

Caution Before changing the operation mode, be sure to stop the serial transmission/reception.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After reset	R/W
ASIM01	TXE01	RXE01	PS011	PS010	CL01	SL01	0	0	FFA8H	00H	R/W

TXE01	RXE01	Operation mode	Function of RxD01/P22 pin	Function of TxD01/P23 pin
0	0	Operation stopped	Port function (P22)	Port function (P23)
0	1	UART mode (reception only)	Serial function (RxD01)	
1	0	UART mode (transmission only)	Port function (P22)	Serial function (TxD01)
1	1	UART mode (transmission/reception)	Serial function (RxD01)	

PS011	PS010	Parity bit specification
0	0	No parity
0	1	Transmission = Always 0 parity addition Reception = Parity not checked (parity error not generated)
1	0	Odd parity
1	1	Even parity

CL01	Character length specification
0	7 bits
1	8 bits

SL01	Transmit data stop bit length specification
0	1 bit
1	2 bits

Caution Before changing the operation mode, be sure to stop the serial transmission/reception.

(b) Asynchronous serial interface status registers 0, 1 (ASIS00, ASIS01)

ASIS00 and ASIS01 can be read with an 8-bit memory manipulation instruction.

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

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS0n	0	0	0	0	0	PE0n	FE0n	OVE0n	FFA1H, FFA9H	00H	R

PE0n	Parity error flag
0	Parity error does not occur.
1	Parity error occurs (transmit data parity specification and receive data parity do not match).

FE0n	Framing error flag
0	Framing error does not occur.
1	Framing error occurs[Note 1] (stop bit not detected).

OVE0n	Overrun error flag
0	Overrun error does not occur.
1	Overrun error occurs[Note 2] (Next receive completed before data is read from receive buffer register).

Notes 1. Even if the stop bit length is set to 2 bits using bit 2 (SL0n) of asynchronous serial interface mode register n (ASIM0n), only 1 stop bit is detected during reception.

2. If an overrun error occurs, be sure to read receive buffer register n (RXB0n). Until RXB0n is read, an overrun error persistently occurs each time data is received.

Remark n = 0, 1

★ **(c) Baud rate generator control registers 0, 1 (BRGC00, BRGC01)**

BRGC00 and BRGC01 are set by an 8-bit memory manipulation instruction.

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

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC00	0	TPS002	TPS001	TPS000	MDL003	MDL002	MDL001	MDL000	FFA2H	00H	R/W

TPS002	TPS001	TPS000	5-bit counter source clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	$f_X/2$	6 MHz	4.19 MHz
0	0	1	$f_X/2^2$	3 MHz	2.1 MHz
0	1	0	$f_X/2^3$	1.5 MHz	1.05 MHz
0	1	1	$f_X/2^4$	750 kHz	524 kHz
1	0	0	$f_X/2^5$	375 kHz	262 kHz
1	0	1	$f_X/2^6$	187 kHz	131 kHz
1	1	0	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	1	$f_X/2^8$	46.8 kHz	32.7 kHz

MDL003	MDL002	MDL001	MDL000	Baud rate generator input clock selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	–

Note Expanded-specification products only.

Cautions 1. If a write to BRGC00 is performed during communication, the output of the baud rate generator may be disrupted, preventing normal communication from continuing. BRGC00 should therefore not be written to during communication.

2. Set BRGC00 to 00H in the infrared data transfer mode.

Remarks 1. f_X: System clock oscillation frequency

2. f_{SCK}: 5-bit counter source clock

3. k: Value set in bits MDL000 to MDL003 ($0 \le k \le 14$)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC01	0	TPS012	TPS011	TPS010	MDL013	MDL012	MDL011	MDL010	FFAAH	00H	R/W

TPS012	TPS011	TPS010	5-bit counter source clock selection		
				At f_X = 12 MHz[Note]	At f_X = 8.38 MHz
0	0	0	$f_X/2^2$	3 MHz	2 MHz
0	0	1	$f_X/2^3$	1.5 MHz	1 MHz
0	1	0	$f_X/2^4$	750 kHz	524 kHz
0	1	1	$f_X/2^5$	375 kHz	262 kHz
1	0	0	$f_X/2^6$	187 kHz	131 kHz
1	0	1	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	0	$f_X/2^8$	46.8 kHz	32.7 kHz
1	1	1	$f_X/2^9$	23.4 kHz	16.4 kHz

MDL013	MDL012	MDL011	MDL010	Baud rate generator input clock selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	–

Note Expanded-specification products only.

Caution If a write to BRGC01 is performed during communication, the output of the baud rate generator may be disrupted, preventing normal communication from continuing. BRGC01 should therefore not be written to during communication.

Remarks 1. f_X: System clock oscillation frequency

2. f_{SCK}: 5-bit counter source clock

3. k: Value set in bits MDL010 to MDL013 ($0 \leq k \leq 14$)

The transmit/receive clock for the baud rate to be generated is obtained by dividing the system clock.

- **Generating transmit/receive clock for baud rate from system clock**

  The transmit/receive clock is generated by dividing the system clock. The baud rate generated from the system clock can be calculated from the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{m+1}(k+16)} [\text{Hz}]$$

fX: System clock oscillation frequency
m: Value set by TPS0n0 to TPS0n2 (n = 0, 1)
(1 ≤ m ≤ 8 for UART00, 2 ≤ m ≤ 9 for UART01)
k: Value set by MDL0n0 to MDL0n3 (n = 0, 1) (0 ≤ k ≤ 14)

Tables 12-2 and 12-3 show the relationship between the source clock of the 5-bit counter and the value of m. Table 12-4 shows the relationship between the system clock and baud rate.

★ **Table 12-2. Relationship Between Source Clock of 5-Bit Counter and Value of m (with UART00)**

TPS002	TPS001	TPS000	5-Bit Counter Source Clock Selection			m
				At fX = 12 MHz[Note]	At fX = 8.38 MHz	
0	0	0	fX/2	6 MHz	4.19 MHz	1
0	0	1	$f_X/2^2$	3 MHz	2.1 MHz	2
0	1	0	$f_X/2^3$	1.5 MHz	1.05 MHz	3
0	1	1	$f_X/2^4$	750 kHz	524 kHz	4
1	0	0	$f_X/2^5$	375 kHz	262 kHz	5
1	0	1	$f_X/2^6$	187 kHz	131 kHz	6
1	1	0	$f_X/2^7$	93.7 kHz	65.5 kHz	7
1	1	1	$f_X/2^8$	46.8 kHz	32.7 kHz	8

Note Expanded-specification products only.

★ **Table 12-3. Relationship Between Source Clock of 5-Bit Counter and Value of m (with UART01)**

TPS012	TPS011	TPS010	5-Bit Counter Source Clock Selection			m
				At fX = 12 MHz[Note]	At fX = 8.38 MHz	
0	0	0	$f_X/2^2$	3 MHz	2 MHz	2
0	0	1	$f_X/2^3$	1.5 MHz	1 MHz	3
0	1	0	$f_X/2^4$	750 kHz	524 kHz	4
0	1	1	$f_X/2^5$	375 kHz	262 kHz	5
1	0	0	$f_X/2^6$	187 kHz	131 kHz	6
1	0	1	$f_X/2^7$	93.7 kHz	65.5 kHz	7
1	1	0	$f_X/2^8$	46.8 kHz	32.7 kHz	8
1	1	1	$f_X/2^9$	23.4 kHz	16.4 kHz	9

Note Expanded-specification products only.

★ Table 12-4. Relationship Between System Clock and Baud Rate

System Clock f_X (MHz)	12.000[Note]			10.000[Note]			8.386			8.000			7.3728			5.000			4.1943		
Baud Rate (bps)	BRGC 00	BRGC 01	Error (%)	BRGC 00	BRGC 01	Error (%)	BRGC 00	BRGC 01	Error (%)	BRGC 00	BRGC 01	Error (%)	BRGC 00	BRGC 01	Error (%)	BRGC 00	BRGC 01	Error (%)	BRGC 00	BRGC 01	Error (%)
150	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	7BH	1.14
300	–	–	–	–	–	–	–	7BH	1.10	–	7AH	0.16	–	78H	0	–	70H	1.73	7BH	6BH	1.14
600	–	74H	–2.34	–	70H	1.73	7BH	6BH	1.10	7AH	6AH	0.16	78H	68H	0	70H	60H	1.73	6BH	5BH	1.14
1,200	79H	64H	–2.34	70H	60H	1.73	6BH	5BH	1.10	6AH	5AH	0.16	68H	58H	0	60H	50H	1.73	5BH	4BH	1.14
2,400	69H	54H	–2.34	60H	50H	1.73	5BH	4BH	1.10	5AH	4AH	0.16	58H	48H	0	50H	40H	1.73	4BH	3BH	1.14
4,800	59H	44H	–2.34	50H	40H	1.73	4BH	3BH	1.10	4AH	3AH	0.16	48H	38H	0	40H	30H	1.73	3BH	2BH	1.14
9,600	49H	34H	–2.34	40H	30H	1.73	3BH	2BH	1.10	3AH	2AH	0.16	38H	28H	0	30H	20H	1.73	2BH	1BH	1.14
19,200	39H	24H	–2.34	30H	20H	1.73	2BH	1BH	1.10	2AH	1AH	0.16	28H	18H	0	20H	10H	1.73	1BH	0BH	1.14
31,250	2DH	18H	0	24H	14H	0	21H	11H	–1.34	20H	10H	0	1BH	0BH	1.69	14H	04H	0	11H	01H	–1.31
38,400	29H	14H	–2.34	20H	10H	1.73	1BH	0BH	1.10	1AH	0AH	0.16	18H	08H	0	10H	00H	1.73	0BH	–	1.14
76,800	19H	04H	–2.34	10H	00H	1.73	0BH	–	1.10	0AH	–	0.16	08H	–	0	00H	–	1.73	–	–	–
115,200	0FH	–	0.16	06H	–	–1.36	02H	–	1.10	01H	–	2.12	00H	–	0	–	–	–	–	–	–
153,600	04H	–	–2.34	00H	–	1.73	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

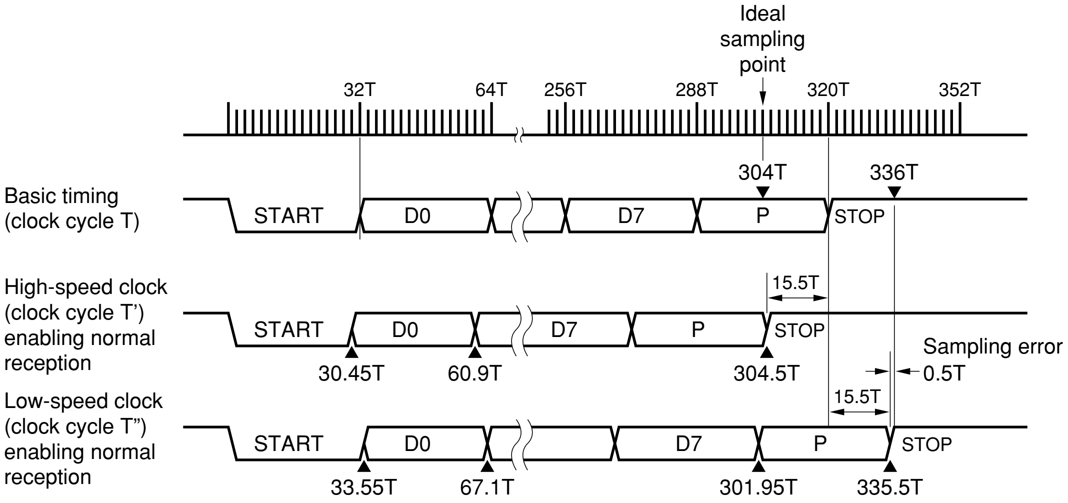
Note Expanded-specification products only.

- **Range of baud rate tolerance**

  The range of baud rate tolerance depends on the number of bits in one frame and division ratio of the counter [1/(16 + k)].

Figure 12-11 shows an example of baud rate tolerance.

Figure 12-11. Baud Rate Tolerance Including Sampling Error (When k = 0)



Remark T: Source clock cycle of 5-bit counter

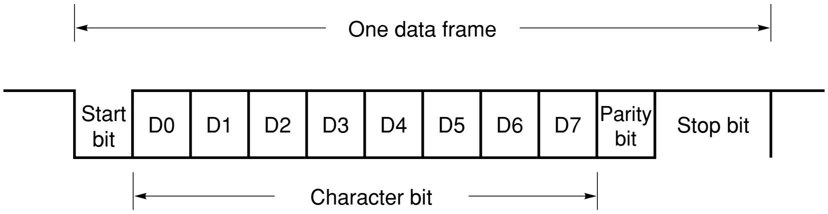
Baud rate tolerance (when k = 0) = ±15.5/320 × 100 = 4.8438 (%)

(2) Communication operation

(a) Data format

Figure 12-12 shows transmit/receive data format.

Figure 12-12. Asynchronous Serial Interface Transmit/Receive Data Format



One data frame consists of following bits:

- Start bits .... 1 bit
- Character bits .... 7 bits/8 bits
- Parity bits .... Even parity/odd parity/0 parity/no parity
- Stop bits .... 1 bit/2 bits

The character bit length, parity bit and stop bit length for each data frame is specified by asynchronous serial interface mode register n (ASIM0n).

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always 0.

The serial transfer rate is selected by means of baud rate generator control register n (BRGC0n).

If a serial data receive error is generated, the receive error contents can be determined by reading the status of asynchronous serial interface status register n (ASIS0n).

Remark n = 0, 1

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **Transmission**

  The number of bits with a value of "1", including the parity bit, in the transmit data is controlled to be even. The value of the parity bit is as follows:

  Number of bits with a value of "1" in transmit data is odd: 1
  Number of bits with a value of "1" in transmit data is even: 0

- **Reception**

  The number of bits with a value of "1", including the parity bit, in the receive data is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- **Transmission**

  Conversely to the situation with even parity, the number of bits with a value of "1", including the parity bit, in the transmit data is controlled to be odd. The value of the parity bit is as follows:

  Number of bits with a value of "1" in transmit data is odd: 0
  Number of bits with a value of "1" in transmit data is even: 1

- **Reception**

  The number of bits with a value of "1", including the parity bit, in the receive data is counted. If it is even, a parity error occurs.

(iii) 0 Parity

When transmitting, the parity bit is set to 0 irrespective of the transmit data.
At reception, a parity bit check is not performed. Therefore, a parity error is not generated, irrespective of whether the parity bit is set to 0 or 1.

(iv) No parity

A parity bit is not added to the transmit data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error is not generated.

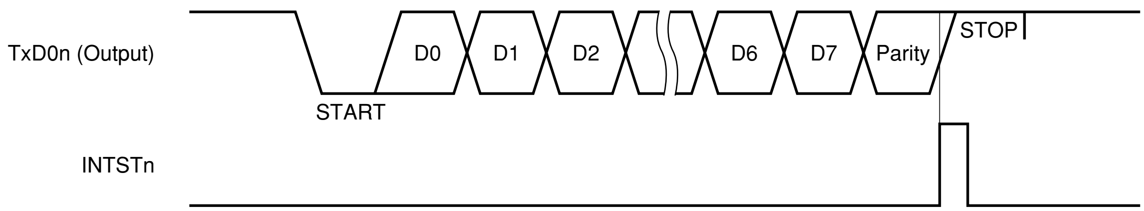
(c) Transmission

A transmit operation is enabled by setting the TXE0n bit of asynchronous serial interface mode n (ASIM0n) to 1 and is started by writing transmit data to transmit shift register n (TXS0n). The start bit, parity bit and stop bit(s) are added automatically.

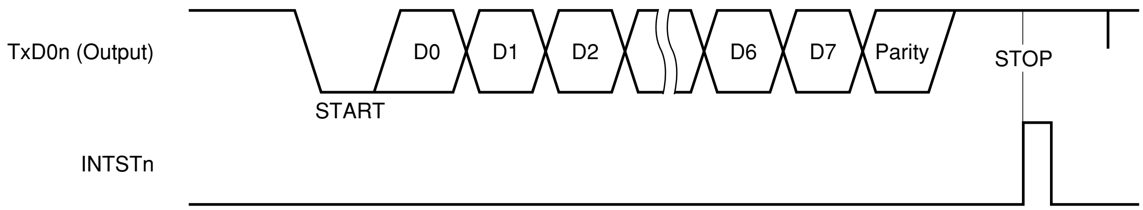
When transmit operation starts, the data in transmit shift register n (TXS0n) is shifted out, and when transmit shift register n (TXS0n) is empty, a transmission completion interrupt request (INTSTn) is generated.

Figure 12-13. Timing of Asynchronous Serial Interface Transmission Completion Interrupt Request Generation

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution **Rewriting of asynchronous serial interface mode register n (ASIM0n) should not be performed during a transmit operation. If rewriting of the ASIM0n register is performed during transmission, subsequent transmit operations may not be possible (the normal state is restored by $\overline{\text{RESET}}$ input).**

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt request (INTSTn) or the interrupt request flag (STIFn) set by INTSTn.

Remark n = 0, 1

(d) Reception

When the RXE0n bit of asynchronous serial interface mode register n (ASIM0n) is set (1), a receive operation is enabled and sampling of the RxD0n pin input is performed.

RxD0n pin input sampling is performed using the serial clock specified by ASIM0n.

When the RxD0n pin input becomes low, the 5-bit counter of the baud rate generator starts counting, and at the time when half the time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD0n pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 5-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to receive buffer register n (RXB0n), and a reception completion interrupt request (INTSRn) is generated.

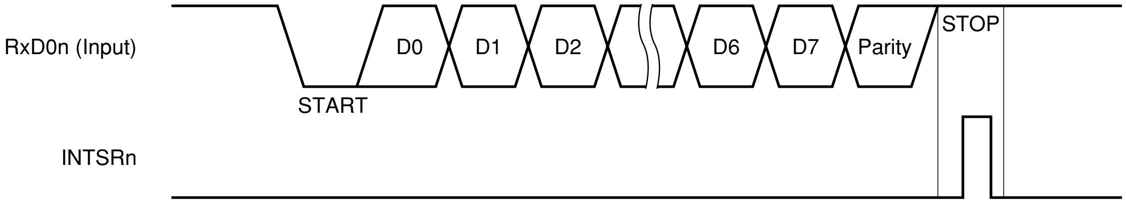
If an error occurs, the receive data in which the error occurred is still transferred to RXB0n.

INTSRn is generated if bit 1 (ISRM0n) of ASIM0n is cleared (0) on occurrence of the error.

If the ISRM0n bit is set (1), INTSRn is not generated.

If the RXE0n bit is reset (0) during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB0n and ASIS0n are not changed, and INTSRn and INTSER0 are not generated.

Figure 12-14. Timing of Asynchronous Serial Interface Reception Completion Interrupt Request Generation



Caution **Receive buffer register n (RXB0n) must be read even if a receive error occurs. If RXB0n is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.**

Remark n = 0, 1

(e) Receive errors

The three types of errors during receive operations are the parity error, framing error and overrun error. With the UART00, setting the data receive result error flag in asynchronous serial interface status register 0 (ASIS00) generates a receive error interrupt request (INTSER0). The receiver error interrupt request is generated before the receive complete interrupt request (INTSR0). With the UART01, the receiver error interrupt request is not generated. Table 12-5 shows the causes of receive errors.

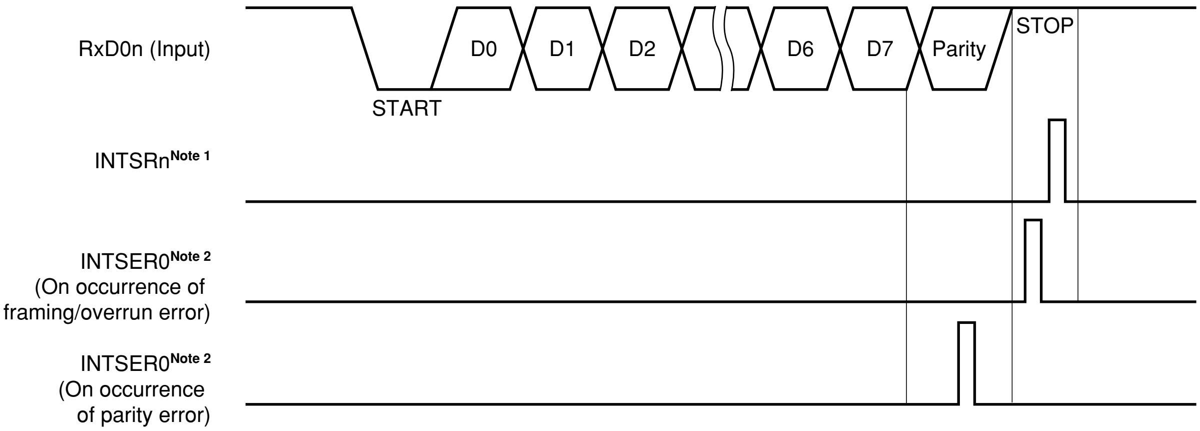
Reading the data in ASIS0n makes it possible to ascertain what error has occurred during reception (see **Figures 12-14** and **12-15**).

The contents of ASIS0n are reset (0) by reading receive buffer register n (RXB0n) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 12-5. Receive Error Causes

Receive Error	Cause	ASIS0n Value
Parity error	Transmission-time parity specification and reception data parity do not match	04H
Framing error	Stop bit not detected	02H
Overrun error	Reception of next data is completed before data is read from receive buffer register	01H

Figure 12-15. Receive Error Timing



Notes 1. INTSRn is not generated if a receive error occurs when the ISRM0n bit is set (1).

2. The receive error interrupt request is not generated with UART01.

Cautions 1. The contents of ASIS0n are reset (to 0) by reading receive buffer register n (RXB0n) or receiving the next data. To ascertain the error contents, ASIS0n must be read before reading RXB0n.

2. Receive buffer register n (RXB0n) must be read even if a receive error has occurred. If RXB0n is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

Remark n = 0, 1

12.4.3 Infrared data transfer mode

Caution The infrared data transfer mode can only be used with UART00.

In the infrared data transfer mode, pulses can be output and received in the following data format.

★ **Remark** The SIR standard is not supported (negotiation at 9,600 bps cannot be performed).

(1) Data format

Figure 12-16 shows the data format of the infrared data transfer mode in comparison with the data format in UART mode.

The IR frame corresponds to the bit string of the UART frame that consists of a start bit, 8 data bits and 1 stop bit.

The length of the electrical pulse transmitted or received in the IR frame is 3/16 of a 1-bit cycle. The pulse 3/16 of a 1-bit cycle rises in the middle of the bit cycle (refer to the figure below).

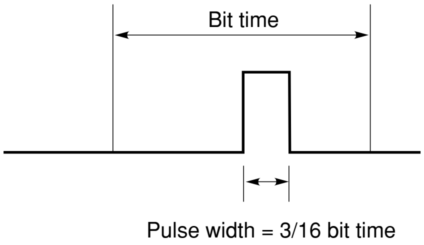
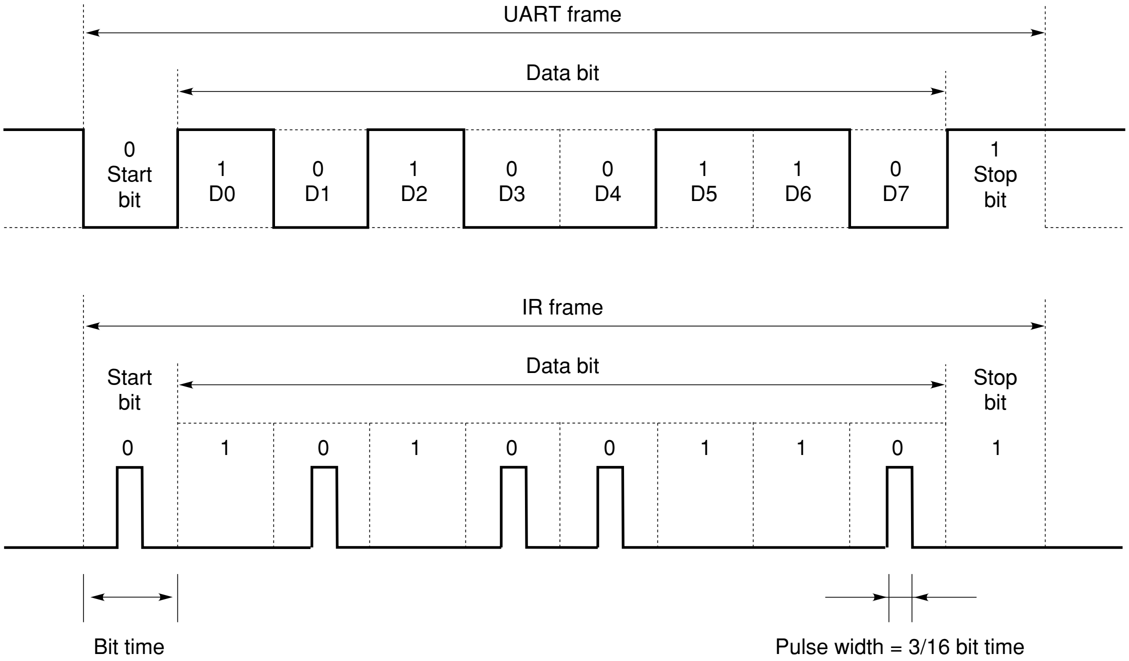


Figure 12-16. Comparison of Data Format in Infrared Data Transfer Mode and UART Mode



(2) Bit rate and pulse width

Table 12-6 shows the values of the bit rate, bit rate tolerance, and pulse width.

Table 12-6. Bit Rate and Pulse Width

Bit Rate (kbits/s)	Bit Rate Tolerance (% of Bit Rate)	Minimum Pulse Width (μs)[Note 2]	Nominal Value of Pulse Width 3/16 (μs)	Maximum Pulse Width (μs)
115.2[Note 1]	±0.87	1.41	1.63	2.71

Notes 1. fx = @ 7.3728 MHz operation

2. Where a digital noise eliminator is used with the microcontroller at a frequency of 1.41 MHz or higher

Caution Set baud rate generator control register 0 (BRGC00) to 00H in the infrared data transfer mode.

Remark fx: System clock oscillation frequency

(3) Baud rate that can be set in infrared data transfer mode

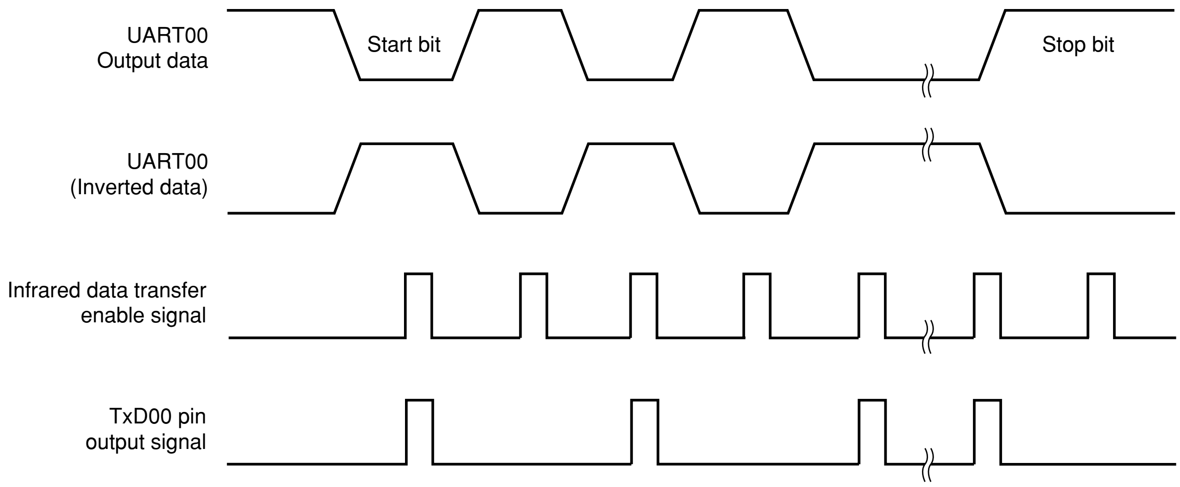
Table 12-7. Baud Rate That Can Be Set in Infrared Data Transfer Mode

System Clock fx (MHz)	Baud Rate (bps)
★ 12.000[Note]	187,500[Note]
8.386	131,031
8.000	125,000
7.3728	115,200
5.000	78,125
4.1943	65,536

Note Expanded-specification products only.

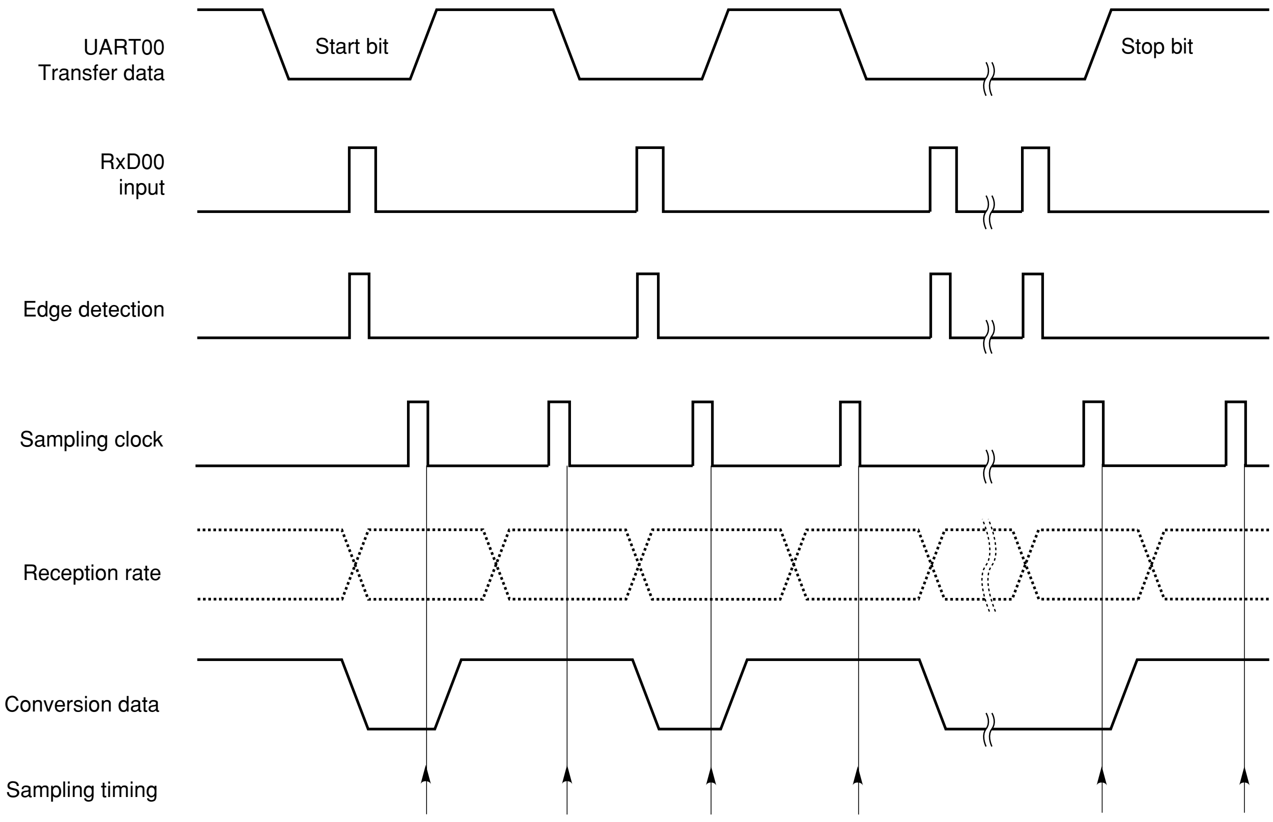
(4) I/O data and internal signal

- **Transmission timing**



- **Reception timing**

Data reception is delayed by half the set baud rate.



CHAPTER 13 SERIAL INTERFACE SIO3

13.1 Function of Serial Interface SIO3

Serial interface SIO3 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **13.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCK}}$), serial output line (SO), and serial input line (SI).

Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit of the serial transferred 8-bit data is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, a display controller, etc.

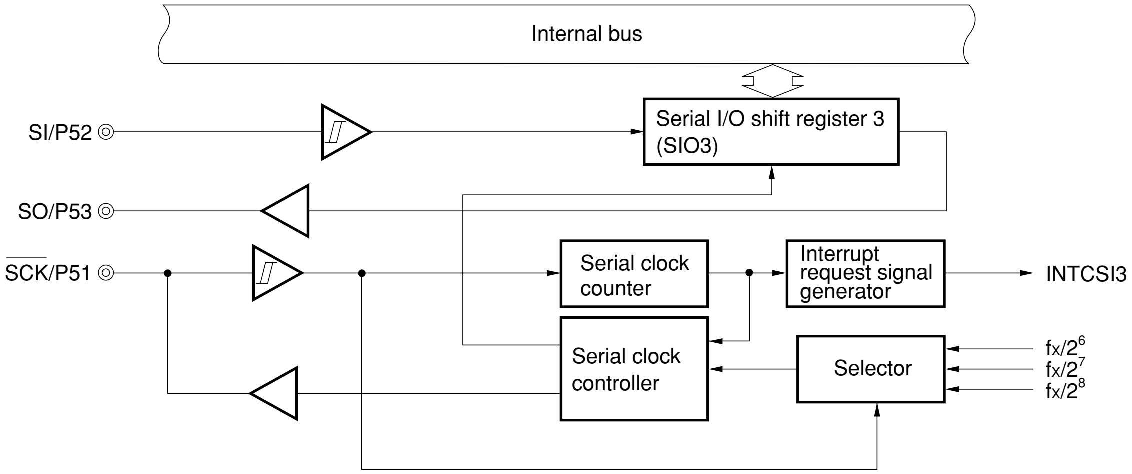
13.2 Configuration of Serial Interface

Serial interface SIO3 includes the following hardware.

Table 13-1. Configuration of Serial Interface 3

Item	Configuration
Register	Serial I/O shift register 3 (SIO3)
Control register	Serial operation mode register 3 (CSIM3)

Figure 13-1. Block Diagram of Serial Interface 3



(1) Serial I/O shift register 3 (SIO3)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) in synchronization with the serial clock.

SIO3 is set by an 8-bit memory manipulation instruction.

When bit 7 (CSIE3) of serial operation mode register 3 (CSIM3) is set to 1, a serial operation can be started by writing data to or reading data from SIO3.

When transmitting, data written to SIO3 is output to the serial output (SO).

When receiving, data is read from the serial input (SI) and written to SIO3.

$\overline{\text{RESET}}$ input makes SIO3 undefined.

Caution Do not access SIO3 during a transfer operation unless the access is triggered by a transfer start (read operations are disabled when MODE = 0 and write operations are disabled when MODE = 1).

13.3 Register Controlling Serial Interface

Serial interface SIO3 is controlled by the following register.

- Serial operation mode register 3 (CSIM3)

(1) Serial operation mode register 3 (CSIM3)

This register is used to enable or disable the SIO3 serial clock, operation modes, and specific operations.
CSIM3 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM3 to 00H.

Caution When using the 3-wire serial I/O mode, set the port mode registers (PM5x) as shown below. Also, set the respective output latches to 0.

- **For serial clock output (master transmit/receive)**
  Set P51 ($\overline{\text{SCK}}$) to the output mode (PM51 = 0).
- **For serial clock input (slave transmit/receive)**
  Set P51 to the input mode (PM51 = 1).
- **For transmit/transmit and receive mode**
  Set P53 (SO) to the output mode (PM53 = 0).
  Set P52 (SI) to the input mode (PM52 = 1) (in transmit/receive mode).
- **For receive mode**
  Set P52 (SI) to the input mode (PM52 = 1).

★ **Figure 13-2. Format of Serial Operation Mode Register 3**

Symbol	⑦	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30	FFB0H	00H	R/W

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation disabled	Cleared	Port function[Note 1]
1	Operation enabled	Count operation enabled	Serial function + port function[Note 2]

MODE	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO output
0	Transmit/transmit and receive mode	Write to SIO3	Normal output
1	Receive-only mode	Read from SIO3	Fixed at low level

SCL31	SCL30	Clock selection		
			At f_X = 12 MHz[Note 3]	At f_X = 8.38 MHz
0	0	External clock input to $\overline{\text{SCK}}$ pin		
0	1	$f_X/2^6$	187 kHz	131 kHz
1	0	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	$f_X/2^8$	46.8 kHz	32.7 kHz

Notes 1. When CSIE3 = 0 (SIO3 operation stopped), the SI, SO, and $\overline{\text{SCK}}$ pins can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled), if only the transmit function is used, the SI pin can be used for a port function, and in the receive-only mode, the SO pin can be used for a port function.

3. Expanded-specification products only

Remark f_X: System clock oscillation frequency

13.4 Operation of Serial Interface

This section explains the two modes of serial interface SIO3.

13.4.1 Operation stop mode

Because serial transfer is not performed during this mode, the power consumption can be reduced.
In addition, pins can be used as normal I/O ports.

(1) Register settings

Operation stop mode is set by serial operation mode register 3 (CSIM3).
CSIM3 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM3 to 00H.

Symbol	⑦	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30	FFB0H	00H	R/W

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation disabled	Cleared	Port function[Note 1]
1	Operation enabled	Count operation enabled	Serial function + port function[Note 2]

Notes 1. When CSIE3 = 0 (SIO3 operation stopped), the SI, SO, and $\overline{\text{SCK}}$ pins can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled), if only the transmit function is used, the SI pin can be used for a port function, and in the receive-only mode, the SO pin can be used for a port function.

13.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{\text{SCK}}$), serial output line (SO), and serial input line (SI).

(1) Register settings

The 3-wire serial I/O mode is set by serial operation mode register 3 (CSIM3).
CSIM3 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM3 to 00H.

Caution **When using the 3-wire serial I/O mode, set the port mode registers (PM5x) as shown below. Also, set the respective output latches to 0.**

- **For serial clock output (master transmit/receive)**
  Set P51 ($\overline{\text{SCK}}$) to the output mode (PM51 = 0).
- **For serial clock input (slave transmit/receive)**
  Set P51 to the input mode (PM51 = 1).
- **For transmit/transmit and receive mode**
  Set P53 (SO) to the output mode (PM53 = 0).
  Set P52 (SI) to the input mode (PM52 = 1) (in transmit/receive mode).
- **For receive mode**
  Set P52 (SI) to the input mode (PM52 = 1).

Symbol	⑦	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30	FFB0H	00H	R/W

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation disabled	Cleared	Port function[Note 1]
1	Operation enabled	Count operation enabled	Serial function + port function[Note 2]

MODE	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO output
0	Transmit/transmit and receive mode	Write to SIO3	Normal output
1	Receive-only mode	Read from SIO3	Fixed at low level

SCL31	SCL30	Clock selection		
			At fx = 12 MHz[Note 3]	At fx = 8.38 MHz
0	0	External clock input to $\overline{\text{SCK}}$ pin		
0	1	$f_X/2^6$	187 kHz	131 kHz
1	0	$f_X/2^7$	93.7 kHz	65.5 kHz
1	1	$f_X/2^8$	46.8 kHz	32.7 kHz

Notes 1. When CSIE3 = 0 (SIO3 operation stopped), the SI, SO, and $\overline{\text{SCK}}$ pins can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled), if only the transmit function is used, the SI pin can be used for a port function, and in the receive-only mode, the SO pin can be used for a port function.

3. Expanded-specification products only

Remark fx: System clock oscillation frequency

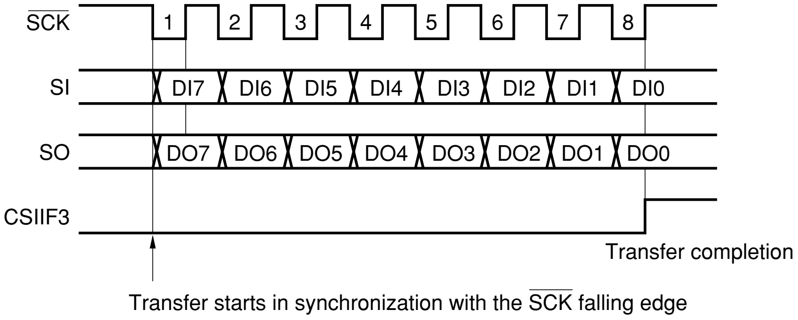
(2) Communication operations

In the 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

Serial I/O shift register 3 (SIO3) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SO latch and is output from the SO pin. Data that is received via the SI pin in synchronization with the rising edge of the serial clock is latched to SIO3.

Completion of an 8-bit transfer automatically stops operation of SIO3 and sets an interrupt request flag (CSIIF3).

Figure 13-3. Timing of 3-Wire Serial I/O Mode



(3) Transfer start

A serial transfer starts when the following conditions have been satisfied and transfer data has been set (or read) to serial I/O shift register 3 (SIO3).

- The SIO3 operation control bit (CSIE3) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or $\overline{SCK}$ is set to high level.
- Transmit/transmit and receive mode
  When CSIE3 = 1 and MODE = 0, transfer starts when writing to SIO3.
- Receive-only mode
  When CSIE3 = 1 and MODE = 1, transfer starts when reading from SIO3.

Caution After data has been written to SIO3, transfer will not start even if the CSIE3 bit value is set to 1.

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets an interrupt request flag (CSIIF3).

CHAPTER 14 INTERRUPT FUNCTIONS

14.1 Types of Interrupt Functions

The following three types of interrupt functions are available.

(1) Non-maskable interrupt

This interrupt is unconditionally acknowledged even in the interrupt disabled status. It is not subject to interrupt priority control and therefore takes precedence over all interrupt requests.

This interrupt generates a standby release signal.

One interrupt request from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts are subject to mask control, and can be divided into two groups according to the setting of the priority specification flag register (PR0L, PR0H, PR1L): one with higher priority and the other with lower priority. Higher-priority interrupts can nest lower-priority interrupts. The priority when two or more interrupt requests with the same priority occur at the same time is predetermined (refer to **Table 14-1**).

This interrupt generates a standby release signal.

Eight external interrupt requests and sixteen internal interrupt requests are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt generated when the BRK instruction is executed and can be acknowledged even in the interrupt disabled status. This interrupt is not subject to interrupt priority control.

14.2 Interrupt Sources and Configuration

A total of 26 interrupt sources including non-maskable, maskable, and software interrupt sources are available (refer to **Table 14-1**).

Remark There are two types of interrupt sources for the watchdog timer (INTWDT): non-maskable interrupts and maskable interrupts (internal). Only one of these interrupts can be selected.

Table 14-1. 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			
Non-maskable	—	INTWDT	Watchdog timer overflow (when non-maskable interrupt is selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (when interval timer mode is selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTP6			0012H	
	8	INTP7			0014H	
	9	INTTM7	TM7 underflow	Internal	0016H	(B)
	10	INTTM000	TM00 and CR000 match signal generation (when compare register is specified) TI010 valid edge detection (when the capture register is specified)		0018H	
	11	INTTM010	TM00 and CR010 match signal generation (when compare register is specified) TI000 valid edge detection (when the capture register is specified)		001AH	
	12	INTTM001	TM01 and CR001 match signal generation (when compare register is specified) TI011 valid edge detection (when the capture register is specified)		001CH	
	13	INTTM011	TM01 and CR011 match signal generation (when compare register is specified) TI001 valid edge detection (when the capture register is specified)		001EH	
	14	INTSER0	UART00 receive error generation		0020H	
	15	INTSR0	End of UART00 reception		0022H	
	16	INTST0	End of UART00 transmission		0024H	
	17	INTSR1	End of UART01 reception		0026H	
	18	INTST1	End of UART01 transmission		0028H	

Notes 1. The default priority is the priority applicable when more than one maskable interrupt is generated at the same time. 0 is the highest priority and 23 the lowest.

2. Basic configuration types (A) to (D) correspond to (A) to (D) on the next pages.

Table 14-1. Interrupt Source List (2/2)

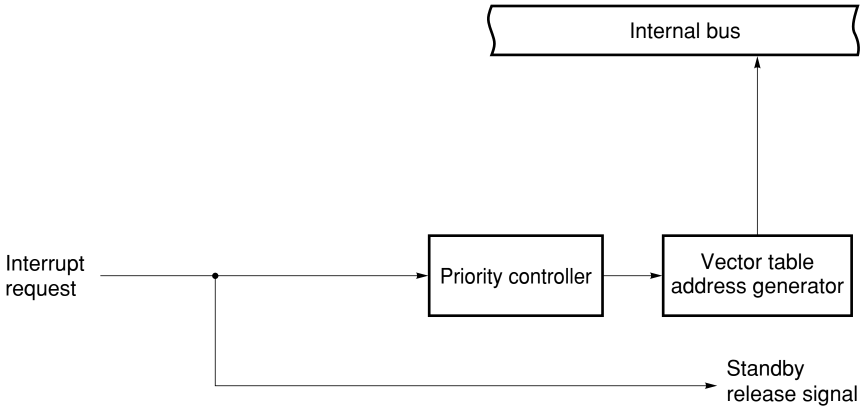
Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Maskable	19	INTTM50	TM50 and CR50 match signal generation	Internal	002AH	(B)
	20	INTTM51	TM51 and CR51 match signal generation		002CH	
	21	INTTM52	TM52 and CR52 match signal generation		002EH	
	22	INTCSI3	End of SIO3 transfer		0030H	
	23	INTAD0	End of A/D conversion		0032H	
Software	—	BRK	Execution of BRK instruction	—	003EH	(D)

Notes **1.** The default priority is the priority applicable when more than one maskable interrupt is generated at the same time. 0 is the highest priority and 23 the lowest.

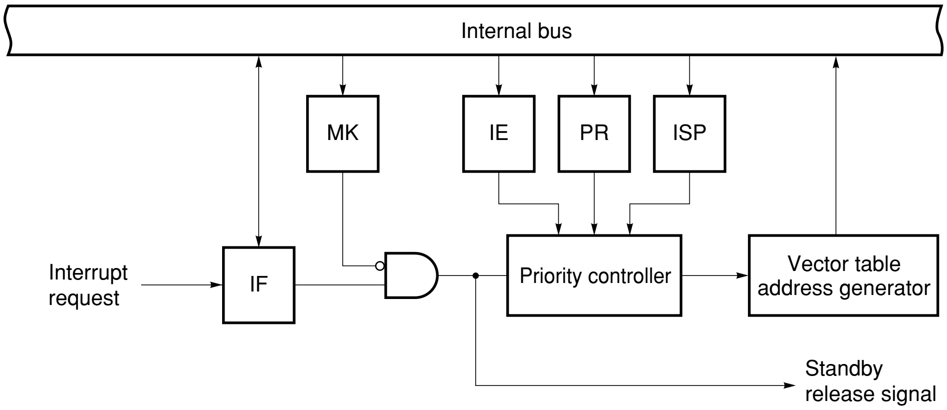
2. Basic configuration types (A) to (D) correspond to (A) to (D) on the next pages.

Figure 14-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt

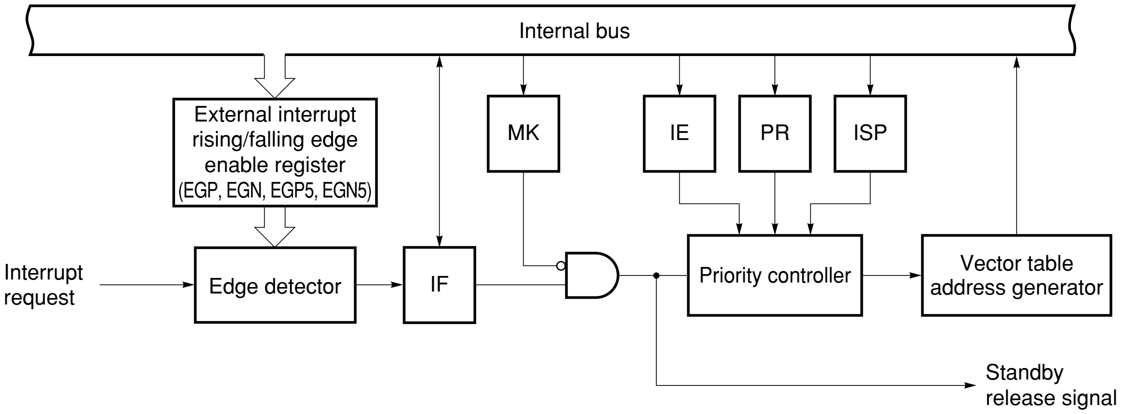
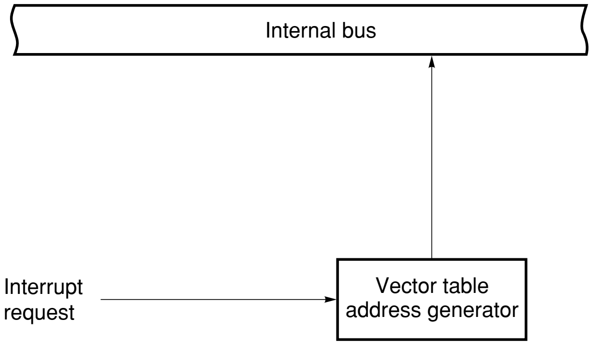

Internal bus
External interrupt rising/falling edge enable register (EGP, EGN, EGP5, EGN5)
MK
IE
PR
ISP
Interrupt request
Edge detector
IF
Priority controller
Vector table address generator
Standby release signal

Figure 14-1. Basic Configuration of Interrupt Function (2/2)

(D) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

14.3 Registers Controlling Interrupt Functions

The following eight types of registers control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L)
- Priority specification flag registers (PR0L, PR0H, PR1L)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- External interrupt rising edge enable register 5 (EGP5)
- External interrupt falling edge enable register 5 (EGN5)
- Program status word (PSW)

Table 14-2 shows the names of the interrupt request flags, interrupt mask flags, and priority specification flags corresponding to the respective interrupt request sources.

Table 14-2. Flags Corresponding to Respective Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	WDTIF[Note]	IF0L	WDTMK[Note]	MK0L	WDTPR[Note]	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	
INTP6	PIF6		PMK6		PPR6	
INTP7	PIF7	IF0H	PMK7	MK0H	PPR7	PR0H
INTTM7	TMIF7		TMMK7		TMPR7	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTTM001	TMIF001		TMMK001		TMPR001	
INTTM011	TMIF011		TMMK011		TMPR011	
INTSER0	SERIF0		SERMK0		SERPR0	
INTSR0	SRIF0		SRMK0		SRPR0	
INTST0	STIF0	IF1L	STMK0	MK1L	STPR0	PR1L
INTSR1	SRIF1		SRMK1		SRPR1	
INTST1	STIF1		STMK1		STPR1	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTTM52	TMIF52		TMMK52		TMPR52	
INTCSI3	CSIIF3		CSIMK3		CSIPR3	
INTAD0	ADIF0		ADMK0		ADPR0	

Note Interrupt control flag when the watchdog timer is used as an interval timer

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

An interrupt request flag is set to 1 when the corresponding interrupt request is generated or when an instruction is executed, and is cleared to 0 when the interrupt request is acknowledged, when the $\overline{\text{RESET}}$ signal is input, or when an instruction is executed.

IF0L, IF0H, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When using IF0L and IF0H as a 16-bit register, IF0, it is set by a 16-bit memory manipulation instruction.

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

Figure 14-2. Format of Interrupt Request Flag Registers

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After reset	R/W
IF0L	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
IF0H	SRIF0	SERIF0	TMIF011	TMIF001	TMIF010	TMIF000	TMIF7	PIF7	FFE1H	00H	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
IF1L	ADIF0	CSIIF3	TMIF52	TMIF51	TMIF50	STIF1	SRIF1	STIF0	FFE2H	00H	R/W

××IF	Interrupt request flag
0	Interrupt request signal is not generated.
1	Interrupt request signal is generated and interrupt is requested.

Cautions 1. The WDTIF flag can be read/written only when the watchdog timer is used as an interval timer. Clear the WDTIF flag to 0 when watchdog timer mode 1 is used.

2. Before restarting the timer, serial interface, or A/D converter in the standby mode, be sure to clear the interrupt request flag. Note that noise may cause an interrupt request flag to be set.

3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared, and then the interrupt routine is started.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

An interrupt mask flag enables or disables the corresponding maskable interrupt servicing and release of the standby mode.

MK0L, MK0H, and MK1L are set by a 1-bit or 8-bit memory manipulation instruction. When using MK0L and MK0H as a 16-bit register, MK0, it is set by a 16-bit memory manipulation instruction.

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

Figure 14-3. Format of Interrupt Mask Flag Register

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After reset	R/W
MK0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
MK0H	SRMK0	SERMK0	TMMK 011	TMMK 001	TMMK 010	TMMK 000	TMMK7	PMK7	FFE5H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
MK1L	ADMK0	CSIMK3	TMMK 52	TMMK 51	TMMK 50	STMK1	SRMK1	STMK0	FFE6H	FFH	R/W

××MK	Interrupt servicing control
0	Enables interrupt servicing
1	Disables interrupt servicing

Cautions 1. If the watchdog timer is used in watchdog timer mode 1, the WDTMK flag will be undefined when read.

2. Because port 0 and P54 to P57 have alternate functions of external interrupt request inputs, the corresponding interrupt request flag is set when the output mode is specified and output level of a port pin is changed.

To use the port in the output mode, therefore, set the corresponding interrupt mask flag to 1 in advance.

(3) Priority specification flag registers (PR0L, PR0H, PR1L)

A priority specification flag sets the priority of the corresponding maskable interrupt.

PR0L, PR0H, and PR1L are set by a 1-bit or 8-bit memory manipulation instruction. When using PR0L and PR0H as a 16-bit register, PR0, it is set by a 16-bit memory manipulation instruction.

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

Figure 14-4. Format of Priority Specification Flag Register

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After reset	R/W
PR0L	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	WDTPR	FFE8H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
PR0H	SRPR 0	SERPR 0	TMPR 011	TMPR 001	TMPR 010	TMPR 000	TMPR7	PPR7	FFE9H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
PR1L	ADPR0	CSIPR3	TMPR 52	TMPR 51	TMPR 50	STPR 1	SRPR 1	STPR 0	FFEAH	FFH	R/W

××PR	Priority level selection
0	High priority level
1	Low priority level

Caution To use the watchdog timer in watchdog timer mode 1, set the WDTPR flag to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

EGP and EGN specify the valid edge to be detected on pins P00 to P03.

EGP and EGN can be read or written to with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 14-5. Format of External Interrupt Rising Edge Enable Register and External Interrupt Falling Edge Enable Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0	FF48H	00H	R/W

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0	FF49H	00H	R/W

EGPn	EGNn	Valid edge of INTPn pin (n = 0 to 3)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

(5) External interrupt rising edge enable register 5 (EGP5), external interrupt falling edge enable register 5 (EGN5)

EGP5 and EGN5 specify the valid edge to be detected on pins P54 to P57.

EGP5 and EGN5 can be read or written to with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 14-6. Format of External Interrupt Rising Edge Enable Register 5 and External Interrupt Falling Edge Enable Register 5

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
EGP5	EGP57	EGP56	EGP55	EGP54	0	0	0	0	FF7CH	00H	R/W

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
EGN5	EGN57	EGN56	EGN55	EGN54	0	0	0	0	FF7DH	00H	R/W

EGP5n	EGN5n	Valid edge of INTPn pin (n = 4 to 7)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

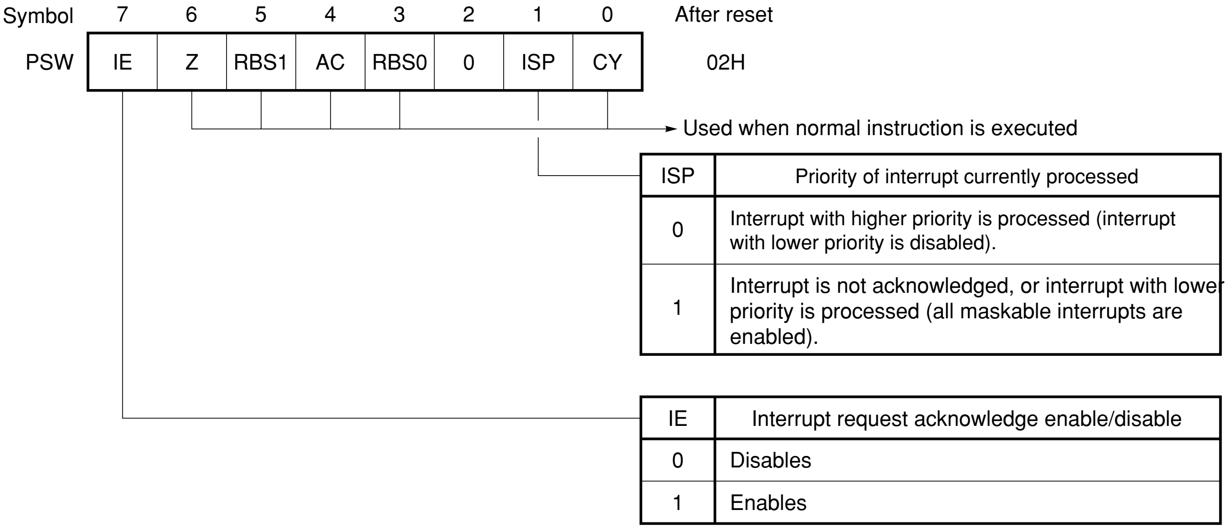
(6) Program status word (PSW)

The program status word is a register that holds the instruction execution result and current status of interrupt request. An IE flag that enables/disables the maskable interrupts and an ISP flag that controls multiple interrupts processing are mapped to this register.

This register can be read or written in 8-bit units. In addition, it can also be manipulated by using a bit manipulation instruction or dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, and when the BRK instruction is executed, the contents of the PSW are automatically saved to the stack. At this time, the IE flag is reset to 0. If a maskable interrupt request has been acknowledged, the contents of the priority flag of that interrupt are transferred to the ISP flag. The contents of the PSW can also be saved to the stack by the PUSH PSW instruction, and restored from the stack by RETI, RETB, or POP PSW instruction.

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

Figure 14-7. Configuration of Program Status Word



Symbol	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY	02H

ISP	Priority of interrupt currently processed
0	Interrupt with higher priority is processed (interrupt with lower priority is disabled).
1	Interrupt is not acknowledged, or interrupt with lower priority is processed (all maskable interrupts are enabled).

IE	Interrupt request acknowledge enable/disable
0	Disables
1	Enables

14.4 Interrupt Servicing Operation

14.4.1 Non-maskable interrupt request acknowledgement operation

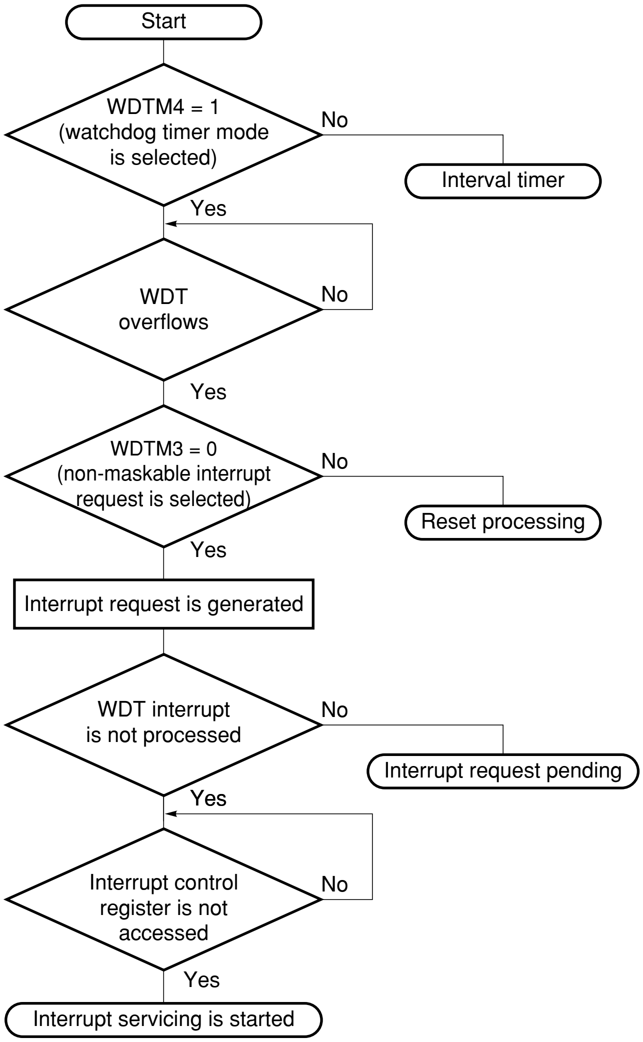
The non-maskable interrupt request is unconditionally acknowledged even when interrupt requests are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the non-maskable interrupt request is acknowledged, the contents are saved to the stack, program status word (PSW) and program counter (PC), in that order, the IE flag and ISP flag are reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

If a new non-maskable interrupt request is generated while the non-maskable interrupt service program is being executed, the interrupt request is acknowledged when the current execution of the non-maskable interrupt service program is complete (after the RETI instruction has been executed) and one instruction in the main routine has been executed. If two or more new non-maskable interrupt requests are generated while the non-maskable interrupt service program is being executed, only one non-maskable interrupt request is acknowledged after execution of the non-maskable interrupt service program is complete.

Figure 14-8 shows the flowchart from non-maskable interrupt request generation to acknowledgement, Figure 14-9 shows the timing of non-maskable interrupt request acknowledgement, and Figure 14-10 shows the acknowledgement operation when multiple non-maskable interrupt requests are generated.

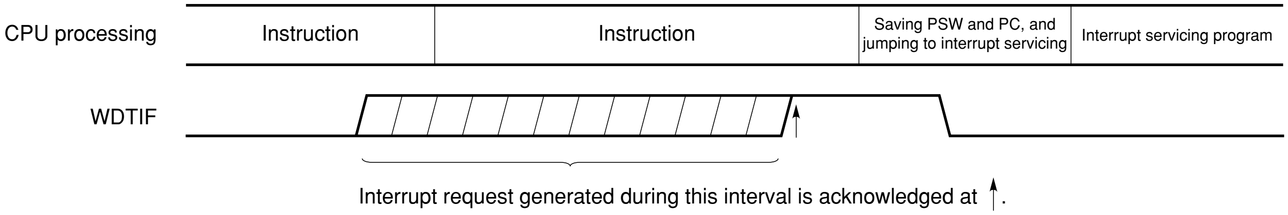
Figure 14-8. Flowchart from Non-Maskable Interrupt Request Generation to Acknowledgement


Start
WDTM4 = 1 (watchdog timer mode is selected)
No
Interval timer
Yes
WDT overflows
No
Yes
WDTM3 = 0 (non-maskable interrupt request is selected)
No
Reset processing
Yes
Interrupt request is generated
WDT interrupt is not processed
No
Interrupt request pending
Yes
Interrupt control register is not accessed
No
Yes
Interrupt servicing is started

WDTM: Watchdog timer mode register

WDT: Watchdog timer

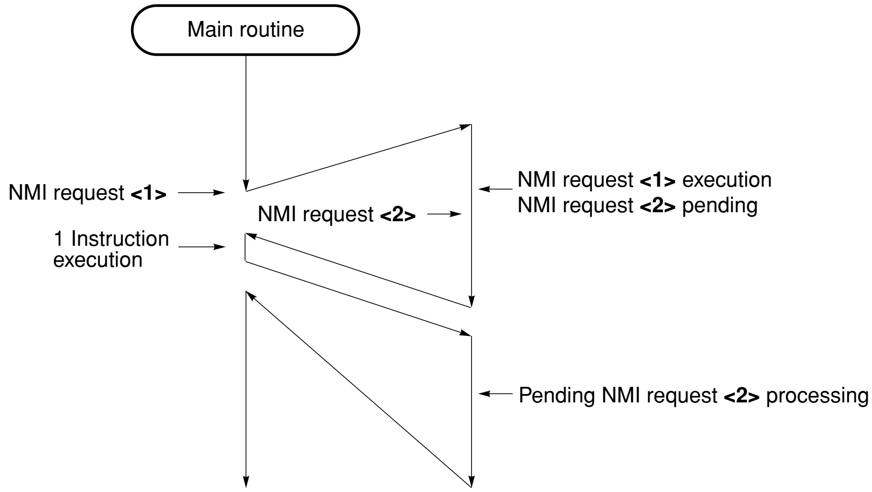
Figure 14-9. Timing of Non-Maskable Interrupt Request Acknowledgement


CPU processing
Instruction
Instruction
Saving PSW and PC, and jumping to interrupt servicing
Interrupt servicing program
WDTIF
Interrupt request generated during this interval is acknowledged at ↑.

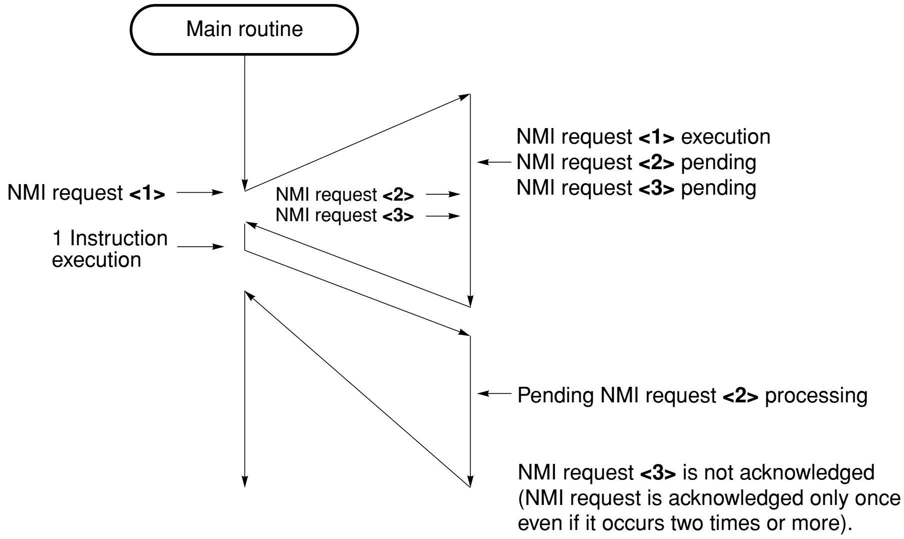
WDTIF: Watchdog timer interrupt request flag

Figure 14-10. Acknowledgement Operation of Non-Maskable Interrupt Request

(a) When new non-maskable interrupt request is generated while non-maskable interrupt service program is being executed



(b) If two new non-maskable interrupt requests are generated while non-maskable interrupt service program is being executed


Main routine
NMI request <1>
NMI request <2>
NMI request <3>
1 Instruction execution
NMI request <1> execution
NMI request <2> pending
NMI request <3> pending
Pending NMI request <2> processing
NMI request <3> is not acknowledged
(NMI request is acknowledged only once
even if it occurs two times or more).

14.4.2 Maskable interrupt request acknowledgement operation

A maskable interrupt request can be acknowledged when the interrupt request flag is set to 1 and the corresponding interrupt request mask (MK) flag is cleared to 0. A vectored interrupt request is acknowledged in the interrupt enabled status (when the IE flag is set to 1). However, an interrupt request with a lower priority cannot be acknowledged while an interrupt with a higher priority is being serviced (when the ISP flag is reset to 0).

The time required to start the interrupt servicing after a maskable interrupt request has been generated is shown in Table 14-3.

For the timing of the interrupt request acknowledgement, refer to **Figures 14-12** and **14-13**.

Table 14-3. Time from Generation of Maskable Interrupt Request to Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note The wait time is the maximum when an interrupt request is generated immediately before a division instruction.

Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

When two or more maskable interrupt requests are generated at the same time, they are acknowledged starting from the one assigned the highest priority by the priority specification flag. If the same priorities are specified by the priority specification flag, the interrupt with the highest default priority is acknowledged first.

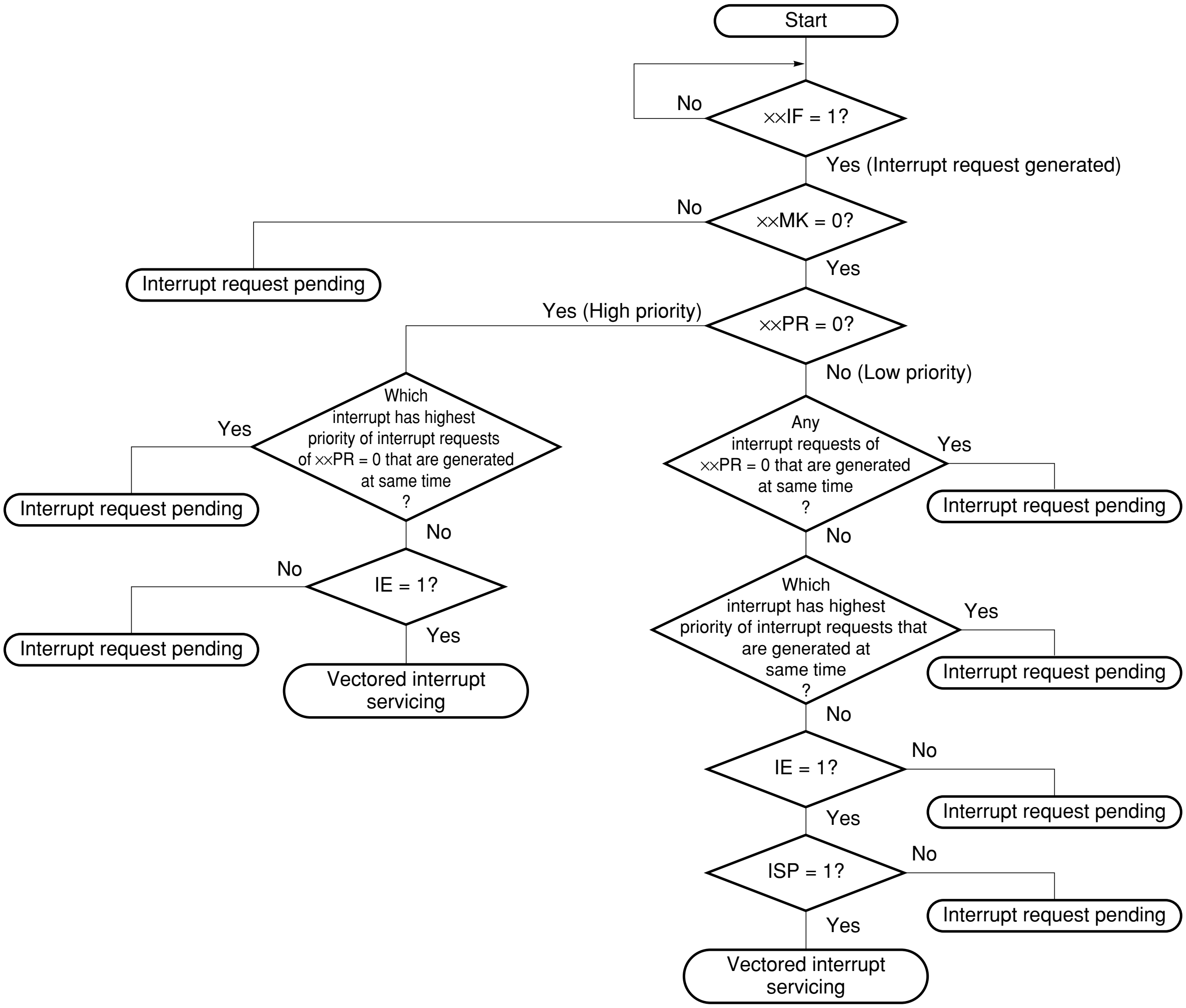
A pending interrupt request is acknowledged when the status in which it can be acknowledged is set.

Figure 14-11 shows the algorithm of acknowledging interrupt requests.

When a maskable interrupt request is acknowledged, the contents are saved to the stack, the program status word (PSW) and the program counter (PC), in that order, the IE flag is reset to 0, and the contents of the interrupt priority specification flag of the acknowledged interrupt request are transferred to the ISP flag. In addition, the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt servicing, use the RETI instruction.

Figure 14-11. Interrupt Request Acknowledgement Program Algorithm


Start
××IF = 1?
No
Yes (Interrupt request generated)
××MK = 0?
No
Interrupt request pending
Yes
××PR = 0?
Yes (High priority)
No (Low priority)
Which interrupt has highest priority of interrupt requests of ××PR = 0 that are generated at same time ?
Yes
Interrupt request pending
No
IE = 1?
No
Interrupt request pending
Yes
Vectored interrupt servicing
Any interrupt requests of ××PR = 0 that are generated at same time ?
Yes
Interrupt request pending
No
Which interrupt has highest priority of interrupt requests that are generated at same time ?
Yes
Interrupt request pending
No
IE = 1?
No
Interrupt request pending
Yes
ISP = 1?
No
Interrupt request pending
Yes
Vectored interrupt servicing

××IF: Interrupt request flag

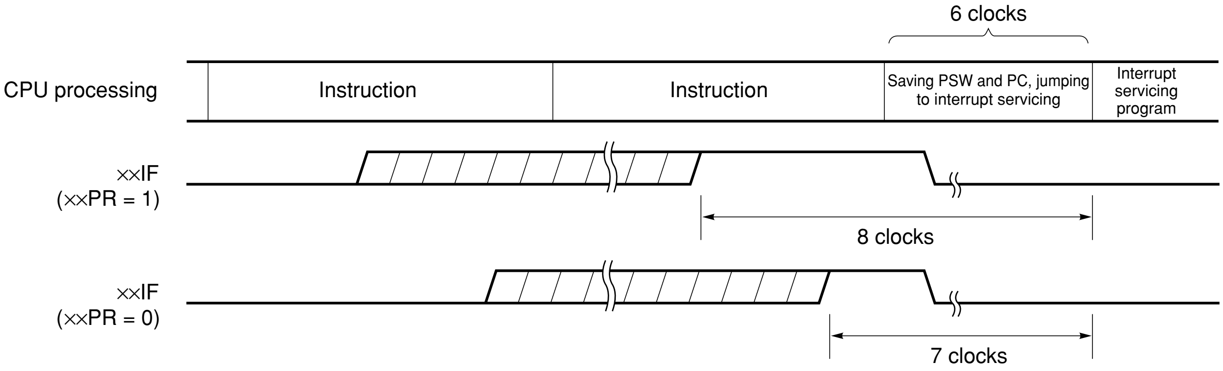
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgement of maskable interrupt request (1 = Enable, 0 = Disable)

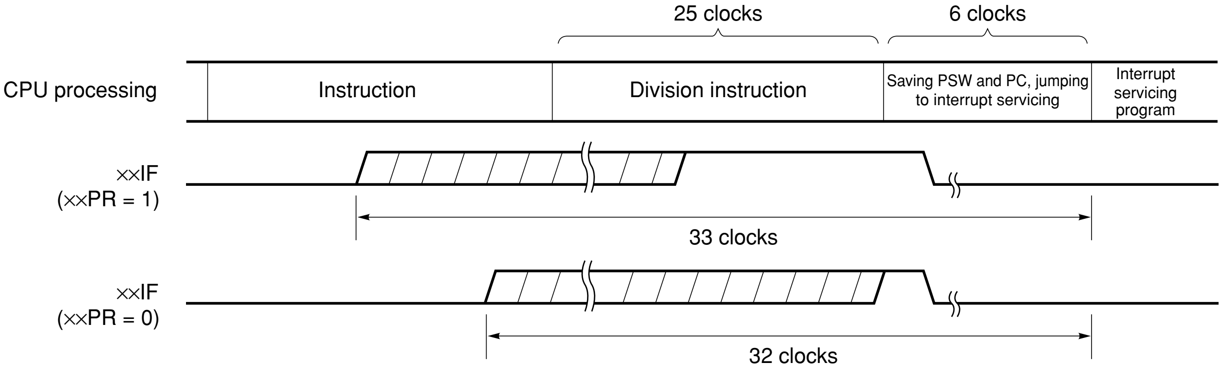
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = Higher priority interrupt servicing, 1 = No interrupt request acknowledged, or lower priority interrupt servicing)

Figure 14-12. Interrupt Request Acknowledgement Timing (Minimum Time)



Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

Figure 14-13. Interrupt Request Acknowledgement Timing (Maximum Time)



Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

14.4.3 Software interrupt request acknowledgement operation

The software interrupt request can be acknowledged when the BRK instruction is executed. This interrupt cannot be disabled.

When the software interrupt request is acknowledged, the contents are saved to the stack, the program status word (PSW) and the program counter (PC), in that order, the IE flag is reset to 0, the contents of the vector table (003EH and 003FH) are loaded to the PC, and execution branches.

To return from the software interrupt servicing, use the RETB instruction.

Caution Do not use the RETI instruction to return from the software interrupt.

14.4.4 Multiple interrupt servicing

Acknowledging another interrupt request while one interrupt is being serviced is called multiple interrupts.

Multiple interrupts are not generated unless interrupt requests are enabled (IE = 1) (except the non-maskable interrupt). When an interrupt request is acknowledged, the other interrupts are disabled (IE = 0). To enable multiple interrupts, therefore, the IE flag must be set to 1 by executing the EI instruction during interrupt servicing and interrupts must be enabled.

Even if interrupt requests are enabled, some multiple interrupts are not acknowledged due to control by the programmable priority. An interrupt has two types of priorities: a default priority and a programmable priority. Multiple interrupts are controlled by the programmable priority.

In the EI status, if an interrupt request having the same as or higher priority than that of the interrupt currently being serviced is generated, the interrupt is acknowledged as multiple interrupt. If an interrupt request with a priority lower than that of the interrupt currently being serviced is generated, the interrupt is not acknowledged as multiple interrupt.

If interrupts are disabled, or if a multiple interrupt is not acknowledged because it has a low priority, the interrupt is held pending. After the servicing of the current interrupt is complete, and after one instruction of the main servicing has been executed, the pending interrupt is acknowledged.

Multiple interrupts are not acknowledged while the non-maskable interrupt is being serviced.

Table 14-4 shows interrupt requests enabled for multiple interrupts. Figure 14-14 shows multiple interrupt examples.

Table 14-4. Interrupt Requests Enabled for Multiple Interrupt During Interrupt Servicing

Multiple Interrupt Request Servicing Interrupt		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			××PR = 0		××PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		×	×	×	×	×
Maskable interrupt	ISP = 0	√	√	×	×	×
	ISP = 1	√	√	×	√	×
Software interrupt		√	√	×	√	×

Remarks 1. √: Multiple interrupt enabled.

×: Multiple interrupt disabled.

2. ISP and IE are flags included in PSW.

ISP = 0: Interrupt with higher priority is serviced.

ISP = 1: Interrupt request is not acknowledged or interrupt with lower priority is being serviced.

IE = 0: Acknowledging interrupt request is disabled.

IE = 1: Acknowledging interrupt request is enabled.

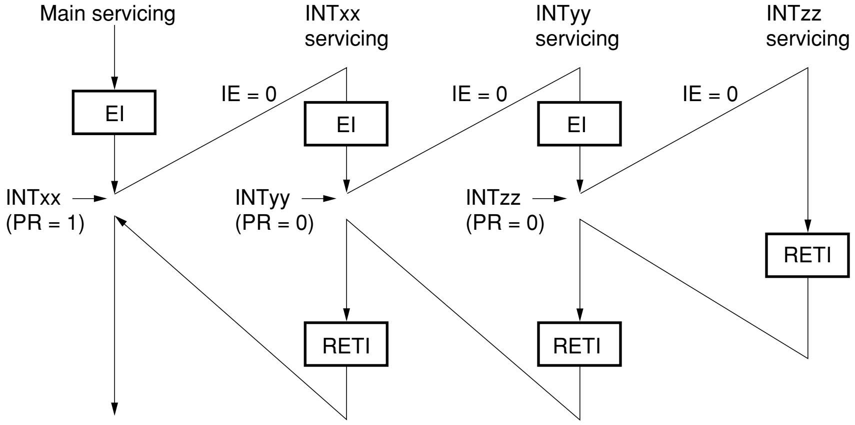
3. ××PR is flag included in PR0L, PR0H, and PR1L.

××PR = 0: Higher priority level

××PR = 1: Lower priority level

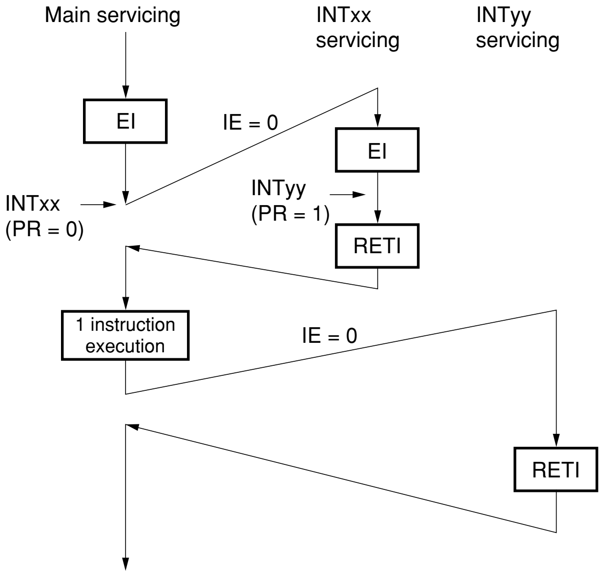
Figure 14-14. Multiple Interrupt Example (1/2)

Example 1. Multiple interrupt is generated twice



This multiple interrupt example shows two interrupt requests, INTyy and INTzz, being acknowledged while interrupt INTxx is being serviced. Before each interrupt request is acknowledged, the EI instruction is always issued and interrupt requests are enabled.

Example 2. Multiple interrupt is not generated because of its priority

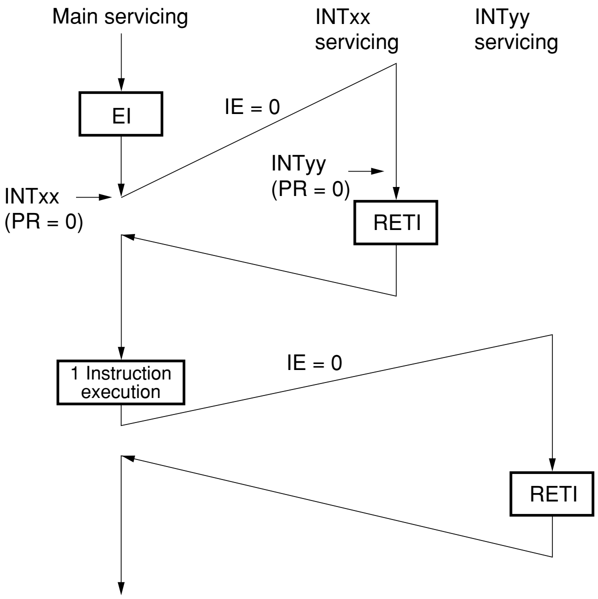


INTyy, which is generated while INTxx is being serviced, is not acknowledged and multiple interrupt servicing is not performed because the priority of INTyy is lower than that of INTxx. INTyy is held pending and is acknowledged after one instruction of the main servicing has been executed.

PR = 0:	Higher priority level
PR = 1:	Lower priority level
IE = 0:	Acknowledging interrupt request is disabled

Figure 14-14. Multiple Interrupt Example (2/2)

Example 3. Multiple interrupt is not generated because interrupts not enabled



While INTxx is serviced, other interrupts are not enabled (the EI instruction has not been executed). Therefore, INTyy is not acknowledged and multiple interrupt servicing is not performed. This interrupt (INTyy) is held pending and is acknowledged after one instruction of the main servicing has been executed.

PR = 0:	Higher priority level
IE = 0:	Acknowledging interrupt request is disabled

14.4.5 Pending interrupt requests

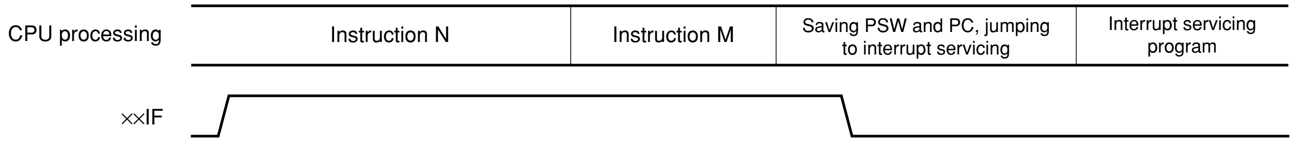
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgement is held pending until the end of execution of the next instruction. These instructions (instructions that have interrupt requests held pending) are listed 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 instruction to IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, PR1L, EGP, EGN, EGP5, EGN5 registers

Caution The BRK instruction is not one of the above-listed instructions that have interrupt requests held pending. 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. However, the non-maskable interrupt request is acknowledged.

The timing with which interrupt requests are held pending is shown in Figure 14-15.

Figure 14-15. Pending Interrupt Request


CPU processing
Instruction N
Instruction M
Saving PSW and PC, jumping to interrupt servicing
Interrupt servicing program
××IF

Remarks 1. Instruction N: Instruction that has interrupt request held pending

2. Instruction M: Instruction other than one which has interrupt request held pending

3. The operation of ××IF (interrupt request) is not affected by the value of ××PR (priority level).

15.1 External Device Expansion Function

The external device expansion function is for connecting an external device to areas other than the internal ROM, RAM, and SFR areas. To connect an external device, ports 4 and 6 are used. Port 4 controls address/data, read/write strobe, wait, and address strobe signals.

The μPD780982, 780983, 780984, and 780986 can be expanded with 256 bytes of external memory. By using an external access area in the SFR area, the μPD780988 can be expanded with 16 bytes of external memory.

★ **Caution The external device expansion function can only be used under conditions of f_X = 8.38 MHz or lower, and V_{DD} = 4.0 to 5.5 V.**

Table 15-1. 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
$\overline{\text{RD}}$	Read strobe signal	P64
$\overline{\text{WR}}$	Write strobe signal	P65
$\overline{\text{WAIT}}$	Wait signal	P66
ASTB	Address strobe signal	P67

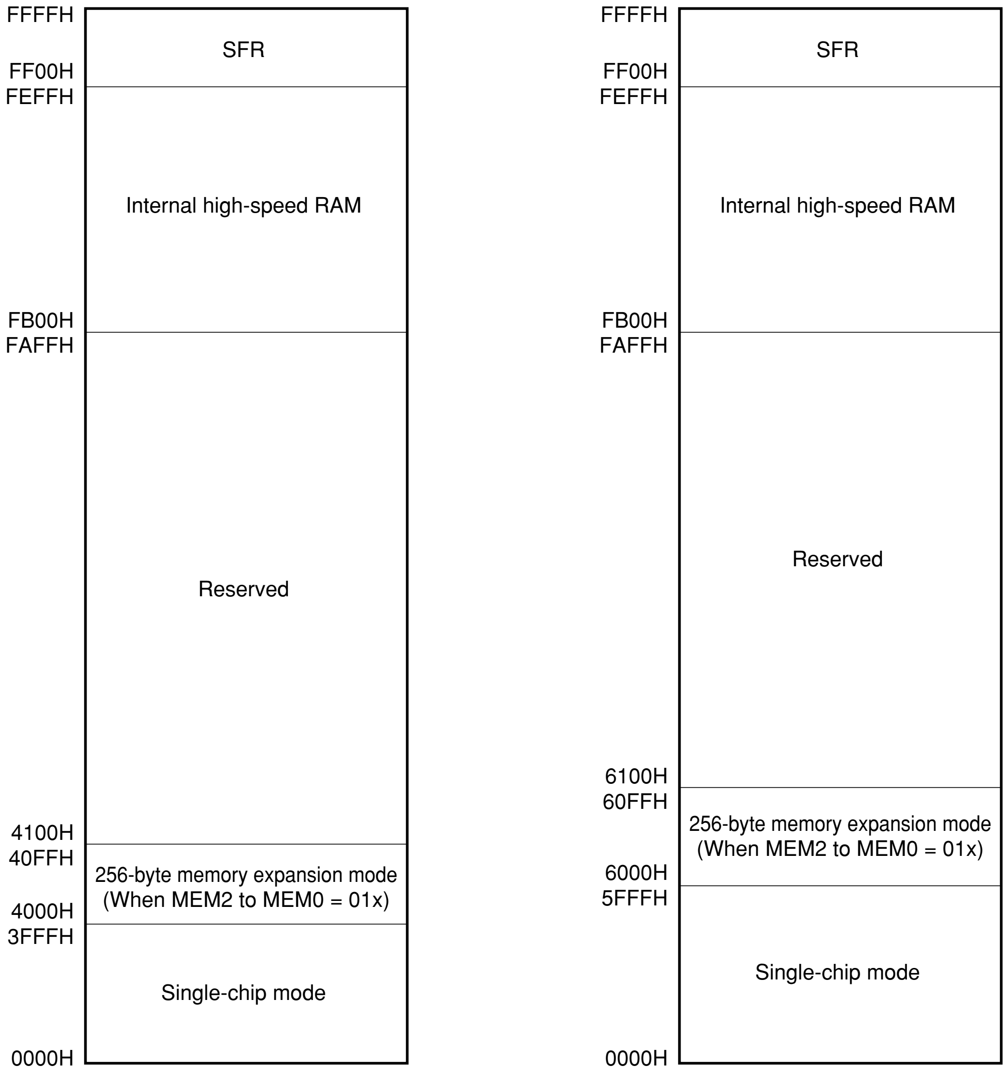
Table 15-2. Status of Ports 4 and 6 in External Memory Expansion Mode

Port	Port 4	Port 6			
External Expansion Mode	0 to 7	4	5	6	7
Single-chip mode	Port	Port			
256-byte memory expansion mode	Address/data	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			

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

The memory map is as follows when the external device expansion function is used.

Figure 15-1. Memory Map When External Device Expansion Function Used (1/3)

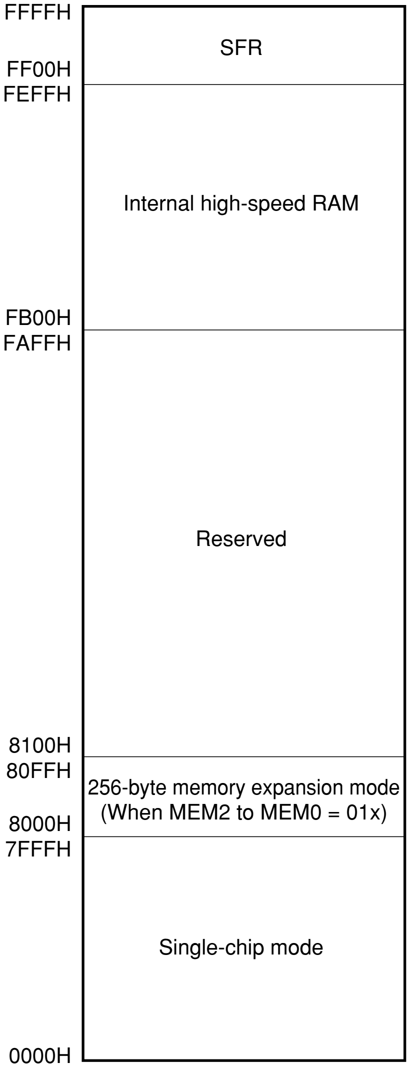


(a) Memory map of the μPD78F0988A when the μPD780982 and flash memory capacity are 16 KB

(b) Memory map of the μPD78F0988A when the μPD780983 and flash memory capacity are 24 KB

Figure 15-1. Memory Map When External Device Expansion Function Used (2/3)

(c) Memory map of the μPD78F0988A when the μPD780984 and flash memory capacity are 32 KB



(d) Memory map of the μPD78F0988A when the μPD780986 and flash memory capacity are 48 KB

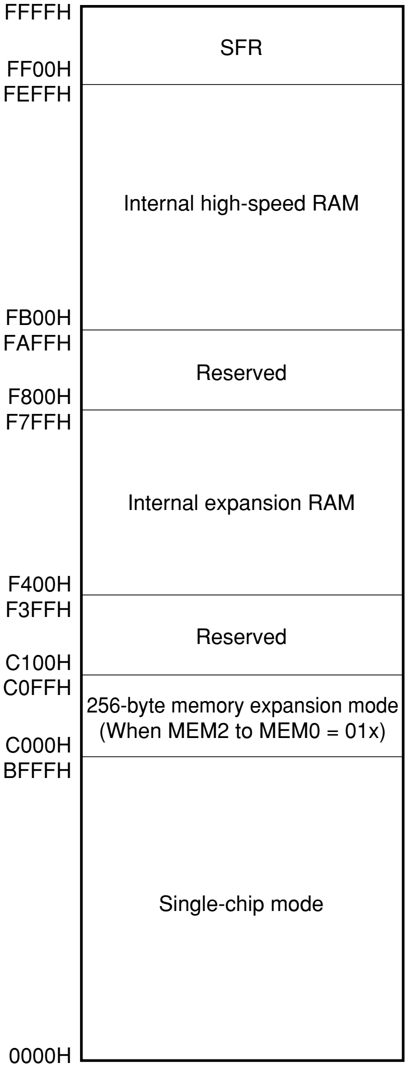
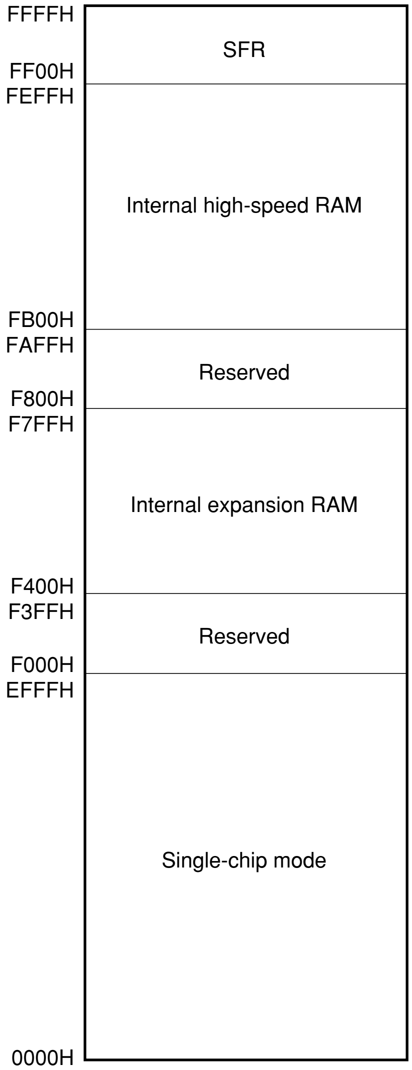


Figure 15-1. Memory Map When External Device Expansion Function Used (3/3)

(e) Memory map of the μPD78F0988A when the μPD780988 and flash memory capacity are 60 KB



Cautions 1. The μPD78F0988A of when the μPD780988 and flash memory capacity is 60 KB cannot be expanded with external memory of 256 bytes. Use of the SFR area's external access area will allow 16-byte external memory expansion.

2. Setting the flash memory capacity to 48 KB or less with the internal memory size switching register (IMS) will allow the μPD78F0988A to be expanded with 256 bytes of external memory.

15.2 Registers Controlling External Device Expansion Function

The external device expansion function is controlled by the following three registers.

- Memory expansion mode register (MEM)
- Memory expansion wait setting register (MM)
- Memory size switching register (IMS)

(1) Memory expansion mode register (MEM)

MEM is a register that sets an external expansion area.
MEM is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 15-2. Format of Memory Expansion Mode Register

	Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
★	MEM	0	0	0	0	0	MEM2	MEM1	MEM0	FF47H	00H	W

MEM2	MEM1	MEM0	Single-chip/memory expansion mode selection	P40 to P47, P64 to P67 pin status	
				P40 to P47	P64 to P67
0	0	0	Single-chip mode	Port mode	
0	1	×	256-byte memory expansion mode	AD0 to AD7	P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = $\overline{\text{WAIT}}$ P67 = ASTB
Other than above			Setting prohibited		

Caution Always set bits 3 to 7 to 0.

Remark ×: don’t care

(2) Memory expansion wait setting register (MM)

MM is a register that sets the number of wait states.

MM is set by an 8-bit memory manipulation instruction.

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

Figure 15-3. Format of Memory Expansion Wait Setting Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
MM	0	0	PW1	PW0	0	0	0	0	FFF8H	10H	R/W

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

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

(3) Memory size switching register (IMS)

This register sets the capacities of the internal ROM and internal high-speed RAM.

IMS is set by an 8-bit memory manipulation instruction.

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

Figure 15-4. Format of Memory Size Switching Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	CFH	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	1	0	0	16 KB
0	1	1	0	24 KB
1	0	0	0	32 KB
1	1	0	0	48 KB^Note
1	1	1	1	60 KB
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1,024 bytes
Other than above			Setting prohibited

Note Make the flash memory capacity 48 KB or less when using the external device expansion function with the μPD78F0988A.

Cautions 1. The value of IMS after reset is the same (CFH) for all the products in the μPD780988 Subseries, regardless of the internal memory capacity. Therefore, be sure to set the value of IMS according to the internal memory capacity of the product used.

2. The external memory space can be expanded in a space other than that specified by IMS, regardless of the internal memory capacity.

Table 15-3. Set Value of Internal Memory Size Switching Register

Part Number	Set Value of IMS
μPD780982	C4H
μPD780983	C6H
μPD780984	C8H
μPD780986	CCH
μPD780988	CFH**Note 1**
μPD78F0988A	**Note 2**

Notes 1. There is no need to change the set value of IMS because the initial value of the μPD780988 is CFH.

2. Set C4H, C6H, C8H, CCH, or CFH according to the mask ROM version used.

15.3 Timing of External Device Expansion Function

The timing control signal output pins used in the external memory expansion mode are as follows.

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

This pin outputs a read strobe signal when an instruction is fetched or data is accessed from the external memory.

When the internal memory is accessed, the read strobe signal is not output (instead, this pin holds a high level).

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

This pin outputs a write strobe signal when the external memory is accessed for data.

When the internal memory is accessed, the write strobe signal is not output (this pin holds a high level).

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

This pin inputs an external wait signal.

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

When the internal memory is accessed, the external wait signal is ignored.

(4) ASTB pin (alternate function: P67)

This pin outputs an address strobe signal which is always output regardless of instruction fetch or data access from the external memory.

The address strobe signal is also output when the internal memory is accessed.

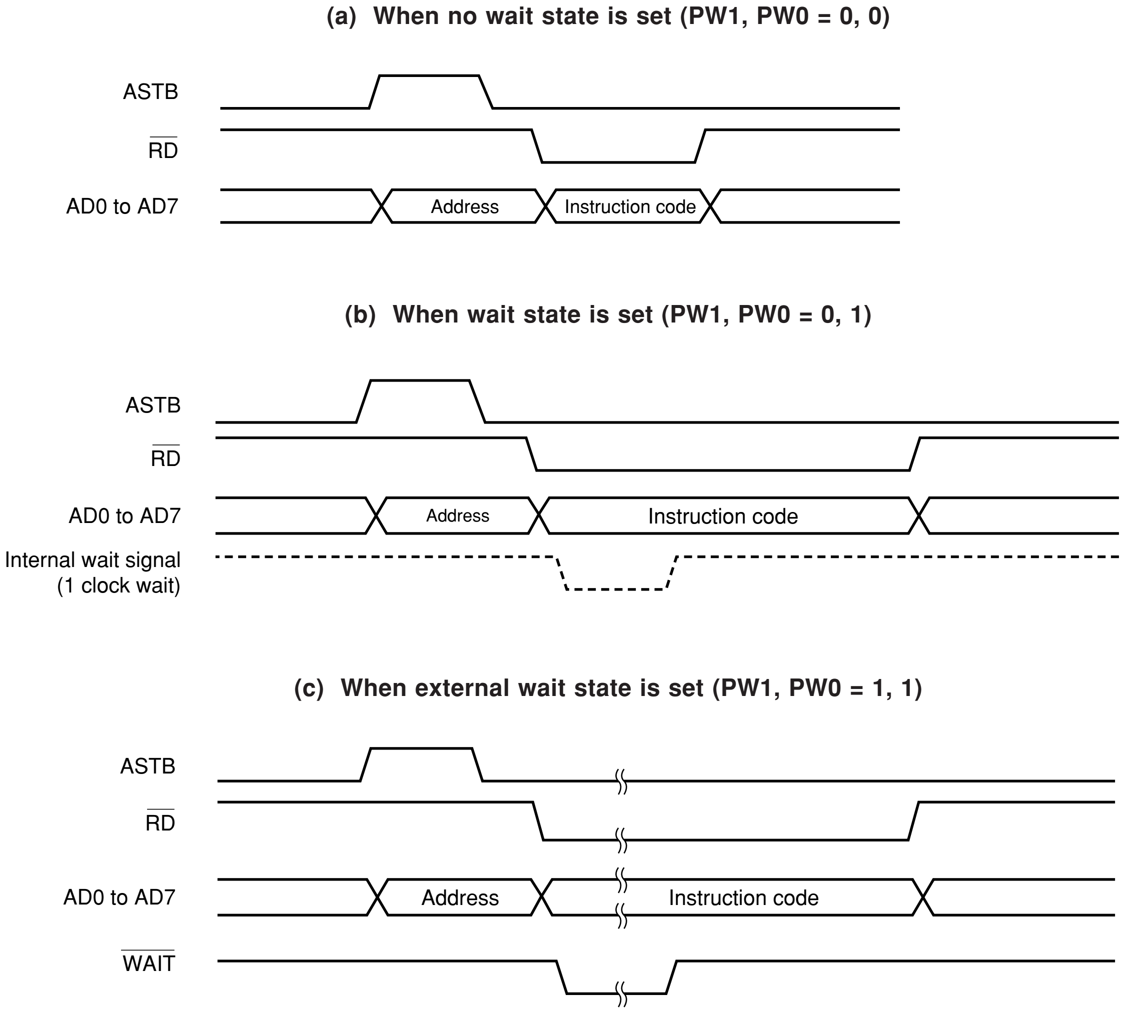
(5) AD0 to AD7 pins (alternate function: P40 to P47)

These pins output address and data signals. The valid signals are output or input when instructions are fetched or data is accessed from the external memory.

The status of the signal also changes when the internal memory is accessed (the output contents are undefined).

Figures 15-5 to 15-8 show the timing charts.

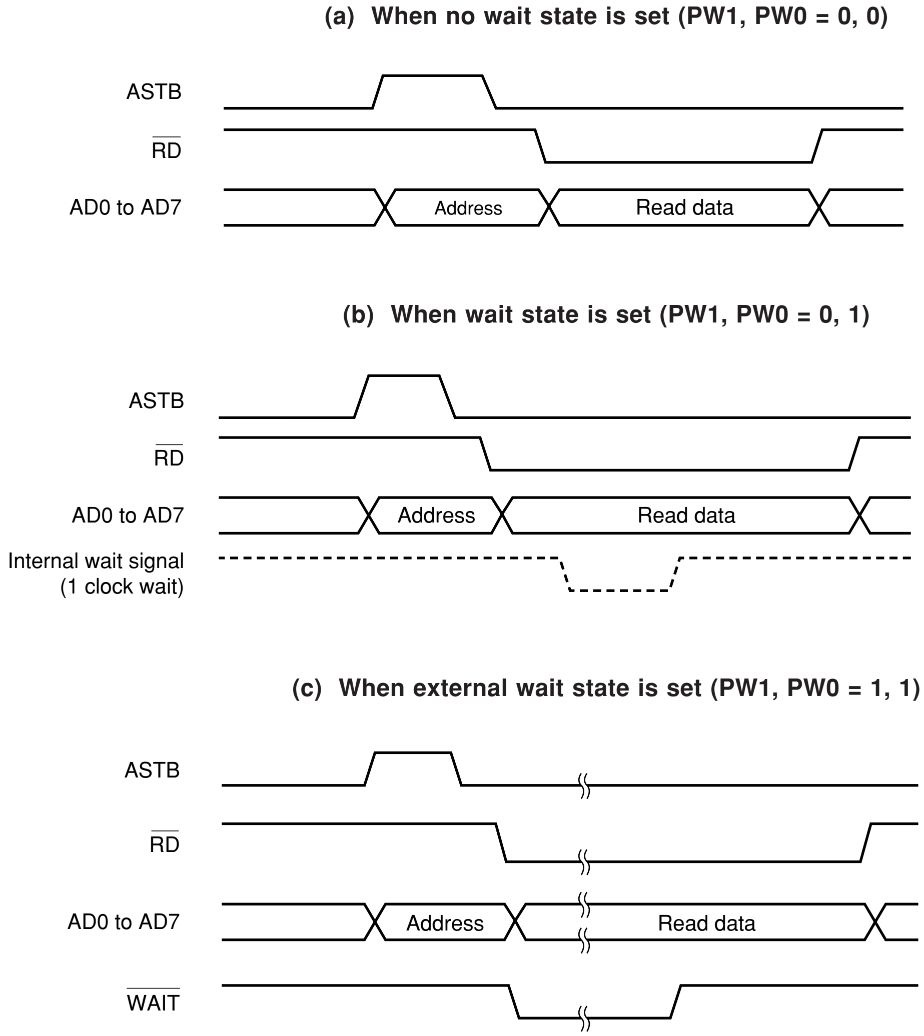
Figure 15-5. Instruction Fetch from External Memory


(a) When no wait state is set (PW1, PW0 = 0, 0)

(b) When wait state is set (PW1, PW0 = 0, 1)

(c) When external wait state is set (PW1, PW0 = 1, 1)

Figure 15-6. Read Timing of External Memory

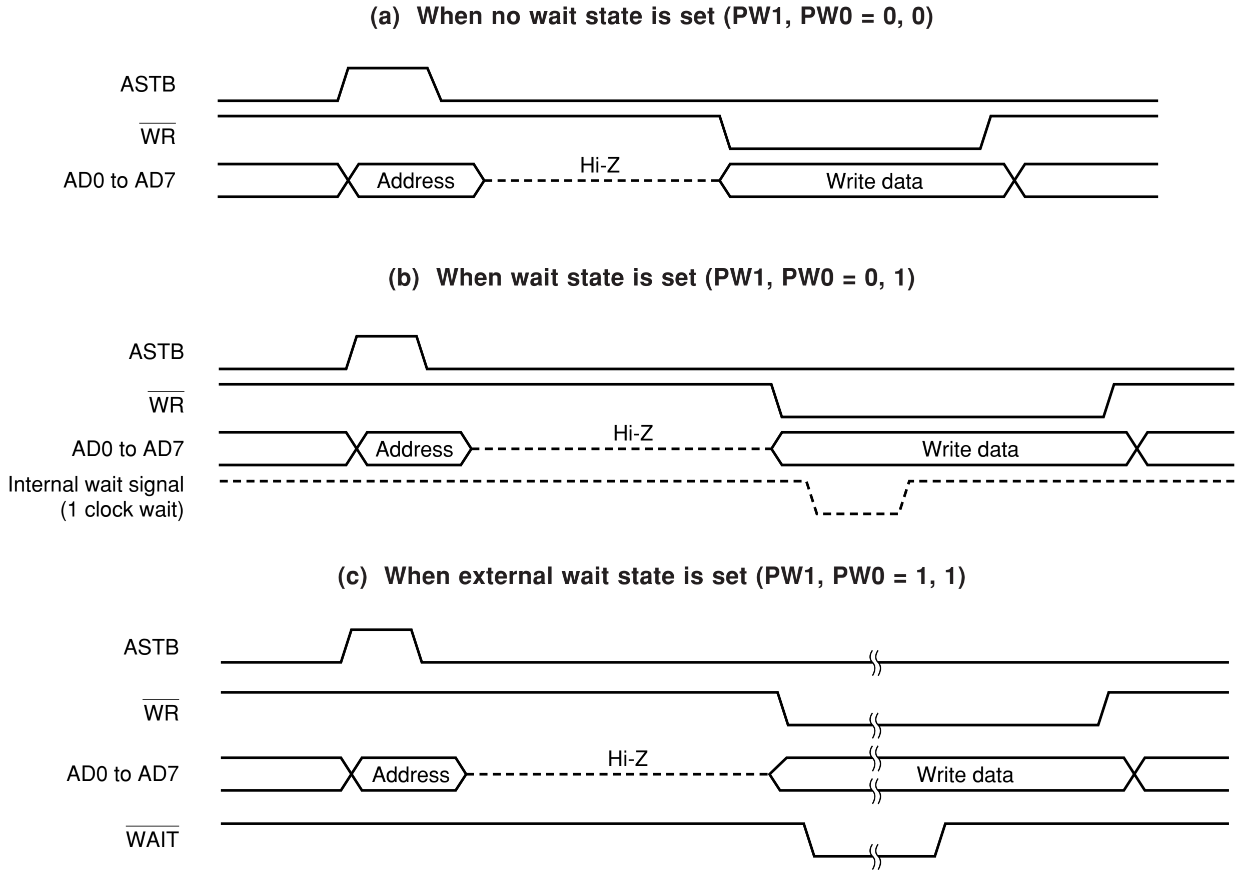


(a) When no wait state is set (PW1, PW0 = 0, 0)

(b) When wait state is set (PW1, PW0 = 0, 1)

(c) When external wait state is set (PW1, PW0 = 1, 1)

Figure 15-7. Write Timing of External Memory

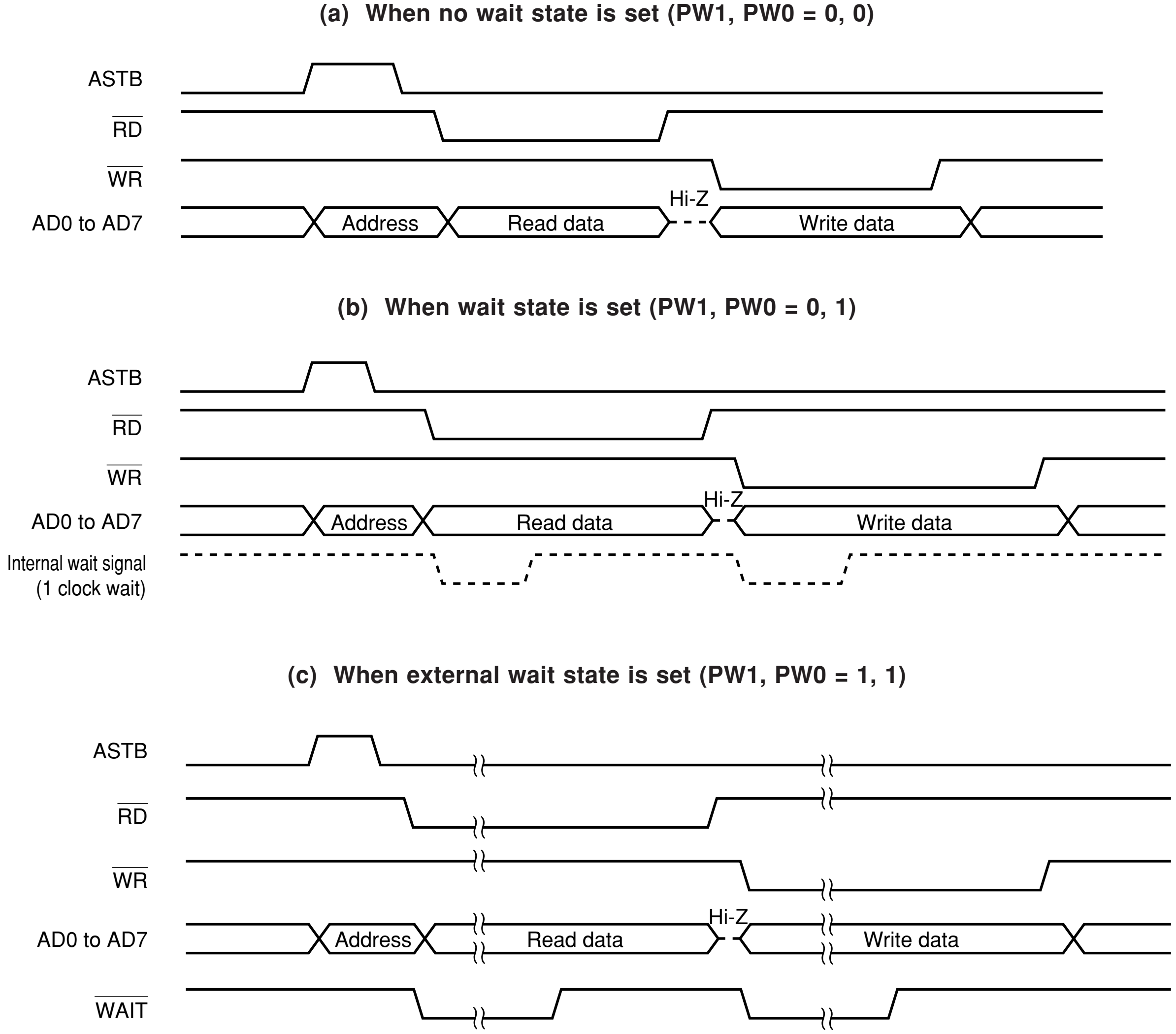


(a) When no wait state is set (PW1, PW0 = 0, 0)

(b) When wait state is set (PW1, PW0 = 0, 1)

(c) When external wait state is set (PW1, PW0 = 1, 1)

Figure 15-8. Read-Modify-Write Timing of External Memory



(a) When no wait state is set (PW1, PW0 = 0, 0)

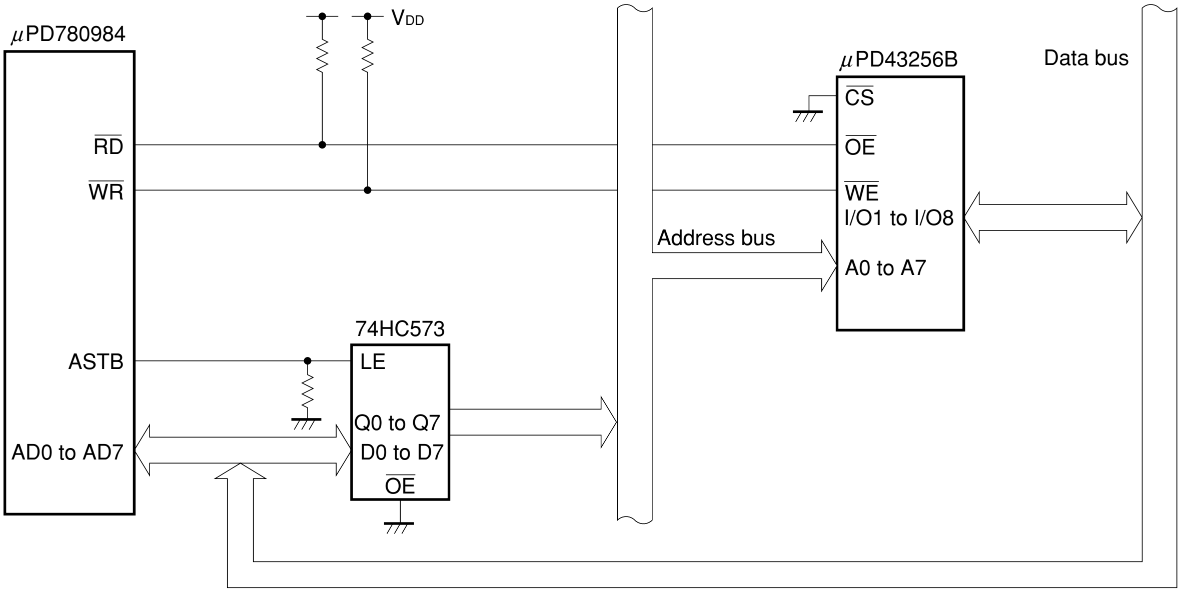
(b) When wait state is set (PW1, PW0 = 0, 1)

(c) When external wait state is set (PW1, PW0 = 1, 1)

15.4 Example of Connection with Memory

Figure 15-9 shows an example of connecting the μPD780984 and an external memory. In this application example, SRAM is connected. In addition, the external device expansion function is used in the full address mode, and 32 KB of addresses, 0000H to 7FFFH, are allocated to internal ROM; addresses 8000H and higher are allocated to SRAM.

Figure 15-9. Example of Connecting μPD780984 and Memory



CHAPTER 16 STANDBY FUNCTION

16.1 Standby Function and Configuration

16.1.1 Standby function

The standby function is used to reduce the current consumption of the system and can be effected in the following two modes.

(1) HALT mode

This mode is set when the HALT instruction is executed. The HALT mode stops the operation clock of the CPU. The system clock oscillator continues oscillating. This mode does not reduce the current consumption as much as the STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations such as a watch operation.

(2) STOP mode

This mode is set when the STOP instruction is executed. The STOP mode stops the system clock oscillator and stops the entire system. The current consumption of the CPU can be substantially reduced in this mode. The low voltage (V_{DD} = 2.0 V) of the data memory can be retained. Therefore, this mode is useful for retaining the contents of the data memory at an extremely low current.

The STOP mode can be released by an interrupt request, so this mode can be used for intermittent operation. However, a certain amount of time is required until the system clock oscillator stabilizes after the STOP mode is released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting the standby mode are all retained. In addition, the statuses of the output latch of the I/O ports and output buffer are also retained.

Cautions 1. To set the STOP mode, be sure to stop the operations of the peripheral hardware before executing the STOP instruction.

2. To reduce the current consumption of the A/D converter, clear bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 0 to stop A/D conversion, and then execute the HALT or STOP instruction.

16.1.2 Register controlling standby function

The wait time during which oscillation is stabilized after the STOP mode is released by an interrupt request is controlled by the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 04H. Therefore, to release the STOP mode by inputting the $\overline{\text{RESET}}$ signal, the time required to release the mode is 2^{17}/fx.

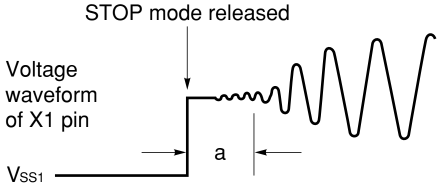
★ **Figure 16-1. Format of Oscillation Stabilization Time Select Register**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selects oscillation stabilization tim when STOP mode released		
				At fx = 12 MHz[Note]	At fx = 8.38 MHz
0	0	0	2^{12}/fx	341.3 μs	488.8 μs
0	0	1	2^{14}/fx	1.36 ms	1.96 ms
0	1	0	2^{15}/fx	2.73 ms	3.91 ms
0	1	1	2^{16}/fx	5.46 ms	7.82 ms
1	0	0	2^{17}/fx	10.9 ms	15.6 ms
Other than above			Setting prohibited		

Note Expanded-specification products only.

Caution The wait time when the STOP mode is released does not include the time required for the clock oscillation to start after the STOP mode has been released (see “a” in the figure below). The same applies when the STOP mode is released by $\overline{\text{RESET}}$ input or generation of an interrupt request.



Remark fx: System clock oscillation frequency

16.2 Operation of Standby Function

16.2.1 HALT mode

(1) Setting and operation status of HALT mode

The HALT mode is set by executing the HALT instruction.

The operation status in the HALT mode is shown in the table below.

Table 16-1. Operation Status in HALT Mode

Item		Operation Status
Clock generator		Oscillatable Supply of clock to CPU is stopped.
CPU		Stops operation.
Port (output latch)		Retains previous status before setting HALT mode.
16-bit timer/event counter		Operable
8-bit timer/event counter		
10-bit inverter control timer		
Watchdog timer		
Real-time output port		
A/D converter		
Serial interface		
External interrupt		
Externally extended bus line	AD0 to AD7	High impedance
	ASTB	Low level
	$\overline{\text{WR}}$, $\overline{\text{RD}}$	High level
	$\overline{\text{WAIT}}$	High impedance

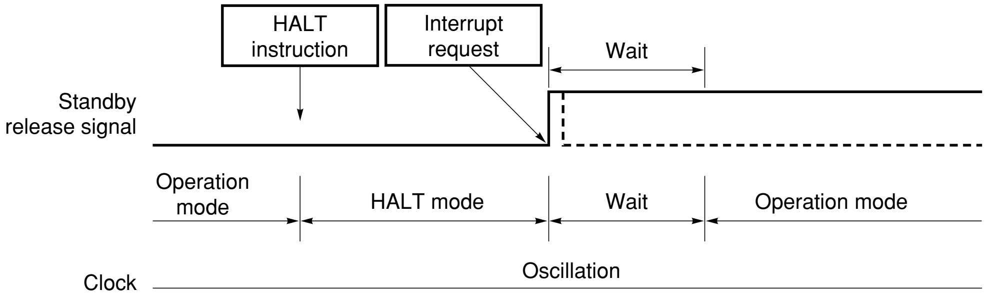
(2) Releasing HALT mode

The HALT mode can be released by the following three sources.

(a) Releasing by unmasked interrupt request

If an unmasked interrupt request is generated, the HALT mode is released. If the interrupt request is enabled at this time, vectored interrupt servicing is performed. If the interrupt request is disabled, the instruction at the next address is executed.

Figure 16-2. Releasing HALT Mode by Interrupt Request



Remarks 1. The dotted lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

2. The wait time is as follows.

• When vectored interrupt servicing is performed:	8 to 9 clocks
• When vectored interrupt servicing is not performed:	2 to 3 clocks

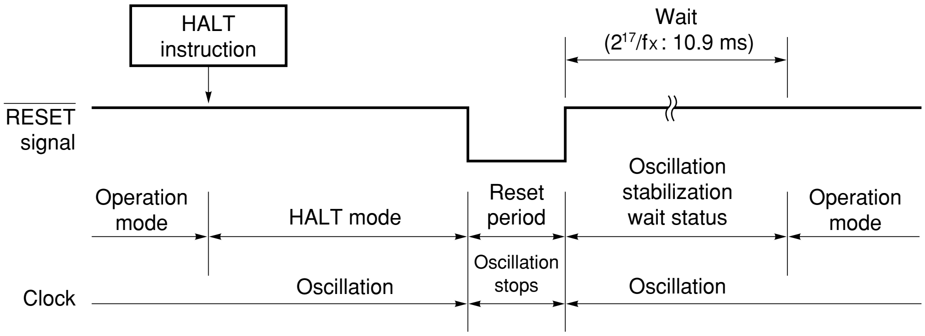
(b) Releasing by non-maskable interrupt request

If a non-maskable interrupt request is generated, the HALT mode is released regardless of whether interrupt requests are enabled or disabled, and vectored interrupt servicing is performed.

(c) Releasing by $\overline{\text{RESET}}$ input

If the $\overline{\text{RESET}}$ signal is input, the HALT mode is released. After branching to the reset vector address in the same manner as the ordinary reset operation, and program execution is started again.

★ **Figure 16-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input**



Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 12 MHz.

Table 16-2. Operation After Release of HALT Mode

Releasing Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Executes next address instruction.
	0	0	1	×	Executes interrupt servicing.
	0	1	0	1	Executes next address instruction.
	0	1	×	0	
	0	1	1	1	Executes interrupt servicing.
	1	×	×	×	Retains HALT mode.
Non-maskable interrupt request	–	–	×	×	Executes interrupt servicing.
$\overline{\text{RESET}}$ input	–	–	×	×	Executes reset processing.

×: don't care

16.2.2 STOP mode

(1) Setting and operation status of STOP mode

The STOP mode is set by executing the STOP instruction.

Cautions 1. When the STOP mode is set, the X2 pin is internally pulled up to V_{DD1} to suppress the current leakage of the crystal oscillator block. Therefore, do not use the STOP mode in a system where the external clock is used as the system clock.

2. Because the standby mode can be released by an interrupt request signal, the standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. When the STOP mode is set, therefore, the HALT mode is set immediately after the STOP instruction has been executed, the wait time set by the oscillation stabilization time select register (OSTS) elapses, and then an operation mode is set.

The following table shows the operation status in the STOP mode.

Table 16-3. Operation Status in STOP Mode

Item		Operation Status
Clock generator		Oscillation stopped.
CPU		Stops operation.
Output port (output latch)		Retains previous status immediately before STOP instruction execution.
16-bit timer/event counter		Operable only when TI000 or TI001 is selected as count clock.
8-bit timer/event counter		Operable only when TI50, TI51, or TI52 is selected as count clock.
10-bit inverter control timer		Stops operation.
Watchdog timer		Stops operation.
Real-time output port		Operable when external trigger is used or when TI010, TI011, or TI52 is selected as count clock of timer/event counter.
A/D converter		Stops operation.
Serial interface		Stops operation.
External interrupt		Operable
Externally extended bus line	AD0 to AD7	High impedance
	ASTB	Low level
	$\overline{\text{WR}}$, $\overline{\text{RD}}$	High level
	$\overline{\text{WAIT}}$	High impedance

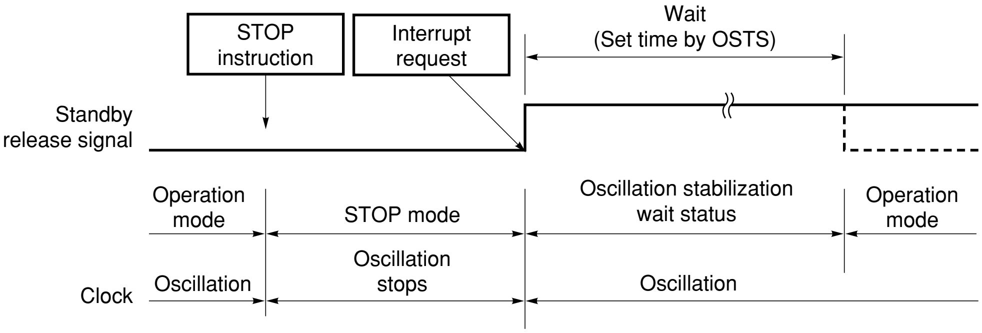
(2) Releasing STOP mode

The STOP mode can be released by the following two sources.

(a) Releasing by unmasked interrupt request

If an unmasked interrupt request is generated, the STOP mode can be released. If interrupt requests are enabled at this time, vectored interrupt servicing is performed, after the oscillation stabilization time has elapsed. If interrupt requests are in the acknowledgement disabled status, the instruction at the next address is executed.

Figure 16-4. Releasing STOP Mode by Interrupt Request

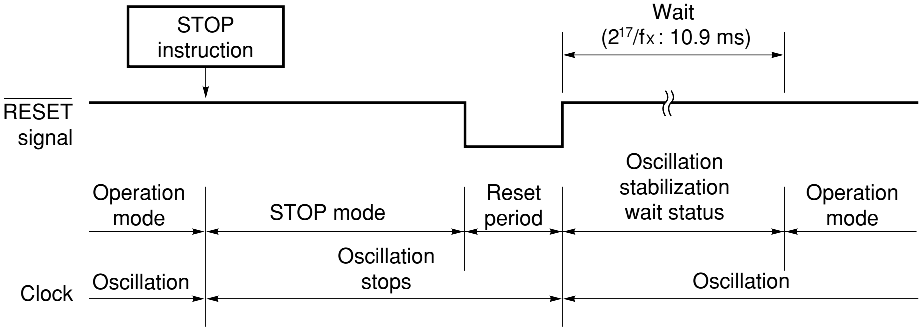


Remark The dotted lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Releasing by $\overline{\text{RESET}}$ input

If the $\overline{\text{RESET}}$ signal is input, the STOP mode is released. The reset operation is performed after the oscillation stabilization time has elapsed.

★ **Figure 16-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input**



Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 12 MHz.

Table 16-4. Operation After Release of STOP Mode

Releasing Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Executes next address instruction.
	0	0	1	×	Executes interrupt servicing.
	0	1	0	1	Executes next address instruction.
	0	1	×	0	
	0	1	1	1	Executes interrupt servicing.
	1	×	×	×	Retains STOP mode.
$\overline{\text{RESET}}$ input	–	–	×	×	Executes reset processing.

×: don’t care

CHAPTER 17 RESET FUNCTION

The reset signal can be effected by the following two methods.

(1) External reset input from $\overline{\text{RESET}}$ pin
(2) Internal reset by inadvertent program loop detection by watchdog timer

There is no functional difference between the external reset and internal reset, and execution of the program is started from addresses written to addresses 0000H and 0001H when the $\overline{\text{RESET}}$ signal is input.

The reset function is effected when a low-level signal is input to the $\overline{\text{RESET}}$ pin or when an overflow occurs in the watchdog timer. As a result, each hardware enters the status shown in Table 17-1. Each pin goes into a high-impedance state while the $\overline{\text{RESET}}$ signal is input, and during the oscillation stabilization time immediately after the reset function has been released.

When a high-level signal is input to the $\overline{\text{RESET}}$ pin, the reset function is released, and program execution is started after oscillation stabilization time (2^{17}/fx) has elapsed. The reset function effected by an overflow in the watchdog timer is automatically released after reset, and program execution is started after the oscillation stabilization time (2^{17}/fx) has elapsed (refer to **Figures 17-2** to **17-4**).

Cautions 1. Input a low-level signal to the $\overline{\text{RESET}}$ pin for 10 μs or longer to execute an external reset.
2. Oscillation of the system clock is stopped while the $\overline{\text{RESET}}$ signal is being input.
3. When releasing the STOP mode by the $\overline{\text{RESET}}$ input, the contents during the STOP mode are retained while the $\overline{\text{RESET}}$ signal is being input. However, the port pins go into a high-impedance state.

Figure 17-1. Reset Function Block Diagram

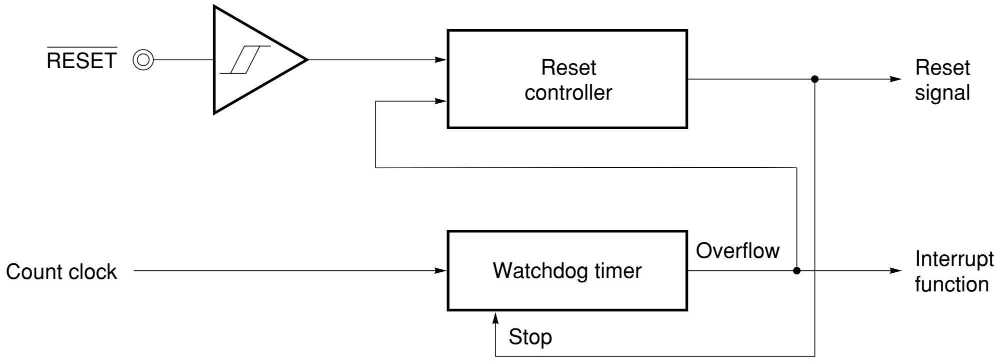


Figure 17-2. Reset Timing by $\overline{\text{RESET}}$ Input

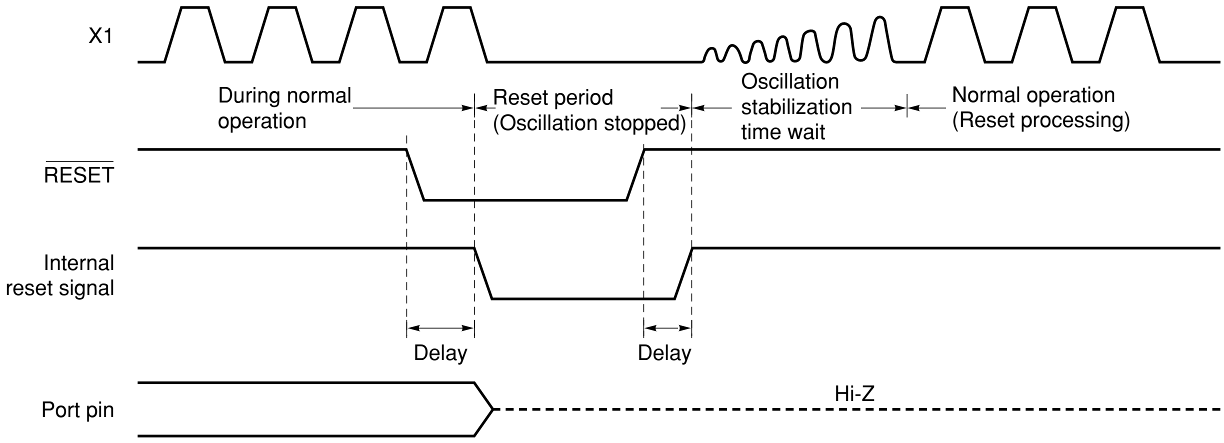

X1
During normal operation
Reset period (Oscillation stopped)
Oscillation stabilization time wait
Normal operation (Reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 17-3. Reset Timing by Overflow in Watchdog Timer

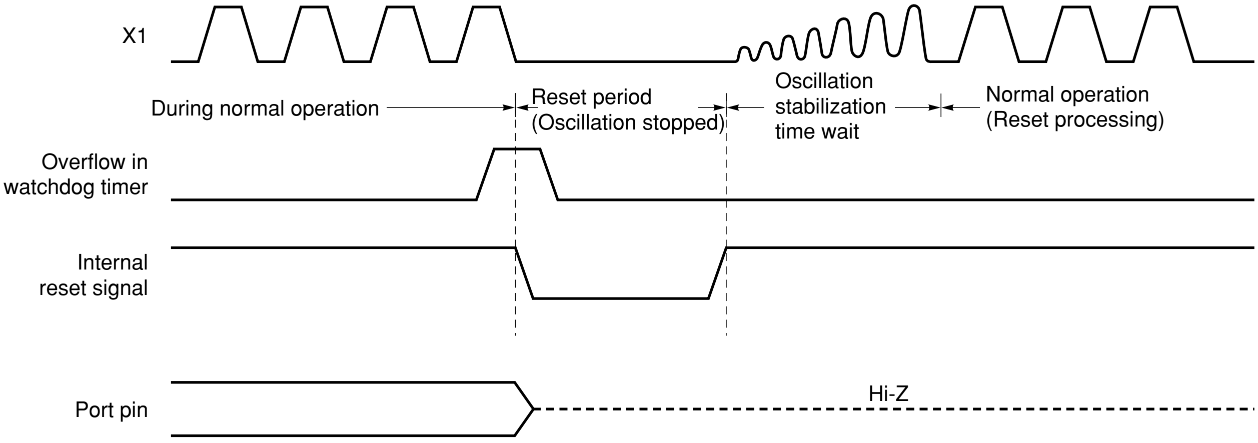

X1
During normal operation
Reset period (Oscillation stopped)
Oscillation stabilization time wait
Normal operation (Reset processing)
Overflow in watchdog timer
Internal reset signal
Port pin
Hi-Z

Figure 17-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

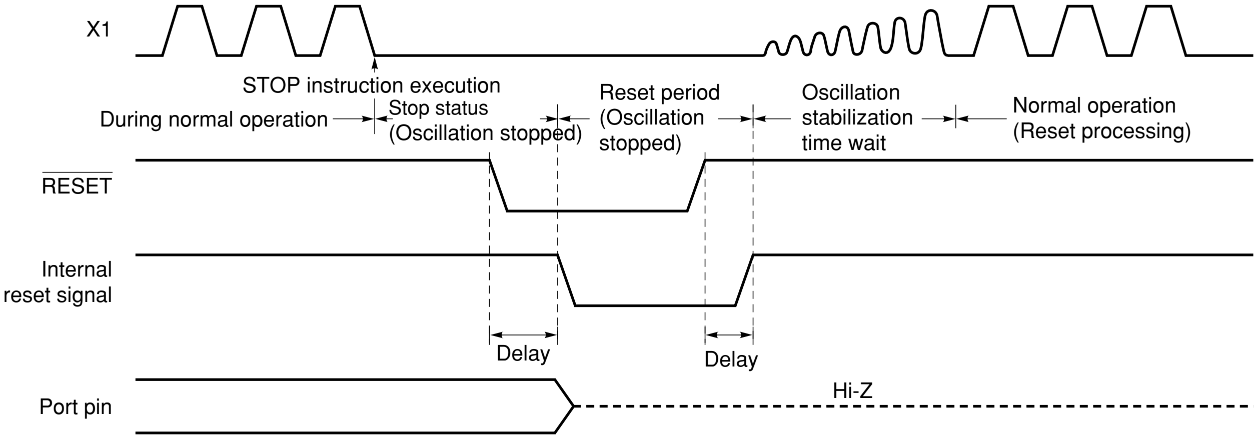

X1
STOP instruction execution
During normal operation
Stop status (Oscillation stopped)
Reset period (Oscillation stopped)
Oscillation stabilization time wait
Normal operation (Reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Table 17-1. Status of Each Hardware After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) are set
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port (output latch)	Ports 0 to 6 (P0 to P6)	00H
Port mode registers (PM0, PM2 to PM6)		FFH
Pull-up resistor option registers (PU0, PU2 to PU6)		00H
Processor clock control register (PCC)		04H
Memory expansion mode register (MEM)		00H
Memory expansion wait setting register (MM)		10H
Internal memory size select register (IMS)		CFH[Note 3]
Internal expansion RAM size select register (IXS)		0CH[Note 4]
Flash programming mode control register (FLPMC)		08H[Note 5]
Oscillation stabilization time select register (OSTS)		04H
Real-time output port	Mode registers (RTPM00, RTPM01)	00H
	Control registers (RTPC00, RTPC01)	00H
	DC control registers (DCCTL0, DCCTL1)	00H
	Buffer registers (RTBL00, RTBH00, RTBL01, RTBH01)	00H

Notes **1.** Only the contents of the PC among hardware become undefined during reset input and oscillation stabilization time wait. The other statuses do not differ from those after reset.

2. If the reset signal is input in the standby mode, the status before reset is retained even after reset.

3. Set the following value before operating each device even though the initial value is CFH.

μPD780982: C4H
μPD780983: C6H
μPD780984: C8H
μPD780986: CCH
μPD780988: CFH (No need to change the set value of IMS because the initial IMS value of the μPD780988 is CFH).
μPD78F0988A: Values corresponding to those of mask ROM versions

4. Set the following value before operating each device even though the initial value is 0CH.

μPD780982, 780983, 780984: 0CH (No need to change the set value of IXS because the initial IXS value of the μPD780982, 780983, 780984 are set to 0CH).
μPD780986, 780988: 0AH
μPD78F0988A: Values corresponding to those of mask ROM versions

5. Bit 2 changes according to V_{PP} voltage level.

Table 17-1. Status of Each Hardware After Reset (2/2)

Hardware		Status After Reset
10-bit inverter control timer	Compare registers (CM0 to CM2)	0000H
	Compare register (CM3)	00FFH
	Buffer registers (BFCM0 to BFCM2)	0000H
	Buffer register (BFCM3)	00FFH
	Dead-time reload register (DTIME)	FFH
	Control register (TMC7)	00H
	Mode register (TMM7)	00H
16-bit timer/event counter	Timer counters (TM00, TM01)	0000H
	Capture/compare control registers (CRC00, CRC01)	00H
	Capture/compare registers (CR000, CR010, CR001, CR011)	Undefined
	Prescaler mode registers (PRM00, PRM01)	00H
	Mode control registers (TMC00, TMC01)	00H
	Timer output control registers (TOC00, TOC01)	00H
8-bit timer/event counter	Timer counters (TM50 to TM52)	00H
	Compare registers (CR50 to CR52)	Undefined
	Clock select registers (TCL50 to TCL52)	00H
	Mode control registers (TMC50 to TMC52)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H
Serial interface	Asynchronous serial interface mode registers (ASIM00, ASIM01)	00H
	Asynchronous serial interface status registers (ASIS00, ASIS01)	00H
	Transmit shift registers (TXS00, TXS01)	FFH
	Receive buffer registers (RXB00, RXB01)	FFH
	Baud rate generator control registers (BRGC00, BRGC01)	00H
	Shift register (SIO3)	Undefined
	Mode register (CSIM3)	00H
A/D converter	Mode register (ADM0)	00H
	Conversion result register (ADCR0)	Undefined
	Analog input channel specification register (ADS0)	00H
Interrupt	Request flag registers (IF0L, IF0H, IF1L)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable registers (EGP, EGP5)	00H
	External interrupt falling edge enable registers (EGN, EGN5)	00H

CHAPTER 18 μPD78F0988A

The μPD78F0988A replaces the on-chip mask ROM of the μPD780988 with flash memory, which can be written, deleted, and rewritten while mounted on the board. Table 18-1 lists the differences between the μPD78F0988A and the mask ROM versions.

Table 18-1. Differences Between μPD78F0988A and Mask ROM Versions

Item	μPD78F0988A	Mask ROM Versions
Internal ROM type	Flash memory	Mask ROM
Internal ROM capacity	60 KB**Note 1**	μPD780982: 16 KB μPD780983: 24 KB μPD780984: 32 KB μPD780986: 48 KB μPD780988: 60 KB
Internal expansion RAM capacity	1,024 bytes**Note 2**	μPD780982, 780983, 780984: None μPD780986, 780988: 1,024 bytes
TEST pin	Not available	Available
V_{PP} pin	Available	Not available

Notes 1. By using the internal memory size switching register (IMS), the flash memory capacity can be set to the same capacity as the memory in the mask ROM versions.

2. By using the internal expansion RAM size switching register (IXS), the flash memory capacity can be set to the same capacity as the memory in the mask ROM versions.

Caution There are differences in noise immunity and noise radiation between the flash memory versions 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.

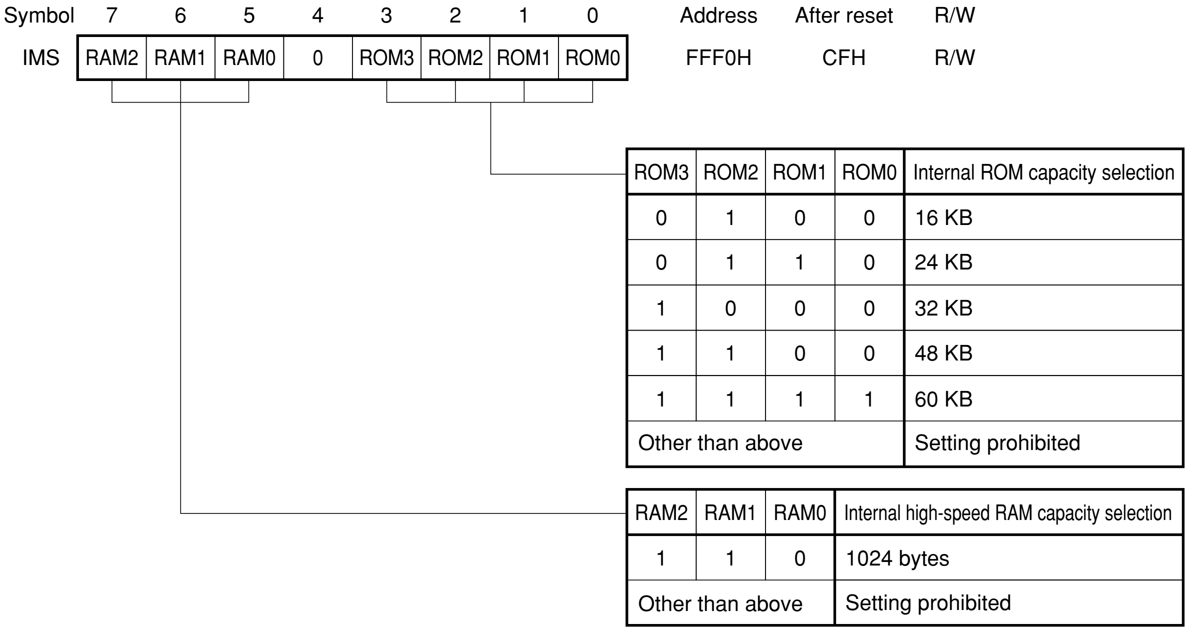
18.1 Internal Memory Size Switching Register

For the μPD78F0988A, it is possible to select the capacity of the internal memory using the internal memory size switching register (IMS). By setting IMS, the internal memory of the μPD78F0988 can be mapped identically to that of a mask ROM version.

IMS is set by an 8-bit memory manipulation instruction.

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

Figure 18-1. Format of Memory Size Switching Register



Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	CFH	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	1	0	0	16 KB
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

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1024 bytes
Other than above			Setting prohibited

The values set to IMS in order to obtain the same memory map as mask ROM versions are shown in Table 18-2.

Table 18-2. Set Values of Memory Size Switching Register

Applicable Mask ROM Versions	Set Value of IMS
μPD780982	C4H
μPD780983	C6H
μPD780984	C8H
μPD780986	CCH
μPD780988	CFH

Caution When mask ROM versions are used, IMS should be set to the values shown in Table 18-2. The setting value for the μPD780988 is CFH, so it is not necessary to change the initial value.

18.2 Internal Expansion RAM Size Switching Register

For the μPD78F0988A, it is possible to select the capacity of the internal expansion RAM using the internal expansion RAM size switching register (IXS). By setting IXS, the same memory map as mask ROM versions with different internal expansion RAM capacities is possible.

IXS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 0CH.

Figure 18-2. Format of Internal Expansion RAM Size Switching Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	FFF4H	0CH	R/W

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
0	1	0	1	0	1024 bytes
0	1	1	0	0	No internal expansion RAM
Other than above					Setting prohibited

The values set to IXS in order to obtain the same memory map as mask ROM versions are shown in Table 18-3.

Table 18-3. Set Values of Internal Expansion RAM Size Switching Register

Applicable Mask ROM Versions	Set Value of IXS
μPD780982	0CH
μPD780983	
μPD780984	
μPD780986	0AH
μPD780988	

Caution When mask ROM versions are used, IXS should be set to the values shown in Table 18-3. The setting value for the μPD780982, 780983, and 780984 is 0CH, so it is not necessary to change the initial value.

★ 18.3 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)) to the target system with the μPD78F0988A mounted (on-board). A flash memory writing adapter (program adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the program adapter are the products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Write or erase the flash memory under the following conditions.

- Expanded-specification products
  - 4.5 V ≤ V_{DD} ≤ 5.5 V: f_X = 10.0 MHz or lower
  - 3.0 V ≤ V_{DD} < 4.5 V: f_X = 8.38 MHz or lower
- Conventional products
  - 4.0 V ≤ V_{DD} ≤ 5.5 V: f_X = 8.38 MHz or lower

Refer to **CHAPTER 20 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)** and **CHAPTER 21 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)** for details of conditions other than the above.

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilities low-quantity, varied model production
- Easy data adjustment when starting mass production

18.3.1 Programming environment

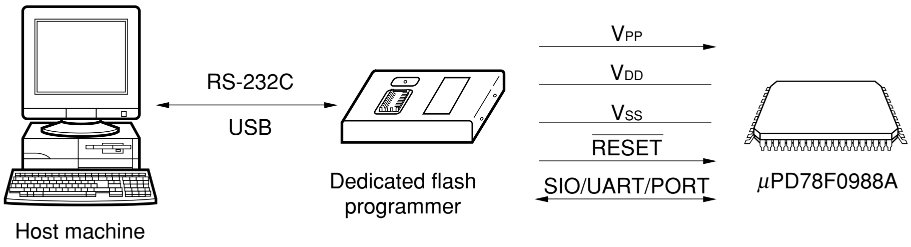
The following shows the environment required for μPD78F0988A flash memory programming.

When Flashpro III or Flashpro IV is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer to the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 18-3. Environment for Writing Program to Flash Memory



18.3.2 Communication mode

Use the communication mode shown in Table 18-4 to perform communication between the dedicated flash programmer and μPD78F0988A.

Table 18-4. Communication Mode List

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU CLOCK	Flash Clock	Multiple Rate		
3-wire serial I/O (SIO3)	SIO ch-0 (3-wired, sync.)	100 Hz to 1.25 MHz **Note 2**	Any	1 to 10 MHz **Note 2**	1.0	SI/P52 SO/P53 $\overline{\text{SCK}}$/P51	0
3-wire serial I/O (SIO3) with handshake	SIO ch-3 + handshake					SI/P52 SO/P53 $\overline{\text{SCK}}$/P51 P50 (HS)	3
UART (UART00)	UART ch-0 (Async.)	4,800 to 76,800 bps **Notes 2, 4**	Any	1 to 10 MHz **Note 2**	1.0	RxD00/P20 TxD00/P21	8
Pseudo 3-wire serial I/O[Note 3]	PORT A (Pseudo-3-wired)	100 Hz to 1 kHz[Note 2]	Any	1 to 10 MHz **Note 2**	1.0	P24/TI50/TO50 (serial data input) P25/TI51/TO51 (serial data output) P26/TI52/TO52 (serial clock input)	12

Notes 1. Selection items for TYPE settings on the dedicated flash programmer (Flashpro III/Flashpro IV).

2. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 20 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)** and **CHAPTER 21 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**.

3. Serial transfer is executed by controlling the port with software.

4. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

Figure 18-4. Communication Mode Selection Format

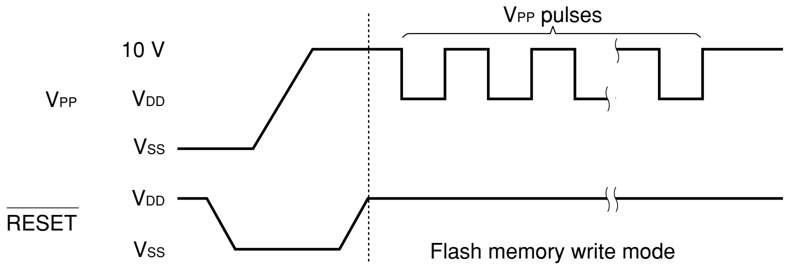
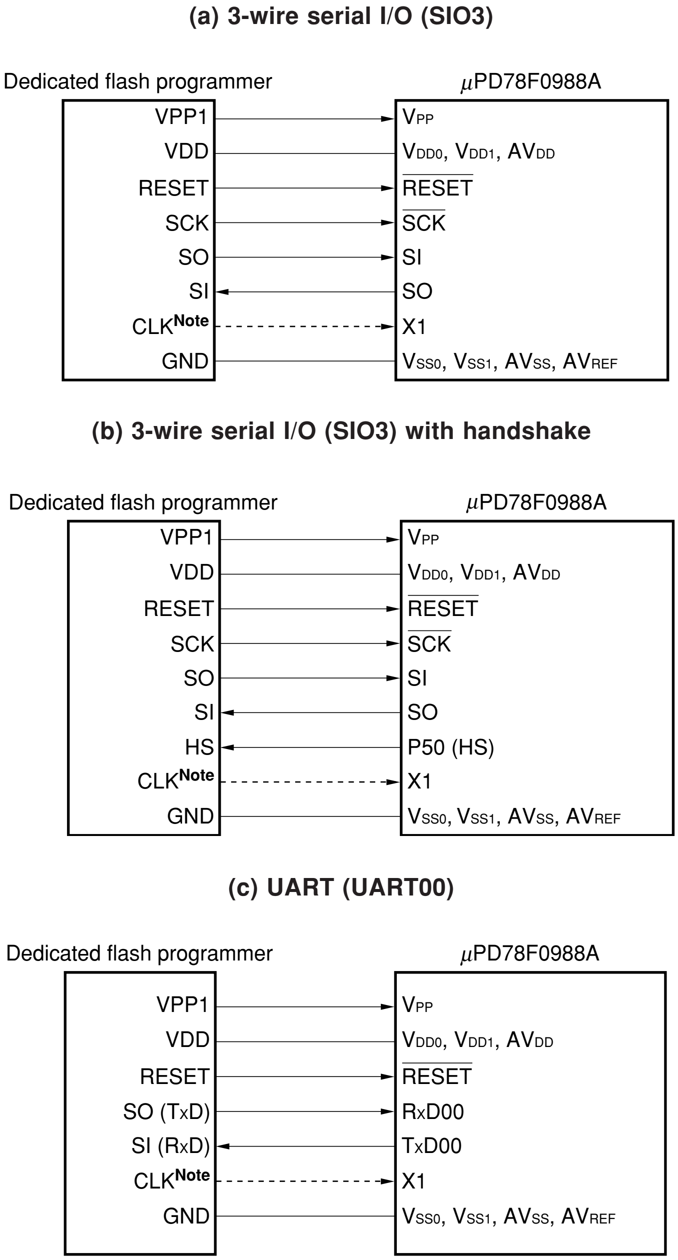


Figure 18-5. Example of Connection with Dedicated Flash Programmer (1/2)



(a) 3-wire serial I/O (SIO3)

(b) 3-wire serial I/O (SIO3) with handshake

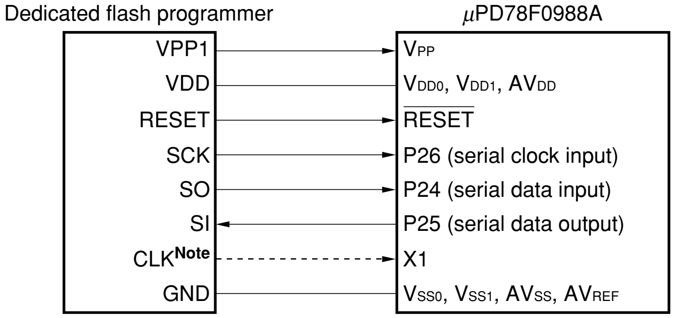
(c) UART (UART00)

Note Connect this pin when the system clock is supplied from the dedicated flash programmer. If a resonator is already connected to the X1 pin, the CLK pin does not need to be connected.

Caution The V_{DD0} and V_{DD1} pins, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. Before using the power supply connected to the V_{DD0} and V_{DD1} pins, supply voltage before starting programming.

Figure 18-5. Example of Connection with Dedicated Flash Programmer (2/2)

(d) Pseudo 3-wire serial I/O



Note Connect this pin when the system clock is supplied from the dedicated flash programmer. If a resonator is already connected to the X1 pin, the CLK pin does not need to be connected.

Caution The V_{DD0} and V_{DD1} pins, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. Before using the power supply connected to the V_{DD0} and V_{DD1} pins, supply voltage before starting programming.

If Flashpro III/Flashpro IV is used as a dedicated flash programmer, the following signals are generated for the μPD78F0988A. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 18-5. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	SIO3	SIO3 (HS)	UART00	Pseudo 3-Wired
VPP1	Output	Write voltage	V_{PP}	◎	◎	◎	◎
VPP2	–	–	–	×	×	×	×
VDD	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD0}, V_{DD1}, AV_{DD}	◎ Note	◎ Note	◎ Note	◎ Note
GND	–	Ground	V_{SS0}, V_{SS1}, AV_{SS}, AV_{REF}	◎	◎	◎	◎
CLK	Output	Clock output	X1	○	○	○	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎	◎
SI (RxD)	Input	Reception signal	SO/TxD00/P25	◎	◎	◎	◎
SO (TxD)	Output	Transmit signal	SI/RxD00/P24	◎	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK}}$/P26	◎	◎	×	◎
HS	Input	Handshake signal	P50 (HS)	×	◎	×	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the signal is supplied on the target board, pin need not be connected.
× : Pin need not be connected.

18.3.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and flash memory programming mode may be required in some cases.

<V$_{PP}$ pin>

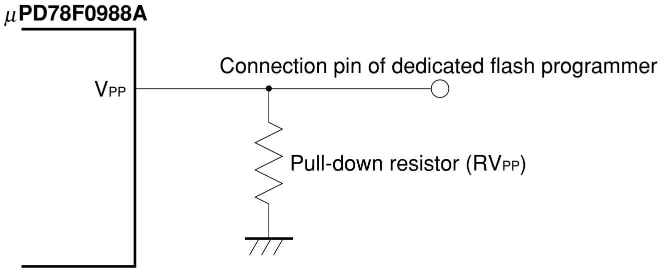
In normal operation mode, input 0 V to the V$_{PP}$ pin. In flash memory programming mode, a write voltage of 10.0 V (TYP.) is supplied to the V$_{PP}$ pin, so perform the following.

(1) Connect a pull-down resistor (RV$_{PP}$ = 10 kΩ) to the V$_{PP}$ pin.

(2) Use the jumper on the board to switch the V$_{PP}$ pin input to either the writer or directly to GND.

A V$_{PP}$ pin connection example is shown below.

Figure 18-6. V$_{PP}$ Pin Connection Example


μPD78F0988A
Connection pin of dedicated flash programmer
VPP
Pull-down resistor (RVPP)

<Serial interface pin>

The following shows the pins used by the serial interface.

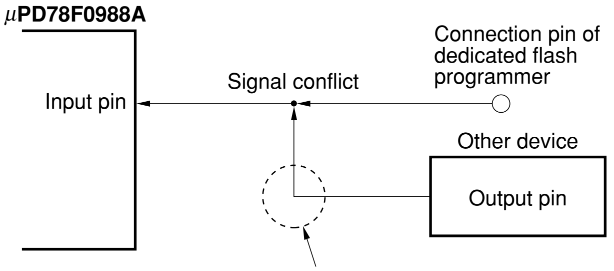
Serial Interface	Pins Used
3-wire serial I/O (SIO3)	SI, SO, $\overline{\text{SCK}}$
3-wire serial I/O (SIO3) with handshake	SI, SO, $\overline{\text{SCK}}$, P50 (HS)
UART (UART00)	RxD00, TxD00
Pseudo 3-wire serial I/O	P24, P25, P26

When connecting the dedicated flash programmer to a serial interface pin that is connected to another device on-board, signal conflict or abnormal operation of the other devices may occur. Care must therefore be taken with such connections.

(1) Signal conflict

If the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a signal conflict occurs. To prevent this, isolate the connection with the other device or set the other device to the output high impedance status.

Figure 18-7. Signal Conflict (Input Pin of Serial Interface)

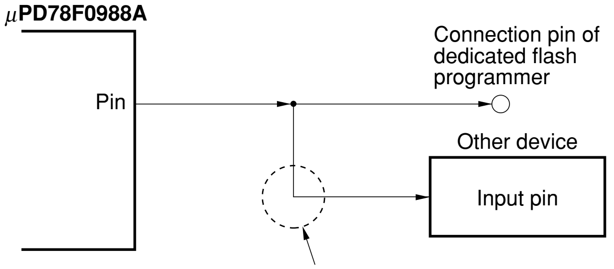


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict, therefore, isolate the signal of the other device.

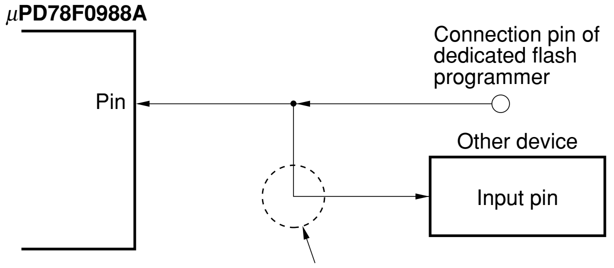
(2) Abnormal operation of other device

If the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), a signal is output to the device, and this may cause an abnormal operation. To prevent this abnormal operation, isolate the connection with the other device or set so that the input signals to the other device are ignored.

Figure 18-8. Abnormal Operation of Other Device



If the signal output by the μPD78F0988A affects another device in the flash memory programming mode, isolate the signals of the other device.

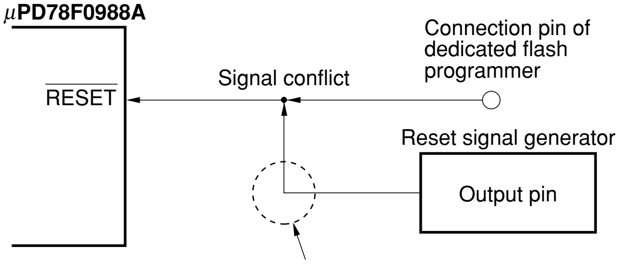


If the signal output by the dedicated flash programmer affects another device in the flash memory programming mode, isolate the signals of the other device.

<$\overline{\text{RESET}}$ pin>

If the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin connected to the reset signal generator on-board, a signal conflict occurs. To prevent this, isolate the connection with the reset signal generator. If the reset signal is input from the user system in the flash memory programming mode, a normal programming operation cannot be performed. Therefore, do not input reset signals from other than the dedicated flash programmer.

Figure 18-9. Signal Conflict ($\overline{\text{RESET}}$ Pin)


μPD78F0988A
$\overline{\text{RESET}}$
Signal conflict
Connection pin of dedicated flash programmer
Reset signal generator
Output pin

The signal output by the reset signal generator and the signal output from the dedicated flash programmer conflict in the flash memory programming mode, so isolate the signal of the reset signal generator.

<Port pins>

When the μPD78F0988A enter the flash memory programming mode, all the pins other than those that communicate in flash programmer are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD0} or V_{SS0} via a resistor.

<Oscillator>

When using the on-board clock, connect X1 and X2 as required in the normal operation mode.

When using the clock output of the flash programmer, connect it directly to X1, disconnecting the main oscillator on-board, and leave the X2 pin open.

<Power supply>

To use the power output from the flash programmer, connect the V_{DD0} and V_{DD1} pins to VDD of the flash programmer, and the V_{SS0} and V_{SS1} pins to GND of the flash programmer.

To use the on-board power supply, make connections that accord with the normal operation mode. However, because the voltage is monitored by the flash programmer, be sure to connect VDD of the flash programmer.

Supply the same power as in the normal operation mode to the other power supply pins (AV_{DD} and AV_{SS}).

<Other pins>

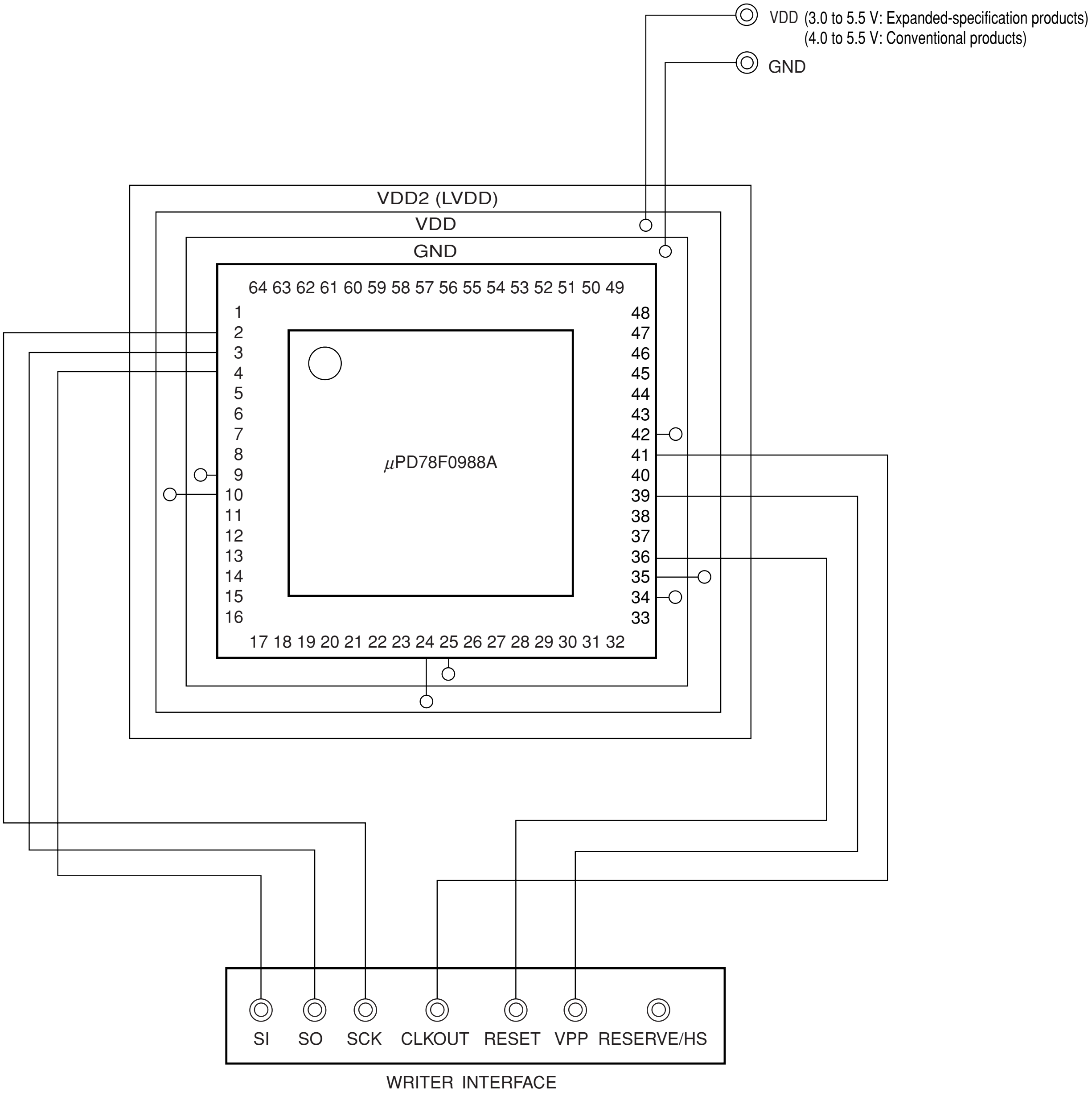
Process the other pins (TO70 to TO75, AV_{REF}, and TEST) in the same manner as in the normal operation mode.

18.3.4 Connection of adapter for flash writing

The following figures show the examples of recommended connection when the adapter for flash writing is used.

Figure 18-10. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3)

(a) 64-pin plastic QFP (14 × 14), 64-pin plastic LQFP (14 × 14)



(b) 64-pin plastic SDIP (19.05 mm (750))

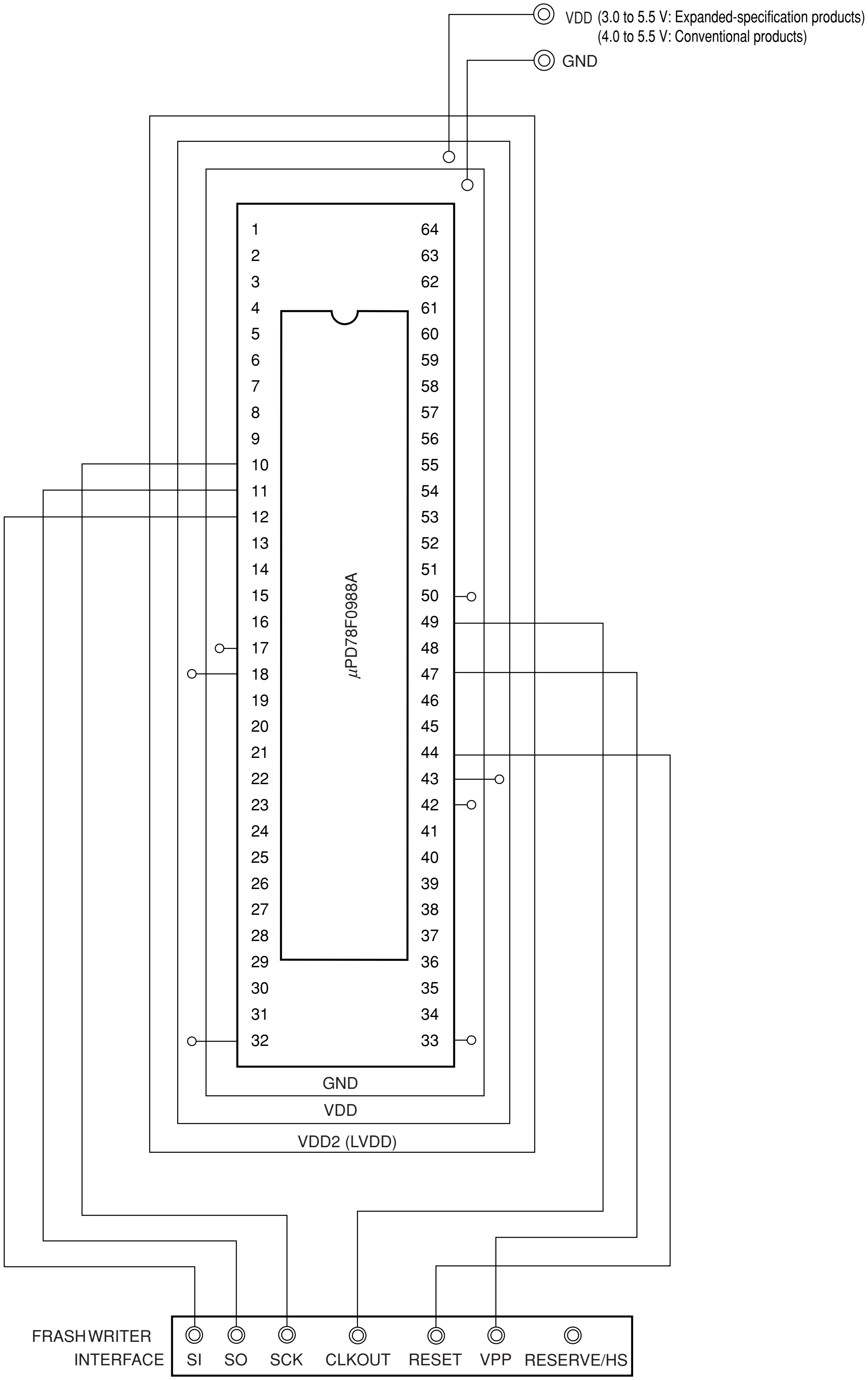
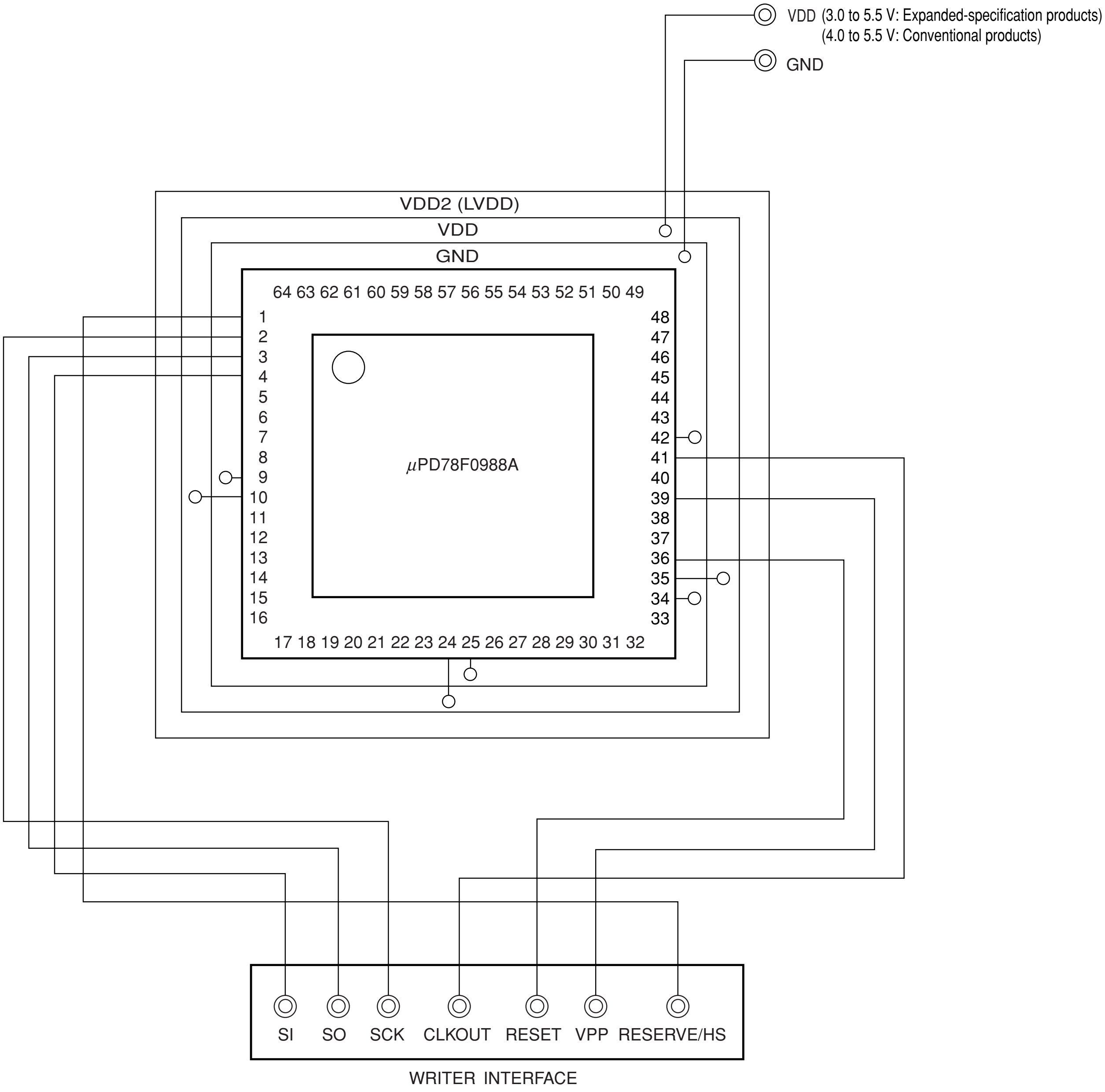


Figure 18-11. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3) with Handshake

(a) 64-pin plastic QFP (14 × 14), 64-pin plastic LQFP (14 × 14)



(b) 64-pin plastic SDIP (19.05 mm (750))

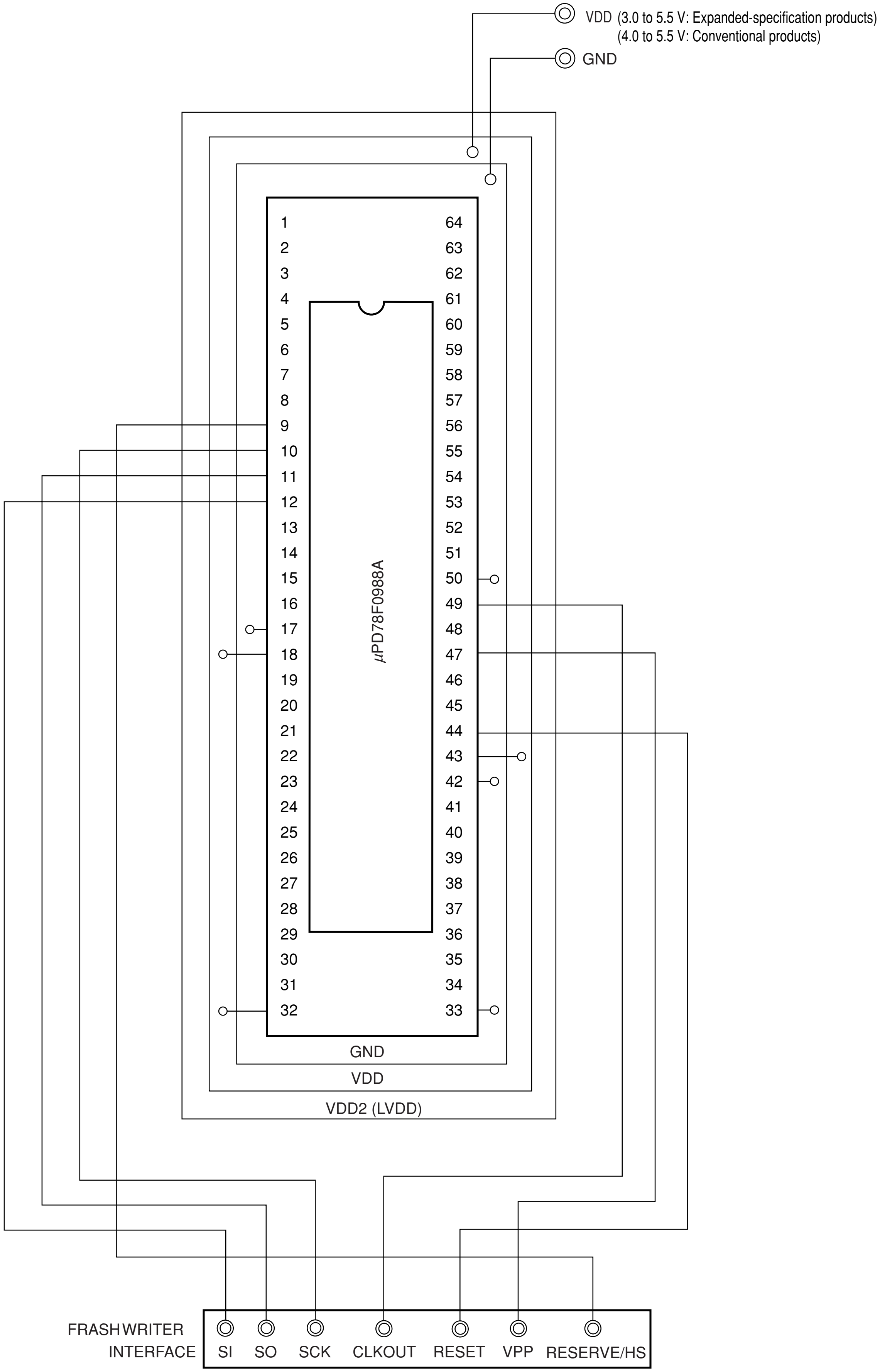
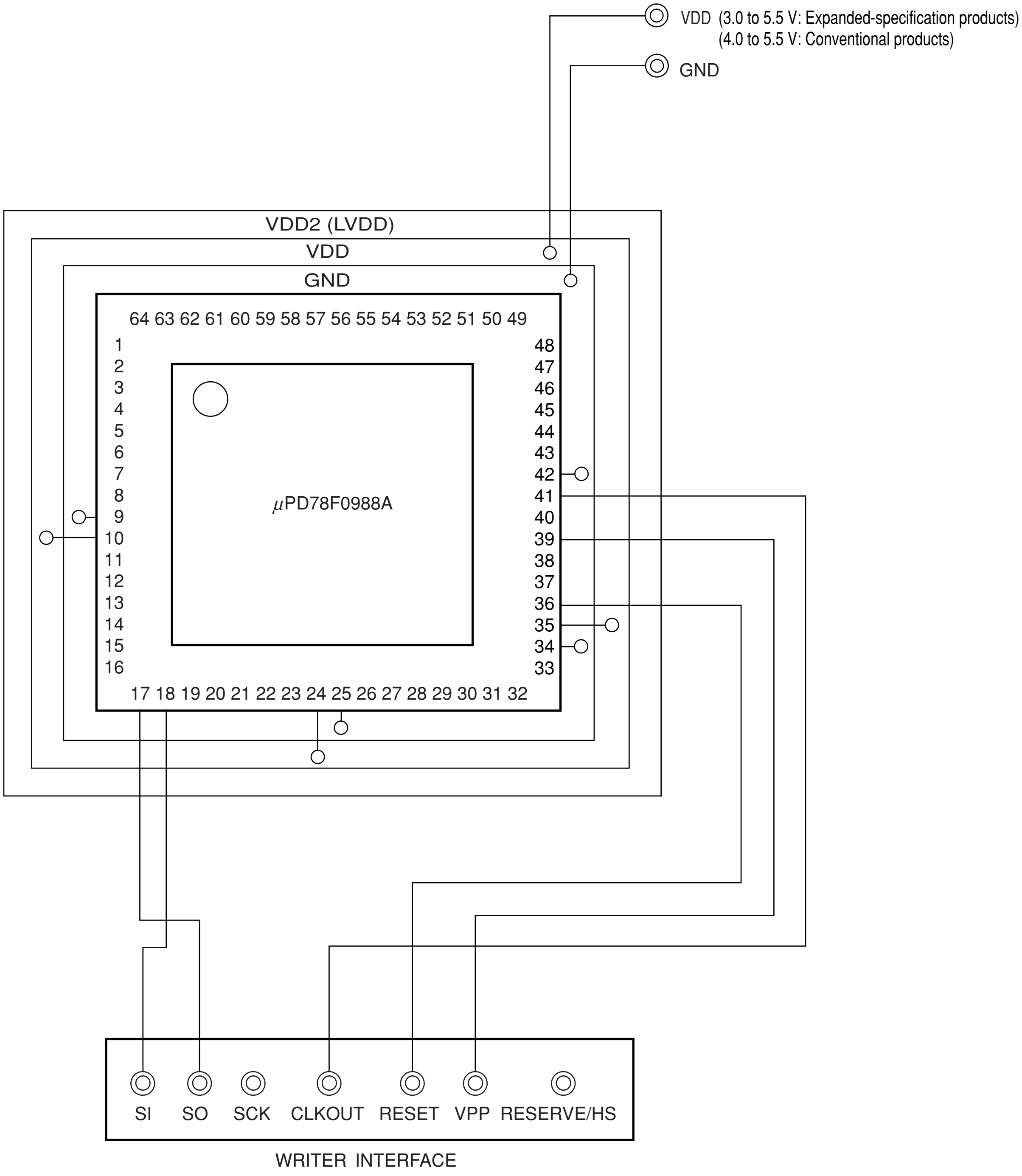


Figure 18-12. Wiring Example for Flash Writing Adapter with UART (UART00)

(a) 64-pin plastic QFP (14 × 14), 64-pin plastic LQFP (14 × 14)



(b) 64-pin plastic SDIP (19.05 mm (750))

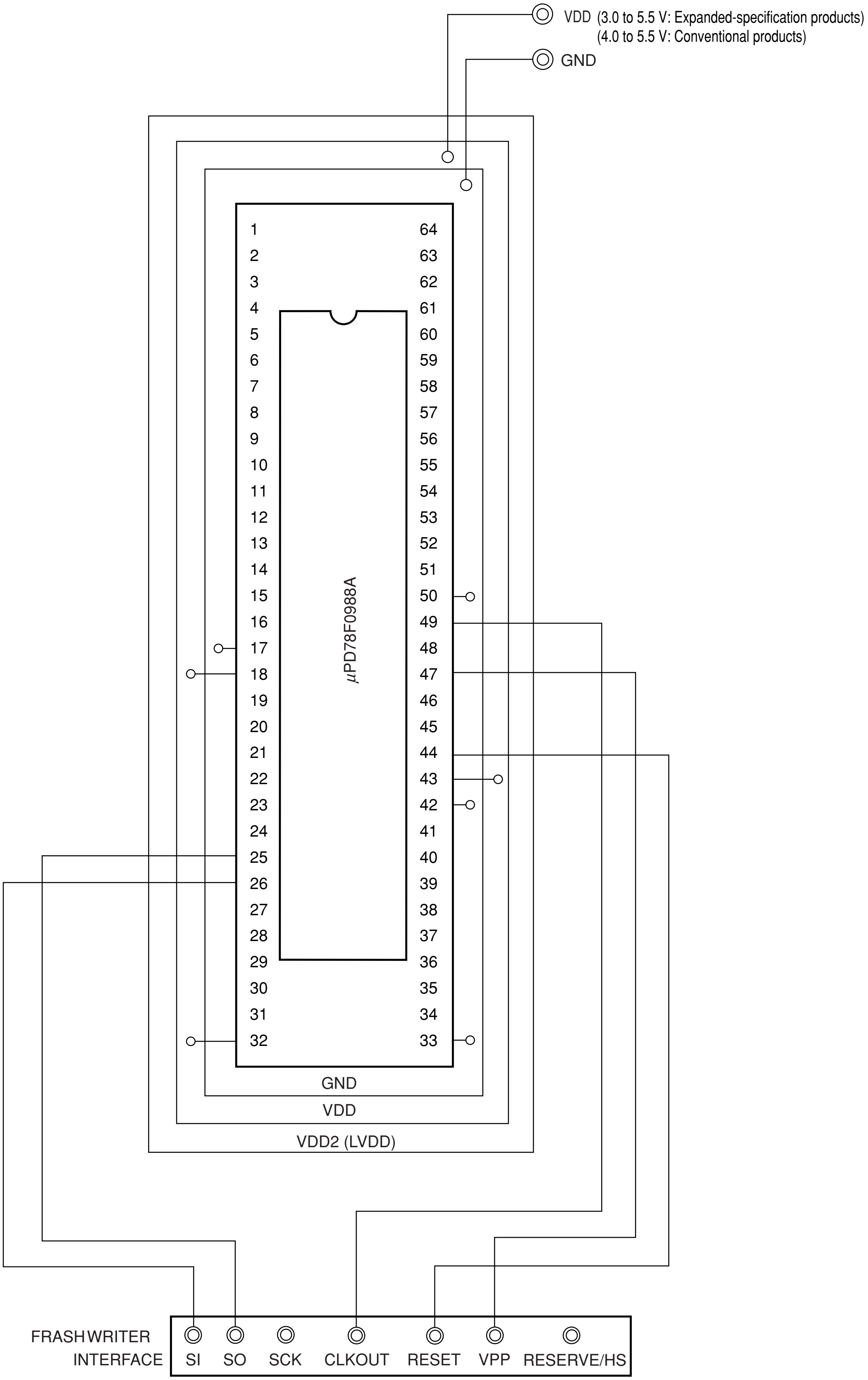
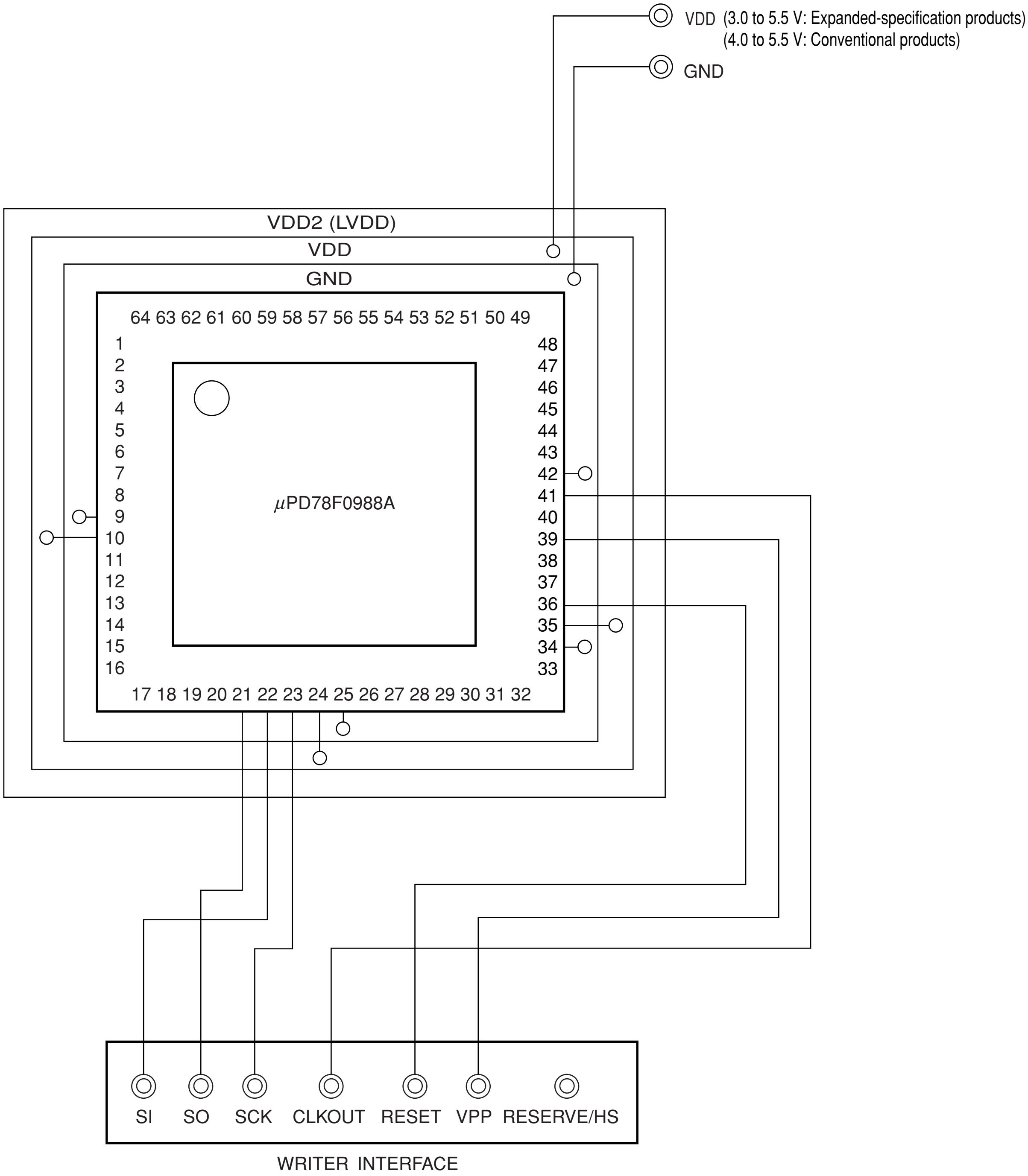
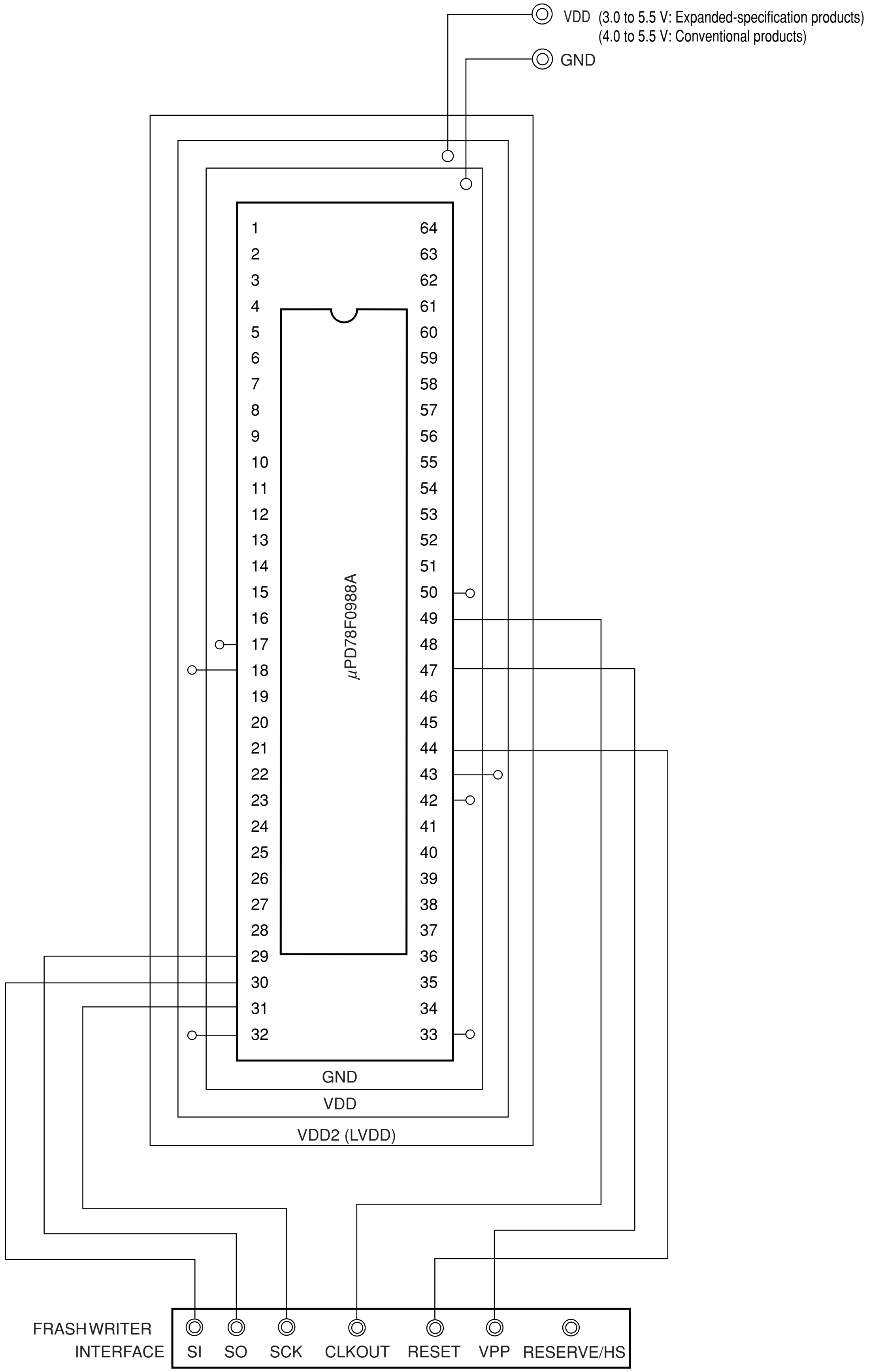


Figure 18-13. Wiring Example for Flash Writing Adapter with Pseudo 3-Wire Serial I/O

(a) 64-pin plastic QFP (14 × 14), 64-pin plastic LQFP (14 × 14)



(b) 64-pin plastic SDIP (19.05 mm (750))



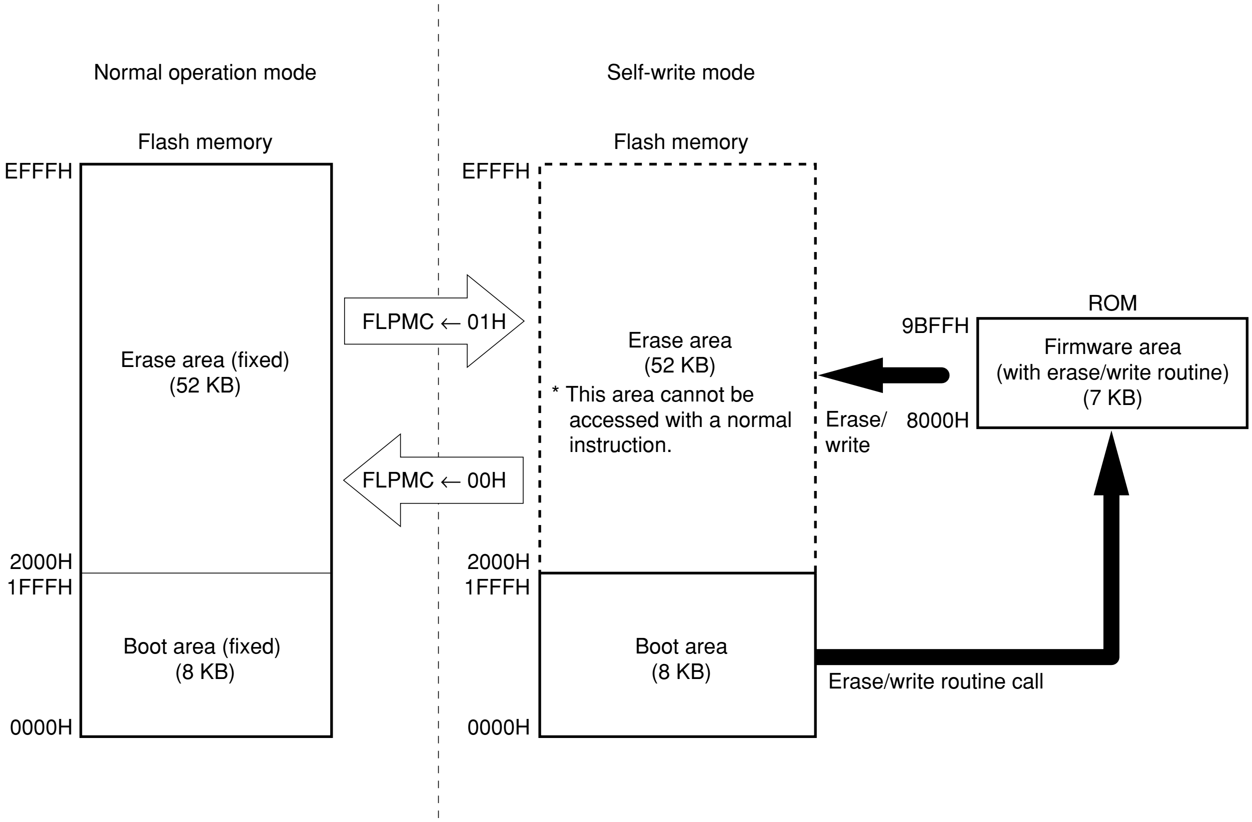
18.4 Flash Memory Programming by Self Write

With the μPD78F0988A, it is possible to rewrite the flash memory by a program.

18.4.1 Flash memory configuration

The configuration of the flash memory is shown in Figure 18-14.

Figure 18-14. Flash Memory Configuration



18.4.2 Flash programming mode control register

The flash programming mode control register (FLPMC) is a register for checking the operation mode selection and V_{PP} pin status.

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

$\overline{\text{RESET}}$ input clears this register to 08H.

Figure 18-15. Format of Flash Programming Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
FLPMC	0	0	0	0	1	VPP	0	FLSPM0	FF89H	08H[Note 1]	R/W[Note 2]

VPP	V_{PP} pin voltage status
0	The voltage required for erase/write is not applied to V_{PP} pin.
1	Voltage greater than that of V_{DD} pin is applied to V_{PP} pin.

FLSPM0	Operation mode selection
0	Normal operation mode
1	Self-write mode

Notes 1. Bit 2 changes depending on the level of V_{PP}.

2. Bit 2 is read only.

Cautions 1. The VPP bit indicates the status of the voltage applied to the V_{PP} pin. If the VPP bit is 0, the voltage required for erase/write is not being applied. However, even if the VPP bit is 1, it does not necessarily mean that the voltage required for erase/write is being applied. Set the hardware so that the voltage required for erase/write is applied to the V_{PP} pin.

Also, if using software in addition to hardware to check that the voltage required for erase/write is being applied, use an external hardware detection circuit and its output.

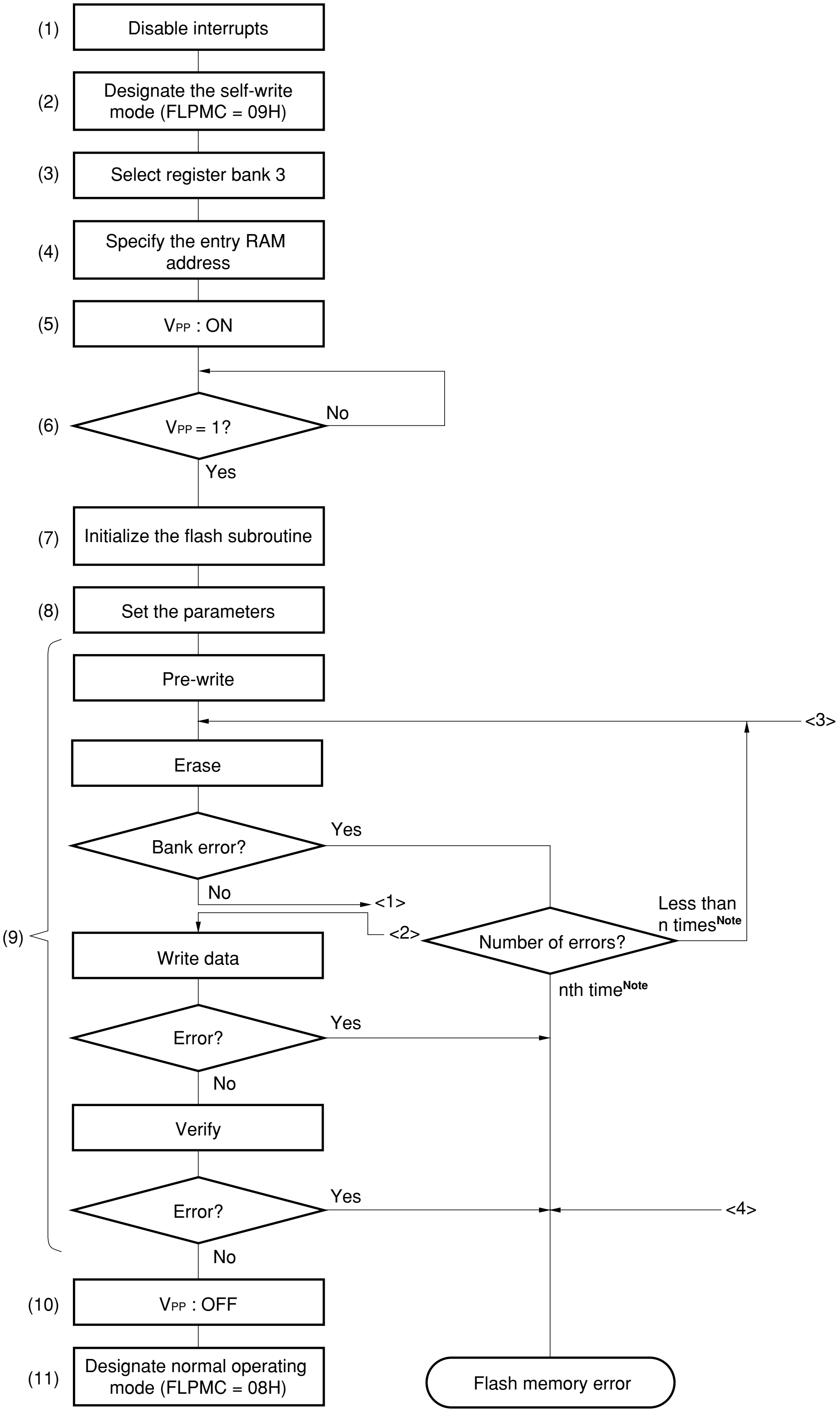
2. The initial values of bits 1 and 3 to 7 should not be changed.

18.4.3 Self-write procedure

The procedure for performing self write is shown below (see **Figure 18-16**).

(1) Disable interrupts.
(2) Designate the self-write mode (FLPMC = 09H).
(3) Select register bank 3.
(4) Specify the start address of the entry RAM for the HL register.
(5) V_{PP}: ON (ON signal for voltage IC)
(6) Check the V_{PP} level.
(7) Initialize the flash subroutine.
(8) Set the parameters.
(9) Control the flash memory (erase, write, etc.).
(10) V_{PP}: OFF (OFF signal for voltage IC)
(11) Designate the normal operating mode (FLPMC = 08H).

Figure 18-16. Self Programming Flowchart (1/2)



Note Differs depending on the user program.

Remark For <1> to <4>, refer to the following flowchart.

Figure 18-16. Self Programming Flowchart (2/2)

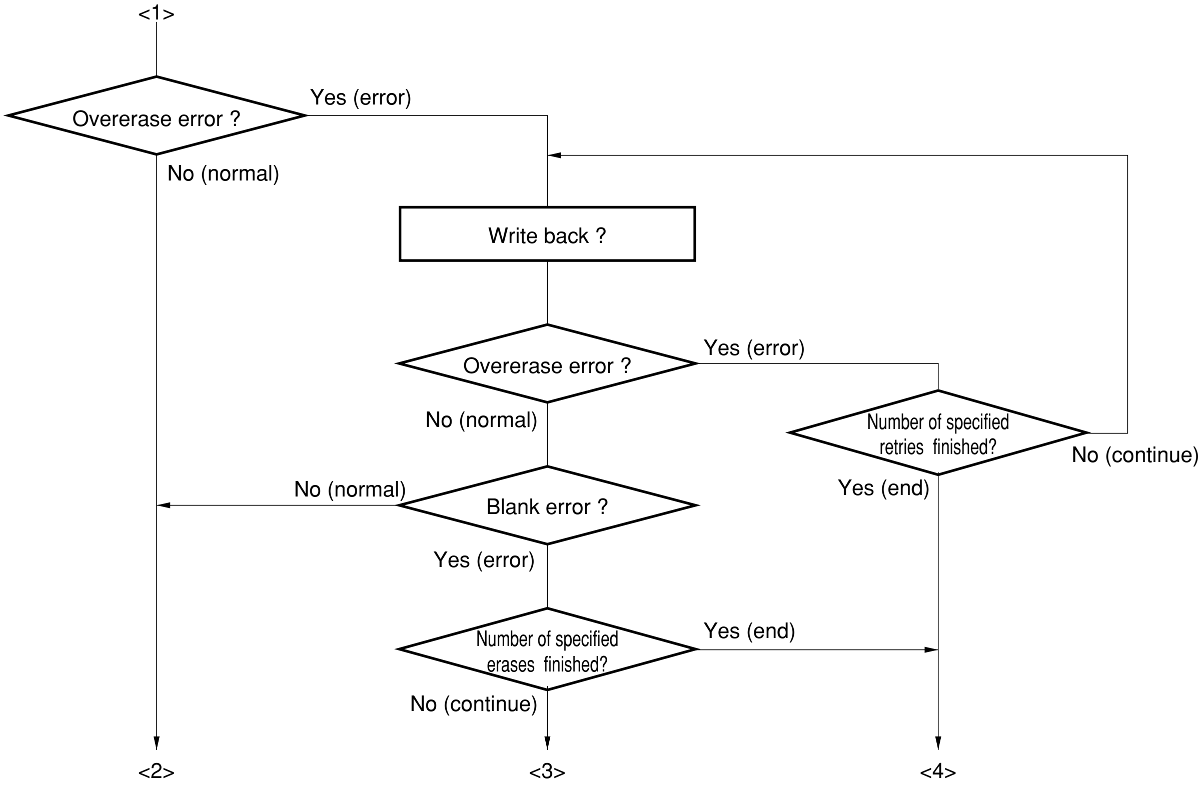
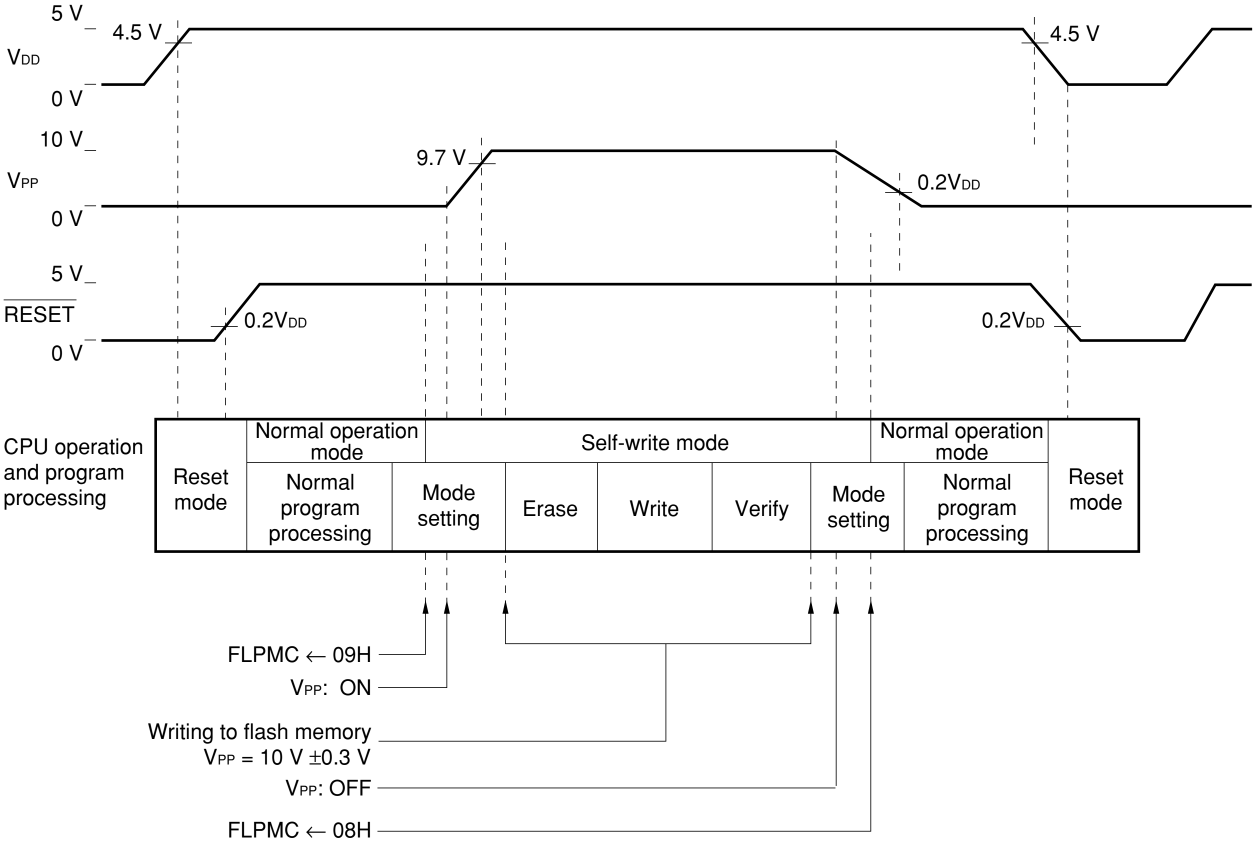


Figure 18-17. Self-Write Timing



18.4.4 CPU resources

The CPU resources used during self write are as follows.

- Register bank: BANK3 (8 bytes)
  B register: Status flag
  C register: Function number
  HL register: Entry RAM area starting address

- Stack area: Maximum 16 bytes

- Write data storage area: 1 to 256 bytes

- Entry RAM area: 32 bytes
  RAM area used by the self-write subroutines.
  Can be specified by the user using the HL register.

- Status flag

7	6	5	4	3	2	1	0
Parameter setting error	—	—	Verify error	Write error	Erase error	Blank check error	—

18.4.5 Entry RAM area

A description of the entry RAM area is shown in Table 18-6.

Table 18-6. Entry RAM Area

Offset Value	Description
+0	Reserved area (1 byte)
+1	Reserved area (1 byte)
+2 and +3	Flash memory start address (2 bytes)
+4 and +5	Reserved area (2 bytes)
+6	No. of bytes written in flash memory (1 byte)
+7	Write time data (1 byte)
+8 to +10	Erase time data (3 bytes)
+11 to +13	Writeback time data (3 bytes)
+14 and +15	Write data storage buffer starting address (2 bytes)
+16 and +17	Total number of blocks and areas (2 bytes)
+18 ⋮	Reserved area (15 bytes)

Example When the value of the HL register of register bank 3 is 0FD00H
0FD00H: Status
0FD02H: Flash memory start address
0FD06H: Number of bytes written in flash memory
.
.

The entry RAM area is explained in detail below.

(a) Flash memory start address

This is the flash memory address value used by the _FlashByteWrite subroutine.

(b) Number of bytes written in flash memory

Area number and number of bytes written in the flash memory.

(c) Write time data

Set the following values according to the operating frequency.

f_X (MHz)	Setting Value
1.00 to 1.28	20H
1.29 to 2.56	40H
2.57 to 5.12	60H
5.13 to 8.38	80H
★ 8.39 to 10.0	A0H

(d) Erase time data

Setting value = Erase time (s) × Operating frequency/2^9 + 1

(Erase time range: 0.2 to 20 seconds, up to 100 times in 20 seconds are possible, assuming that one erase time is 0.2 seconds)

Example Erase time: 0.2 seconds, operating frequency: 5 MHz

Setting value = 0.2 × 5,000,000/512 + 1
= 1954 (decimal)
= 7A2 (hexadecimal)

(e) Write data storage buffer starting address

This area contains the starting address of the write data storage buffer area. The RAM data (write data) specified by the address data in this area is written in the flash memory (_FlashByteWrite subroutine). The data in this area is specified as the starting address and it is possible to specify up to a maximum of 256 bytes of write data.

(f) Writeback time

Setting value = Writeback time (s) × Operating frequency/2^7

(Up to 30 times in 1.5 seconds are possible, assuming that one writeback time is 0.05 seconds.)

Example Writeback time: 0.05 seconds, operating frequency: 5 MHz

Setting value = 0.05 × 5,000,000/128
= 1953 (decimal)
= 7A1 (hexadecimal)

18.4.6 Self-write subroutines

The self-write subroutines and their functions are shown in Table 18-7 below.

Table 18-7. List of Self-Write Subroutines

Function Number		Subroutine Name	Function
Decimal	Hexadecimal		
0	00H	_FlashEnv	Initializes the flash subroutine.
1	01H	_FlashSetEnv	Sets the parameters.
2	02H	_FlashGetInfo	Reads flash information.
16	10H	_FlashAreaBlankCheck	Performs a blank check of a specified area.
32	20H	_FlashAreaPreWrite	Performs prewrite for a specified area.
48	30H	_FlashAreaErase	Erases a specified area.
54	40H	_FlashAreaWriteBack	Writes back to a specified area.
80	50H	_FlashByteWrite	Writes continuously in byte units.
96	60H	_FlashAreaIVerify	Performs internal verification of a specified area.

(1) _FlashEnv subroutine

[Function]

Initializes the flash subroutine.

[Argument]

Entry RAM address ...... 2 bytes (HL register)

[Return value]

None

[Register/memory status after called]

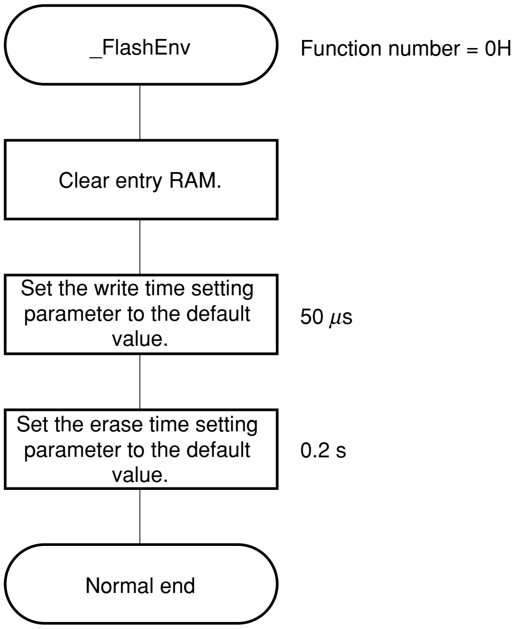
Entry RAM address

[Call example]

When the entry RAM address = 0FC30H

```
                DI
                SET1    FLSPM0
  LOOP:         BF      VPP, $LOOP
                SEL     RB3
                MOVW    HL, #0FC30H             ; Entry RAM address
; * * * * * * * * * * Initialization * * * * * * * * * *
                MOV     C, #0H                  ; FlashEnv (function number setting)
                CALL    !8100H
                        .
                        .
                        .
```

[Flowchart]


_FlashEnv
Function number = 0H
Clear entry RAM.
Set the write time setting parameter to the default value.
50 μs
Set the erase time setting parameter to the default value.
0.2 s
Normal end

(2) _FlashSetEnv subroutine

[Function]

Sets the parameters.

[Argument]

Write time data: 1 byte (offset value: +7)
Erase time data: 3 bytes (offset value: +8 to 10)

[Return value]

Status (B register)
   00H: Normal end
   80H: Parameter setting error

[Register/memory status after called]

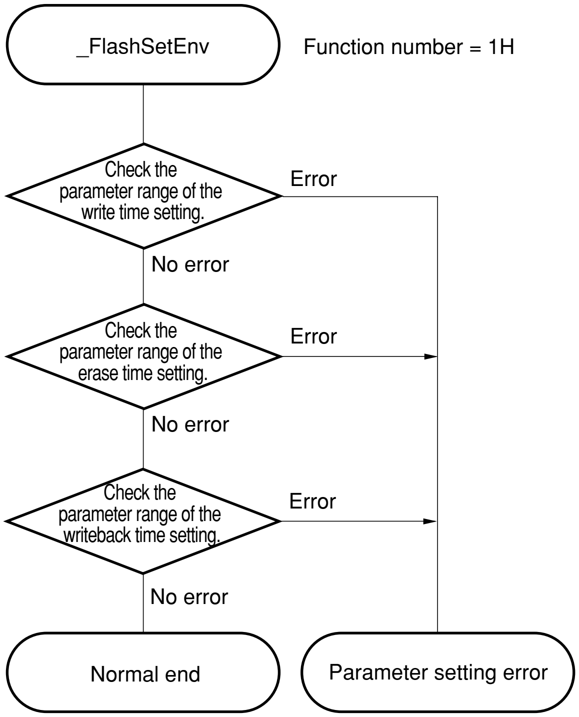
Entry RAM address, write time data, erase time data

[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #20H         ; Write time data
        MOV     !0FC37H, A
        MOV     A, #A2H         ; Erase time data
        MOV     !0FC38H, A      ; 0.2 s : 0007A2H (at 5 MHz)
        MOV     A, #07H
        MOV     !0FC39H, A
        MOV     A, #00H
        MOV     !0FC3AH, A
        MOV     A, #0A1H        ; Writeback time data setting
        MOV     !0FC3BH, A      ; 50 ms : 0007A1H (at 5 MHz)
        MOV     A, #07H
        MOV     !0FC3CH, A
        MOV     A, #00H
        MOV     !0FC3DH, A
        MOV     A, #02H
        MOV     !0FC40H, A      ; Total block number data setting
        MOV     !0FC41H, A      ; Total area number data setting
    ;
        MOV     C, #1H          ; FlashSetEnv (function number setting)
        CALL    !8100H
                .
                .
                .
```

[Flowchart]


_FlashSetEnv
Function number = 1H
Check the parameter range of the write time setting.
Error
No error
Check the parameter range of the erase time setting.
Error
No error
Check the parameter range of the writeback time setting.
Error
No error
Normal end
Parameter setting error

(3) _FlashGetInfo subroutine

[Function]

Reads the flash product identification codes.

- μPD78F0988A signature: 50H
- μPD78F0988 (old product) signature: 40H

[Argument]

Flash product identification code: 1 byte (offset value: +6)

[Return value]

Status (B register)

00H: Normal end
80H: Option specify error

Product identification code (A register)

[Register/memory status after called]

Entry RAM starting address

[Call example]

When the entry RAM address = 0FC30H

```
        MOV   A, #0H
        MOV   !0FC36H, A
;
        MOV   C, #40H              ; FlashGetInfo (function number setting)
        CALL  !8100H
```

[Note]

This function enables new products to be distinguished from old products.

(4) _FlashAreaBlankCheck subroutine

[Function]

Performs a blank check of a specified area.

[Argument]

Area number (= 0, 1): 1 byte (offset value: +6)

0: Blank check of area 0000H to 1FFFH (boot area)
1: Blank check of area 2000H to EFFFH

[Return value]

Status (B register)

00H: Normal end
02H: Blank check error
80H: Area number specification error

[Register/memory status after called]

Entry RAM address, area number

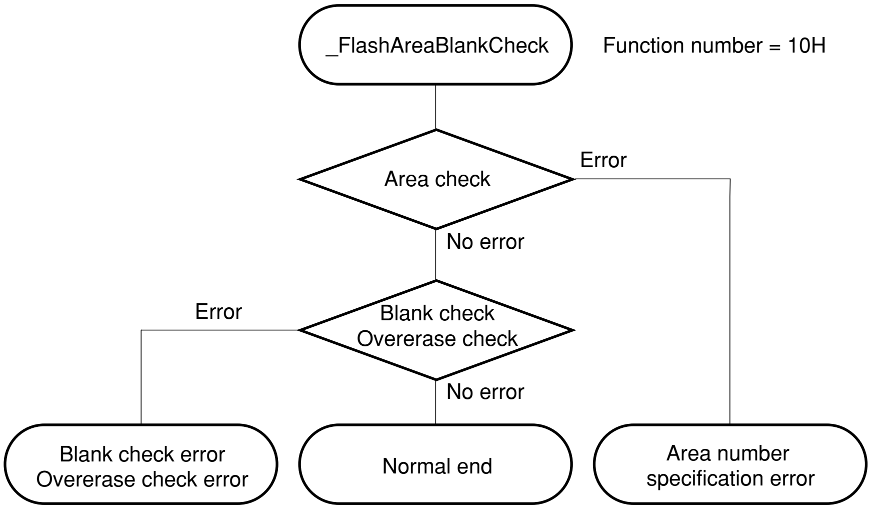
[Call example]

When the entry RAM address = 0FC30H

```
MOV     A, #01H        ; Specifies area 2000H to EFFFH
MOV     !0FC36H, A
MOV     C, #10H        ; FlashAreaBlankCheck (function number setting)
CALL    !8100H
        .
        .
        .
```

Caution Area 0 (0000H to 1FFFH) is a boot area. Do not specify area 0 as an argument.

[Flowchart]



(5) _FlashAreaPreWrite subroutine

[Function]

Performs prewrite for a specified area (writes 00H to a specified area).

[Argument]

Area number (= 0, 1, 2): 1 byte (offset value: +6)
0: Prewrites area 0000H to 1FFFH (boot area)
1: Prewrites area 2000H to EFFFH

[Return value]

Status (B register)
00H: Normal end
08H: Write error
80H: Area number specification error

[Register/memory status after called]

Entry RAM address, area number

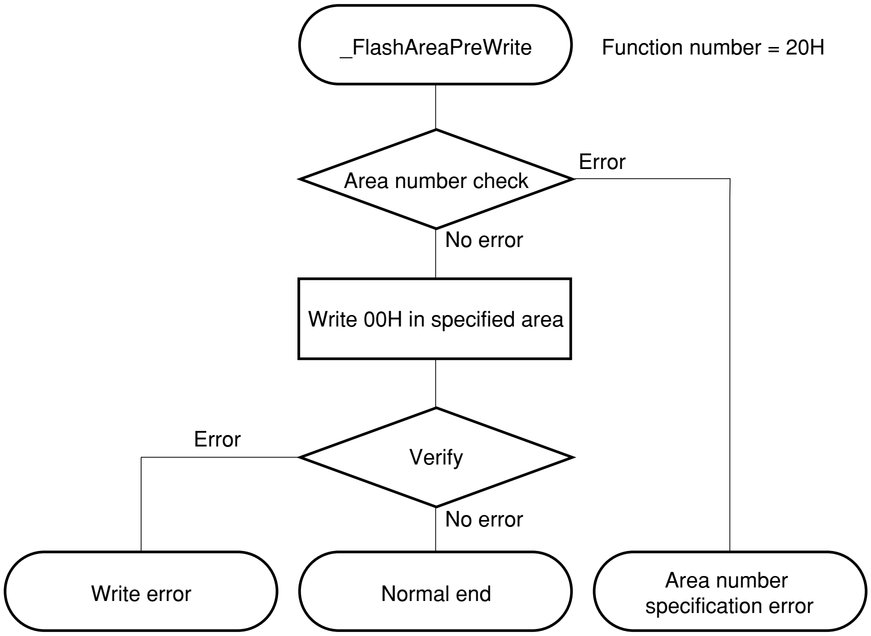
[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #1H          ; Specifies 2000H to EFFFH
        MOV     !0FC36H, A
  ;
        MOV     C, #20H         ; FlashAreaPreWrite (function number setting)
        CALL    !8100H
                .
                .
                .
```

Caution Area 0 (0000H to 1FFFH) is a boot area. Do not specify area 0 as an argument.

[Flowchart]



(6) _FlashAreaErase subroutine

[Function]

Erases a specified area.

[Argument]

Area number (= 0, 1): 1 byte (offset value: +6)
0: Erases area 0000H to 1FFFH (boot area)
1: Erases area 2000H to EFFFH

[Return value]

Status (B register)
00H: Normal end
02H: Blank check error
04H: Overerase check error
80H: Area number specification error

[Register/memory status after called]

Entry RAM address, area number

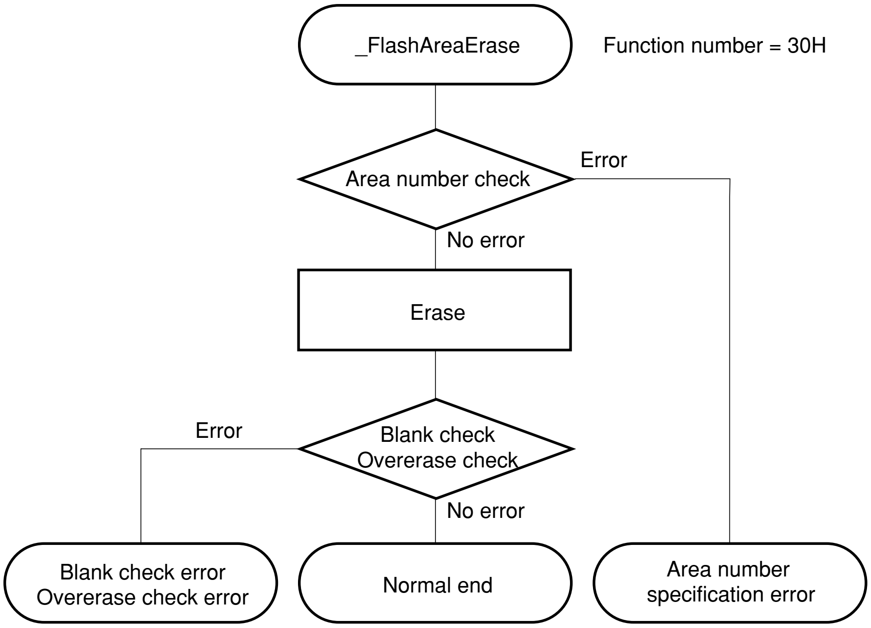
[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #1H          ; Specifies 2000H to EFFFH
        MOV     !0FC36H, A
    ;
        MOV     C, #30H         ; FlashAreaErase (function number setting)
        CALL    !8100H
                .
                .
                .
```

Caution Area 0 (0000H to 1FFFH) is a boot area. Do not specify area 0 as an argument.

[Flowchart]



(7) _FlashAreaWriteBack subroutine

[Function]

Writes back the flash signature codes.
(Writeback is an operation to return a flash area in an overerasure status after erasure to the proper erasure status.)

[Argument]

Area number (= 0, 1):	1 byte (offset value: +6)

[Return value]

Status (B register)

00H:	Normal end
02H:	Blank check error
04H:	Overerase check error
80H:	Write address error

[Register/memory status after called]

Entry RAM starting address and area number

[Call example]

When the entry RAM address = 0FC30H

```
        MOV   A, #1H              ; Area 1 setting
        MOV   !0FC36H, A
;
        MOV   C, #40H             ; FlashAreaWriteBack (function number setting)
        CALL  !8100H
```

[Notes]

Set the writeback time to 50 ms/writeback.
Set the number of writebacks to 30 max., assuming 50 ms/writeback.

(8) _FlashByteWrite subroutine

[Function]

Writes continuously in byte units.

[Argument]

Flash memory write start address: 2 bytes (offset value: +2)
Number of bytes[Note] written in flash memory: 1 byte (offset value: +6)
Write data storage buffer starting address: 2 bytes (offset value: +14)

Note If 0 is set, it is possible to set a maximum of 256 bytes.

[Return value]

Status (B register)

00H: Normal end
08H: Write error
80H: Write address error

[Register/memory status after called]

Entry RAM address, number of bytes written in flash memory

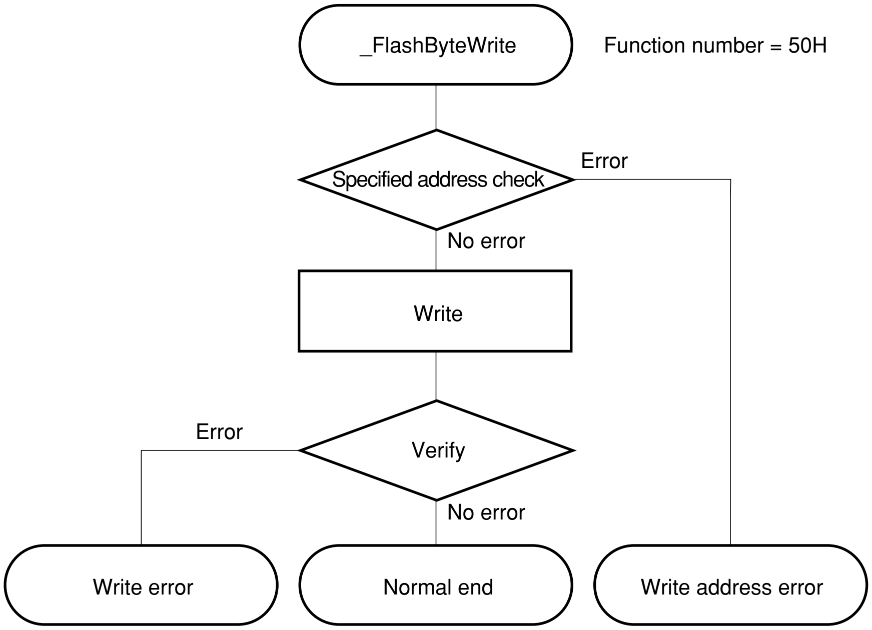
The flash memory write start address is updated to the address at the end of writing.

[Call example]

When the entry RAM address = 0FC30H

```
        MOVW    AX, #0FD00H  ; Write data storage buffer starting address
        MOVW    !0FC3EH, AX
        MOVW    AX, #2000H   ; Flash memory write start address
        MOVW    !0FC32H, AX
        MOV     A, #0H       ; Number of bytes written in flash memory (256 bytes)
        MOV     !0FC36H, A
  ;
        MOV     C, #50H      ; FlashByteWrite (function number setting)
        CALL    !8100H
                .
                .
                .
```

[Flowchart]


_FlashByteWrite
Function number = 50H
Specified address check
Error
No error
Write
Error
Verify
No error
Write error
Normal end
Write address error

(9) _FlashArealVerify subroutine

[Function]

Performs internal verification of a specified area (reads the flash memory of a specified area in a different mode, and compares it).

[Argument]

Area number (= 0, 1): 1 byte (offset value: +6)

0: Performs internal verification of area 0000H to 1FFFH (boot area)

1: Performs internal verification of area 2000H to EFFFH

[Return value]

Status (B register)

00H: Normal end

10H: Verify error

80H: Area number specification error

[Register/memory status after called]

Entry RAM address, area number

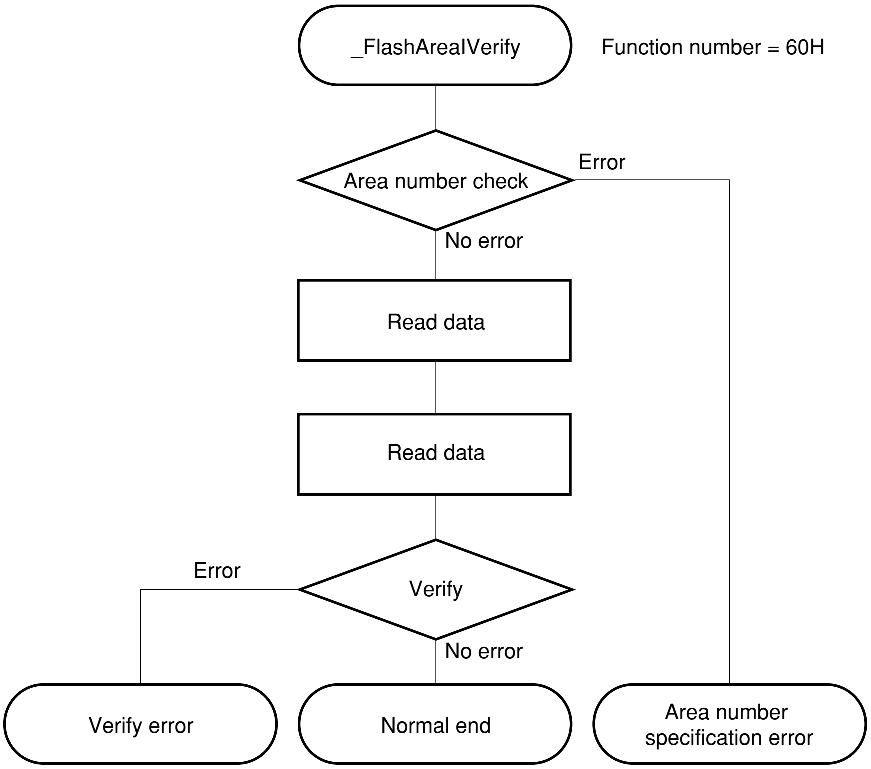
[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #01H         ; Specifies 2000H to EFFFH
        MOV     !0FC36H, A
    ;
        MOV     C, #60H         ; FlashArealVerify (function number setting)
        CALL    !8100H
                .
                .
                .
```

Caution Area 0 (0000H to 1FFFH) is a boot area. Do not specify area 0 as an argument.

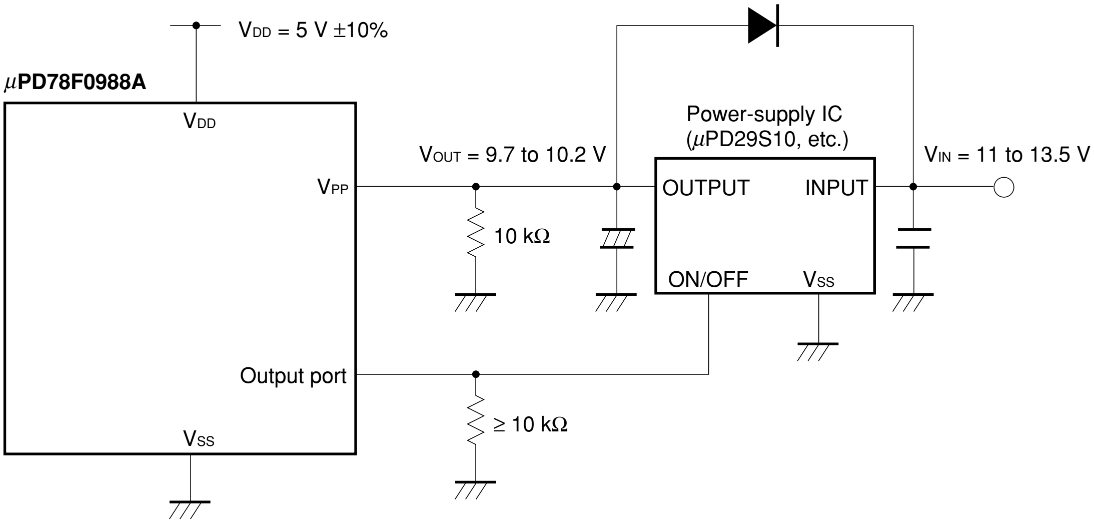
[Flowchart]



18.4.7 Self-write circuit configuration

The configuration of the self-write circuit is shown in Figure 18-18.

Figure 18-18. Self-Write Circuit Configuration



CHAPTER 19 INSTRUCTION SET

This chapter lists the instruction set of the μPD780988 Subseries. For the details of the operation and machine language (instruction code) of each instruction, refer to **78K/0 Series User's Manual Instructions (U12326E)**.

19.1 Conventions

19.1.1 Operand representation and description formats

In the operand field of each instruction, an operand is described according to the description format for operand representation of that instruction (for details, refer to the assembler specifications). Some operands may be described in two or more description formats. In this case, select one of them. Uppercase characters, #, !, $, and [ ] are keywords and must be described as is. The meanings of the symbols are as follows:

- #: Immediate data
- !: Absolute address
- $: Relative address
- [ ]: Indirect address

When describing immediate data, also describe an appropriate numeric value or label. When describing a label, be sure to describe #, !, $, or [ ].

Register description formats r or rp for an operand can be described as a function name (such as X, A, or C) or absolute name (the name in parentheses in the table below, such as R0, R1, or R2).

Table 19-1. Operand Representation and Description Formats

Representation	Description Format
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol[Note]
sfrp	Special function register symbol (only even address of register that can be manipulated in 16-bit units)[Note]
saddr	FE20H to FF1FH immediate data or label
saddrp	FE20H to FF1FH immediate data or label (even address only)
addr16	0000H to FFFFH immediate data or label (even address only for 16-bit data transfer instruction)
addr11	0800H to 0FFFH immediate data or label
addr5	0040H to 007FH immediate data or label (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note FFD0H to FFDFH cannot be addressed.

Remark For the symbols of the special function registers, refer to **Table 3-4 Special Function Register List**.

19.1.2 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 contents of address or register in ( )
$\times_H$, $\times_L$:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
$\overline{\quad\quad}$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

19.1.3 Description of flag operation column

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to result
R:	Value saved before is restored

19.2 Operation List

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte			
		saddr, #byte	3	6	7	(saddr) ← byte			
		sfr, #byte	3	–	7	sfr ← byte			
		A, r **Note 3**	1	2	–	A ← r			
		r, A **Note 3**	1	2	–	r ← A			
		A, saddr	2	4	5	A ← (saddr)			
		saddr, A	2	4	5	(saddr) ← A			
		A, sfr	2	–	5	A ← sfr			
		sfr, A	2	–	5	sfr ← A			
		A, !addr16	3	8	9 + 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			

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

4. m indicates the number of wait states when the external memory expansion area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	**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)			
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 4**	1	4	–	AX ← rp			
		rp, AX **Note 4**	1	4	–	rp ← AX			
		AX, !addr16	3	10	12 + 2n	AX ← (addr16)			
		!addr16, AX	3	10	12 + 2m	(addr16) ← AX			
	XCHW	AX, rp **Note 4**	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 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 + (saddr16)	×	×	×
		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)	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

4. Only when rp = BC, DE, HL

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

4. m indicates the number of wait states when the external memory expansion area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**ADDC**	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	×	×	×
	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	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	**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)	×		
	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)	×		

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	**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)	×	×	×
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}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

4. m indicates the number of wait states when the external memory expansion area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulation	**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			
	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			×

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

4. m indicates the number of wait states when the external memory expansion area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulation	**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 ← $\overline{CY}$			×
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

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

4. m indicates the number of wait states when the external memory expansion area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulation	**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			
	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			

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Conditional branch	**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 when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory expansion area is read.

4. m indicates the number of wait states when the external memory expansion area is written.

19.3 Instruction List by Addressing

(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

2nd Operand 1st 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 for r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

2nd Operand 1st 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

2nd Operand 1st 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/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

2nd Operand / 1st 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

Absolute Maximum Ratings (T_A = 25°C)

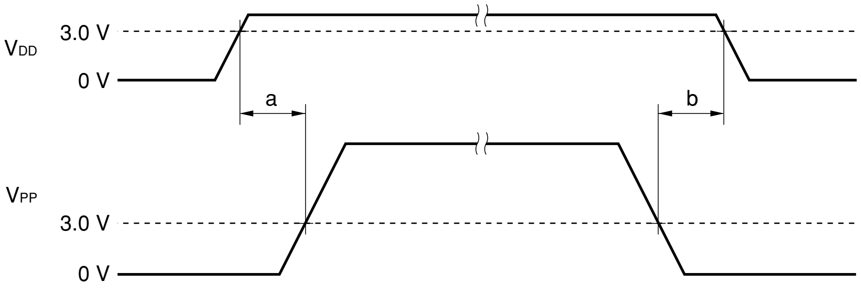
Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	V_{PP}	μPD78F0988A, 78F0988A(A) only **Note 1**		–0.3 to +10.5	V
	AV_{DD}			–0.3 to V_{DD} + 0.3	V
	AV_{REF}			–0.3 to V_{DD} + 0.3	V
	AV_{SS}			–0.3 to +0.3	V
Input voltage	V_I	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P50 to P57, P64 to P67, TO70 to TO75, X1, X2, $\overline{RESET}$		–0.3 to V_{DD} + 0.3	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3	V
Analog input voltage	V_{AN}	P10 to P17	Analog input pin	AV_{SS} – 0.3 to AV_{REF} + 0.3 and –0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin		–10	mA
		P00, P01, P30 to P37, P40 to P47, P50 to P57, P64 to P67 total		–15	mA
		P02, P03, P20 to P26, TO70 to TO75 total		–15	mA
Output current, low	I_{OL} **Note 2**	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P40 to P47, P64 to P67 per pin	Peak value	20	mA
			rms value	10	mA
		P50 to P57, TO70 to TO75 per pin	Peak value	30	mA
			rms value	15	mA
		P00, P01, P30 to P37, P40 to P47, P64 to P67 total	Peak value	50	mA
			rms value	20	mA
		P02, P03, P20 to P26 total	Peak value	30	mA
			rms value	15	mA
		TO70 to TO75 total	Peak value	100	mA
			rms value	70	mA
		P50 to P57 total	Peak value	100	mA
			rms value	70	mA
Operating ambient temperature	T_A	In normal operating mode		–40 to +85	°C
		In flash memory programming mode (μPD78F0988A, 78F0988A(A) only)		+10 to +40	°C
Storage temperature	T_{stg}	Mask ROM products		–65 to +150	°C
		Flash memory products		–40 to +125	°C

(The notes are explained on the following 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.

Notes 1. When writing in the flash memory, be sure to satisfy the following conditions on the V_{PP} voltage supply timing.

- At rising edge of power supply voltage
  More than 10 μs after V_{DD} reaches the lower limit voltage (3.0 V) of the operating voltage range, V_{PP} should exceed V_{DD} (a in the figure below).
- At falling edge of power supply voltage
  More than 10 μs after V_{PP} falls below the lower limit voltage (3.0 V) of the V_{DD} operating voltage range, start up V_{DD} (b in the figure below).



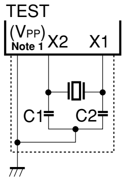
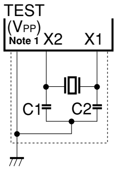
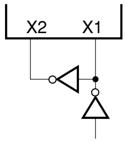
2. The rms value should be calculated as follows: [rms value] = [Peak value] × $\sqrt{\text{Duty}}$

Capacitance (T_A = 25°C, V_{DD} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_{IN}	f = 1 MHz Unmeasured pins returned to 0 V				15	pF
I/O capacitance	C_{IO}	f = 1 MHz Unmeasured pins returned to 0 V	P00 to P03, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67, TO70 to TO75			15	pF

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

System Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 3.0 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)Note 2	4.5 V ≤ V_{DD} ≤ 5.5 V	1.0		12.0	MHz
			3.0 V ≤ V_{DD} < 4.5 V	1.0		8.38	MHz
		Oscillation stabilization timeNote 3	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency (f_X)Note 2	4.5 V ≤ V_{DD} ≤ 5.5 V	1.0		12.0	MHz
			3.0 V ≤ V_{DD} < 4.5 V	1.0		8.38	MHz
		Oscillation stabilization timeNote 3	4.0 V ≤ V_{DD} ≤ 5.5 V			10	ms
			3.0 V ≤ V_{DD} < 4.0 V			30	ms
External clock		X1 input frequency (f_X)Note 2	4.5 V ≤ V_{DD} ≤ 5.5 V	1.0		12.0	MHz
			3.0 V ≤ V_{DD} < 4.5 V	1.0		8.38	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	4.5 V ≤ V_{DD} ≤ 5.5 V	38		500	ns
			3.0 V ≤ V_{DD} < 4.5 V	50		500	ns

Notes **1.** In the case of the μPD78F0988A and 78F0988A(A)

2. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

3. Time required to stabilize oscillation after reset or STOP mode release.

Caution **When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.**

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS1}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

Recommended Oscillator Constant

(1) μPD780982, 780983, 780984, 780986, 780988, 780982(A), 780983(A), 780984(A), 780986(A), 780988(A)

System clock: Ceramic resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	Frequency	Type	Recommended Circuit Constant		Oscillation Voltage Range	
		(MHz)		C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Mfg. Co., Ltd.	CSTCC2M00G56-R0	2.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS2M00G56-B0	2.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCR4M00G53-R0	4.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS4M00G53-B0	4.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCR4M19G53-R0	4.19	SMD	On-chip	On-chip	3.0	5.5
	CSTLS4M19G53-B0	4.19	Lead	On-chip	On-chip	3.0	5.5
	CSTCR4M91G53-R0	4.91	SMD	On-chip	On-chip	3.0	5.5
	CSTLS4M91G53-B0	4.91	Lead	On-chip	On-chip	3.0	5.5
	CSTCR5M00G53-R0	5.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS5M00G53-B0	5.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCE8M00G52-R0	8.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS8M00G53-B0	8.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCE8M38G52-R0	8.38	SMD	On-chip	On-chip	3.0	5.5
	CSTLS8M38G53-B0	8.38	Lead	On-chip	On-chip	3.0	5.5
	CSTCE10M0G52-R0	10.00	SMD	On-chip	On-chip	4.5	5.5
	CSTLS10M0G53-B0	10.00	Lead	On-chip	On-chip	4.5	5.5
	CSTCE12M0G52-R0	12.00	SMD	On-chip	On-chip	4.5	5.5
	CSTLA12M0T55-B0	12.00	Lead	On-chip	On-chip	4.5	5.5
	CSTLA12M0T55093-B0	12.00	Lead	On-chip	On-chip	4.5	5.5

Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit.

Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD780988 Subseries within the specifications of the DC and AC characteristics.

(2) μPD78F0988A, 78F0988A(A)

System clock: Ceramic resonator (T_A = −40 to +85°C)

Manufacturer	Part Number	Frequency	Type	Recommended Circuit Constant		Oscillation Voltage Range	
		(MHz)		C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Mfg. Co., Ltd.	CSTCC2M00G56-R0	2.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS2M00G56-B0	2.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCR4M00G53-R0	4.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS4M00G53-B0	4.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCR4M19G53-R0	4.19	SMD	On-chip	On-chip	3.0	5.5
	CSTLS4M19G53-B0	4.19	Lead	On-chip	On-chip	3.0	5.5
	CSTCR4M91G53-R0	4.91	SMD	On-chip	On-chip	3.0	5.5
	CSTLS4M91G53-B0	4.91	Lead	On-chip	On-chip	3.0	5.5
	CSTCR5M00G53-R0	5.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS5M00G53-B0	5.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCE8M00G52-R0	8.00	SMD	On-chip	On-chip	3.0	5.5
	CSTLS8M00G53-B0	8.00	Lead	On-chip	On-chip	3.0	5.5
	CSTLS8M00G53093-B0	8.00	Lead	On-chip	On-chip	3.0	5.5
	CSTCE8M38G52-R0	8.38	SMD	On-chip	On-chip	3.0	5.5
	CSTLS8M38G53-B0	8.38	Lead	On-chip	On-chip	3.0	5.5
	CSTLS8M38G53093-B0	8.38	Lead	On-chip	On-chip	3.0	5.5
	CSTCE10M0G52-R0	10.00	SMD	On-chip	On-chip	4.5	5.5
	CSTLS10M0G53-B0	10.00	Lead	On-chip	On-chip	4.5	5.5
	CSTLS10M0G53093-B0	10.00	Lead	On-chip	On-chip	4.5	5.5
	CSTCE12M0G52-R0	12.00	SMD	On-chip	On-chip	4.5	5.5
	CSTLA12M0T55-B0	12.00	Lead	On-chip	On-chip	4.5	5.5
	CSTLA12M0T55093-B0	12.00	Lead	On-chip	On-chip	4.5	5.5

Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit.

Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD780988 Subseries within the specifications of the DC and AC characteristics.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 3.0 to 5.5 V) (1/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P10 to P17, P21, P23, P30 to P37, P40 to P47, P50, P53, P64 to P67		$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P00 to P03, P20, P22, P24 to P26, P51, P52, P54 to P57		$0.8V_{DD}$		V_{DD}	V
	V_{IH3}	X1, X2		$V_{DD} - 0.5$		V_{DD}	V
Input voltage, low	V_{IL1}	P10 to P17, P21, P23, P30 to P37, P40 to P47, P50, P53, P64 to P67		0		$0.3V_{DD}$	V
	V_{IL2}	$\overline{\text{RESET}}$, P00 to P03, P20, P22, P24 to P26, P51, P52, P54 to P57		0		$0.2V_{DD}$	V
	V_{IL3}	X1, X2		0		0.4	V
Output voltage, high	V_{OH1}	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –1 mA		$V_{DD} - 1.0$		V_{DD}	V
		I_{OH} = –100 μA		$V_{DD} - 0.5$		V_{DD}	V
Output voltage, low	V_{OL1}	P50 to P57, TO70 to TO75	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 15 mA		0.4	2.0	V
		P00 to P03, P20 to P26, P30 to P37, P40 to P47, P64 to P67	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 1.6 mA			0.4	V
	V_{OL2}	I_{OL} = 400 μA				0.5	V
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67, TO70 to TO75, $\overline{\text{RESET}}$			3	μA
	I_{LIH2}		X1, X2			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67, TO70 to TO75, $\overline{\text{RESET}}$			–3	μA
	I_{LIL2}		X1, X2			–20	μA
Output leakage current, high	I_{LOH}	V_{OUT} = V_{DD}				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				–3	μA
Software pull-up resistor	R_2	V_{IN} = 0 V P00 to P03, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67		15	30	90	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 3.0 to 5.5 V) (2/3)

(1) μPD780982, 780983, 780984, 780986, 780988, 780982(A), 780983(A), 780984(A), 780986(A), 780988(A)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current	I_{DD1}	12.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 1]	When A/D converter stopped		9	18[Note 2]	mA
				When A/D converter operating		10	20[Note 2]	mA
		8.38 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 1]	When A/D converter stopped		6.5	13[Note 2]	mA
				When A/D converter operating		7.5	15[Note 2]	mA
			V_{DD} = 3.0 V ±10%[Note 1, 3]	When A/D converter stopped		3.5	7[Note 2]	mA
				When A/D converter operating		4.5	9[Note 2]	mA
	I_{DD2}	12.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 1]	When peripheral function stopped		2	4	mA
				When peripheral function operating			10	mA
		8.38 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 1]	When peripheral function stopped		1	2	mA
				When peripheral function operating			7	mA
			V_{DD} = 3.0 V ±10%[Note 1, 3]	When peripheral function stopped		0.8	1.5	mA
				When peripheral function operating			4.5	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA
			V_{DD} = 3.0 V ±10%[Note 3]			0.05	10	μA

Notes

1. High-speed mode operation (when the processor clock control register (PCC) is set to 00H).

2. Refers to the total current flowing to the internal power supply (V_{DD0} and V_{DD1}). The peripheral operation current is included, but the current flowing to the pull-up resistors of ports and the AV_{REF} pin is not.

3. Specification when V_{DD} = 3.0 to 3.3 V. The TYP. value is the value at V_{DD} = 3.0 V.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 3.0 to 5.5 V) (3/3)

(2) μPD78F0988A, 78F0988A(A)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current	I_{DD1}	12.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 1]	When A/D converter stopped		25	36[Note 2]	mA
				When A/D converter operating		26	38[Note 2]	mA
		8.38 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 1]	When A/D converter stopped		15	25[Note 2]	mA
				When A/D converter operating		16	27[Note 2]	mA
			V_{DD} = 3.0 V ±10%[Note 1, 3]	When A/D converter stopped		12	17[Note 2]	mA
				When A/D converter operating		13	19[Note 2]	mA
	I_{DD2}	12.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 1]	When peripheral function stopped		2	4	mA
				When peripheral function operating			10	mA
		8.38 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 1]	When peripheral function stopped		1.3	2.6	mA
				When peripheral function operating			7.3	mA
			V_{DD} = 3.0 V ±10%[Note 1, 3]	When peripheral function stopped		1	2	mA
				When peripheral function operating			5	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA
			V_{DD} = 3.0 V ±10%[Note 3]			0.05	10	μA
V_{PP} supply voltage	V_{PP1}	In normal operation mode			0		0.2V_{DD}	V

Notes
1. High-speed mode operation (when the processor clock control register (PCC) is set to 00H).
2. Refers to the total current flowing to the internal power supply (V_{DD0} and V_{DD1}). The peripheral operation current is included, but the current flowing to the pull-up resistors of ports and the AV_{REF} pin is not.
3. Specification when V_{DD} = 3.0 to 3.3 V. The TYP. value is the value at V_{DD} = 3.0 V.

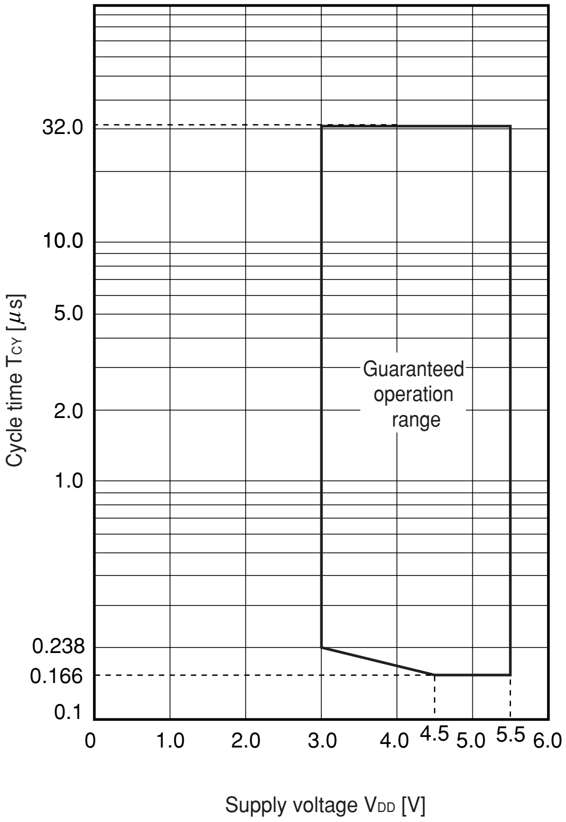
AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 3.0 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (Min. instruction execution time)	T_{CY}	Operating with system clock	4.5 V ≤ V_{DD} ≤ 5.5 V	0.166		32	μs
			3.0 V ≤ V_{DD} < 4.5 V	0.238		32	μs
TI000, TI001, TI010, TI011 input frequency	f_{TI0}			0		f_X/64	MHz
TI000, TI001, TI010, TI011 input high-/low-level width	t_{TIH0} t_{TIL0}			2/f_{sam} + 0.1 Note			μs
TI50, TI51, TI52 input frequency	f_{TI5}	8-/16-bit precision		0		4	MHz
TI50, TI51, TI52 input high-/low-level width	t_{TIH5} t_{TIL5}	8-/16-bit precision		100			ns
Interrupt request input high-/low-level width	t_{INTH} t_{INTL}	INTP0 to INTP7		1			μs
TOFF input high-/low-level width	t_{TOFFH} t_{TOFFL}			2			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}			10			μs

Note Selection of f_{sam} = f_X, f_X/4, f_X/32 is possible with bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00) or with bits 0 and 1 (PRM010, PRM011) of prescaler mode register 01 (PRM01). Note that when selecting TI000 (TM00) or TI001 (TM01) valid edge as the count clock, f_{sam} = f_X/16.

T_{CY} vs V_{DD} (System clock operation)



(2) Read/write operation (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

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 $\overline{\text{RD}}$↓	t_{RDAD}		0	100	ns
Data input time from $\overline{\text{RD}}$↓	t_{RDD1}			$(2 + 2n)t_{CY} - 87$	ns
	t_{RDD2}			$(3 + 2n)t_{CY} - 93$	ns
Read data hold time	t_{RDH}		0		ns
$\overline{\text{RD}}$ low-level width	t_{RDL1}		$(1.5 + 2n)t_{CY} - 33$		ns
	t_{RDL2}		$(2.5 + 2n)t_{CY} - 33$		ns
$\overline{\text{WAIT}}$↓ input time from $\overline{\text{RD}}$↓	t_{RDWT1}			$t_{CY} - 43$	ns
	t_{RDWT2}			$t_{CY} - 43$	ns
$\overline{\text{WAIT}}$↓ input time from $\overline{\text{WR}}$↓	t_{WRWT}			$t_{CY} - 25$	ns
$\overline{\text{WAIT}}$ 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
$\overline{\text{WR}}$ low-level width	t_{WRL}		$(1.5 + 2n)t_{CY} - 15$		ns
Delay time from ASTB↓ to $\overline{\text{RD}}$↓	t_{ASTRD}		6		ns
Delay time from ASTB↓ to $\overline{\text{WR}}$↓	t_{ASTWR}		$2t_{CY} - 15$		ns
Delay time from $\overline{\text{RD}}$↑ at external fetch to ASTB↑	t_{RDAST}		$0.8t_{CY} - 15$	$1.2t_{CY}$	ns
Address hold output time from $\overline{\text{WR}}$↑	t_{WRADH}		$0.8t_{CY} - 15$	$1.2t_{CY} + 30$	ns
Write data output time from $\overline{\text{RD}}$↑	t_{RDWD}		40		ns
Write data output time from $\overline{\text{WR}}$↓	t_{WRWD}		10	60	ns
Address hold time from $\overline{\text{RD}}$↑ at external fetch	t_{RDADH}		$0.8t_{CY} - 15$	$1.2t_{CY} + 30$	ns
Delay time from $\overline{\text{WAIT}}$↑ to $\overline{\text{RD}}$↑	t_{WTRD}		$0.8t_{CY}$	$2.5t_{CY} + 25$	ns
Delay time from $\overline{\text{WAIT}}$↑ to $\overline{\text{WR}}$↑	t_{WTWR}		$0.8t_{CY}$	$2.5t_{CY} + 25$	ns

Remarks **1.** $t_{CY} = T_{CY}/4$

2. n indicates the number of waits.

3. C_L = 100 pF (C_L is the load capacitance of the AD0 to AD7, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, and ASTB pins.)

Caution **T_{CY} can only be used when the MIN. value is 0.238 μs.**

(3) Serial interface (T_A = –40 to +85°C, V_{DD} = 3.0 to 5.5 V)

(a) 3-wire serial I/O mode ($\overline{\text{SCK}}$... Internal clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK}}$ cycle time	t_{KCY1}	4.5 V ≤ V_{DD} ≤ 5.5 V		5.32			μs
		3.0 V ≤ V_{DD} < 4.5 V		7.63			μs
$\overline{\text{SCK}}$ high-/low-level width	t_{KH1} t_{KL1}			$t_{KCY1}/2 - 50$			ns
SI setup time (to $\overline{\text{SCK}}$↑)	t_{SIK1}			100			ns
SI hold time (from $\overline{\text{SCK}}$↑)	t_{KSI1}	4.5 V ≤ V_{DD} ≤ 5.5 V		300			ns
		3.0 V ≤ V_{DD} < 4.5 V		400			ns
Delay time from $\overline{\text{SCK}}$↓ to SO output	t_{KSO1}	C = 100 pF$^{\text{Note}}$	4.5 V ≤ V_{DD} ≤ 5.5 V			200	ns
			3.0 V ≤ V_{DD} < 4.5 V			300	ns

Note C is the load capacitance of the $\overline{\text{SCK}}$ and SO output lines.

(b) 3-wire serial I/O mode ($\overline{\text{SCK}}$... External clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK}}$ cycle time	t_{KCY2}	4.5 V ≤ V_{DD} ≤ 5.5 V		666			ns
		3.0 V ≤ V_{DD} < 4.5 V		800			ns
$\overline{\text{SCK}}$ high-/low-level width	t_{KH2}	4.5 V ≤ V_{DD} ≤ 5.5 V		333			ns
	t_{KL2}	3.0 V ≤ V_{DD} < 4.5 V		400			ns
SI setup time (to $\overline{\text{SCK}}$↑)	t_{SIK2}			100			ns
SI hold time (from $\overline{\text{SCK}}$↑)	t_{KSI2}	4.5 V ≤ V_{DD} ≤ 5.5 V		300			ns
		3.0 V ≤ V_{DD} < 4.5 V		400			ns
Delay time from $\overline{\text{SCK}}$↓ to SO output	t_{KSO2}	C = 100 pF$^{\text{Note}}$	4.5 V ≤ V_{DD} ≤ 5.5 V			200	ns
			3.0 V ≤ V_{DD} < 4.5 V			300	ns

Note C is the load capacitance of the $\overline{\text{SCK}}$ and SO output lines.

(c) UART mode (UART00) (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		4.5 V ≤ V_{DD} ≤ 5.5 V			187500	bps
		3.0 V ≤ V_{DD} < 4.5 V			131031	bps

(d) UART mode (UART00) (Infrared data transfer mode)

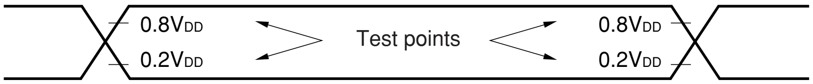
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		4.0 V ≤ V_{DD} ≤ 5.5 V			115200	bps
Bit rate allowable error		4.0 V ≤ V_{DD} ≤ 5.5 V			±0.87	%
Output pulse width		4.0 V ≤ V_{DD} ≤ 5.5 V	1.2		0.24/fbr$^{\text{Note}}$	μs
Input pulse width		4.0 V ≤ V_{DD} ≤ 5.5 V	4/fx			μs

Note fbr: Set baud rate

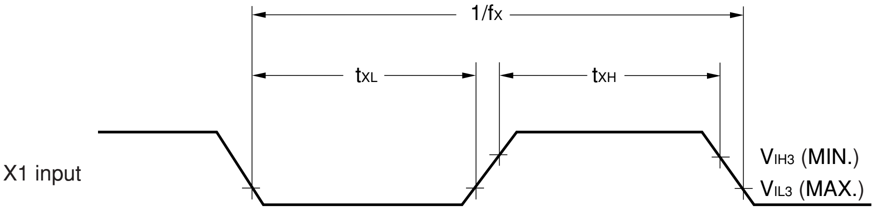
(e) UART mode (UART01) (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		4.5 V ≤ V_{DD} ≤ 5.5 V			93750	bps
		3.0 V ≤ V_{DD} < 4.5 V			65516	bps

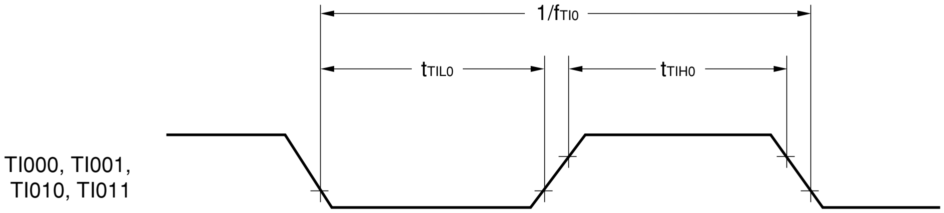
AC Timing Test Points (Excluding X1 Input)

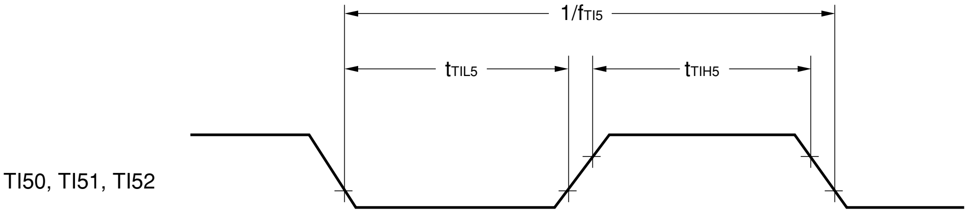


Clock Timing

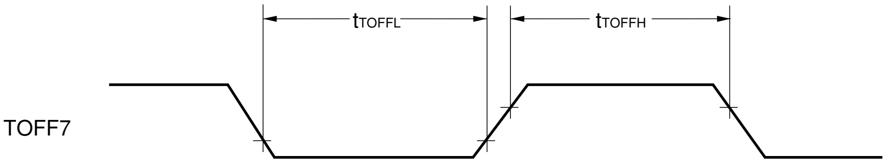


TI Timing



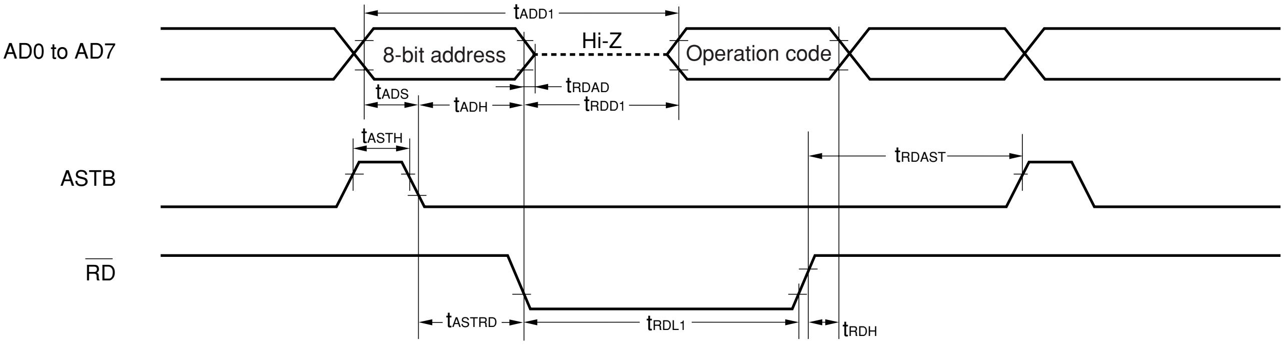


TOFF Timing

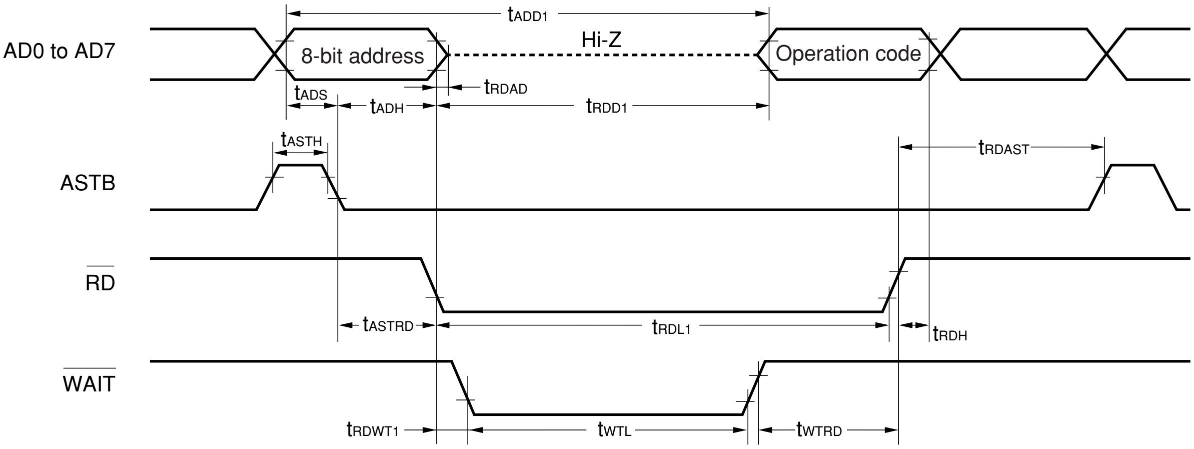


Read/Write Operation

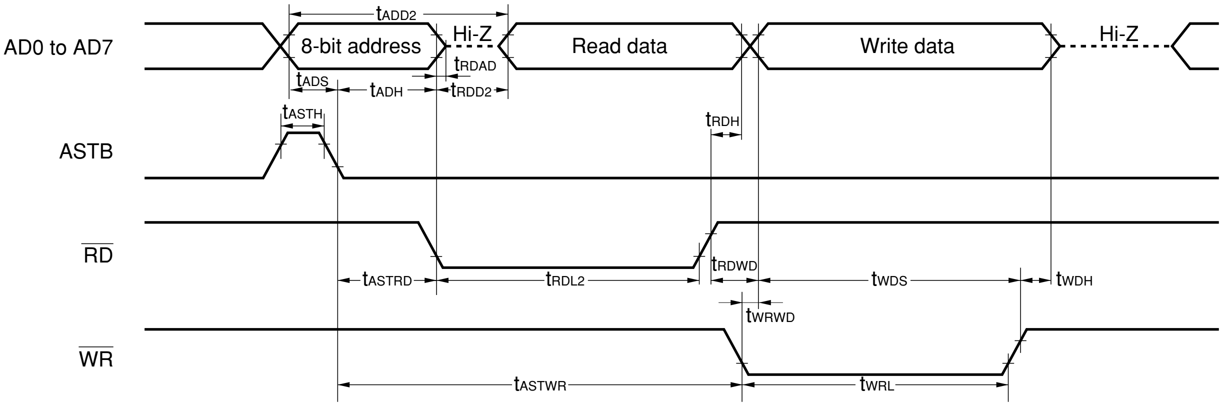
External fetch (no wait):



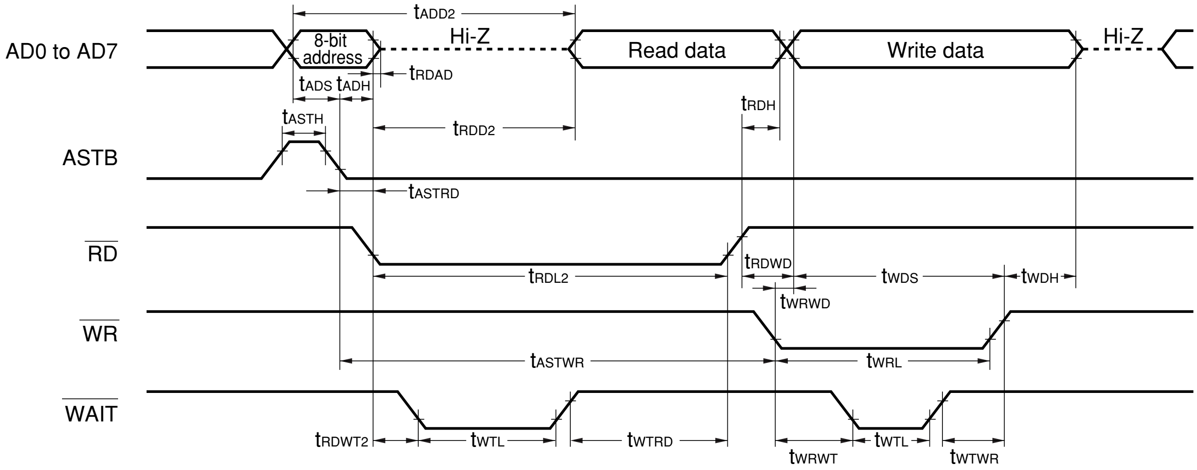
External fetch (wait insertion):



External data access (no wait):

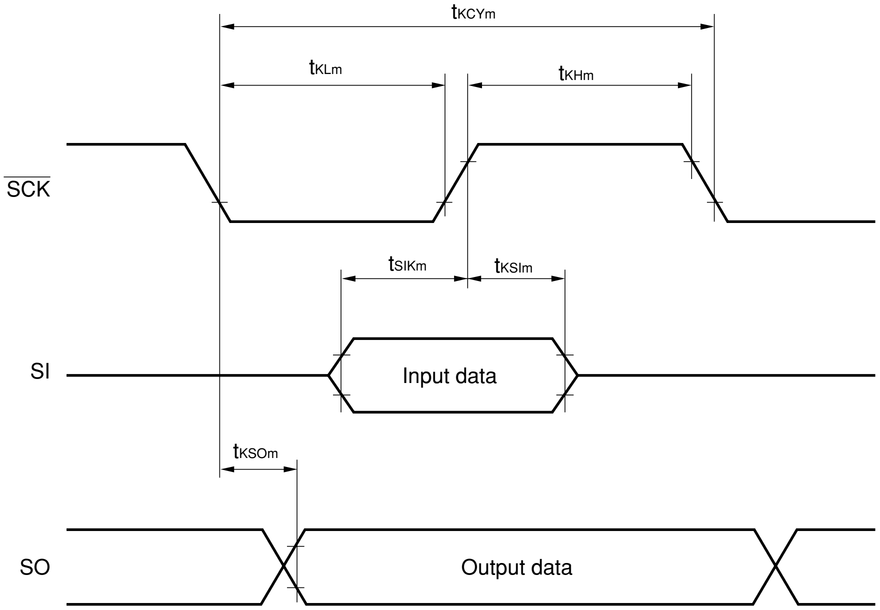


External data access (wait insertion):



Serial Transfer Timing

3-wire serial I/O mode:



m = 1, 2

A/D Converter Characteristics (T_A = –40 to +85°C, V_{DD} = AV_{DD} = 3.0 to 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 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}	4.5 V ≤ AV_{DD} ≤ 5.5 V	12		96	μs
		4.0 V ≤ AV_{DD} < 4.5 V	14		96	μs
		3.0 V ≤ AV_{DD} < 4.0 V	17		96	μs
Zero-scale error[Note 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[Note 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 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 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}		0		AV_{REF}	V
Reference voltage	AV_{REF}		2.7		AV_{DD}	V
Resistance between AV_{REF} and AV_{SS}	R_{REF}	When A/D converter is not operating	20	40		kΩ

Notes 1. Excludes quantization error (±1/2 LSB).

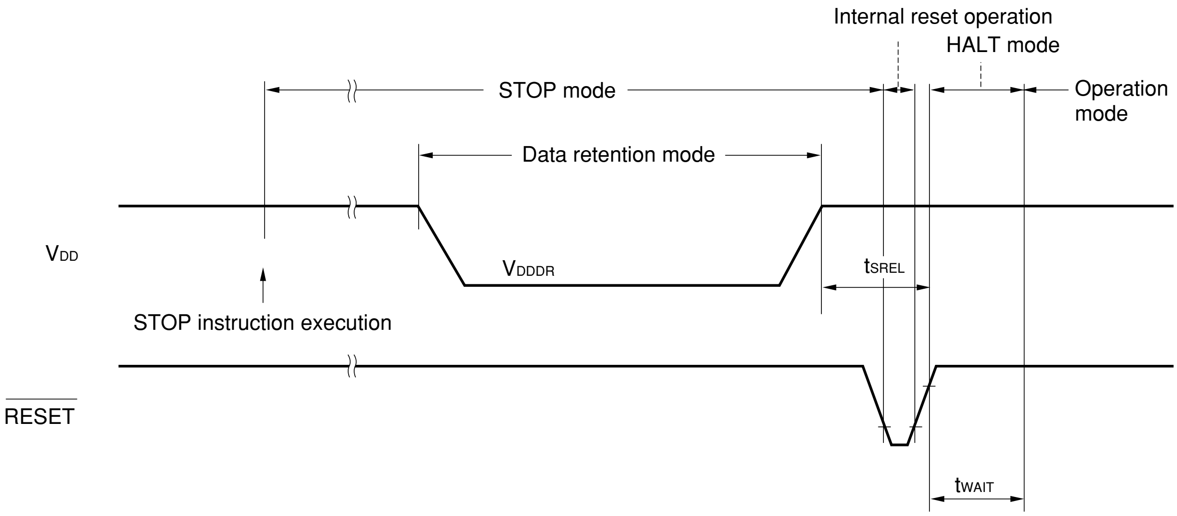
2. This value is indicated as a ratio (%FSR) to the full-scale value.

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

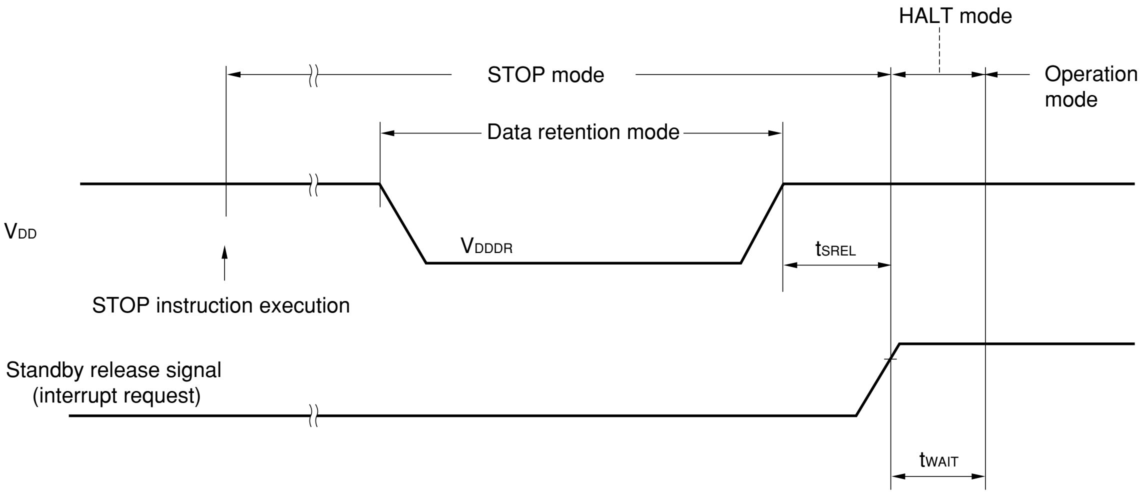
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		2.0		5.5	V
Data retention power supply current	I_{DDDR}	V_{DDDR} = 2.0 V		0.1	10	μA
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time	t_{WAIT}	Release by $\overline{RESET}$		$2^{17}/f_X$		ms
		Release by interrupt request		**Note**		ms

Note Selection of $2^{12}/f_X$ and $2^{14}/f_X$ to $2^{17}/f_X$ is possible with bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

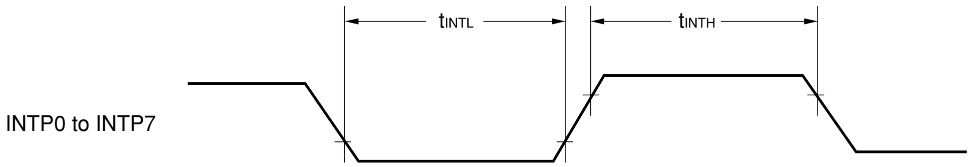
Data Retention Timing (STOP Mode Release by RESET)



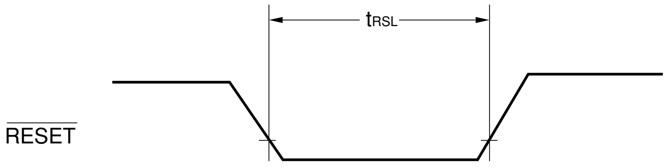
Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Request Signal)



Interrupt Request Input Timing



RESET Input Timing



Flash Memory Programming Characteristics (μPD78F0988A, 78F0988A(A) only)
(T_A = 10 to 40°C, V_{DD} = AV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, V_{PP} = 9.7 to 10.3 V)

(1) Write erase characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_X	4.5 V ≤ V_{DD} ≤ 5.5 V	1.0		10	MHz
		3.0 V ≤ V_{DD} < 4.5 V	1.0		8.38	MHz
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_X = 8.38 MHz			40	mA
V_{PP} supply current	I_{PP}	When V_{PP} = V_{PP2}			100	mA
Step erase time	T_{er}	**Note 1**	0.199	0.2	0.201	s
Overall erase time per area	T_{era}	When step erase time = 0.2 s **Note 2**			20	s/area
Writeback time	T_{wb}	**Note 3**	49.4	50	50.6	ms
Number of writebacks per writeback command	C_{wb}	When writeback time = 50 ms **Note 4**			60	Times/ write- back command
Number of erase/ writebacks	C_{erwb}				16	Times
Step write time	T_{wr}	**Note 5**	48	50	52	μs
Overall write time per word	T_{wrw}	When step write time = 50 μs (1 word = 1 byte) **Note 6**	48		520	μs/ word
Number of rewrites per area	C_{erwr}	1 erase + 1 write after erase = 1 rewrite **Note 7**	20			Times/ area

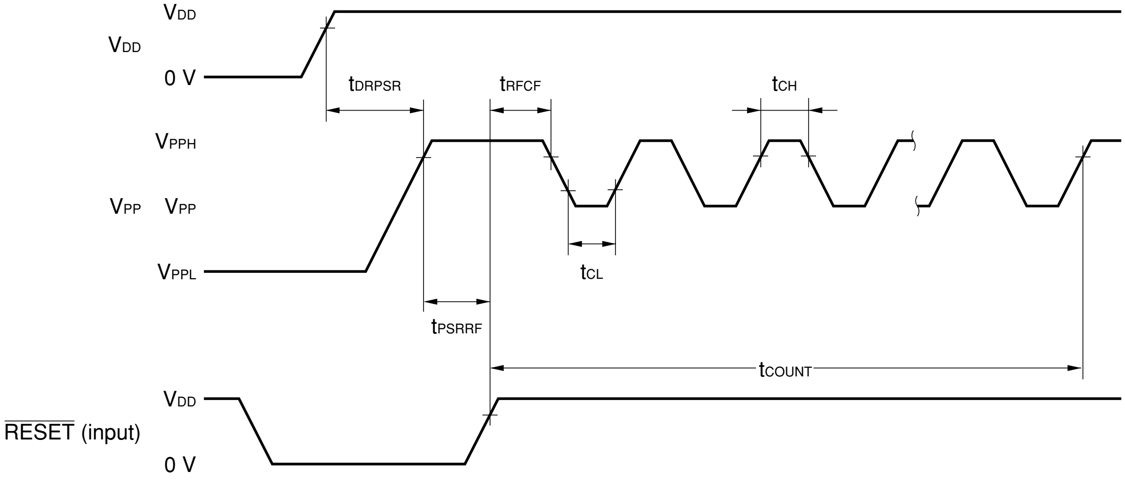
Notes
1. The recommended setting value for the step erase time is 0.2 s.
2. The prewrite time before erasure and the erase verify time (writeback time) is not included.
3. The recommended setting value for the writeback time is 50 ms.
4. Writeback is executed once by the issuance of the writeback command. Therefore, the number of retries must be the maximum value minus the number of commands issued.
5. Recommended step write time setting value 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

Remarks
1. The range of the operating clock during flash memory programming is not the same as the range during normal operation.
2. When using the PG-FP3, FL-PR3 (made by Naito Densei), PG-FP4, or FL-PR4 (made by Naito Densei) the time parameters that need to be downloaded from the parameter files for write/erase are automatically set. Unless otherwise directed, do not change the set values.

(2) Serial write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Set time from V_{DD}↑ to V_{PP}↑	t_{DRPSR}	V_{PP} high voltage	10			μs
Set time from V_{PP}↑ to $\overline{RESET}$↑	t_{PSRRF}	V_{PP} high voltage	1.0			μs
V_{PP} count start time from $\overline{RESET}$↑	t_{RFCF}	V_{PP} high voltage	1.0			μs
Count execution time	t_{COUNT}				20	ms
V_{PP} counter high-level width	t_{CH}		8.0			μs
V_{PP} counter low-level width	t_{CL}		8.0			μs
V_{PP} counter noise elimination width	t_{NFW}			40		ns

Flash Write Mode Setting Timing



Absolute Maximum Ratings (T_A = 25°C)

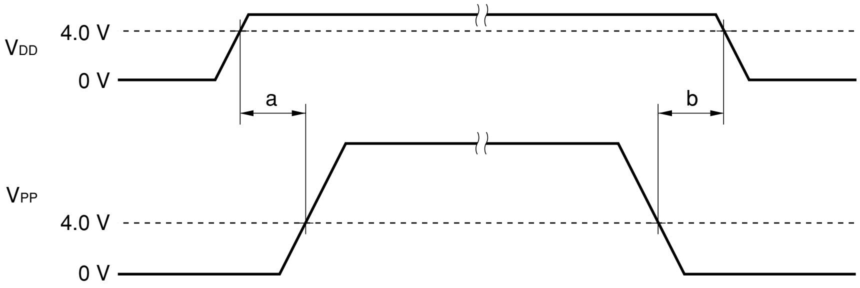
Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	V_{PP}	μPD78F0988A, 78F0988A(A) only **Note 1**		–0.3 to +10.5	V
	AV_{DD}			–0.3 to V_{DD} + 0.3	V
	AV_{REF}			–0.3 to V_{DD} + 0.3	V
	AV_{SS}			–0.3 to +0.3	V
Input voltage	V_I	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P50 to P57, P64 to P67, TO70 to TO75, X1, X2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3	V
Analog input voltage	V_{AN}	P10 to P17	Analog input pin	AV_{SS} – 0.3 to AV_{REF} + 0.3 and –0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin		–10	mA
		P00, P01, P30 to P37, P40 to P47, P50 to P57, P64 to P67 total		–15	mA
		P02, P03, P20 to P26, TO70 to TO75 total		–15	mA
Output current, low	I_{OL} **Note 2**	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P40 to P47, P64 to P67 per pin	Peak value	20	mA
			rms value	10	mA
		P50 to P57, TO70 to TO75 per pin	Peak value	30	mA
			rms value	15	mA
		P00, P01, P30 to P37, P40 to P47, P64 to P67 total	Peak value	50	mA
			rms value	20	mA
		P02, P03, P20 to P26 total	Peak value	30	mA
			rms value	15	mA
		TO70 to TO75 total	Peak value	100	mA
			rms value	70	mA
		P50 to P57 total	Peak value	100	mA
			rms value	70	mA
Operating ambient temperature	T_A	In normal operating mode		–40 to +85	°C
		In flash memory programming mode (μPD78F0988A, 78F0988A(A) only)		+10 to +40	°C
Storage temperature	T_{stg}	Mask ROM products		–65 to +150	°C
		Flash memory products		–40 to +125	°C

(The notes are explained on the following 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.

Notes 1. When writing in the flash memory, be sure to satisfy the following conditions on the V_{PP} voltage supply timing.

- At rising edge of power supply voltage
  More than 10 μs after V_{DD} reaches the lower limit voltage (4.0 V) of the operating voltage range, V_{PP} should exceed V_{DD} (a in the figure below).
- At falling edge of power supply voltage
  More than 10 μs after V_{PP} falls below the lower limit voltage (4.0 V) of the V_{DD} operating voltage range, start up V_{DD} (b in the figure below).



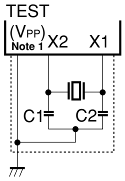
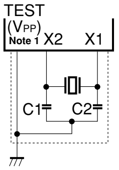
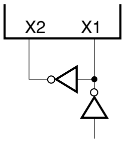
2. The rms value should be calculated as follows: [rms value] = [Peak value] × $\sqrt{\text{Duty}}$

Capacitance (T_A = 25°C, V_{DD} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_{IN}	f = 1 MHz Unmeasured pins returned to 0 V				15	pF
I/O capacitance	C_{IO}	f = 1 MHz Unmeasured pins returned to 0 V	P00 to P03, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67, TO70 to TO75			15	pF

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

System Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)Note 2		1.0		8.38	MHz
		Oscillation stabilization timeNote 3	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency (f_X)Note 2		1.0		8.38	MHz
		Oscillation stabilization timeNote 3	After V_{DD} reaches oscillation voltage range MIN.			10	ms
External clock		X1 input frequency (f_X)Note 2		1.0		8.38	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		50		500	ns

Notes
1. In the case of the μPD78F0988A and 78F0988A(A)
2. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.
3. Time required to stabilize oscillation after reset or STOP mode release.

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

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS1}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

Recommended Oscillator Constant

(1) μPD780982, 780983, 780984, 780986, 780988, 780982(A), 780983(A), 780984(A), 780986(A), 780988(A)

System clock: Ceramic resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Type	Recommended Circuit Constant		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Mfg. Co., Ltd.	CSTCC2M00G56-R0	2.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS2M00G56-B0	2.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCR4M00G53-R0	4.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS4M00G53-B0	4.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCR4M19G53-R0	4.19	SMD	On-chip	On-chip	4.0	5.5
	CSTLS4M19G53-B0	4.19	Lead	On-chip	On-chip	4.0	5.5
	CSTCR4M91G53-R0	4.91	SMD	On-chip	On-chip	4.0	5.5
	CSTLS4M91G53-B0	4.91	Lead	On-chip	On-chip	4.0	5.5
	CSTCR5M00G53-R0	5.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS5M00G53-B0	5.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCE8M00G52-R0	8.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS8M00G53-B0	8.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCE8M38G52-R0	8.38	SMD	On-chip	On-chip	4.0	5.5
	CSTLS8M38G53-B0	8.38	Lead	On-chip	On-chip	4.0	5.5

Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit.
Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD780988 Subseries within the specifications of the DC and AC characteristics.

(2) μPD78F0988A, 78F0988A(A)

System clock: Ceramic resonator (T_A = −40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Type	Recommended Circuit Constant		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Mfg. Co., Ltd.	CSTCC2M00G56-R0	2.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS2M00G56-B0	2.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCR4M00G53-R0	4.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS4M00G53-B0	4.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCR4M19G53-R0	4.19	SMD	On-chip	On-chip	4.0	5.5
	CSTLS4M19G53-B0	4.19	Lead	On-chip	On-chip	4.0	5.5
	CSTCR4M91G53-R0	4.91	SMD	On-chip	On-chip	4.0	5.5
	CSTLS4M91G53-B0	4.91	Lead	On-chip	On-chip	4.0	5.5
	CSTCR5M00G53-R0	5.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS5M00G53-B0	5.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCE8M00G52-R0	8.00	SMD	On-chip	On-chip	4.0	5.5
	CSTLS8M00G53-B0	8.00	Lead	On-chip	On-chip	4.0	5.5
	CSTLS8M00G53093-B0	8.00	Lead	On-chip	On-chip	4.0	5.5
	CSTCE8M38G52-R0	8.38	SMD	On-chip	On-chip	4.0	5.5
	CSTLS8M38G53-B0	8.38	Lead	On-chip	On-chip	4.0	5.5
	CSTLS8M38G53093-B0	8.38	Lead	On-chip	On-chip	4.0	5.5

Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit. Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD780988 Subseries within the specifications of the DC and AC characteristics.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P10 to P17, P21, P23, P30 to P37, P40 to P47, P50, P53, P64 to P67		$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P00 to P03, P20, P22, P24 to P26, P51, P52, P54 to P57		$0.8V_{DD}$		V_{DD}	V
	V_{IH3}	X1, X2		V_{DD} – 0.5		V_{DD}	V
Input voltage, low	V_{IL1}	P10 to P17, P21, P23, P30 to P37, P40 to P47, P50, P53, P64 to P67		0		$0.3V_{DD}$	V
	V_{IL2}	$\overline{\text{RESET}}$, P00 to P03, P20, P22, P24 to P26, P51, P52, P54 to P57		0		$0.2V_{DD}$	V
	V_{IL3}	X1, X2		0		0.4	V
Output voltage, high	V_{OH1}	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –1 mA		V_{DD} – 1.0		V_{DD}	V
		I_{OH} = –100 μA		V_{DD} – 0.5		V_{DD}	V
Output voltage, low	V_{OL1}	P50 to P57, TO70 to TO75	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 15 mA		0.4	2.0	V
		P00 to P03, P20 to P26, P30 to P37, P40 to P47, P64 to P67	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 1.6 mA			0.4	V
	V_{OL2}	I_{OL} = 400 μA				0.5	V
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67, TO70 to TO75, $\overline{\text{RESET}}$			3	μA
	I_{LIH2}		X1, X2			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	P00 to P03, P10 to P17, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67, TO70 to TO75, $\overline{\text{RESET}}$			–3	μA
	I_{LIL2}		X1, X2			–20	μA
Output leakage current, high	I_{LOH}	V_{OUT} = V_{DD}				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				–3	μA
Software pull-up resistor	R_2	V_{IN} = 0 V P00 to P03, P20 to P26, P30 to P37, P40 to P47, P50 to P57, P64 to P67		15	30	90	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V) (2/2)

(1) μPD780982, 780983, 780984, 780986, 780988, 780982(A), 780983(A), 780984(A), 780986(A), 780988(A)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current	I_{DD1}	8.38 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%$^{\text{Note 1}}$	When A/D converter stopped		6.5	$13^{\text{Note 2}}$	mA
				When A/D converter operating		7.5	$15^{\text{Note 2}}$	mA
	I_{DD2}	8.38 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%$^{\text{Note 1}}$	When peripheral function stopped		1	2	mA
				When peripheral function operating			7	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA

Notes 1. High-speed mode operation (when the processor clock control register (PCC) is set to 00H).

2. Refers to the total current flowing to the internal power supply (V_{DD0} and V_{DD1}). The peripheral operation current is included, but the current flowing to the pull-up resistors of ports and the AV_{REF} pin is not.

(2) μPD78F0988A, 78F0988A(A)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current	I_{DD1}	8.38 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%$^{\text{Note 1}}$	When A/D converter stopped		15	$25^{\text{Note 2}}$	mA
				When A/D converter operating		16	$27^{\text{Note 2}}$	mA
	I_{DD2}	8.38 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%$^{\text{Note 1}}$	When peripheral function stopped		1.3	2.6	mA
				When peripheral function operating			7.3	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA
V_{PP} supply voltage	V_{PP1}	In normal operation mode			0		$0.2V_{DD}$	V

Notes 1. High-speed mode operation (when the processor clock control register (PCC) is set to 00H).

2. Refers to the total current flowing to the internal power supply (V_{DD0} and V_{DD1}). The peripheral operation current is included, but the current flowing to the pull-up resistors of ports and the AV_{REF} pin is not.

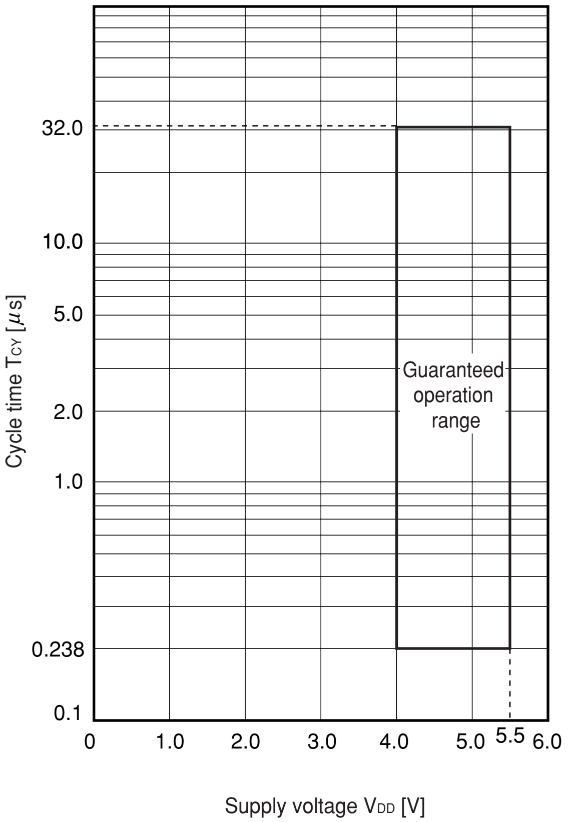
AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (Min. instruction execution time)	T_{CY}	Operating with system clock	0.238		32	μs
TI000, TI001, TI010, TI011 input frequency	f_{TI0}		0		$f_X/64$	MHz
TI000, TI001, TI010, TI011 input high-/low-level width	t_{TIH0} t_{TIL0}		$2/f_{sam} + 0.1^{\text{Note}}$			μs
TI50, TI51, TI52 input frequency	f_{TI5}	8-/16-bit precision	0		4	MHz
TI50, TI51, TI52 input high-/low-level width	t_{TIH5} t_{TIL5}	8-/16-bit precision	100			ns
Interrupt request input high-/low-level width	t_{INTH} t_{INTL}	INTP0 to INTP7	1			μs
TOFF input high-/low-level width	t_{TOFFH} t_{TOFFL}		2			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}		10			μs

Note Selection of f_{sam} = f_X, $f_X/4$, $f_X/32$ is possible with bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00) or with bits 0 and 1 (PRM010, PRM011) of prescaler mode register 01 (PRM01). Note that when selecting TI000 (TM00) or TI001 (TM01) valid edge as the count clock, f_{sam} = $f_X/16$.

T_{CY} vs V_{DD} (System clock operation)



(2) Read/write operation (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

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 $\overline{\text{RD}}$↓	t_{RDAD}		0	100	ns
Data input time from $\overline{\text{RD}}$↓	t_{RDD1}			$(2 + 2n)t_{CY} – 87$	ns
	t_{RDD2}			$(3 + 2n)t_{CY} – 93$	ns
Read data hold time	t_{RDH}		0		ns
$\overline{\text{RD}}$ low-level width	t_{RDL1}		$(1.5 + 2n)t_{CY} – 33$		ns
	t_{RDL2}		$(2.5 + 2n)t_{CY} – 33$		ns
$\overline{\text{WAIT}}$↓ input time from $\overline{\text{RD}}$↓	t_{RDWT1}			$t_{CY} – 43$	ns
	t_{RDWT2}			$t_{CY} – 43$	ns
$\overline{\text{WAIT}}$↓ input time from $\overline{\text{WR}}$↓	t_{WRWT}			$t_{CY} – 25$	ns
$\overline{\text{WAIT}}$ 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
$\overline{\text{WR}}$ low-level width	t_{WRL}		$(1.5 + 2n)t_{CY} – 15$		ns
Delay time from ASTB↓ to $\overline{\text{RD}}$↓	t_{ASTRD}		6		ns
Delay time from ASTB↓ to $\overline{\text{WR}}$↓	t_{ASTWR}		$2t_{CY} – 15$		ns
Delay time from $\overline{\text{RD}}$↑ at external fetch to ASTB↑	t_{RDAST}		$0.8t_{CY} – 15$	$1.2t_{CY}$	ns
Address hold output time from $\overline{\text{WR}}$↑	t_{WRADH}		$0.8t_{CY} – 15$	$1.2t_{CY} + 30$	ns
Write data output time from $\overline{\text{RD}}$↑	t_{RDWD}		40		ns
Write data output time from $\overline{\text{WR}}$↓	t_{WRWD}		10	60	ns
Address hold time from $\overline{\text{RD}}$↑ at external fetch	t_{RDADH}		$0.8t_{CY} – 15$	$1.2t_{CY} + 30$	ns
Delay time from $\overline{\text{WAIT}}$↑ to $\overline{\text{RD}}$↑	t_{WTRD}		$0.8t_{CY}$	$2.5t_{CY} + 25$	ns
Delay time from $\overline{\text{WAIT}}$↑ to $\overline{\text{WR}}$↑	t_{WTWR}		$0.8t_{CY}$	$2.5t_{CY} + 25$	ns

Remarks **1.** $t_{CY} = T_{CY}/4$

2. n indicates the number of waits.

3. C_L = 100 pF (C_L is the load capacitance of the AD0 to AD7, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, and ASTB pins.)

Caution **T_{CY} can only be used when the MIN. value is 0.238 μs.**

(3) Serial interface (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

(a) 3-wire serial I/O mode ($\overline{\text{SCK}}$... Internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK}}$ cycle time	t_{KCY1}		954			ns
$\overline{\text{SCK}}$ high-/low-level width	t_{KH1} t_{KL1}		$t_{KCY1}/2 - 50$			ns
SI setup time (to $\overline{\text{SCK}}$↑)	t_{SIK1}		100			ns
SI hold time (from $\overline{\text{SCK}}$↑)	t_{KSI1}		400			ns
Delay time from $\overline{\text{SCK}}$↓ to SO output	t_{KSO1}	C = 100 pF Note			300	ns

Note C is the load capacitance of the $\overline{\text{SCK}}$ and SO output lines.

(b) 3-wire serial I/O mode ($\overline{\text{SCK}}$... External clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK}}$ cycle time	t_{KCY2}		800			ns
$\overline{\text{SCK}}$ high-/low-level width	t_{KH2} t_{KL2}		400			ns
SI setup time (to $\overline{\text{SCK}}$↑)	t_{SIK2}		100			ns
SI hold time (from $\overline{\text{SCK}}$↑)	t_{KSI2}		400			ns
Delay time from $\overline{\text{SCK}}$↓ to SO output	t_{KSO2}	C = 100 pF Note			300	ns

Note C is the load capacitance of the $\overline{\text{SCK}}$ and SO output lines.

(c) UART mode (UART00) (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					125000	bps

(d) UART mode (UART00) (Infrared data transfer mode)

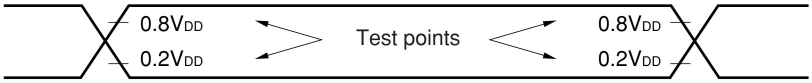
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					115200	bps
Bit rate allowable error					±0.87	%
Output pulse width			1.2		$0.24/f_{br}$ Note	μs
Input pulse width			$4/f_X$			μs

Note fbr: Set baud rate

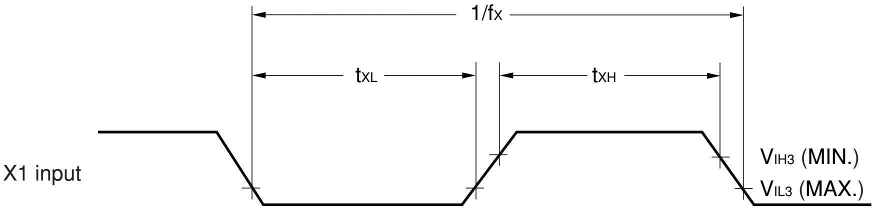
(e) UART mode (UART01) (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					38400	bps

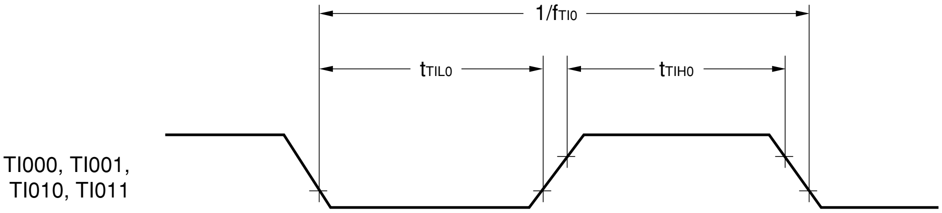
AC Timing Test Points (Excluding X1 Input)

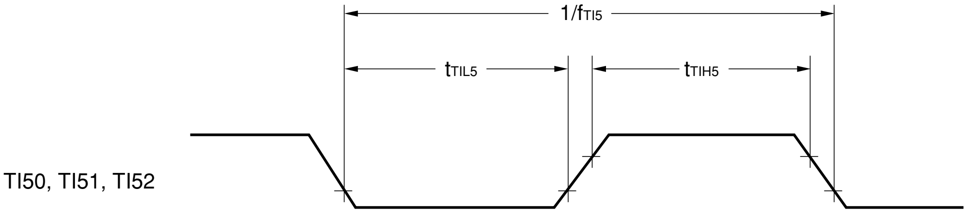


Clock Timing

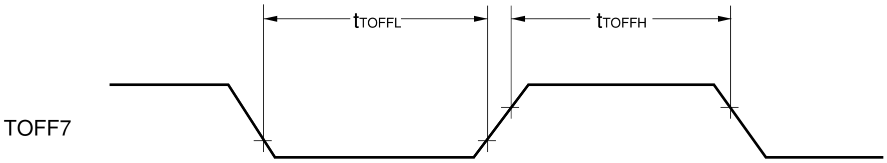


TI Timing



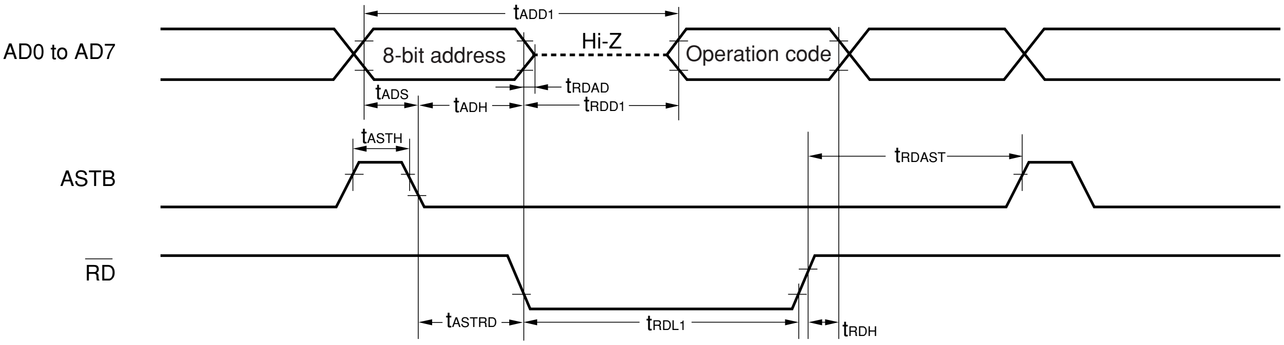


TOFF Timing

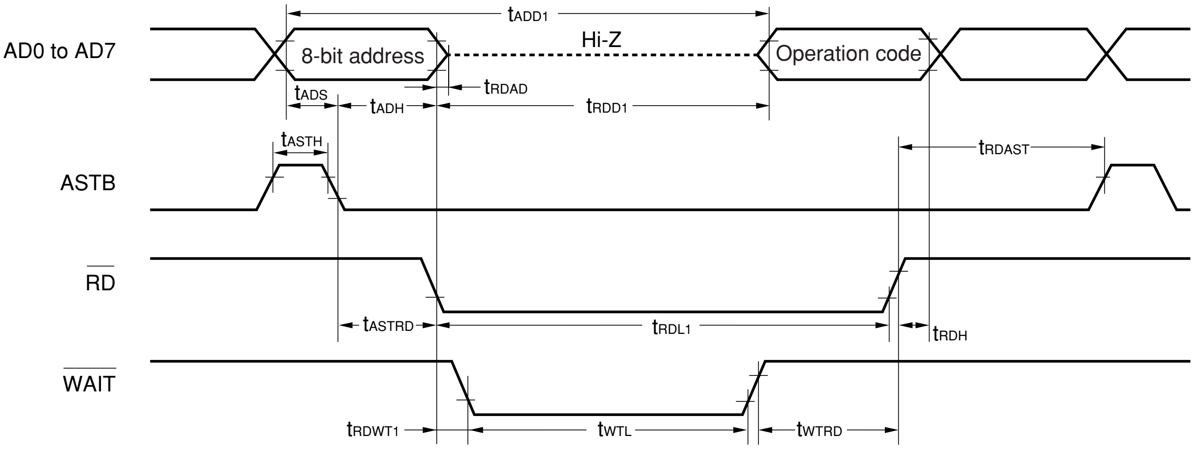


Read/Write Operation

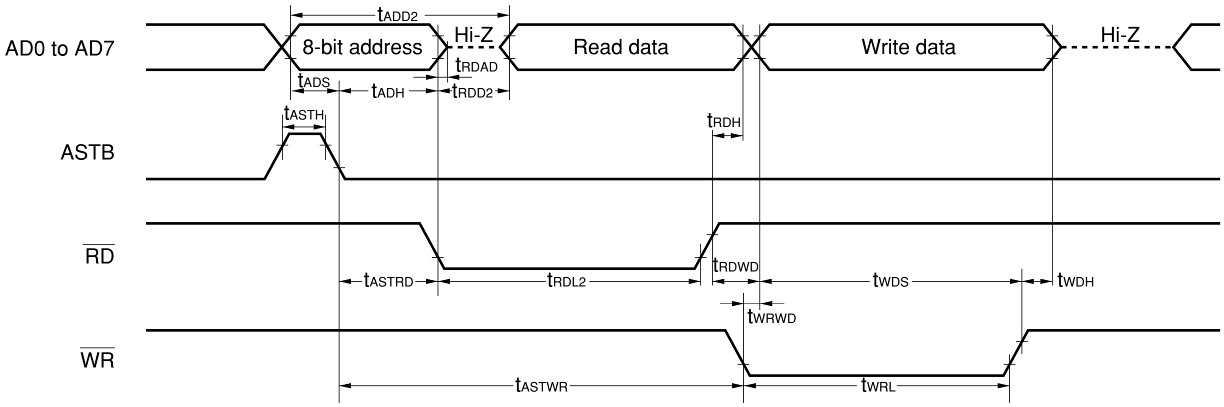
External fetch (no wait):



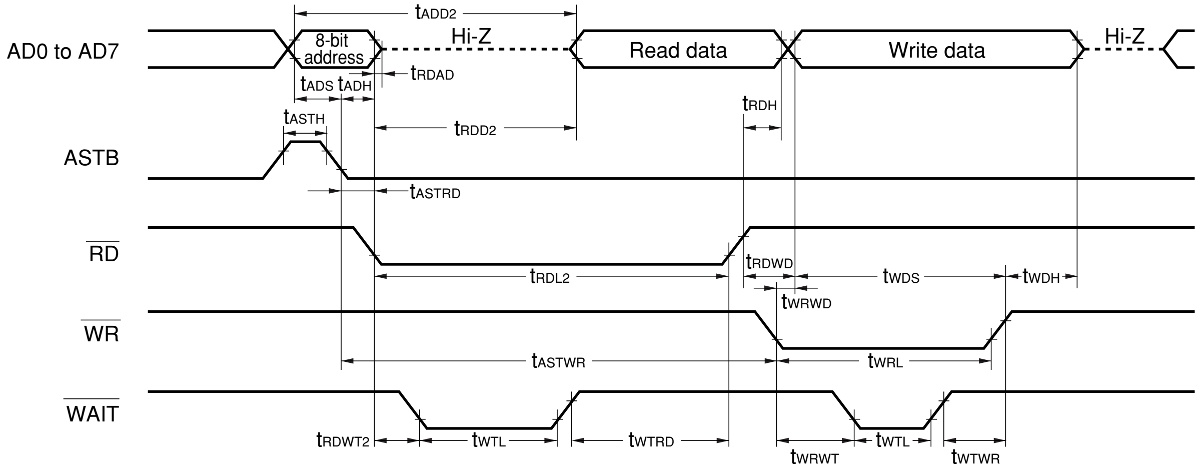
External fetch (wait insertion):



External data access (no wait):

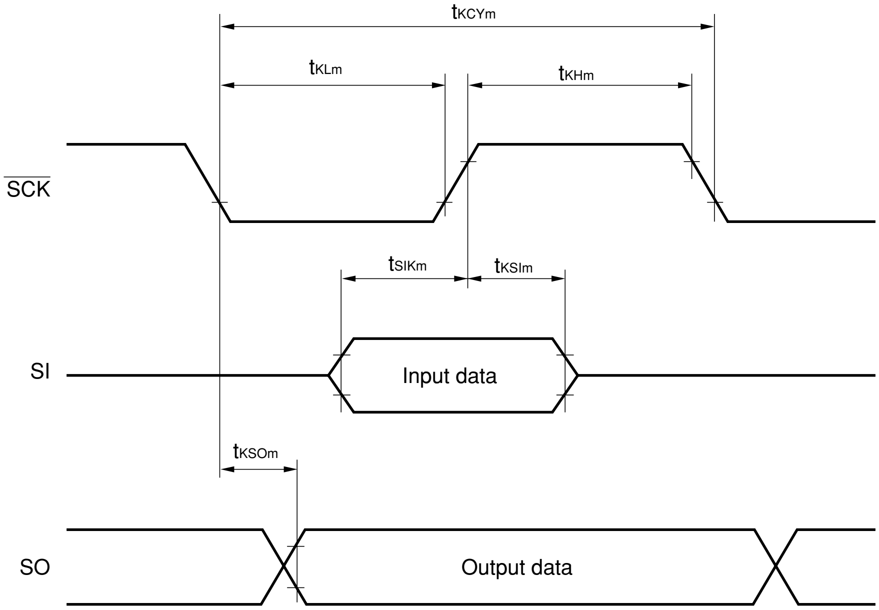


External data access (wait insertion):



Serial Transfer Timing

3-wire serial I/O mode:



m = 1, 2

A/D Converter Characteristics (T_A = –40 to +85°C, V_{DD} = AV_{DD} = 4.0 to 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 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_{DD} ≤ 5.5 V	14		96	μs
Zero-scale error[Note 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[Note 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 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 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}		0		AV_{REF}	V
Reference voltage	AV_{REF}		2.7		AV_{DD}	V
Resistance between AV_{REF} and AV_{SS}	R_{REF}	When A/D converter is not operating	20	40		kΩ

Notes 1. Excludes quantization error (±1/2 LSB).

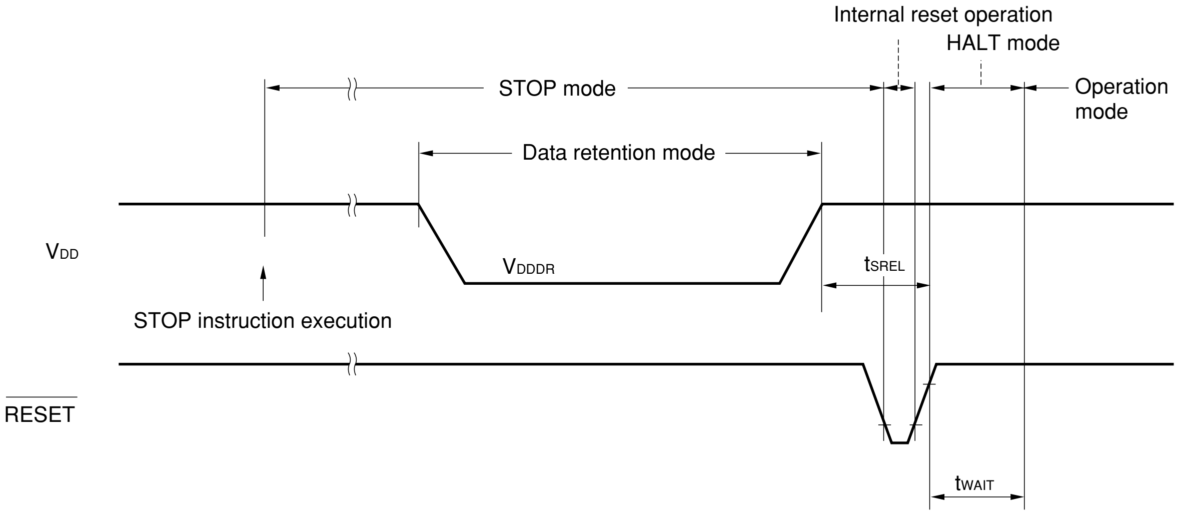
2. This value is indicated as a ratio (%FSR) to the full-scale value.

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

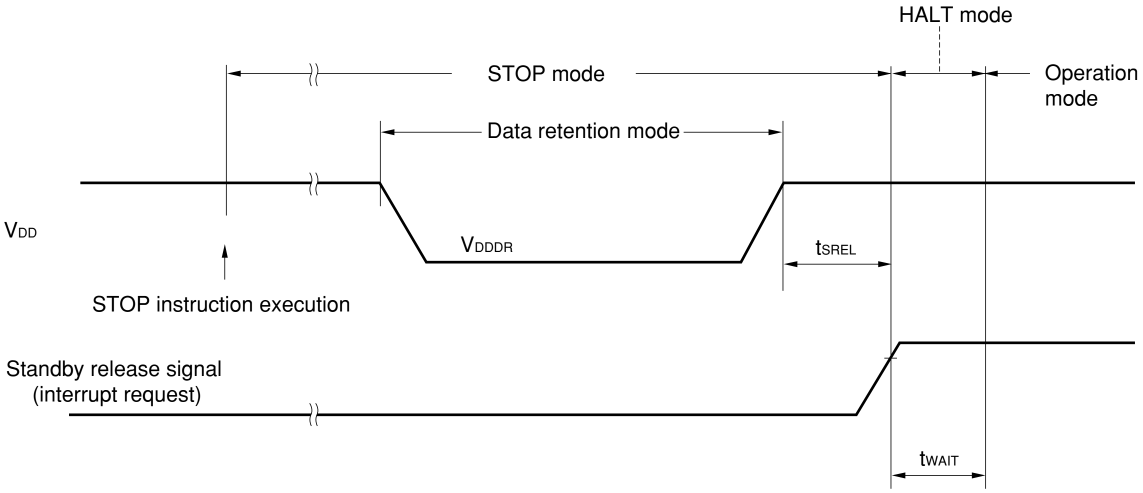
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		2.0		5.5	V
Data retention power supply current	I_{DDDR}	V_{DDDR} = 2.0 V		0.1	10	μA
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{17}/f_X$		ms
		Release by interrupt request		**Note**		ms

Note Selection of $2^{12}/f_X$ and $2^{14}/f_X$ to $2^{17}/f_X$ is possible with bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

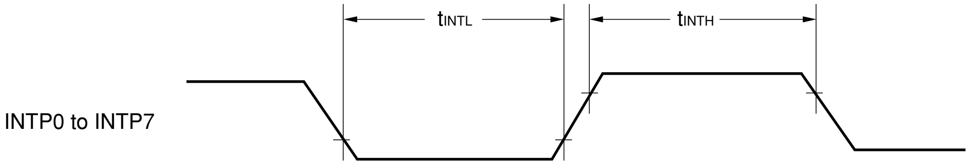
Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)



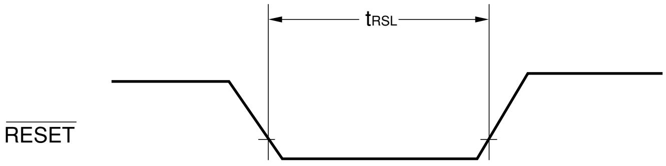
Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Request Signal)



Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



Flash Memory Programming Characteristics (μPD78F0988A, 78F0988A(A) only)
(T_A = 10 to 40°C, V_{DD} = AV_{DD} = 4.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, V_{PP} = 9.7 to 10.3 V)

(1) Write erase characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_X		1.0		8.38	MHz
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_X = 8.38 MHz			40	mA
V_{PP} supply current	I_{PP}	When V_{PP} = V_{PP2}			100	mA
Step erase time	T_{er}	**Note 1**	0.199	0.2	0.201	s
Overall erase time per area	T_{era}	When step erase time = 0.2 s **Note 2**			20	s/area
Writeback time	T_{wb}	**Note 3**	49.4	50	50.6	ms
Number of writebacks per writeback command	C_{wb}	When writeback time = 50 ms **Note 4**			60	Times/ write-back command
Number of erase/ writebacks	C_{erwb}				16	Times
Step write time	T_{wr}	**Note 5**	48	50	52	μs
Overall write time per word	T_{wrw}	When step write time = 50 μs (1 word = 1 byte) **Note 6**	48		520	μs/ word
Number of rewrites per area	C_{erwr}	1 erase + 1 write after erase = 1 rewrite **Note 7**	20			Times/ area

Notes

1. The recommended setting value for the step erase time is 0.2 s.
2. The prewrite time before erasure and the erase verify time (writeback time) is not included.
3. The recommended setting value for 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. Recommended step write time setting value 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

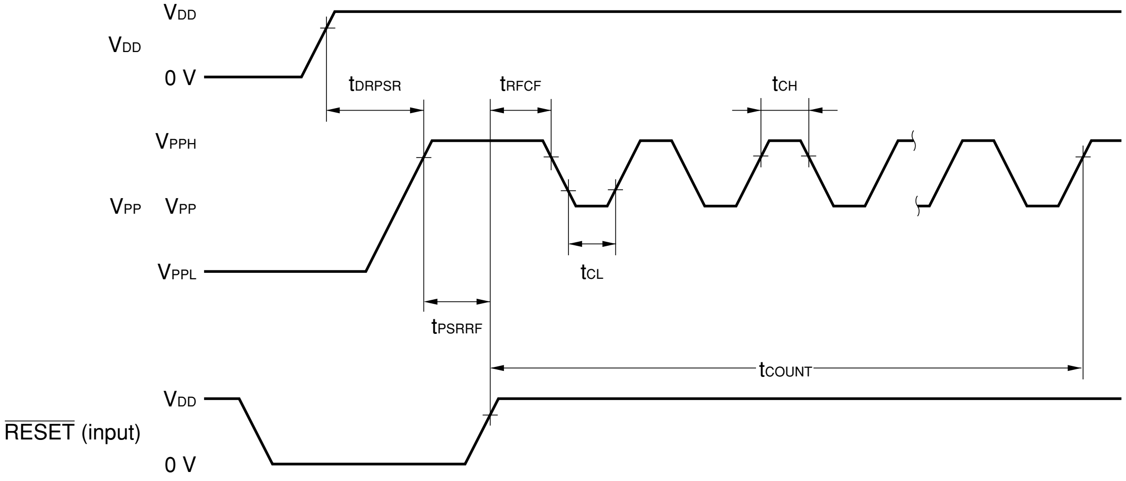
Remarks

1. The range of the operating clock during flash memory programming is the same as the range during normal operation.
2. When using the PG-FP3, FL-PR3 (made by Naito Densei), PG-FP4, or FL-PR4 (made by Naito Densei) the time parameters that need to be downloaded from the parameter files for write/erase are automatically set. Unless otherwise directed, do not change the set values.

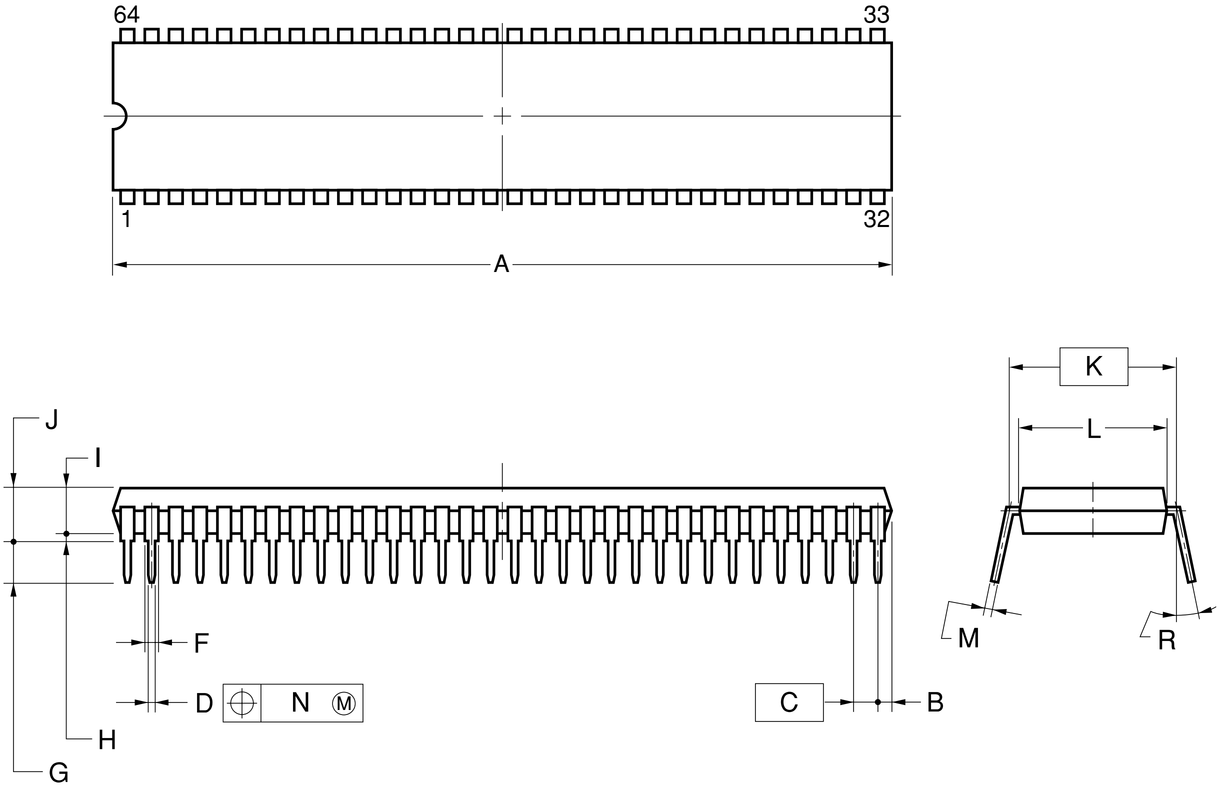
(2) Serial write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Set time from V_{DD}↑ to V_{PP}↑	t_{DRPSR}	V_{PP} high voltage	10			μs
Set time from V_{PP}↑ to $\overline{RESET}$↑	t_{PSRRF}	V_{PP} high voltage	1.0			μs
V_{PP} count start time from $\overline{RESET}$↑	t_{RFCF}	V_{PP} high voltage	1.0			μs
Count execution time	t_{COUNT}				20	ms
V_{PP} counter high-level width	t_{CH}		8.0			μs
V_{PP} counter low-level width	t_{CL}		8.0			μs
V_{PP} counter noise elimination width	t_{NFW}			40		ns

Flash Write Mode Setting Timing



64-PIN PLASTIC SDIP (19.05mm(750))



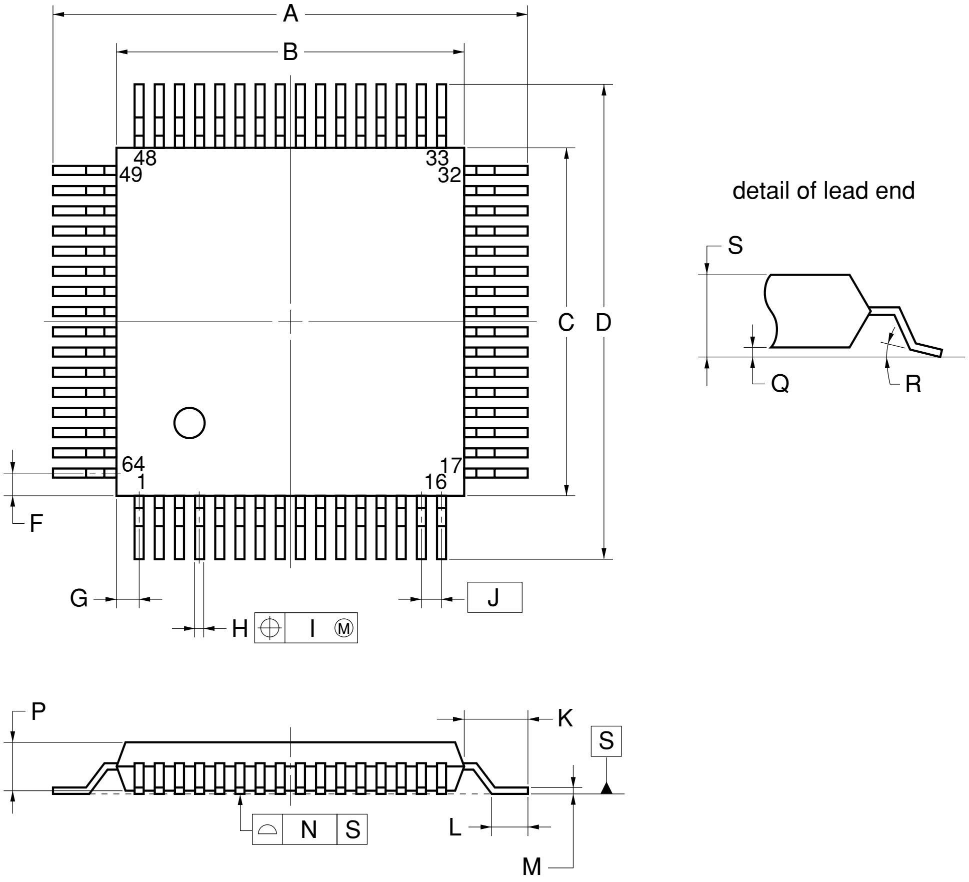
NOTES

1. Each lead centerline is located within 0.17 mm of its true position (T.P.) at maximum material condition.
2. Item "K" to center of leads when formed parallel.

ITEM	MILLIMETERS
A	$58.0^{+0.68}_{-0.20}$
B	1.78 MAX.
C	1.778 (T.P.)
D	0.50±0.10
F	0.9 MIN.
G	3.2±0.3
H	0.51 MIN.
I	$4.05^{+0.26}_{-0.20}$
J	5.08 MAX.
K	19.05 (T.P.)
L	17.0±0.2
M	$0.25^{+0.10}_{-0.05}$
N	0.17
R	0 ~ 15°

P64C-70-750A,C-4

64-PIN PLASTIC QFP (14x14)



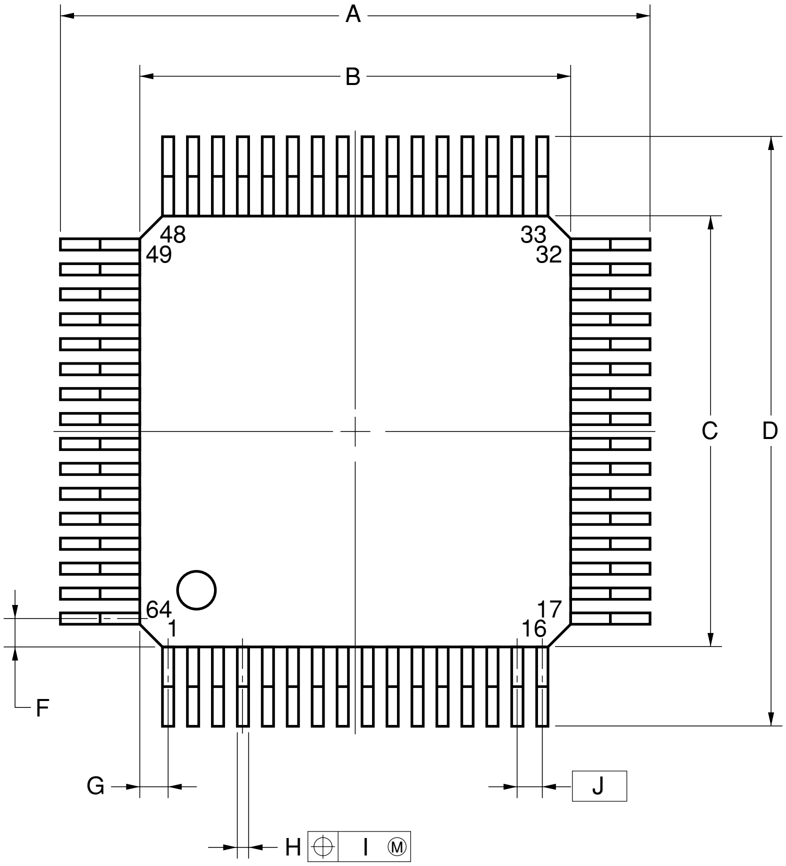
NOTE

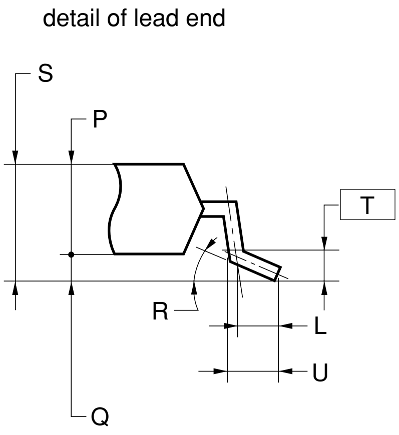
Each lead centerline is located within 0.15 mm of its true position (T.P.) at maximum material condition.

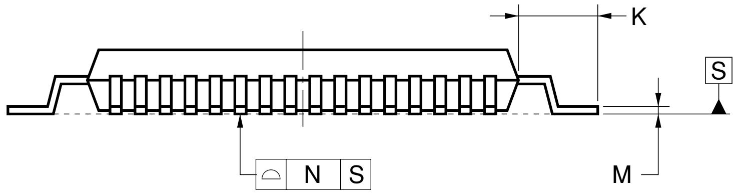
ITEM	MILLIMETERS
A	17.6±0.4
B	14.0±0.2
C	14.0±0.2
D	17.6±0.4
F	1.0
G	1.0
H	$0.37^{+0.08}_{-0.07}$
I	0.15
J	0.8 (T.P.)
K	1.8±0.2
L	0.8±0.2
M	$0.17^{+0.08}_{-0.07}$
N	0.10
P	2.55±0.1
Q	0.1±0.1
R	5°±5°
S	2.85 MAX.

P64GC-80-AB8-5

64-PIN PLASTIC LQFP (14x14)







ITEM	MILLIMETERS
A	17.2±0.2
B	14.0±0.2
C	14.0±0.2
D	17.2±0.2
F	1.0
G	1.0
H	$0.37^{+0.08}_{-0.07}$
I	0.20
J	0.8 (T.P.)
K	1.6±0.2
L	0.8
M	$0.17^{+0.03}_{-0.06}$
N	0.10
P	1.4±0.1
Q	0.127±0.075
R	$3°^{+4°}_{-3°}$
S	1.7 MAX.
T	0.25
U	0.886±0.15

P64GC-80-8BS

NOTE

Each lead centerline is located within 0.20 mm of its true position (T.P.) at maximum material condition.

★

CHAPTER 23 RECOMMENDED SOLDERING CONDITIONS

This product should be soldered and mounted under the following recommended conditions.
For soldering methods and conditions other than those recommended below, contact an NEC sales representative.
For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 23-1. Surface Mounting Type Soldering Conditions (1/2)

(1) μPD78F0988AGC-AB8: 64-pin plastic QFP (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Three times or less	IR35-00-3
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Three times or less	VP15-00-3
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	—

Caution Do not use different soldering methods together (except for partial heating).

(2) μPD780982GC-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780983GC-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780984GC-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780986GC-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780988GC-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780982GC(A)-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780983GC(A)-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780984GC(A)-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780986GC(A)-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD780988GC(A)-xxx-8BS:	**64-pin plastic LQFP (14 × 14)**
μPD78F0988AGC(A)-AB8:	**64-pin plastic QFP (14 × 14)**

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Two times or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Two times or less	VP15-00-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	—

Caution Do not use different soldering methods together (except for partial heating).

Table 23-1. Surface Mounting Type Soldering Conditions (2/2)

(3) μPD780982GC-xxx-8BS-A: 64-pin plastic LQFP (14 × 14)
μPD780983GC-xxx-8BS-A: 64-pin plastic LQFP (14 × 14)
μPD780984GC-xxx-8BS-A: 64-pin plastic LQFP (14 × 14)
μPD780986GC-xxx-8BS-A: 64-pin plastic LQFP (14 × 14)
μPD780988GC-xxx-8BS-A: 64-pin plastic LQFP (14 × 14)
μPD78F0988AGC-AB8-A: 64-pin plastic SDIP (19.05 mm (750))
μPD78F0988AGC-8BS-A: 64-pin plastic LQFP (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Wave soldering	For details, contact an NEC Electronics sales representative.	—
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	—

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remark Products that have the part numbers suffixed by “-A” are lead-free products.

Table 23-2. Insertion Type Soldering Conditions

(1) μPD780982CW-xxx: 64-pin plastic SDIP (19.05 mm (750))
μPD780983CW-xxx: 64-pin plastic SDIP (19.05 mm (750))
μPD780984CW-xxx: 64-pin plastic SDIP (19.05 mm (750))
μPD780986CW-xxx: 64-pin plastic SDIP (19.05 mm (750))
μPD780988CW-xxx: 64-pin plastic SDIP (19.05 mm (750))
μPD78F0988ACW: 64-pin plastic SDIP (19.05 mm (750))

Soldering Method	Soldering Condition
Wave soldering (only for pins)	Solder bath temperature: 260°C max., Time: 10 seconds max.
Partial heating	Pin temperature: 300°C max., Time: 3 seconds max. (per pin row)

Caution   Apply wave soldering only to the pins and be careful not to bring solder into direct contact with the package.

(2) μPD780982CW-xxx-A: 64-pin plastic SDIP (19.05 mm (750))
μPD780983CW-xxx-A: 64-pin plastic SDIP (19.05 mm (750))
μPD780984CW-xxx-A: 64-pin plastic SDIP (19.05 mm (750))
μPD780986CW-xxx-A: 64-pin plastic SDIP (19.05 mm (750))
μPD780988CW-xxx-A: 64-pin plastic SDIP (19.05 mm (750))
μPD78F0988ACW-A: 64-pin plastic SDIP (19.05 mm (750))

Soldering Method	Soldering Condition
Wave soldering (only for pins)	For details, contact an NEC Electronics sales representative.
Partial heating	Pin temperature: 300°C max., Time: 3 seconds max. (per pin row)

Caution   Only the pins of the THD are heated when performing wave soldering.
Make sure that flow solder does not come in contact with the packge.

Remark   Products that have the part numbers suffixed by “-A” are lead-free products.

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD780988 Subseries. Figure A-1 shows the configuration example of the tools.

- **Support for PC98-NX series**

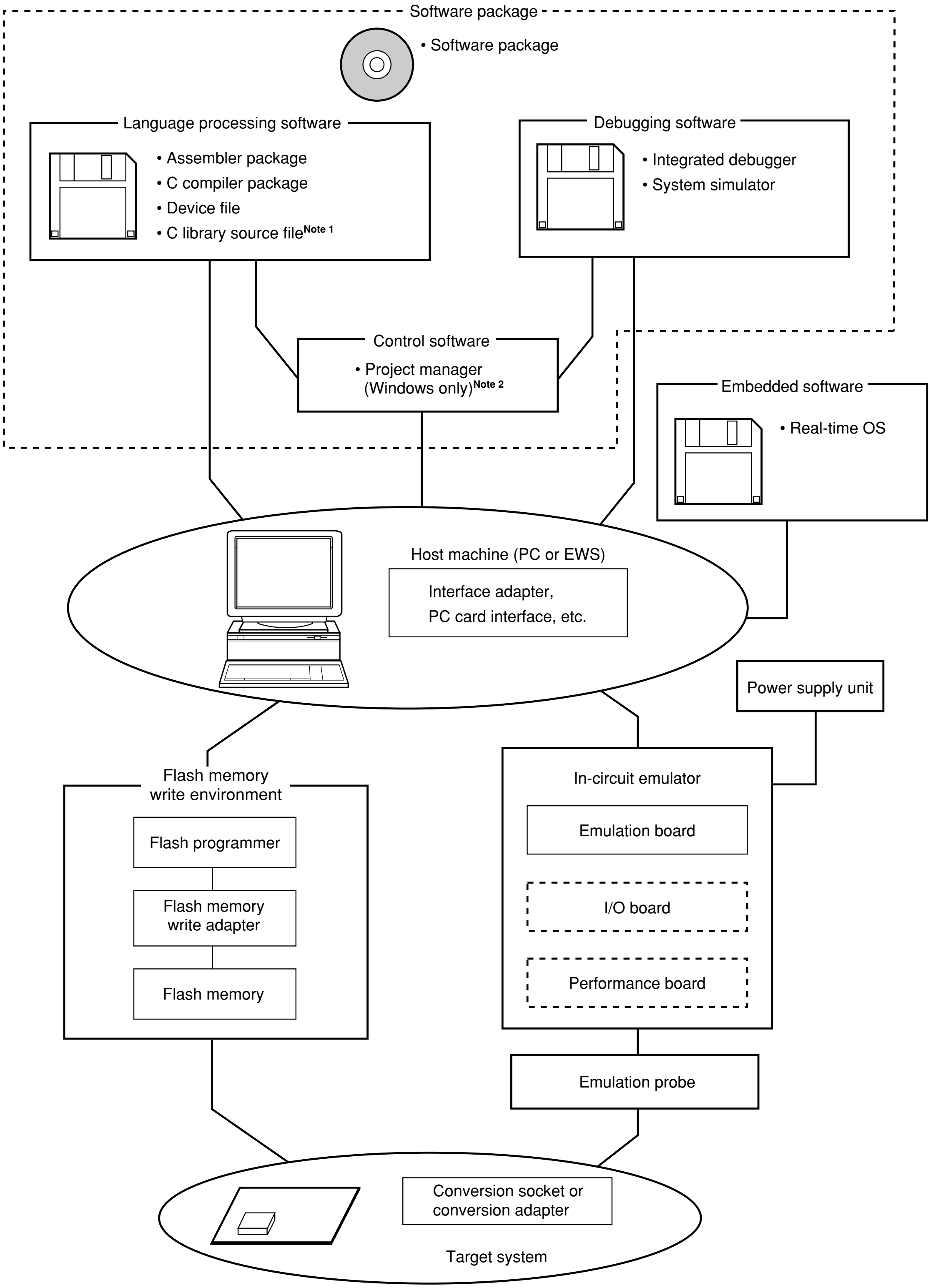
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles can be used for PC98-NX series computers. When using PC98-NX series computers, refer to the description for IBM PC/AT compatibles.

- **Windows**

  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows 2000
  - Windows NT™ Ver. 4.0

★

Figure A-1. Configuration of Development Tools



Notes 1. The C library source file is not included in the software package.

2. The project manager is included in the assembler package.
The project manager is only used for Windows.

★ A.1 Software Package

SP78K0 Software package	This package contains various software tools for 78K/0 Series development. The following tools are included. RA78K0, CC78K0, ID78K0-NS, SM78K0, and various device files
	Part Number: μS××××SP78K0

Remark ×××× in the part number differs depending on the OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

RA78K0 Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with an optional device file (DF780988). **<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 optional assembler package and device file. **<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
DF780988**Note 1** Device file	This file contains information peculiar to the device. This device file should be used in combination with an optional tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, ID78K0, and RX78K0). Corresponding OS and host machine differ depending on the tool used.
	Part Number: μS××××DF780988
CC78K0-L**Note 2** C library source file	This is a source file of functions configuring the object library included in the C compiler package. This file is required to match the object library included in C compiler package to the user's specifications. It does not depend on the operating environment because it is a source file.
	Part Number: μS××××CC78K0-L

Notes 1. The DF780988 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, ID78K0, and RX78K0.

★ **2.** 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$^{\text{TM}}$	HP-UX$^{\text{TM}}$ (Rel. 10.10)	
3K17	SPARCstation$^{\text{TM}}$	SunOS$^{\text{TM}}$ (Rel. 4.1.4), Solaris$^{\text{TM}}$ (Rel. 2.5.1)	

μS××××DF780988
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

★ A.3 Control Software

Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0). It can only be used in Windows.

★ A.4 Flash Memory Writing Tools

Flashpro III (Part number: FL-PR3, PG-FP3) Flashpro IV (Part number: FL-PR4, PG-FP4) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-64CW FA-64GC FA-64GC-8BS-A Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro III/Flashpro IV. • FA-64CW: 64-pin plastic SDIP (CW type) • FA64-GC: 64-pin plastic QFP (GC-AB8 type) • FA-64GC-8BS-A: 64-pin plastic LQFP (GC-8BS type)

Remark FL-PR3, FL-PR4, FA-64CW, FA-64GC, and FA-64GC-8BS-A are products of Naito Densei Machida Mfg. Co., Ltd.
Contact: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

A.5 Debugging Tools (Hardware)

★ A.5.1 When using the in-circuit emulator IE-78K0-NS or IE-78K0-NS-A

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 is supported by the integrated debugger (ID78K0-NS). This emulator should be used in combination with a power supply unit, emulation probe, and interface adapter, which is required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance board		This board is used to enhance the functions of the IE-78K0-NS. By connecting this board to the IE-78K0-NS-PA before use, debugging functions, such as coverage function addition and the enhancement of tracer and timer functions, are enhanced.
IE-78K0-NS-A In-circuit emulator		In-circuit emulator that combines 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 receptacle of 100 V to 240 V AC.
IE-70000-98-IF-C Interface adapter		This adapter is required when using a PC-9800 Series computer (except notebook type) as the host machine (C bus supported).
IE-70000-CD-IF-A PC card interface		This PC card and interface cable are required when using a notebook-type computer as the host machine (PCMCIA socket supported).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using an IBM PC/AT or compatible computer as the host machine (ISA bus supported).
IE-70000-PCI-IF-A Interface adapter		This adapter is required when using a PCI bus integrated computer as the host machine.
IE-780988-NS-EM4 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 and IE-78K0-NS-P01.
	IE-78K0-NS-P01 I/O board	This board is used in combination with the IE-780988-NS-EM4 and IE-78K0-NS to perform emulation.
NP-64CW Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for a 64-pin plastic SDIP (CW type).
NP-64GC-TQ NP-H64GC-TQ Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for a 64-pin plastic QFP (GC-AB8 type) and 64-pin plastic LQFP (GC-8BS type).
	TGC-064SAP Conversion adapter (Refer to **Figure A-4**)	This conversion adapter connects the NP-64GC-TQ and NP-H64GC-TQ to the target system board designed to mount a 64-pin plastic QFP (GC-AB8 type) and 64-pin plastic LQFP (GC-8BS type).

Remarks 1. NP-64CW, NP-64GC, NP-64GC-TQ, and NP-H64GC-TQ are products of Naito Densei Machida Mfg. Co., Ltd.
For further information, contact Naito Densei Machida Mfg. Co., Ltd. (+81-45-475-4191)

2. TGC-064SAP is a product made by TOKYO ELETECH CORPORATION.
For further information, contact Daimaru Kogyo Co., Ltd.
Tokyo Electronics Department (+81-3-3820-7112)
Osaka Electronics Department (+81-6-6244-6672)

3. The TGC-064SAP is sold in single units.

A.5.2 When using the in-circuit emulator IE-78001-R-A

IE-78001-R-A In-circuit emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It is supported by the integrated debugger (ID78K0). This emulator should be used in combination with an emulation probe and interface adapter, which is required to connect this emulator to the host machine.
IE-70000-98-IF-C Interface adapter		This adapter is required when using a PC-9800 Series computer (except notebook type) as the IE-78001-R-A host machine (C bus supported).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using an IBM PC/AT or compatible computer as the IE-78001-R-A host machine (ISA bus supported).
★ IE-70000-PCI-IF-A Interface adapter		This adapter is required when using a PCI bus integrated computer as the IE-78001-R-A host machine.
IE-78000-R-SV3 Interface adapter		This is the adapter and cable required when using an EWS computer as the IE-78001-R-A host machine, and is used connected to the board in the IE-78001-R-A. 10Base-5 is supported for Ethernet^{TM}. For other methods, a commercially available conversion adapter is required.
IE-780988-NS-EM4 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, IE-78K0-NS-P01, and IE-78K0-R-EX1.
	IE-78K0-NS-P01 I/O board	This board is used in combination with the IE-780988-NS-EM4 and IE-78K0-NS to perform emulation.
IE-78K0-R-EX1 Emulation probe conversion board		This board is required when using the IE-780988-NS-EM4 and IE-78K0-NS-P01 on the IE-78001-R-A.
EP-78240CW-R Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for a 64-pin plastic SDIP (CW type).
EP-78240GC-R Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for a 64-pin plastic QFP (GC-AB8 type).
	EV-9200GC-64 Conversion socket (Refer to **Figures A-2** and **A-3**)	This conversion socket connects the EP-78240GC-R to the target system board designed to mount a 64-pin plastic QFP (GC-AB8 type).

Remark The EV-9200GC-64 is sold in packages of 5 units.

A.6 Debugging Tools (Software)

SM78K0 System simulator	This is a system simulator for the 78K/0 Series. The SM78K0 is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with the device file (DF780988) (sold separately).
	Part Number: μS××××SM78K0
ID78K0-NS Integrated debugger (supporting in-circuit emulators IE-78K0-NS and IE-78K0-NS-A)	This debugger supports the in-circuit emulators for the 78K/0 Series. The ID78K0-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file (DF780988) (sold separately).
ID78K0 Integrated debugger (supporting in-circuit emulator IE-78001-R-A)	Part Number: μS××××ID78K0-NS, μS××××ID78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0
μS××××ID78K0-NS
μS××××ID78K0

××××	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)	

★ A.7 Embedded Software

RX78K0 Real-time OS	RX78K0 is a real-time OS conforming to the μITRON specifications. Tool (configurator) for generating nucleus of RX78K0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K0) and device file (DF780988). **<Precaution when using RX78K0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K0, fill in the purchase application form in advance 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	Source program for mass-produced object

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)	
BB13		Windows (English version)	

A.8 Upgrading from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A

If you already have a former in-circuit emulator for 78K/0 Series microcontrollers (IE-78000-R or IE-78000-R-A), that in-circuit emulator can operate as equivalent to the IE-78001-R-A by replacing its internal break board with the IE-78001-R-BK.

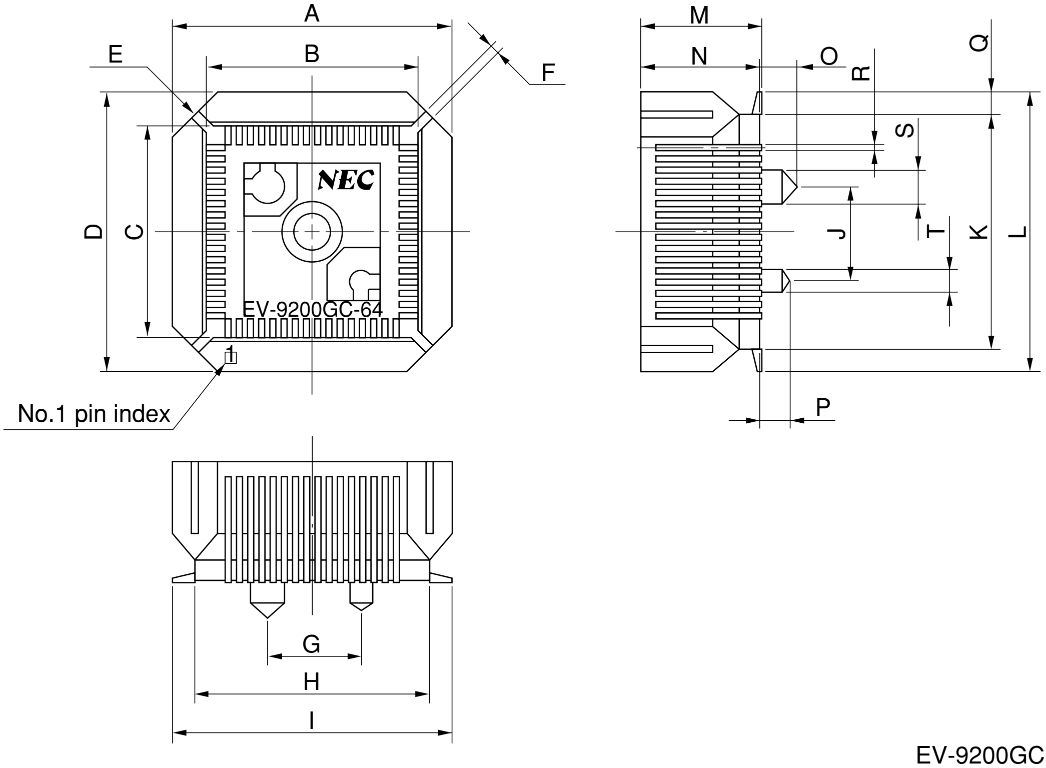
Table A-1. Upgrading from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A

In-Circuit Emulator Owned	In-Circuit Emulator Cabinet System Upgrade[Note]	Board to Be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note To replace the cabinet, send your in-circuit emulator to NEC Electronics Corporation.

A.9 Package Drawings for Conversion Socket and Conversion Adapter

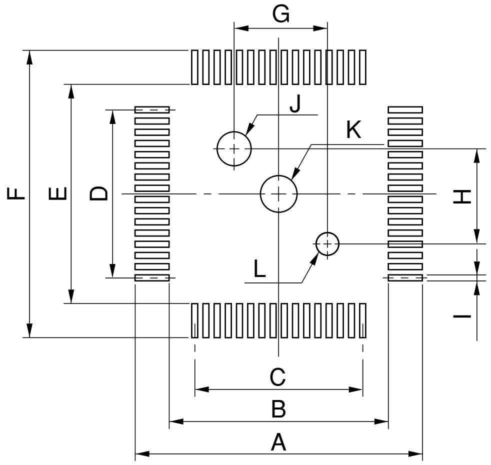
Figure A-2. EV-9200GC-64 Package Drawing (For Reference Only)



EV-9200GC-64-G0

ITEM	MILLIMETERS	INCHES
A	18.8	0.74
B	14.1	0.555
C	14.1	0.555
D	18.8	0.74
E	4-C 3.0	4-C 0.118
F	0.8	0.031
G	6.0	0.236
H	15.8	0.622
I	18.5	0.728
J	6.0	0.236
K	15.8	0.622
L	18.5	0.728
M	8.0	0.315
N	7.8	0.307
O	2.5	0.098
P	2.0	0.079
Q	1.35	0.053
R	0.35±0.1	$0.014^{+0.004}_{-0.005}$
S	ϕ2.3	ϕ0.091
T	ϕ1.5	ϕ0.059

Figure A-3. EV-9200GC-64 Footprints (For Reference Only)



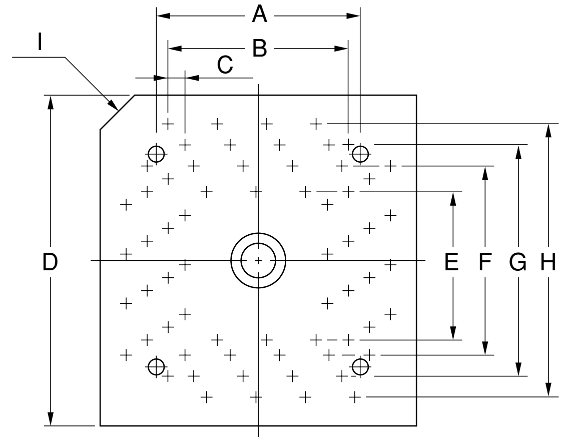
EV-9200GC-64-P1E

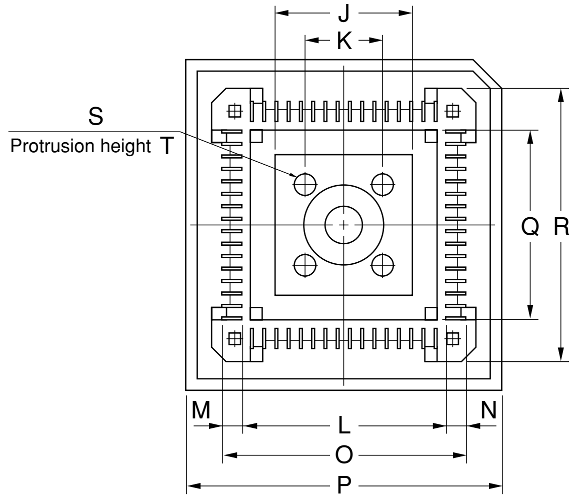
ITEM	MILLIMETERS	INCHES
A	19.5	0.768
B	14.8	0.583
C	0.8±0.02 × 15=12.0±0.05	$0.031^{+0.002}_{-0.001} \times 0.591=0.472^{+0.003}_{-0.002}$
D	0.8±0.02 × 15=12.0±0.05	$0.031^{+0.002}_{-0.001} \times 0.591=0.472^{+0.003}_{-0.002}$
E	14.8	0.583
F	19.5	0.768
G	6.00±0.08	$0.236^{+0.004}_{-0.003}$
H	6.00±0.08	$0.236^{+0.004}_{-0.003}$
I	0.5±0.02	$0.197^{+0.001}_{-0.002}$
J	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	ϕ2.2±0.1	$\phi 0.087^{+0.004}_{-0.005}$
L	ϕ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

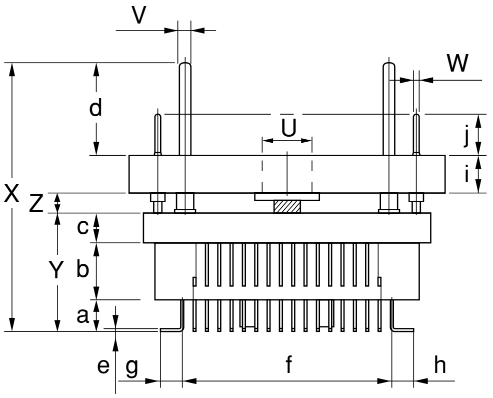
Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to “SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL” (http://www.necel.com/pkg/en/mount/index.html).

Figure A-4. TGC-064SAP Package Drawing (For Reference Only)

Reference diagram: TGC-064SAP (TQPACK064SA+TQSOCKET064SAP) Package dimension (unit: mm)







ITEM	MILLIMETERS	INCHES
A	14.12	0.556
B	0.8x15=12.0	0.031x0.591=0.472
C	0.8	0.031
D	20.65	0.813
E	10.0	0.394
F	12.4	0.488
G	14.8	0.583
H	17.2	0.677
I	C 2.0	C 0.079
J	9.05	0.356
K	5.0	0.197
L	13.35	0.526
M	1.325	0.052
N	1.325	0.052
O	16.0	0.630
P	20.65	0.813
Q	12.5	0.492
R	17.5	0.689
S	4-ϕ1.3	4-ϕ0.051
T	1.8	0.071
U	ϕ3.55	ϕ0.140
V	ϕ0.9	ϕ0.035
W	ϕ0.3	ϕ0.012
X	(19.65)	(0.667)
Y	7.35	0.289
Z	1.2	0.047

ITEM	MILLIMETERS	INCHES
a	1.85	0.073
b	3.5	0.138
c	2.0	0.079
d	6.0	0.236
e	0.25	0.010
f	13.6	0.535
g	1.2	0.047
h	1.2	0.047
i	2.4	0.094
j	2.7	0.106

TGC-064SAP-G0E

Note Made by TOKYO ELETECH CORPORATION.

★

APPENDIX B NOTES ON DESIGNING TARGET SYSTEM

The connection condition diagrams for an emulation probe, conversion connector, and conversion socket or conversion adapter are shown below. Design the system taking into consideration the dimension or shape, etc. of the parts to be mounted on the target system.

Table B-1. Distance Between In-Circuit Emulator and Conversion Socket or Conversion Adapter

Emulation Probe	Conversion Adapter, Conversion Socket	Distance Between In-Circuit Emulator and Conversion Socket or Conversion Adapter
NP-64GC-TQ	TGC-064SAP	170 mm
NP-H64GC-TQ		370 mm
NP-64CW	–	160 mm

Figure B-1. Distance Between In-Circuit Emulator and Conversion Socket or Conversion Adapter (1)

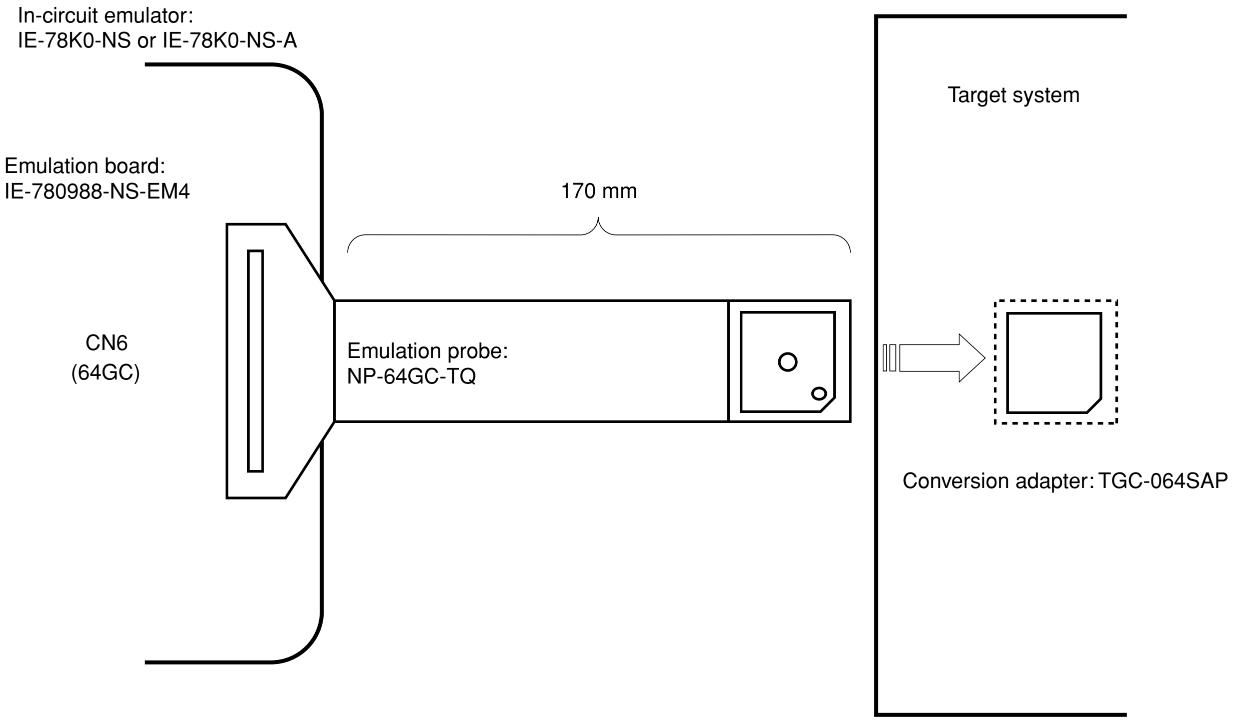


Figure B-2. Distance Between In-Circuit Emulator and Conversion Socket or Conversion Adapter (2)

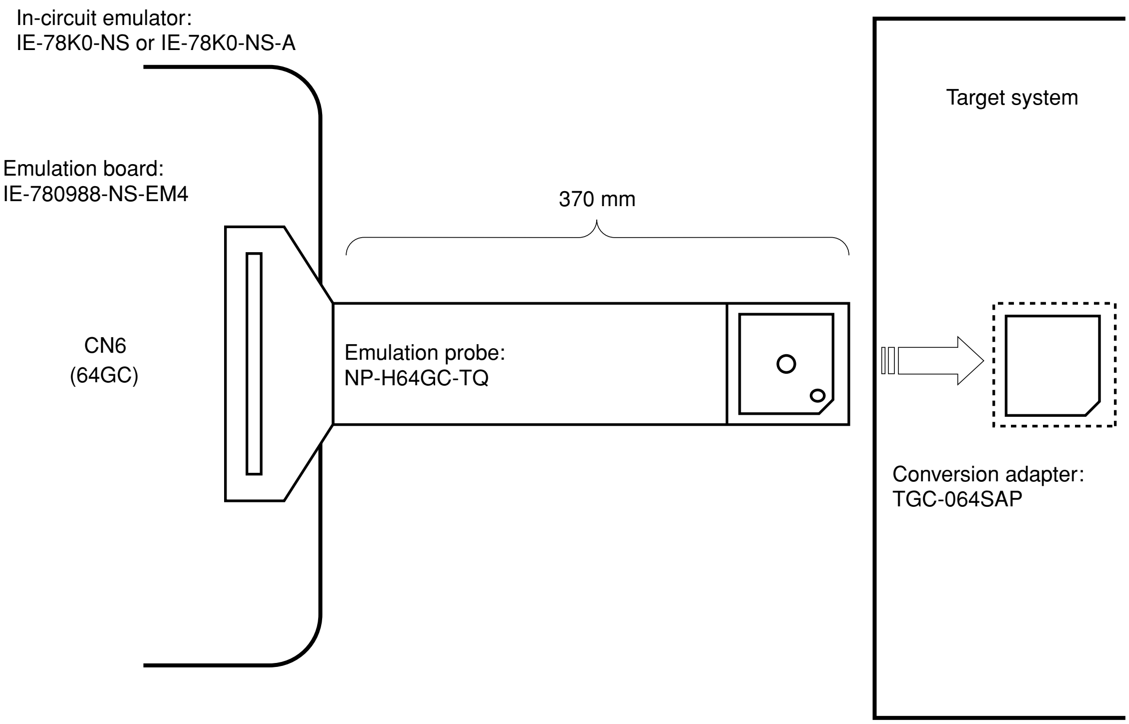


Figure B-3. Distance Between In-Circuit Emulator and Conversion Socket or Conversion Adapter (3)

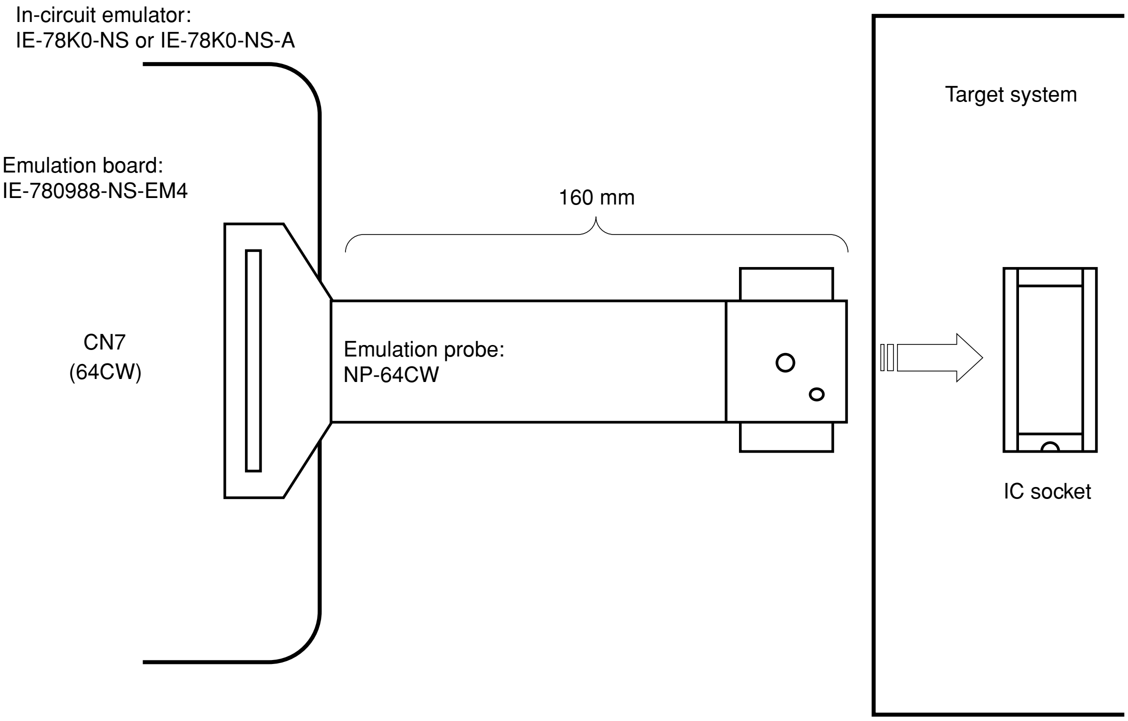


Figure B-4. Connection Condition of Target System (1)

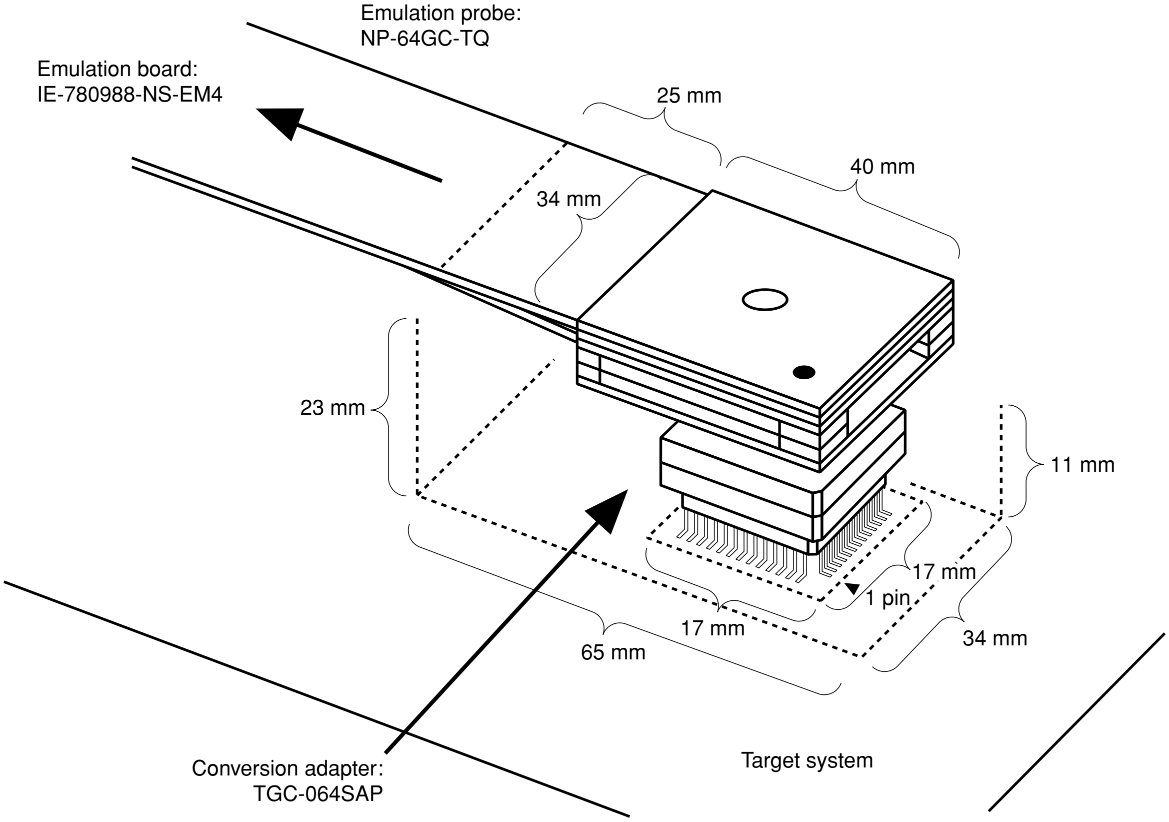


Figure B-5. Connection Condition of Target System (2)

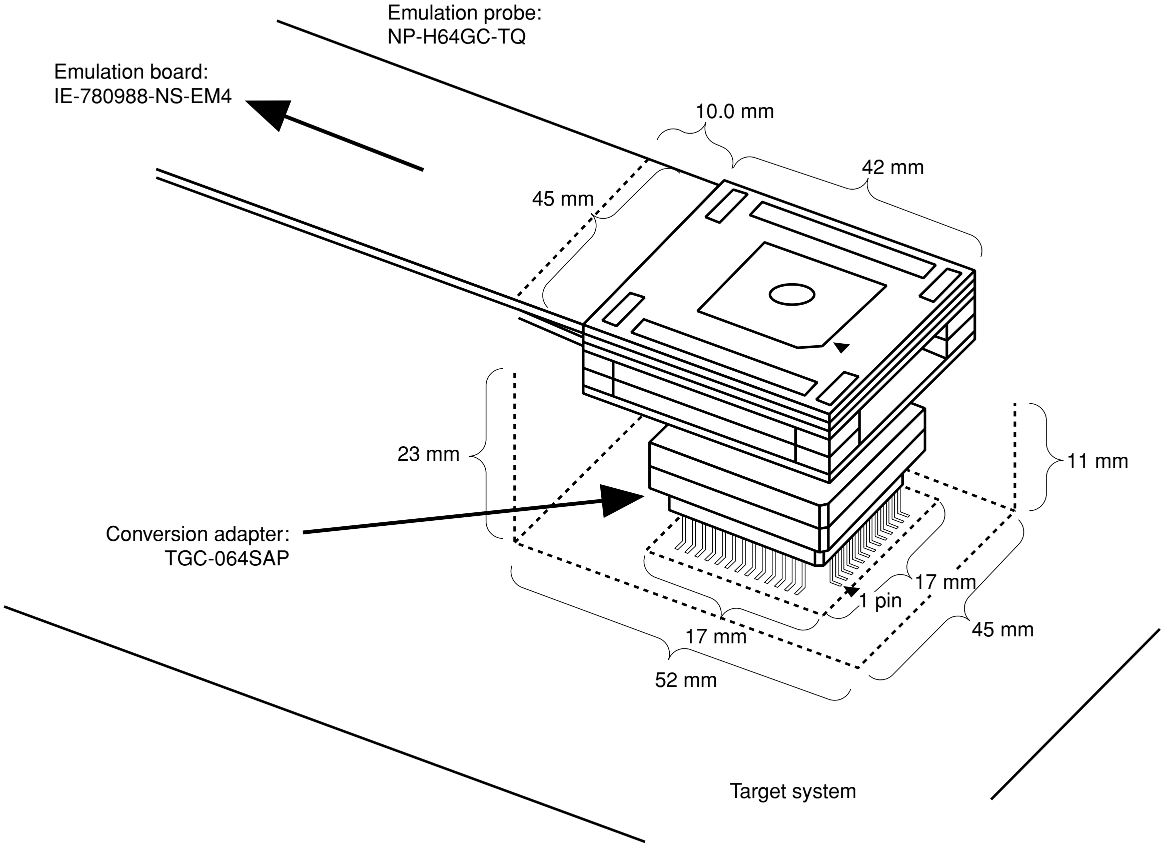
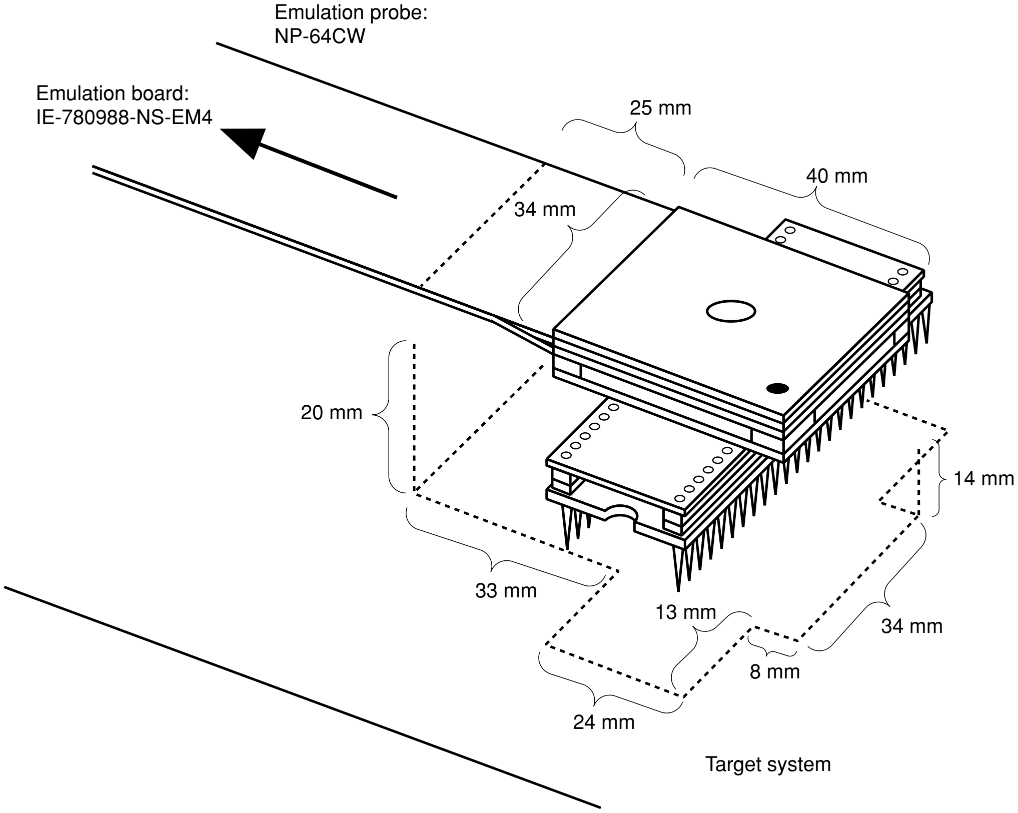


Figure B-6. Connection Condition of Target System (3)



APPENDIX C REGISTER INDEX

C.1 Register Index (In Alphabetical Order with Respect to Register Name)

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

APPENDIX D REVISION HISTORY

A history of the revisions up to this edition is shown below. “Applied to:” indicates the chapters to which the revision was applied.

(1/6)

Edition	Contents	Applied to:
3rd	Change of status from “under development” to “development completed” for the following products μPD780982, 780983, and 780984	Throughout
	1.4 Pin Configuration (Top View) • Deletion of Cautions for AV_{DD} and AV_{SS} connections	**CHAPTER 1 GENERAL**
	2.1 List of Pin Functions (2) Non-port pins • Modification of descriptions for TI000, TI001, AV_{DD}, AV_{SS} pins **2.2.5 P40 to P47 (Port 4)** Modification of description for **(2) Control mode** Addition of I/O circuit type description to **Table 2-1 Types of Pin I/O Circuits** Addition of **Figure 2-1 Pin I/O Circuit**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of description for TMC50, TMC51, TMC52 in **Table 3-4 Special Function Register List**	**CHAPTER 3 CPU ARCHITECTURE**
	4.3 Registers Controlling Port Functions Addition of Caution to **(1) Port mode registers (PM0, PM2, PM3, PM4, PM5, PM6)** Addition of Caution to **(2) Pull-up resistor option registers (PU0, PU2, PU3, PU4, PU5, PU6)** Modification of **Figure 4-11 Format of Pull-Up Resistor Option Register**	**CHAPTER 4 PORT FUNCTIONS**
	6.3 Configuration of 16-Bit Timer/Event Counter Modification of **Table 6-3 TI00n Pin Valid Edge and CR00n, CR01n Capture Triggers** Modification of Caution in **(3) 16-bit capture/compare register 010, 011 (CR010, CR011)** Addition of Caution to **Figure 6-5 Format of Capture/Compare Control Register 00** Addition of Caution to **Figure 6-6 Format of Capture/Compare Control Register 01** Addition of Cautions to **Figure 6-7 Format of Timer Output Control Register 00** Addition of Cautions to **Figure 6-8 Format of Timer Output Control Register 01** Addition of Note to **Figure 6-9 Format of Prescaler Mode Register 00** Addition of Note to **Figure 6-10 Format of Prescaler Mode Register 01** Modification of **Figure 6-13 Interval Timer Configuration Diagram** Modification of description in **6.5.4 External event counter operation** Modification of **Figure 6-27 External Event Counter Configuration Diagram** **6.6 Notes on 16-Bit Timer/Event Counter** Addition of description to **(6) One-shot pulse output** Addition of description to **(10) Capture operation** Addition of description to **(12) Edge detection**	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER**
	7.2 Configuration of 8-Bit Timer/Event Counter Modification of **(2) 8-bit compare registers 50, 51, and 52 (CR50, CR51, and CR52)** **7.3 Registers Controlling 8-Bit Timer/Event Counter** Modification of **(1) 8-bit timer mode control registers 50, 51, and 52 (TMC50, TMC51, and TMC52)** Modification of **Figure 7-4 Format of 8-Bit Timer Mode Control Register 50** Modification of **Figure 7-5 Format of 8-Bit Timer Mode Control Register 51** Modification of **Figure 7-6 Format of 8-Bit Timer Mode Control Register 52**	**CHAPTER 7 8-BIT TIMER/EVENT COUNTER**

(2/6)

Edition	Contents	Applied to:
3rd	**8.4 Operation of 10-Bit Inverter Control Timer** Modification of Caution in **(2) Output waveform widths corresponding to set values** Modification of **Figure 8-6 TM7 Operation Timing (CMn(BFCMn) = 000H)**	**CHAPTER 8 10-BIT INVERTER CONTROL TIMER**
	Modification of **Table 10-5 Real-Time Output Port Operating Mode and Output Trigger**	**CHAPTER 10 REAL-TIME OUTPUT PORT**
	Modification of **Figure 11-10 A/D Conversion End Interrupt Request Generation Timing**	**CHAPTER 11 A/D CONVERTER**
	Addition of **Figure 12-2 Block Diagram of UART00 Baud Rate Generator** Addition of **Figure 12-4 Block Diagram of UART01 Baud Rate Generator**	**CHAPTER 12 SERIAL INTERFACE (UART00, UART01)**
	13.2 Configuration of Serial Interface Modification of **(1) Serial I/O shift register 3 (SIO3)**	**CHAPTER 13 SERIAL INTERFACE (SIO3)**
	Modification of **Table 17-1 Status of Each Hardware After Reset**	**CHAPTER 17 RESET FUNCTION**
	Modification of **Figure 18-4 Flashpro II/III Connection Using 3-Wire Serial I/O** Modification of **Figure 18-5 Flashpro II/III Connection in UART Mode** Modification of **Figure 18-6 Flashpro II/III Connection Using Pseudo 3-Wire Serial I/O**	**CHAPTER 18 μPD78F0988**
	Modification of **Figure A-1 Development Tool Configuration** **A.1 Language Processing Software** • Change in the status of device file DF780988 from “under development” to “development completed” **A.3.1 Hardware** • Addition of the following products Performance board IE-78K0-NS-PA (under development) Emulation probe NP-64GC-TQ (development completed) Conversion adapter TGC-064SAP (development completed) • Change in the status of the following products from “under development” to “development completed” PC card interface IE-70000-CD-IF-A Interface adapter IE-70000-PCI-IF Addition of **Conversion Adapter (TGC-064SAP) Package Drawing**	**APPENDIX A DEVELOPMENT TOOLS**
4th	Addition of the following products to the target products μPD780982(A), 780983(A), 780984(A), 780986(A), 780988(A) Change of part number μPD780982(A)GC-×××-AB8 → μPD780982GC(A)-×××-AB8 μPD780983(A)GC-×××-AB8 → μPD780983GC(A)-×××-AB8 μPD780984(A)GC-×××-AB8 → μPD780984GC(A)-×××-AB8 μPD780986(A)GC-×××-AB8 → μPD780986GC(A)-×××-AB8 μPD780988(A)GC-×××-AB8 → μPD780988GC(A)-×××-AB8 Change of package name 64-pin plastic shrink DIP (750 mil) → 64-pin plastic SDIP (19.05 mm (750))	Throughout
	Modification of bit names and addition of Caution to **Figure 10-6 Format of Real-Time Output Port Mode Register 1**	**CHAPTER 10 REAL-TIME OUTPUT PORT**
	Addition of **11.6 How to Read A/D Converter Characteristics Tables**	**CHAPTER 11 A/D CONVERTER**

(3/6)

Edition	Contents	Applied to:
4th	Addition of **Figure 12-11 Baud Rate Allowance Error Including Sampling Error (When k = 0)** Addition of ASIS0n value description to **Table 12-5 Receive Error Causes**	**CHAPTER 12 SERIAL INTERFACE (UART00, UART01)**
	Addition of Remark to **14.2 Interrupt Sources and Configuration** Addition of Caution to **Figure 14-3 Format of Interrupt Mask Flag Register** Addition of Caution to **Figure 14-4 Format of Priority Specification Flag Register**	**CHAPTER 14 INTERRUPT FUNCTIONS**
	Addition of Caution to **Figure 15-3 Format of Memory Expansion Wait Setting Register**	**CHAPTER 15 EXTERNAL DEVICE EXPANSION FUNCTION**
	Addition of Caution to **18.3.1 Selecting communication mode**	**CHAPTER 18 μPD78F0988**
	A.3.1 Hardware (1) When using the in-circuit emulator IE-78K0-NS: • Change in the status of IE-78K0-NS-PA from “under development” to “development completed”	**APPENDIX A DEVELOPMENT TOOLS**
5th	Deletion of one-shot pulse output from 16-bit timer/event counter function	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER**
	Modification of **Figure 11-17 Differential Linearity Error**	**CHAPTER 11 A/D CONVERTER**
6th	Change of flash memory product from μPD78F0988 to μPD78F0988A, 78F0988A(A)	Throughout
	Change of INTTM01n timing in **Figure 6-18 Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)**	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER**
	Change of INTTM01n timing in **Figure 6-20 CR01n Capture Operation with Rising Edge Specified**	
	Change of INTTM00n and INTTM01n timing in **Figure 6-21 Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)**	
	Change of INTTM01n timing in **Figure 6-23 Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)**	
	Change of INTTM01n timing in **Figure 6-25 Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)**	
	Change of CR01n interrupt value timing in **Figure 6-33 Capture Register Data Retention Timing**	
	Change of INTTM00n timing in **Figure 6-34 Operation Timing of OVF0n Flag**	
	Addition of (11) to (14) to **11.5 Notes on A/D Converter**	**CHAPTER 11 A/D CONVERTER**
	Addition of **Caution** to **Figure 14-2 Format of Interrupt Request Flag Registers**	**CHAPTER 14 INTERRUPT FUNCTIONS**
	Addition of **18.3 Characteristics of Flash Memory**	**CHAPTER 18 μPD78F0988A**
	Addition of “3-wire serial I/O + HS” to **Table 18-4 Communication Modes**	
	Addition of “Writeback” to **Table 18-5 Major Functions of Flash Memory Programming**	

(4/6)

Edition	Contents	Applied to:
6th	Addition of **Figure 18-5 Flashpro III Connection Using 3-Wire Serial I/O (When Handshake Function Is Used)**	**CHAPTER 18** **μPD78F0988A**
	Addition of **18.5 Pin Connection**	
	Modification of **Figure 18-14 Self Programming Flowchart**	
	Modification of **Table 18-7 Entry RAM Area**	
	18.6.5 Entry RAM area Change of (d) Erase time data Addition of (f) Writeback time	
	Modification of **Table 18-8 List of Self-Write Subroutines**	
	Modification of **18.6.6 Self-write subroutines**	
	Development tools Addition of SP78K0 Change of part numbers of RA78K0, CC78K0, DF780988, CC78K0-L, SM78K0, ID78K0-NS, ID78K0, RX78K0, and MX78K0	**APPENDIX A DEVELOPMENT TOOLS**
7th	• Addition of package μPD780982GC-×××-8BS, 780983GC-×××-8BS, 780984GC-×××-8BS μPD780986GC-×××-8BS, 780988GC-×××-8BS, 78F0988AGC-8BS μPD780982GC(A)-×××-8BS, 780983GC(A)-×××-8BS, 780984GC(A)-×××-8BS μPD780986GC(A)-×××-8BS, 780988GC(A)-×××-8BS • Change of power supply voltage range as shown below. V_{DD} = 4.0 to 5.5 V → V_{DD} = 3.0 to 5.5 V (expanded-specification products), V_{DD} = 4.0 to 5.5 V (conventional products) • Change of system clock oscillation frequency (f_X) as shown below. f_X = 8.38 MHz → f_X = 12 MHz (expanded-specification products only), f_X =8.38 MHz • Change of minimum instruction execution time	Throughout
	Addition of **1.1 Expanded-Specification Products and Conventional Products**	**CHAPTER 1 GENERAL**
	1.6 Pin Configuration (Top View) • Addition of **Cautions 2** and **3** to 64-pin plastic SDIP (19.05 mm (750)) • Addition of **Cautions 2** and **3** to 64-pin plastic QFP (14 x 14), 64-pin plastic LQFP (14 x 14)	
	3.1.2 Internal data memory space Addition of description on **(1) Internal high-speed RAM** and **(2) Internal expansion RAM**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of **Table 5-2 Relationship Between CPU Clock and Minimum Instruction Execution Time**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **Figure 5-5 Switching Between System Clock and CPU Clock**	
	Modification of **Figure 6-9 Format of Prescaler Mode Register 00**	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER**
	Modification of **Figure 6-10 Format of Prescaler Mode Register 01**	
	Addition of **Figure 6-16 Configuration Diagram of PPG Output**	
	Addition of **Figure 6-17 PPG Output Operation Timing**	
	Modification of **Figure 7-7 Format of Timer Clock Select Register 50**	**CHAPTER 7 8-BIT TIMER/EVENT COUNTER**
	Modification of **Figure 7-8 Format of Timer Clock Select Register 51**	
	Modification of **Figure 7-9 Format of Timer Clock Select Register 52**	

(5/6)

Edition	Contents	Applied to:
7th	Modification of **Figure 8-2 Format of Inverter Timer Control Register 7**	**CHAPTER 8 10-BIT INVERTER CONTROL TIMER**
	Modification of **Table 9-1 Loop Detection Time of Watchdog Timer**	**CHAPTER 9 WATCHDOG TIMER**
	Modification of **Table 9-2 Interval Time**	
	Modification of **Figure 9-2 Format of Watchdog Timer Clock Select Register**	
	Modification of **Figure 9-4 Format of Oscillation Stabilization Time Select Register**	
	Modification of **Table 9-4 Loop Detection Time of Watchdog Timer**	
	Modification of **Table 9-5 Interval Time of Interval Timer**	
	11.2 Configuration of A/D Converter Addition of register figure to **(2) A/D conversion result register 0 (ADCR0)**	**CHAPTER 11 A/D CONVERTER**
	Modification of **Figure 11-2 Format of A/D Converter Mode Register 0**	
	11.5 Notes on A/D Converter Addition of **(6) Input impedance of ANI0 to ANI7 pins**	
	Modification of **Figure 12-9 Format of Baud Rate Generator Control Register 0**	**CHAPTER 12 SERIAL INTERFACES UART00 AND UART01**
	Modification of **Figure 12-10 Format of Baud Rate Generator Control Register 1**	
	12.4.2 Asynchronous serial interface (UART) mode • Modification of description • Modification of **(1) Register setting (c) Baud rate generator control registers 0, 1 (BRGC00, BRGC01)**	
	Modification of **Table 12-2 Relationship Between Source Clock of 5-Bit Counter and Value of m (with UART00)**	
	Modification of **Table 12-3 Relationship Between Source Clock of 5-Bit Counter and Value of m (with UART01)**	
	Modification of **Table 12-4 Relationship Between System Clock and Baud Rate**	
	Addition of **Remark** to **12.4.3 Infrared data transfer mode**	
	Modification of **Table 12-7 Baud Rate That Can Be Set in Infrared Data Transfer Mode**	
	Modification of **Figure 13-2 Format of Serial Operation Mode Register 3**	**CHAPTER 13 SERIAL INTERFACE SIO3**
	Addition of **Caution** to **15.1 External Device Expansion Function**	**CHAPTER 15 EXTERNAL DEVICE EXPANSION FUNCTION**
	Change of R/W to W in **Figure 15-2 Format of Memory Expansion Mode Register**	
	Modification of **Figure 16-1 Format of Oscillation Stabilization Time Select Register**	**CHAPTER 16 STANDBY FUNCTION**
	Modification of **Figure 16-3 Releasing HALT Mode by $\overline{\text{RESET}}$ Input**	
	Modification of **Figure 16-5 Releasing STOP Mode by $\overline{\text{RESET}}$ Input**	
	Revision of descriptions on flash memory programming as **18.3 Flash Memory Features**	**CHAPTER 18 μPD78F0988A**
	18.4.5 Entry RAM area Modification of **(c) Write time data**	

(6/6)

Edition	Contents	Applied to:
7th	Addition of **CHAPTER 20 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)**	**CHAPTER 20 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)**
	Addition of **CHAPTER 21 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**	**CHAPTER 21 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**
	Addition of **CHAPTER 22 PACKAGE DRAWINGS**	**CHAPTER 22 PACKAGE DRAWINGS**
	Addition of **CHAPTER 23 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 23 RECOMMENDED SOLDERING CONDITIONS**
	Modification of **APPENDIX A DEVELOPMENT TOOLS**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of **APPENDIX B NOTES ON DESIGNING TARGET SYSTEM**	**APPENDIX B NOTES ON DESIGNING TARGET SYSTEM**
7th Edition (Modification Version)	Modification of **1.4 Ordering Information**	**CHAPTER 1 GENERAL**
	Addition of **CHAPTER 23 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 23 RECOMMENDED SOLDERING CONDITIONS**