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

PD789407A, 789417A Subseries

8-Bit Single-Chip Microcontrollers

PD789405A
PD789406A
PD789407A
PD789415A
PD789416A
PD789417A
PD78F9418A

Document No. U13952EJ3V1UD00 (3rd edition)
Date Published October 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.

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

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 August, 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 Electronics product in your application, please contact the NEC Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

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

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

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

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

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

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-265 40 10
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 87 200
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

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

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

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

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

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

J05.6

Major Revisions in This Edition

Page	Description
U13952EJ2V0UD00 → U13952EJ3V0UD00	
pp.38, 39, 41	Modification of pin handling of AV_{REF} pin and V_{PP} pin in **CHAPTER 2 PIN FUNCTIONS**
p.92	Addition of **Note** related to feedback resistor in **Figure 5-3 Format of Suboscillation Mode Register**
pp.112, 113	Addition of **6.5 Cautions on Using 16-Bit Timer 50**
pp.151, 164	Addition of **(8) Input impedance of ANI0 to ANI6 pins** in **10.5 Cautions on Using 8-Bit A/D Converter** and **11.5 Cautions on Using 10-Bit A/D Converter**
p.154	Modification of description of **(2) A/D conversion result register 0 (ADCR0)** in **11.2 Configuration of 10-Bit A/D Converter**
p.196	Addition of description on reading receive data of UART in **13.4.2 Asynchronous serial interface (UART) mode**
p.232	Addition of **Caution** in **Figure 15-2 Format of Interrupt Request Flag Register**
p.237	Addition of **Caution** in **Figure 15-7 Format of Key Return Mode Register 00**
p.256	Addition of description on pull-up resistor and divider resistor for LCD driving in **Table 18-1 Differences Between μPD78F9418A and Mask ROM Versions**
pp.257 to 266	Overall revision of contents related to flash memory programming as **18.1 Flash Memory Characteristics**
pp.278 to 292	Addition of **CHAPTER 21 ELECTRICAL SPECIFICATIONS**
pp.293 to 295	Addition of **CHAPTER 22 CHARACTERISTICS CURVES (REFERENCE VALUES)**
pp.296, 297	Addition of **CHAPTER 23 PACKAGE DRAWINGS**
pp.298, 299	Addition of **CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS**
pp.301 to 310	Overall revision of contents of **APPENDIX A DEVELOPMENT TOOLS** Deletion of embedded software
pp.311 to 314	Addition of **APPENDIX B NOTES ON TARGET SYSTEM DESIGN**
U13952EJ3V0UD00 → U13952EJ3V1UD00	
p. 24	Modification of **1.3 Ordering Information**
p. 300	Addition of **Table 24-1. Surface Mounting Type Soldering Conditions (3/3)**

The mark ★ shows major revised points.

INTRODUCTION

Target Readers

This manual is intended for users who wish to understand the functions of the μPD789407A and μPD789417A Subseries and to design and develop application systems and programs using these microcontrollers.

Target products:

- μPD789407A Subseries: μPD789405A, μPD789406A, and μPD789407A
- μPD789417A Subseries: μPD789415A, μPD789416A, μPD789417A, and μPD78F9418A

Purpose

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

Organization

The μPD789407A and μPD789417A Subseries User's Manual is divided into two parts: this manual and instructions (common to the 78K/0S Series).

μPD789407A and μPD789417A Subseries User's Manual	78K/0S Series User's Manual Instructions
• Pin functions • Internal block functions • Interrupt functions • Other on-chip peripheral functions • Electrical specifications	• CPU function • Instruction set • Explanation of each instruction

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 understand the functions in general:
  → Read this manual in the order of the **CONTENTS**.
- How to interpret the register formats:
  → The name of a bit whose number is enclosed in brackets is reserved for the assembler and is defined for the C compiler by the header file sfrbit.h.
- When you know a register name and want to confirm its details:
  → Read **APPENDIX C REGISTER INDEX**.
- To know the 78K/0S Series instructions functions in detail:
  → Refer to **78K/0S Series Instructions User's Manual (U11047E).**
- To learn the electrical specifications of the μPD789407A and μPD789417A Subseries
  → Refer to **CHAPTER 21 ELECTRICAL SPECIFICATIONS**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (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 ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH

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.
μPD789407A, 789417A Subseries User's Manual	This manual
78K/0S Series Instructions User's Manual	U11047E

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

Document Name		Document No.
RA78K0S Assembler Package	Operation	U14876E
	Language	U14877E
	Structured Assembly Language	U11623E
CC78K0S C Compiler	Operation	U14871E
	Language	U14872E
SM78K Series System Simulator Ver. 2.30 or Later	Operation (Windows™ Based)	U15373E
	External Part User Open Interface Specifications	U15802E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E
IE-789418-NS-EM1 Emulation Board	U14364E

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 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 Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

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

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

CONTENTS

LIST OF FIGURES (1/5)

LIST OF FIGURES (2/5)

LIST OF FIGURES (3/5)

LIST OF FIGURES (4/5)

LIST OF FIGURES (5/5)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL

1.1 Features

- ROM and RAM capacities

Item / Part Number	Program Memory		Data Memory	
			Internal High-Speed RAM	LCD Data RAM
μPD789405A, 789415A	ROM	12 KB	512 bytes	28 × 4 bits
μPD789406A, 789416A		16 KB		
μPD789407A, 789417A		24 KB		
μPD78F9418A	Flash memory	32 KB		

- Minimum instruction execution time can be changed from high speed (0.4 μs: @ 5.0 MHz operation with main system clock) to ultra low speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- 43 I/O ports
- Serial interface channel: Switchable between 3-wire serial I/O and UART modes
- LCD controller/driver:
  - Up to 28 segment signal outputs
  - Up to 4 common signal outputs
  - Bias switchable between 1/2 and 1/3
- Seven A/D converters with an 8-bit resolution (for μPD789407A Subseries only)
- Seven A/D converters with a 10-bit resolution (for μPD789417A Subseries only)
- Six timers:
  - 16-bit timer
  - Two 8-bit timer/event counters
  - 8-bit timer
  - Watch timer
  - Watchdog timer
- 17 vectored interrupt sources
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = –40 to +85°C

1.2 Applications

APS compact cameras, manometers, rice cookers, etc.

★ 1.3 Ordering Information

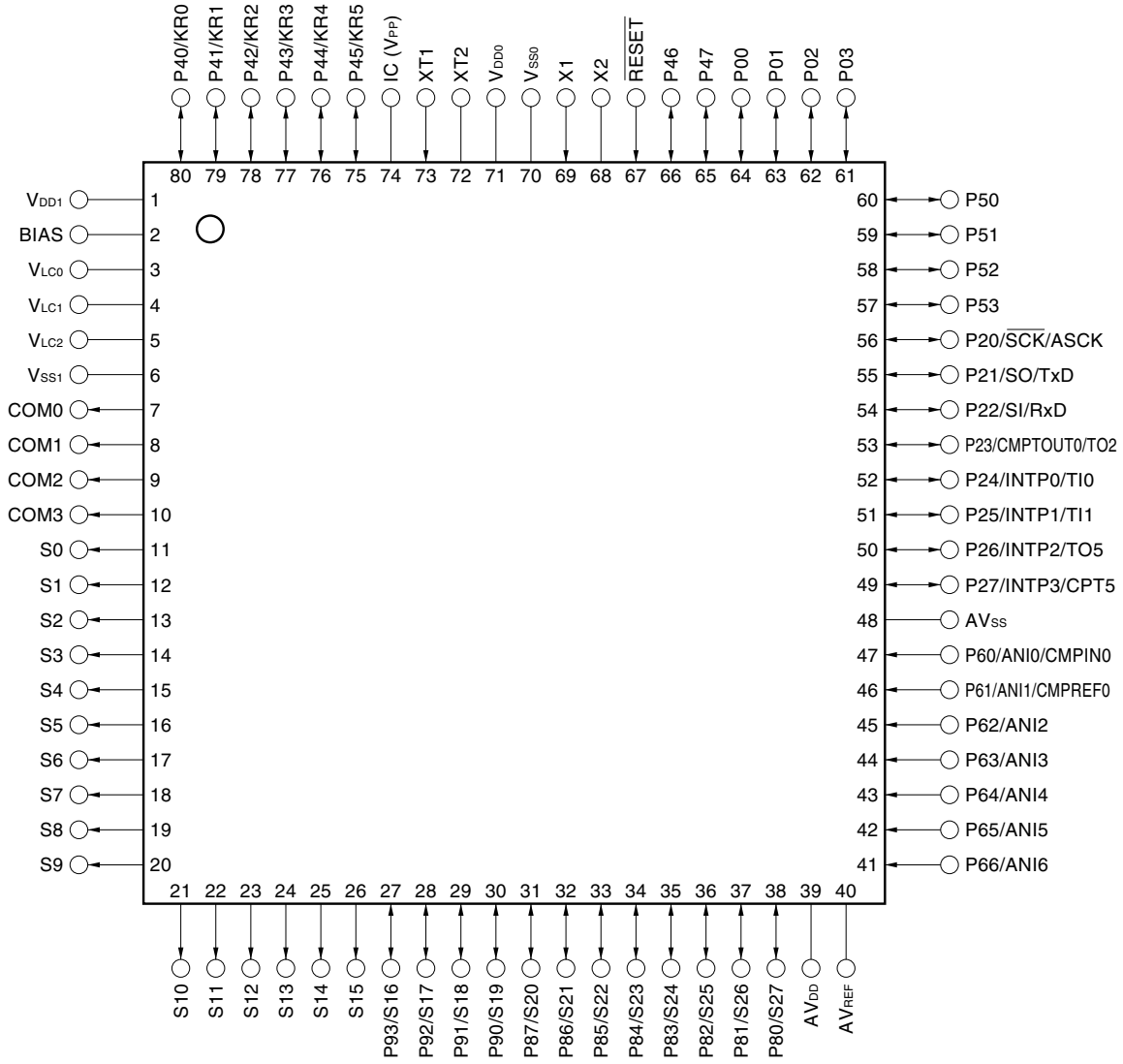
Part Number	Package	Internal ROM
μPD789405AGC-xxx-8BT	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789405AGK-xxx-9EU	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789406AGC-xxx-8BT	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789406AGK-xxx-9EU	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789407AGC-xxx-8BT	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789407AGK-xxx-9EU	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789415AGC-xxx-8BT	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789415AGK-xxx-9EU	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789416AGC-xxx-8BT	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789416AGK-xxx-9EU	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789417AGC-xxx-8BT	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789417AGK-xxx-9EU	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD78F9418AGC-8BT	80-pin plastic QFP (14 x 14)	Flash memory
μPD78F9418AGK-9EU	80-pin plastic TQFP (fine pitch) (12 x 12)	Flash memory
μPD789405AGC-xxx-8BT-A	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789405AGK-xxx-9EU-A	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789406AGC-xxx-8BT-A	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789406AGK-xxx-9EU-A	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789407AGC-xxx-8BT-A	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789407AGK-xxx-9EU-A	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789415AGC-xxx-8BT-A	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789415AGK-xxx-9EU-A	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789416AGC-xxx-8BT-A	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789416AGK-xxx-9EU-A	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD789417AGC-xxx-8BT-A	80-pin plastic QFP (14 x 14)	Mask ROM
μPD789417AGK-xxx-9EU-A	80-pin plastic TQFP (fine pitch) (12 x 12)	Mask ROM
μPD78F9418AGC-8BT-A	80-pin plastic QFP (14 x 14)	Flash memory
μPD78F9418AGK-9EU-A	80-pin plastic TQFP (fine pitch) (12 x 12)	Flash memory

Remarks 1. xxx indicates ROM code suffix.

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

1.4 Pin Configuration (Top View)

- 80-pin plastic QFP (14 x 14)
- 80-pin plastic TQFP (fine pitch) (12 x 12)



Cautions 1. Connect the IC (Internally Connected) pin directly to V_{SS0} or V_{SS1}.

2. Connect the AV_{DD} pin to V_{DD0}.

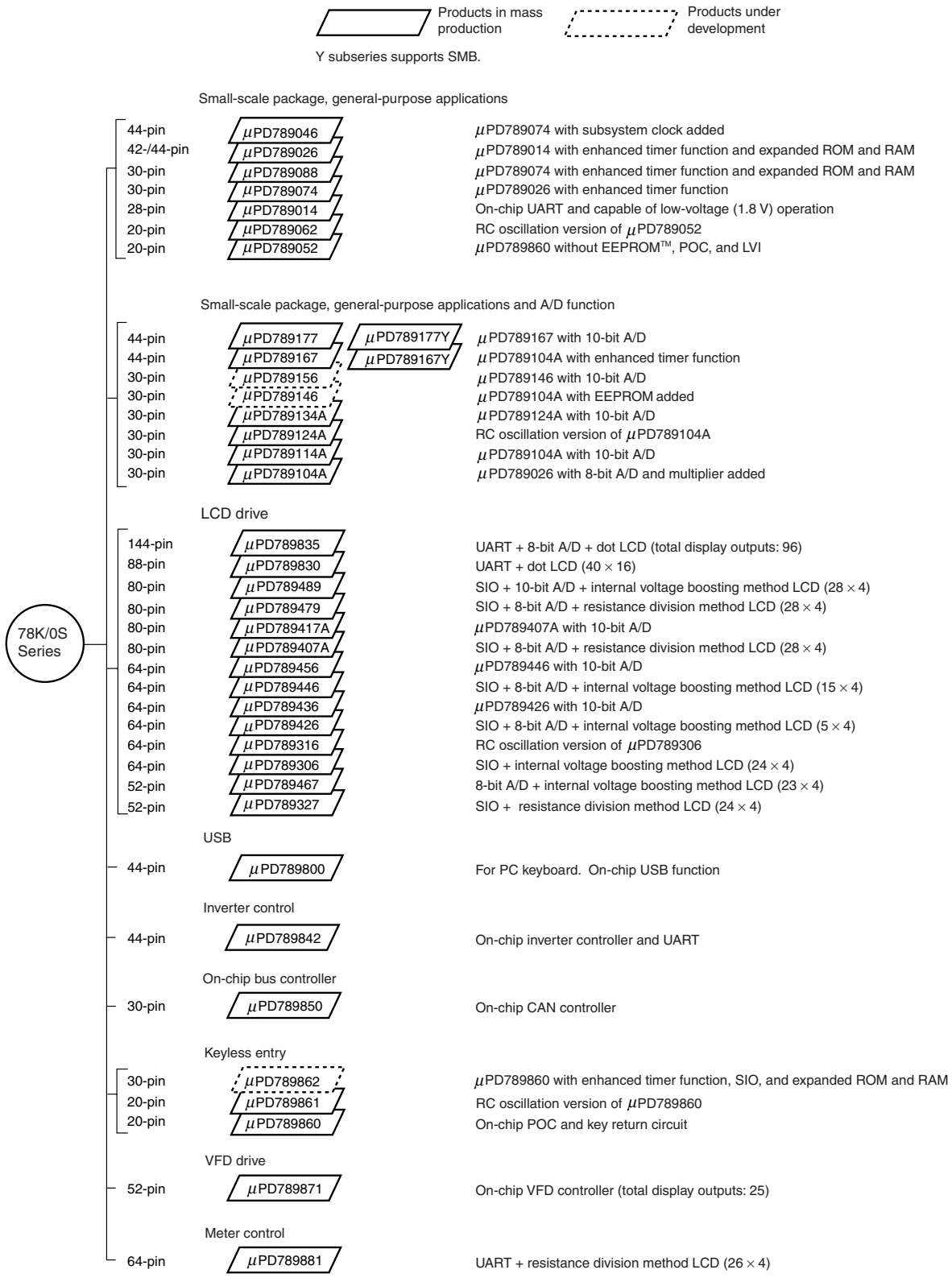
3. Connect the AV_{SS} pin to V_{SS0}.

Remark The parenthesized values apply to the μPD78F9418A.

ANI0 to ANI6:	Analog input
ASCK:	Asynchronous serial input
AV_{DD}:	Analog power supply
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog ground
BIAS:	LCD power supply bias control
CMPIN0:	Comparator input
CMPREF0:	Comparator reference
CMPTOUT0:	Comparator output
COM0 to COM3:	Common output
CPT5:	Capture trigger input
IC:	Internally connected
INTP0 to INTP3:	Interrupt from peripherals
KR0 to KR5:	Key return
P00 to P03:	Port 0
P20 to P27:	Port 2
P40 to P47:	Port 4
P50 to P53:	Port 5
P60 to P66:	Port 6
P80 to P87:	Port 8
P90 to P93:	Port 9
$\overline{\text{RESET}}$:	Reset
RxD:	Receive data
S0 to S27:	Segment output
$\overline{\text{SCK}}$:	Serial clock
SI:	Serial input
SO:	Serial output
TI0, TI1:	Timer input
TO2, TO5:	Timer output
TxD:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{LC0} to V_{LC2}:	LCD power supply
V_{PP}:	Programming power supply
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

★ 1.5 78K/0S Series Lineup

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



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

The major functional differences between the subseries are listed below.

Series for general-purpose applications and LCD drive

Subseries \ Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN.Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
Small-scale package, general-purpose applications	μPD789046	16 K	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1 ch)	34	1.8 V	–
	μPD789026	4 K to 16 K			–							
	μPD789088	16 K to 32 K	3 ch							24		
	μPD789074	2 K to 8 K	1 ch									
	μPD789014	2 K to 4 K	2 ch	–						22		
	μPD789062	4 K							–	14		RC-oscillation version
	μPD789052											–
Small-scale package, general-purpose applications + A/D converter	μPD789177	16 K to 24 K	3 ch	1 ch	1 ch	1 ch	–	8 ch	1 ch (UART: 1 ch)	31	1.8 V	–
	μPD789167						8 ch	–				
	μPD789156	8 K to 16 K	1 ch		–		–	4 ch		20		On-chip EEPROM
	μPD789146						4 ch	–				
	μPD789134A	2 K to 8 K					–	4 ch				RC-oscillation version
	μPD789124A						4 ch	–				
	μPD789114A						–	4 ch				–
	μPD789104A						4 ch	–				
LCD drive	μPD789835	24 K to 60 K	6 ch	–	1 ch	1 ch	3 ch	–	1 ch (UART: 1 ch)	37	1.8 V[Note]	Dot LCD supported
	μPD789830	24 K	1 ch	1 ch			–			30	2.7 V	
	μPD789489	32 K to 48 K	3 ch					8 ch	2 ch (UART: 1 ch)	45	1.8 V	–
	μPD789479	24 K to 48 K					8 ch	–				
	μPD789417A	12 K to 24 K					–	7 ch	1 ch (UART: 1 ch)	43		
	μPD789407A						7 ch	–				
	μPD789456	12 K to 16 K	2 ch				–	6 ch		30		
	μPD789446						6 ch	–				
	μPD789436						–	6 ch		40		
	μPD789426						6 ch	–				
	μPD789316	8 K to 16 K					–		2 ch (UART: 1 ch)	23		RC-oscillation version
	μPD789306											–
	μPD789467	4 K to 24 K		–			1 ch		–	18		
	μPD789327						–		1 ch	21		

Note Flash memory version: 3.0 V

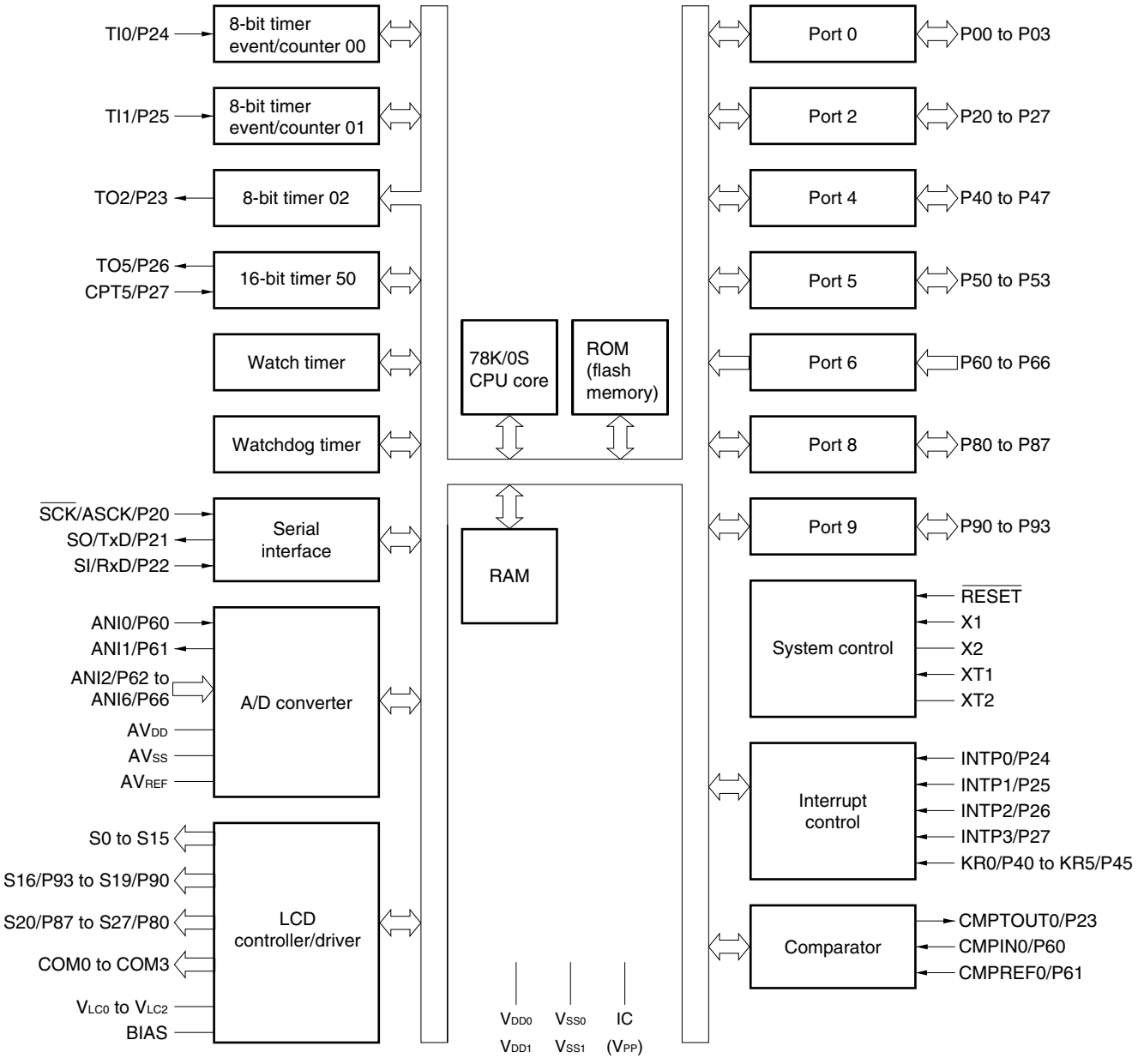
Series for ASSP

Subseries \ Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN.Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
USB	μPD789800	8 K	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	31	4.0 V	–
Inverter control	μPD789842	8 K to 16 K	3 ch	**Note 1**	1 ch	1 ch	8 ch	–	1 ch (UART: 1 ch)	30	4.0 V	–
On-chip bus controller	μPD789850	16 K	1 ch	1 ch	–	1 ch	4 ch	–	2 ch (UART: 1 ch)	18	4.0 V	–
Keyless entry	μPD789861	4 K	2 ch	–	–	1 ch	–	–	–	14	1.8 V	RC-oscillation version, on-chip EEPROM
	μPD789860											On-chip EEPROM
	μPD789862	16 K	1 ch	2 ch					1 ch (UART: 1 ch)	22		
VFD drive	μPD789871	4 K to 8 K	3 ch	–	1 ch	1 ch	–	–	1 ch	33	2.7 V	–
Meter control	μPD789881	16 K	2 ch	1 ch	–	1 ch	–	–	1 ch (UART: 1 ch)	28	2.7 V[Note 2]	–

Notes 1. 10-bit timer: 1 channel

2. Flash memory version: 3.0 V

1.6 Block Diagram



Remarks 1. The internal ROM capacity varies depending on the product.

2. The parenthesized values apply to the μPD78F9418A.

1.7 Overview of Functions

Item \ Part Number		μPD789405A μPD789415A	μPD789406A μPD789416A	μPD789407A μPD789417A	μPD78F9418A
Internal memory	ROM	Mask ROM			Flash memory
		12 KB	16 KB	24 KB	32 KB
	High-speed RAM	512 bytes			
	LCD data RAM	28 × 4 bits			
Minimum instruction execution time		• 0.4/1.6 μs (@ 5.0 MHz operation with main system clock) • 122 μs (@ 32.768 kHz operation with subsystem clock)			
General-purpose registers		8 bits × 8 registers			
Instruction set		• 16-bit operations • Bit manipulation (set, reset, and test)			
I/O ports		Total of 43 port pins • 7 CMOS input pins • 32 CMOS I/O pins • 4 N-ch open-drain pins (12 V withstanding voltage)			
A/D converters		• Seven channels with 8-bit resolution (for μPD789407A Subseries) • Seven channels with 10-bit resolution (for μPD789417A Subseries)			
Comparator		With timer output control function			
Serial interface		Switchable between 3-wire serial I/O and UART modes			
LCD controller/driver		• Up to 28 segment signal outputs • Up to 4 common signal outputs • Bias switchable between 1/2 and 1/3			
Timers		• 16-bit timer: 1 channel • 8-bit timer: 1 channel • 8-bit timer/event counters: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel			
Timer output		2 outputs			
Vectored interrupt sources	Maskable	Internal: 11, external: 5			
	Non-maskable	Internal: 1			
Power supply voltage		V_{DD} = 1.8 to 5.5 V			
Operating ambient temperature		T_A = −40 to +85°C			
Package		• 80-pin plastic QFP (14 x 14) • 80-pin plastic TQFP (fine pitch) (12 x 12)			

An outline of the timer is shown below.

		16-Bit Timer 50	8-Bit Timer/Event Counters 00, 01	8-Bit Timer 02	Watch Timer	Watchdog Timer
Operation mode	Interval timer	–	1 channel	1 channel	1 channel[Note 1]	1 channel[Note 2]
	External event counter	–	1 channel	–	–	–
Function	Timer outputs	1	–	1	–	–
	Square-wave outputs	–	–	1	–	–
	Capture	1 input	–	–	–	–
	Interrupt sources	1	1	1	2	2

Notes 1. The watch timer can perform both watch timer and interval timer functions at the same time.

2. The watchdog timer has watchdog timer and interval timer functions. However, use the watchdog timer by selecting either the watchdog timer function or interval timer function.

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 1 (PU1).	Input	$\overline{\text{SCK}}$/ASCK
P21				SO/TxD
P22				SI/RxD
P23				CMPTOUT0/TO2
P24				INTP0/TI0
P25				INTP1/TI1
P26				INTP2/TO5
P27				INTP3/CPT5
P40 to P45	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	KR0 to KR5
P46, P47				–
P50 to P53	I/O	Port 5. 4-bit N-ch open-drain I/O port. Input/output can be specified in 1-bit units. For a mask ROM version, use of an on-chip pull-up resistor can be specified by the mask option.	Input	–
P60	Input	Port 6. 7-bit input port.	Input	ANI0/CMPIN0
P61				ANI1/CMPREF0
P62 to P66				ANI2 to ANI6
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 2 (PU2).	Input	S27 to S20
P90 to P93	I/O	Port 9. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 2 (PU2).	Input	S19 to S16

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

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P24/TI0
INTP1				P25/TI1
INTP2				P26/TO5
INTP3				P27/CPT5
KR0 to KR5	Input	Key return signal detection	Input	P40 to P45
SI	Input	Serial interface serial data input	Input	P22/RxD
SO	Output	Serial interface serial data output	Input	P21/TxD
$\overline{SCK}$	I/O	Serial interface serial clock input/output	Input	P20/ASCK
ASCK	Input	Serial clock input for asynchronous serial interface	Input	P20/$\overline{SCK}$
RxD	Input	Serial data input for asynchronous serial interface	Input	P22/SI
TxD	Output	Serial data output for asynchronous serial interface	Input	P21/SO
TI0	Input	External count clock input to 8-bit timer (TM00)	Input	P24/INTP0
TI1	Input	External count clock input to 8-bit timer (TM01)	Input	P25/INTP1
TO2	Output	8-bit timer (TM02) output	Input	P23/CMPTOUT0
TO5	Output	16-bit timer (TM50) output	Input	P26/INTP2
CPT5	Input	Capture edge input	Input	P27/INTP3
CMPTOUT0	Output	Comparator output	Input	P23/TO2
CMPIN0	Input	Comparator input	Input	P60/ANI0
CMPREF0	Input	Comparator reference voltage input	Input	P61/ANI1
ANI0	Input	A/D converter analog input	Input	P60/CMPIN0
ANI1				P61/CMPREF0
ANI2 to ANI6				P62 to P66
AV_{REF}	–	A/D converter reference voltage	–	–
AV_{SS}	–	A/D converter ground potential	–	–
AV_{DD}	–	A/D converter analog power supply	–	–
S0 to S15	Output	LCD controller/driver segment signal output	Output	–
S16 to S19			Input	P93 to P90
S20 to S27				P87 to P80
COM0 to COM3	Output	LCD controller/driver common signal output	Output	–
V_{LC0} to V_{LC2}	–	LCD driving voltage	–	–
BIAS	–	Supply voltage for LCD driving	–	–
X1	Input	Connecting crystal resonator for main system clock oscillation	–	–
X2	–		–	–
XT1	Input	Connecting crystal resonator for subsystem clock oscillation	–	–
XT2	–		–	–
RESET	Input	System reset input	Input	–

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

Pin Name	I/O	Function	After Reset	Alternate Function
V_{DD0}	–	Positive power supply for ports	–	–
V_{DD1}	–	Positive power supply for circuits other than ports	–	–
V_{SS0}	–	Ground potential for ports	–	–
V_{SS1}	–	Ground potential of circuits other than ports	–	–
IC	–	Internally connected. Connect directly to V_{SS0} or V_{SS1}.	–	–
V_{PP}	–	Sets flash memory programming mode. Applies high voltage when a program is written or verified.	–	–

2.2 Description of Pin Functions

2.2.1 P00 to P03 (Port 0)

These pins constitute a 4-bit I/O port and can be set to input or output port mode in 1-bit units by using port mode register 0 (PM0). When these pins are used as input port pins, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

2.2.2 P20 to P27 (Port 2)

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

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

(1) Port mode

In this mode, P20 to P27 function as an 8-bit I/O port. These pins can be set to input or output mode in 1-bit units by using port mode register 2 (PM2). When used as input port pins, an on-chip pull-up resistor can be used by setting pull-up resistor option register 1 (PU1).

(2) Control mode

In this mode, P20 to P27 function as the data I/O and the clock I/O of the serial interface, the external interrupt input, and timer I/O.

(a) SI, SO

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

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

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

(c) RxD, TxD

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

(d) ASCK

This is the serial clock input pin of the asynchronous serial interface.

(e) TI0, TI1

These are external clock input pins for the 8-bit timer/event counter.

(f) TO2

This is the output pin of the 8-bit timer.

(g) TO5

This is the output pin of the 16-bit timer.

(h) CPT5

This is the capture edge input pin.

(i) INTP0 to INTP3

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

(j) CMPTOUT0

This is the comparator output pin.

Caution When using P20 to P27 as serial interface pins, the I/O mode and output latch must be set according to the function to be used. For details of the setting, refer to Table 13-2.

2.2.3 P40 to P47 (Port 4)

These pins constitute an 8-bit I/O port. In addition to I/O port pins, these pins can also function as key return signal detection pins.

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

(1) Port mode

In this mode, P40 to P47 function as an 8-bit I/O port. These pins can be set to input or output mode in 1-bit units by using port mode register 4 (PM4). When used as input port pins, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, the pins function as key return signal detection pins (KR0 to KR5).

2.2.4 P50 to P53 (Port 5)

These pins constitute a 4-bit N-channel open-drain I/O port. In the mask ROM version, it is possible to specify that pull-up resistors be used, via a mask option.

2.2.5 P60 to P66 (Port 6)

These pins constitute a 7-bit input-only port. In addition to general-purpose input port pins, these pins can also function as A/D converter analog input pins and comparator input pins.

(1) Port mode

In this port mode, P60 to P66 function as a 7-bit input-only port.

(2) Control mode

In this mode, the pins can be used as A/D converter analog inputs and comparator inputs.

(a) ANI0 to ANI6

These are the A/D converter analog input pins.

(b) CMPIN0

This is the comparator input pin.

(c) CMPREF0

This is the comparator reference voltage input pin.

2.2.6 P80 to P87 (Port 8)

These pins constitute an 8-bit I/O port. In addition to I/O port pins, these pins can also function as LCD controller/driver segment signal.

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

(1) Port mode

In this port mode, P80 to P87 function as an 8-bit I/O port. These pins can be set to input or output mode in 1-bit units by using port mode register 8 (PM8). When used as an input port pins, an on-chip pull-up resistor can be used by setting pull-up resistor option register 2 (PU2).

(2) Control mode

In this mode, P80 to P87 function as segment signal output pins (S20 to S27) for the LCD controller/driver.

2.2.7 P90 to P93 (Port 9)

These pins constitute a 4-bit I/O port. In addition to I/O port pins, these pins can also function as LCD controller/driver segment signal.

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

(1) Port mode

In this mode, P90 to P93 function as a 4-bit I/O port. These pins can be set to input or output mode in 1-bit units by using port mode register 9 (PM9). When used as input port pins, an on-chip pull-up resistor can be used by setting pull-up resistor option register 2 (PU2).

(2) Control mode

In this mode, P90 to P93 function as segment signal output pins (S16 to S19) for the LCD controller/driver.

2.2.8 S0 to S15

These pins are segment signal output pins for the LCD controller/driver.

2.2.9 COM0 to COM3

These pins are common signal output pins for the LCD controller/driver.

2.2.10 V_{LC0} to V_{LC2}

These pins are power supply voltage pins to drive the LCD.

2.2.11 BIAS

This pin supplies power to drive the LCD.

2.2.12 AV_{REF}

★ This pin is the A/D converter reference voltage pin. Connect it to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} when not using the A/D converter.

2.2.13 AV_{DD}

This pin is the A/D converter analog circuit power supply pin. Always keep it at the same potential as the V_{DD0} pin (even when the A/D converter is not used).

2.2.14 AV_{SS}

This pin is the A/D converter ground potential pin. Always keep it at the same potential as the V_{SS0} pin (even when the A/D converter is not used).

2.2.15 $\overline{RESET}$

This pin inputs an active-low system reset signal.

2.2.16 X1, X2

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

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

2.2.17 XT1, XT2

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

To supply an external clock, input the clock to XT1 and input the inverted signal to XT2.

2.2.18 V_{DD0}, V_{DD1}

V_{DD0} is the positive power supply pin for ports, while V_{DD1} is the positive power supply pin for other than ports.

2.2.19 V_{SS0}, V_{SS1}

V_{SS0} is the ground potential pin for ports, while the V_{SS1} is the ground potential pin for other than ports.

2.2.20 V_{PP} (μPD78F9418A only)

A high voltage should be applied to this pin when the flash memory programming mode is set and when the program is written or verified.

★ Handle the pins in either of the following ways.

- Independently connect a 10 kΩ pull-down resistor.
- Switch this pin to be directly connected to the dedicated flash programmer in programming mode or to V_{SS0} or V_{SS1} in normal operation mode using a jumper on the board.

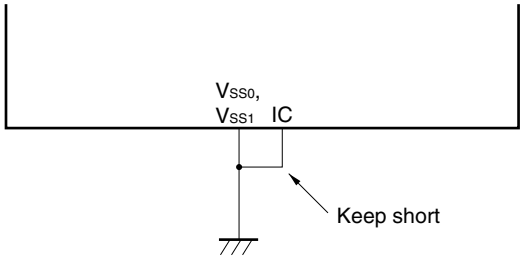
If the wiring between the V_{PP} pin and V_{SS0} or V_{SS1} pin is long, or external noise is superimposed on the V_{PP} pin, the user program may not run correctly.

2.2.21 IC (mask ROM version only)

The IC (internally connected) pin is used to set the μPD789407A and μPD789417A Subseries in the test mode before shipment. In the normal operation mode, directly connect this pin to the V_{SS0} or V_{SS1} pin with as short a wiring length as possible.

If a potential difference is generated between the IC pin and V_{SS0} or V_{SS1} pin due to a long wiring length between these pin, or due to external noise superimposed on the IC pin, the user program may not run correctly.

- Directly connect the IC pin to the V_{SS0} or V_{SS1} pin.



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

The I/O circuit type of each pin and recommended connection of unused pins are shown in Table 2-1. For the I/O circuit configuration of each type, see 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 to P03	5-H	I/O	Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P20/$\overline{SCK}$/ASCK	8-C		
P21/SO/TxD			
P22/SI/RxD			
P23/CMPTOUT0/TO2	10-B		
P24/INTP0/TI0	8-C		Input: Independently connect to V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
P25/INTP1/TI1			
P26/INTP2/TO5			
P27/INTP3/CPT5			
P40/KR0 to P45/KR5			Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P46, P47	5-H		
P50 to P53 (Mask ROM version)	13-U		Input: Independently connect to V_{DD0} or V_{DD1} via a resistor. Output: Leave open.
P50 to P53 (μPD78F9418A)	13-T		
P60/ANI0/CMPIN0	9-D	Input	Connect directly to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1}.
P61/ANI1/CMPREF0			
P62/ANI2 to P66/ANI6	9-C		
P80/S27 to P87/S20	17-F	I/O	Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P90/S19 to P93/S16			
S0 to S15	17-B	Output	Leave open.
COM0 to COM3	18-A		
V_{LC0} to V_{LC2}	–	–	
BIAS			Leave open. However, independently connect to V_{SS0} or V_{SS1} via a resistor when none of V_{LC0} to V_{LC2} are used.
AV_{DD}			Connect directly to V_{DD0} or V_{DD1}.
★ AV_{REF}			Connect directly to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1}.
AV_{SS}			Connect directly to V_{SS0} or V_{SS1}.
XT1		Input	Connect directly to V_{SS0} or V_{SS1}.
XT2		–	Leave open.
$\overline{RESET}$	2	Input	–
IC (Mask ROM version)	–	–	Connect directly to V_{SS0} or V_{SS1}.
★ V_{PP} (μPD78F9418A)			Independently connect to a 10 kΩ pull-down resistor or connect directly to V_{SS0} or V_{SS1}.

Figure 2-1. Pin I/O Circuits (1/2)

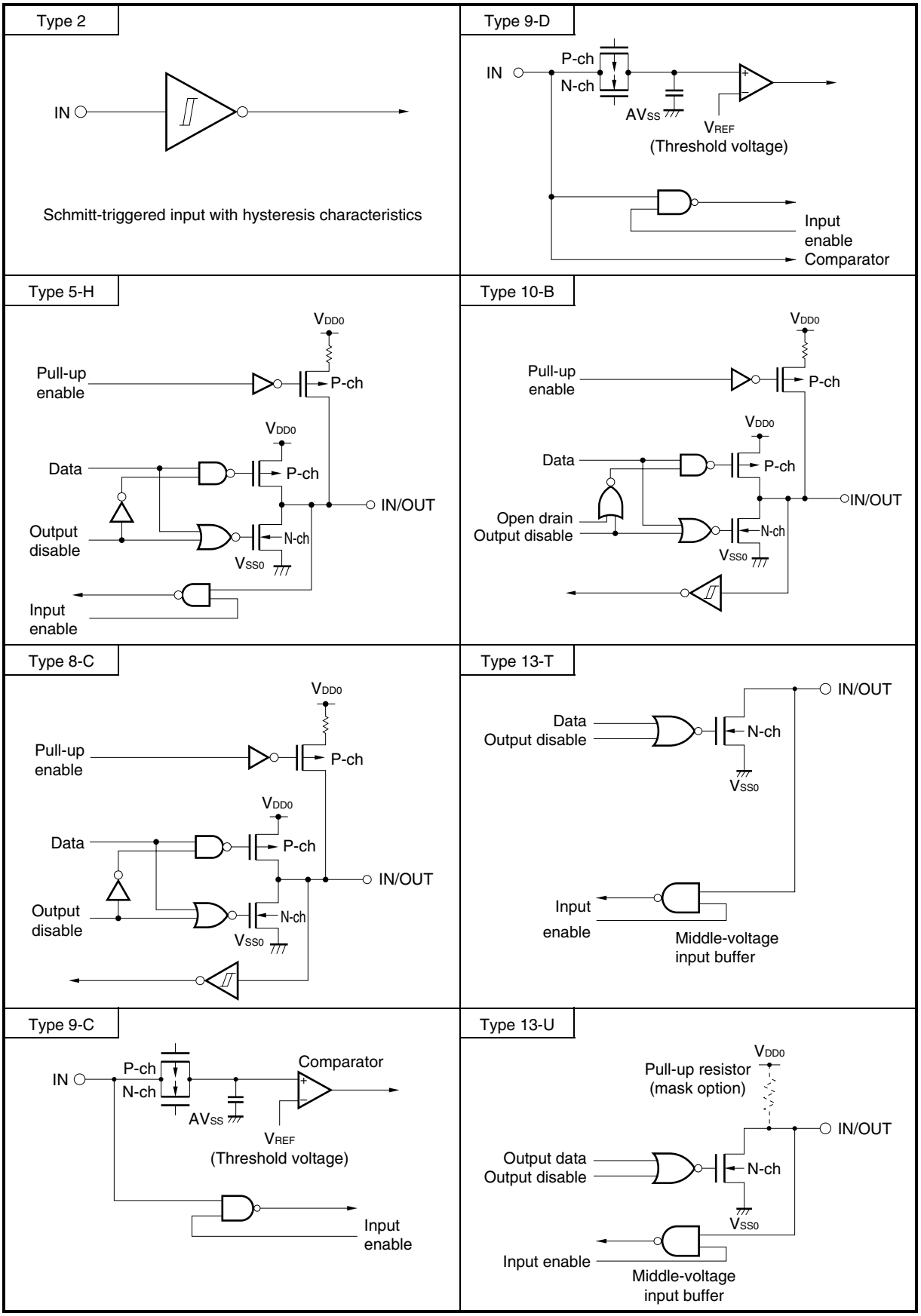
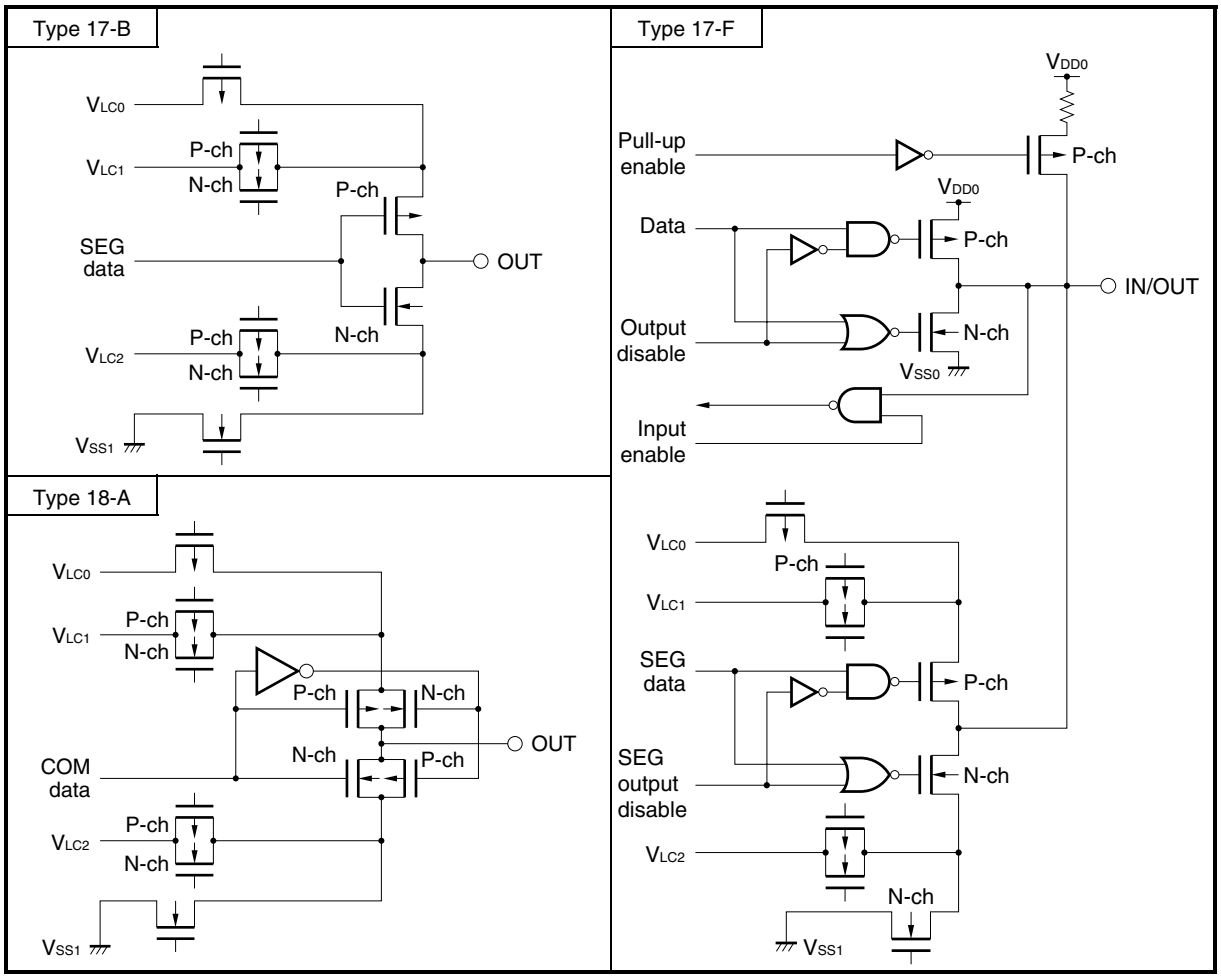


Figure 2-1. Pin I/O Circuits (2/2)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

The μPD789407A and μPD789417A Subseries can access 64 KB of memory space. Figures 3-1 through 3-4 show the memory maps.

Figure 3-1. Memory Map (μPD789405A and μPD789415A)

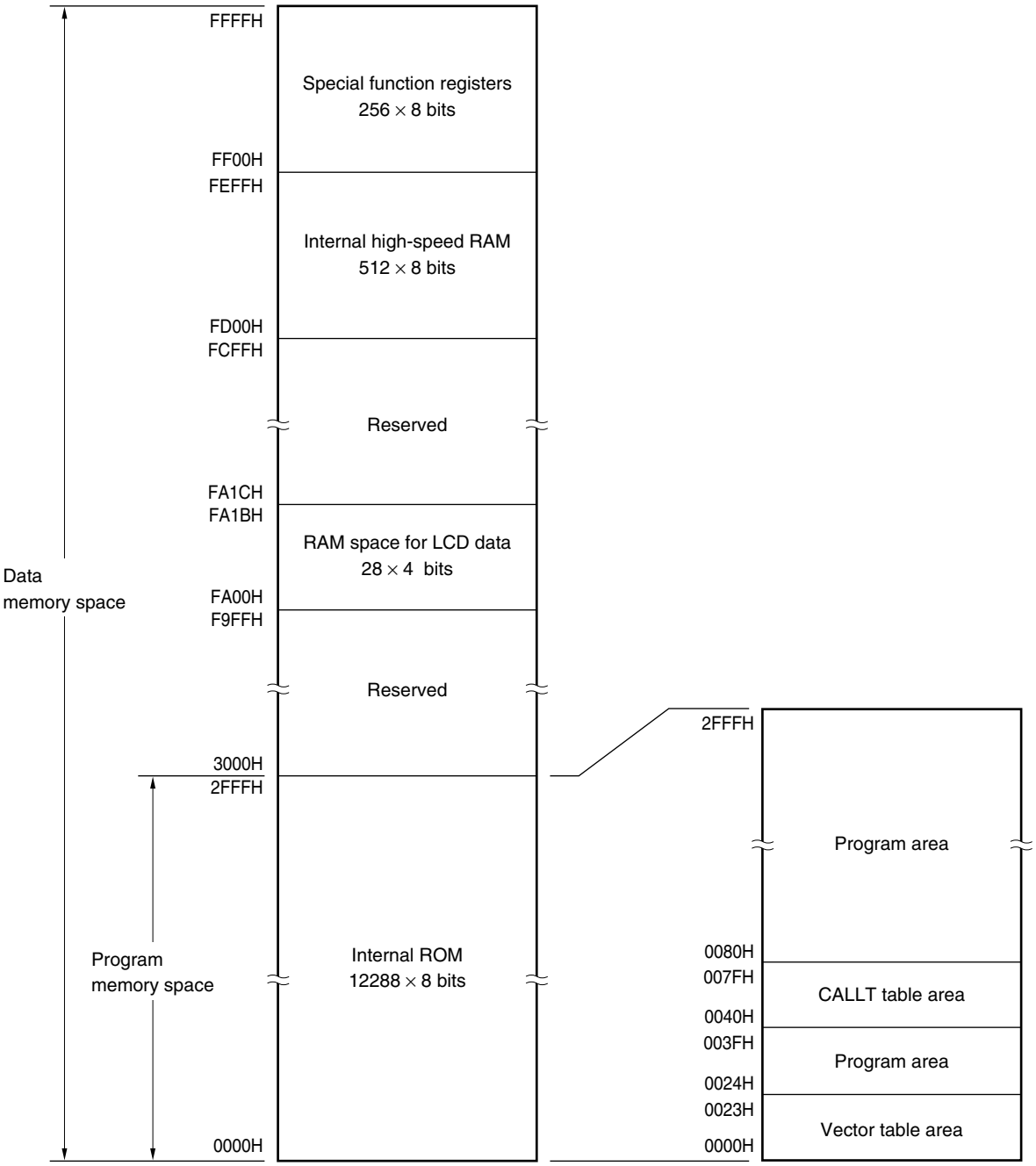


Figure 3-2. Memory Map (μPD789406A and μPD789416A)

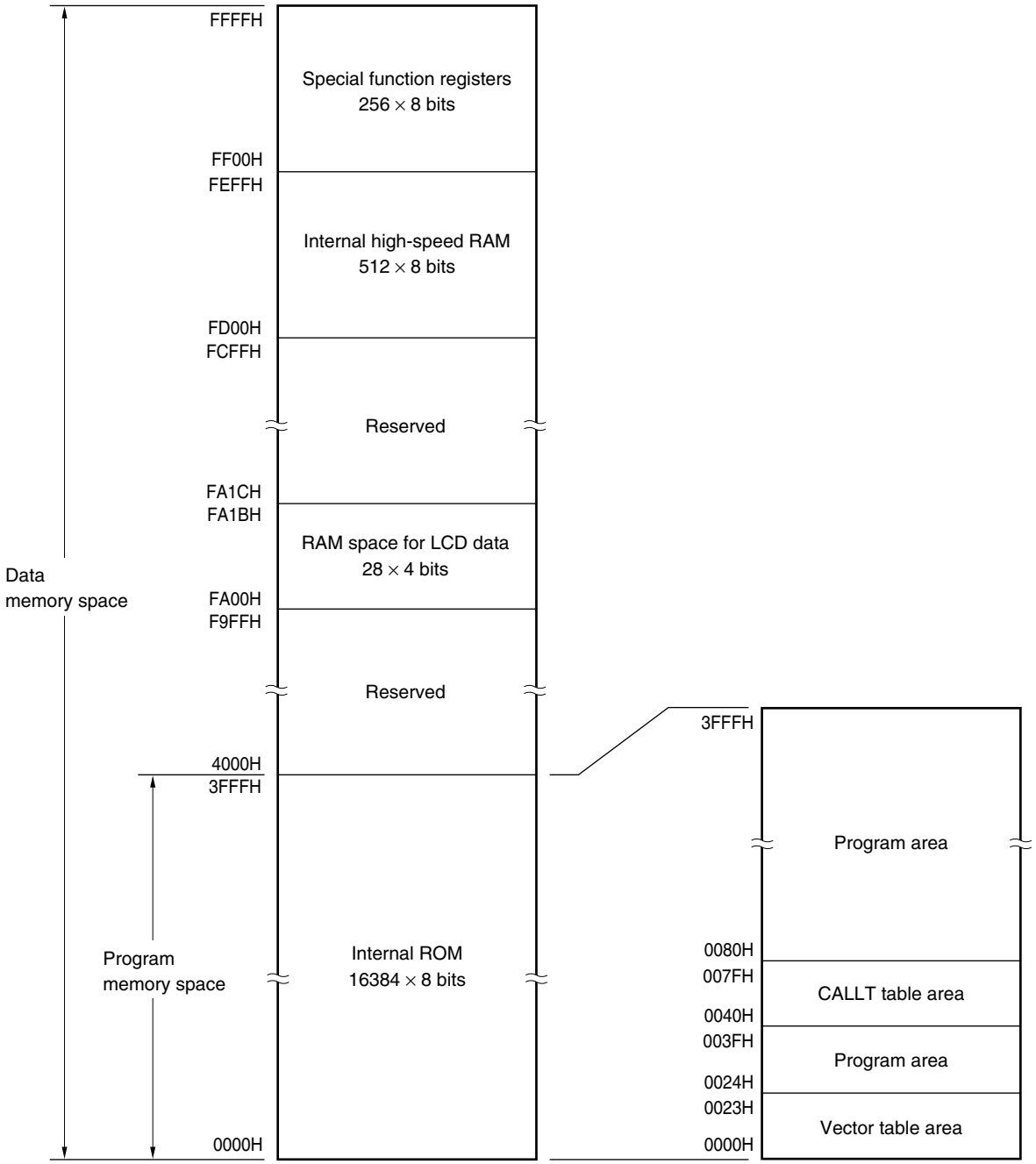


Figure 3-3. Memory Map (μPD789407A and μPD789417A)

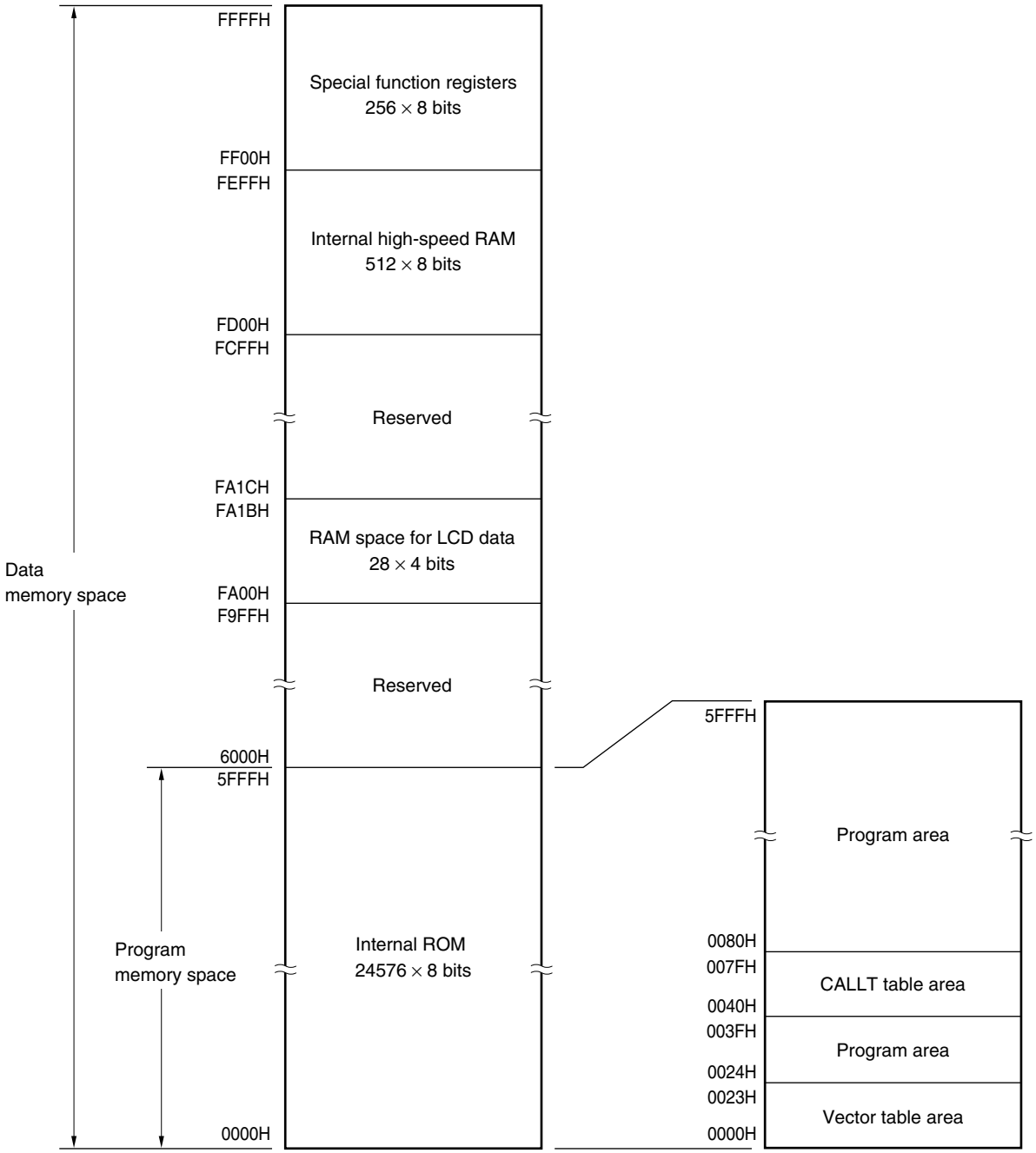
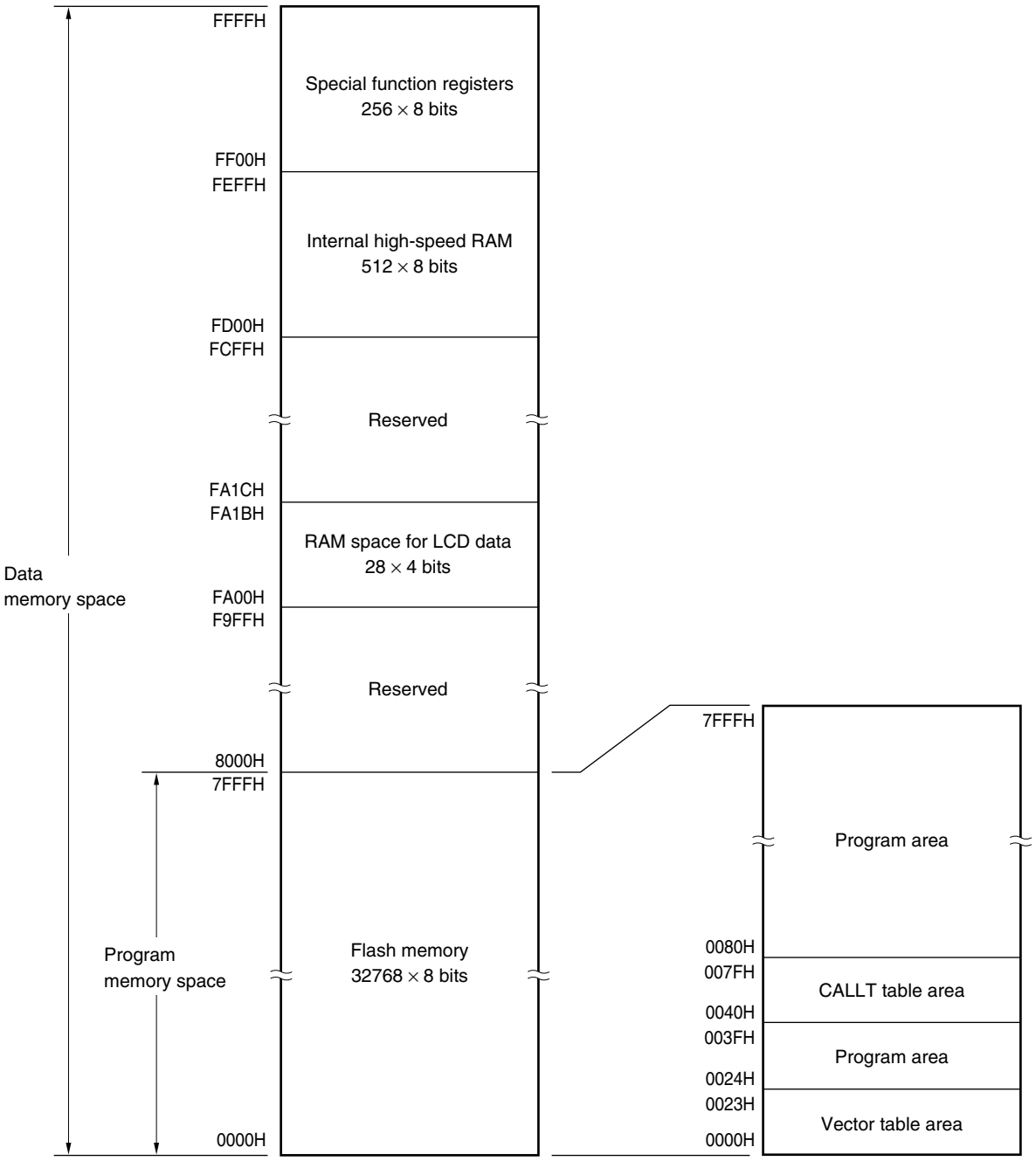


Figure 3-4. Memory Map (μPD78F9418A)



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).

The products in the PD789407A and PD789417A Subseries contain the following internal ROM (or flash memory) capacities.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
PD789405A, 789415A	Mask ROM	12288 8 bits
PD789406A, 789416A		16384 8 bits
PD789407A, 789417A		24576 8 bits
PD78F9418A	Flash memory	32768 8 bits

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

(1) Vector table area

The 36-byte area of addresses 0000H to 0023H is reserved as a vector table area. This area stores program start addresses to be used when branching by $\overline{\text{RESET}}$ input or interrupt request generation. Of a 16-bit program 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 Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	0014H	INTWTI
0004H	INTWDT	0016H	INTTM00
0006H	INTP0	0018H	INTTM01
0008H	INTP1	001AH	INTTM02
000AH	INTP2	001CH	INTTM50
000CH	INTP3	001EH	INTKR00
000EH	INTSR00/INTCSI00	0020H	INTAD0
0010H	INTST00	0022H	INTCMP0
0012H	INTWT		

(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.

3.1.2 Internal data memory space

The μPD789407A and μPD789417A Subseries products incorporate the following RAM:

(1) Internal high-speed RAM

An internal high-speed RAM is allocated to the area between FD00H and FEFFH.

The internal high-speed RAM is also used as a stack.

(2) LCD data RAM

An LCD data RAM is allocated to the area between FA00H and FA1BH.

The LCD display RAM can also be used as ordinary RAM.

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 (see **Table 3-3**).

3.1.4 Data memory addressing

The μPD789407A and μPD789417A Subseries are provided with a variety of addressing modes to make memory manipulation as efficient as possible. In the area that holds data memory (FD00H to FFFFH) especially, specific modes of addressing that correspond to the particular function of an area, such as the special function registers (SFR) or general-purpose registers, are available. Figures 3-5 through 3-8 show the data memory addressing modes.

Figure 3-5. Data Memory Addressing (μPD789405A and μPD789415A)

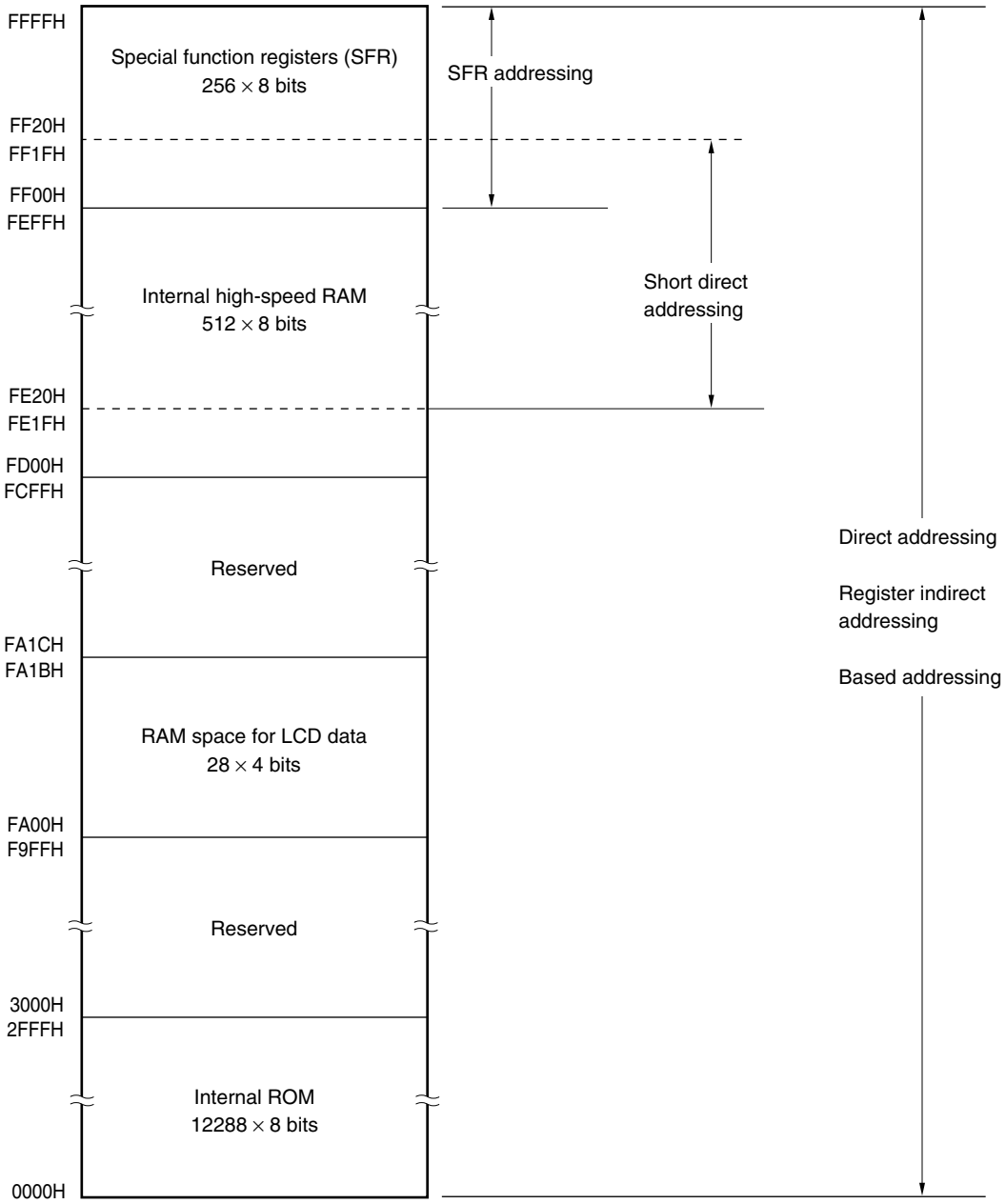


Figure 3-6. Data Memory Addressing (μPD789406A and μPD789416A)

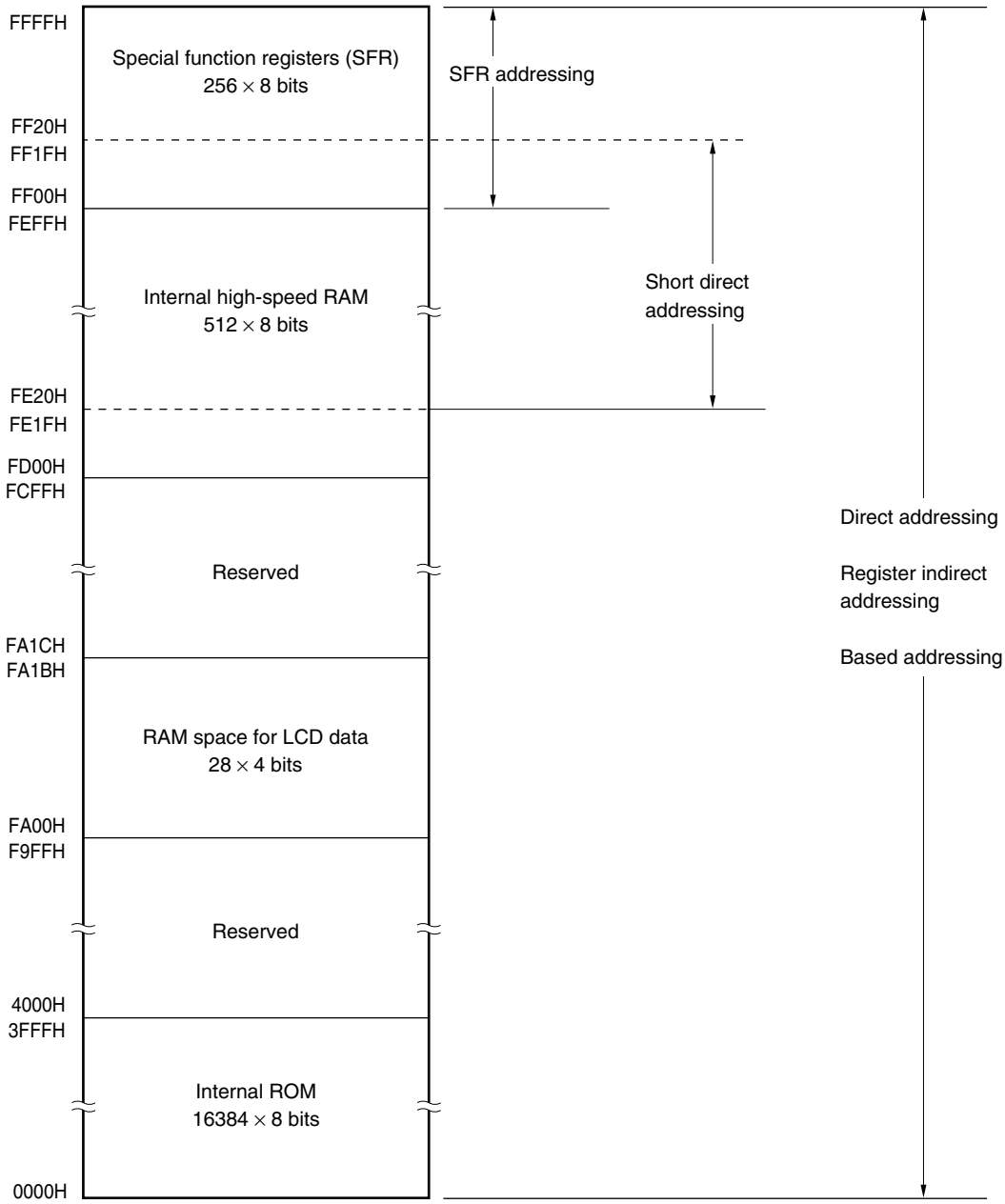


Figure 3-7. Data Memory Addressing (μPD789407A and μPD789417A)

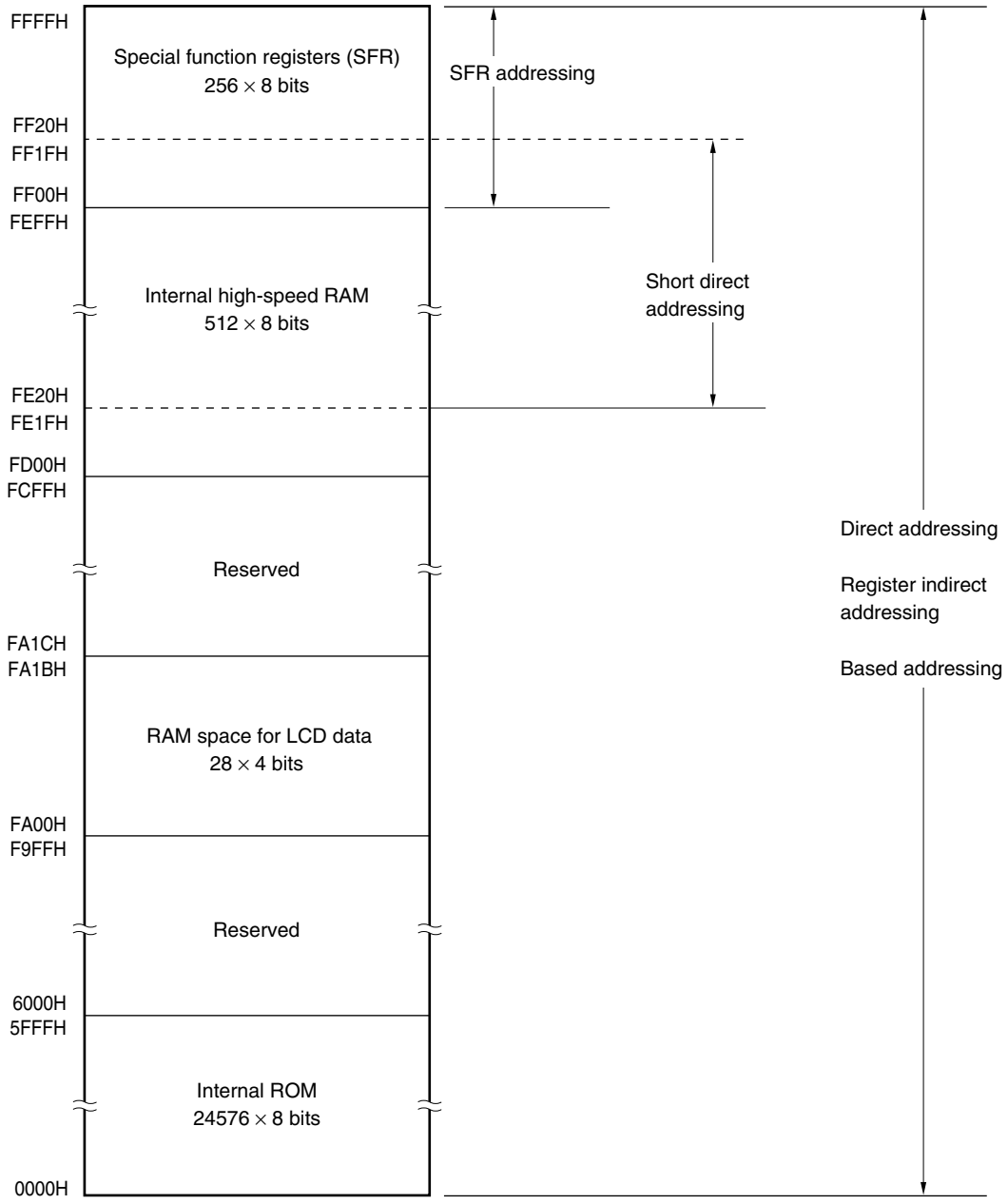
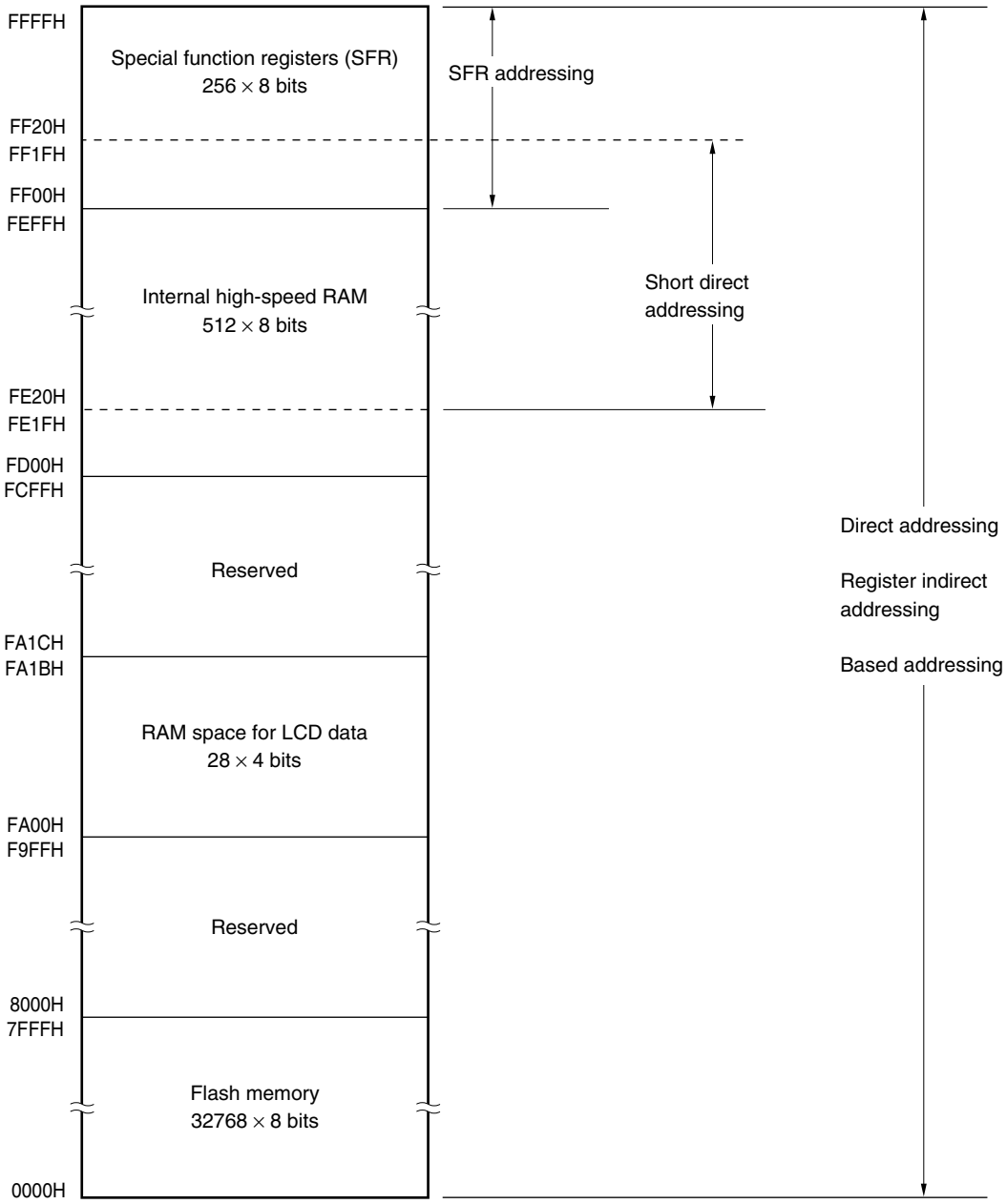


Figure 3-8. Data Memory Addressing (μPD78F9418A)



3.2 Processor Registers

The PD789407A and PD789417A Subseries are provided with the following on-chip processor registers.

3.2.1 Control registers

The control registers contains special functions to control the program sequence statuses and stack memory. A program counter, a program status word, and a stack pointer constitute the control registers.

(1) Program counter (PC)

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

In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data or register contents are set.

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

Figure 3-9. 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 various flags to be set/reset by instruction execution.

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

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

Figure 3-10. Program Status Word Configuration

7							0
IE	Z	0	AC	0	0	1	CY

(a) Interrupt enable flag (IE)

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

When 0, IE is set to the interrupt disable status (DI), and all interrupt requests other than non-maskable interrupts are disabled.

When 1, IE is set to the interrupt enable status (EI). At this time, interrupt request acknowledgment is controlled by an interrupt mask flag corresponding to the interrupt source.

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

(b) Zero flag (Z)

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

(c) Auxiliary carry flag (AC)

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

(d) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-11. Stack Pointer Configuration

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

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

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

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

Figure 3-12. Data Saved to Stack Memory

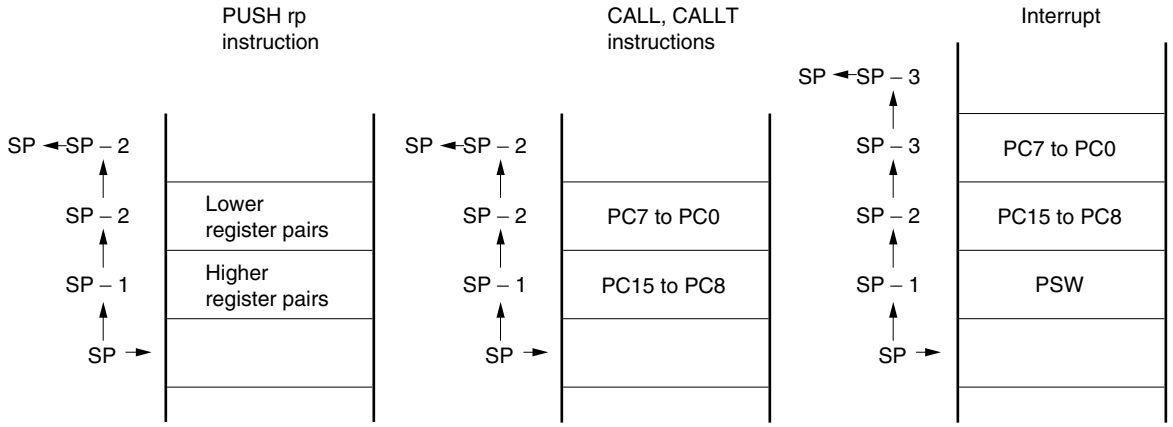
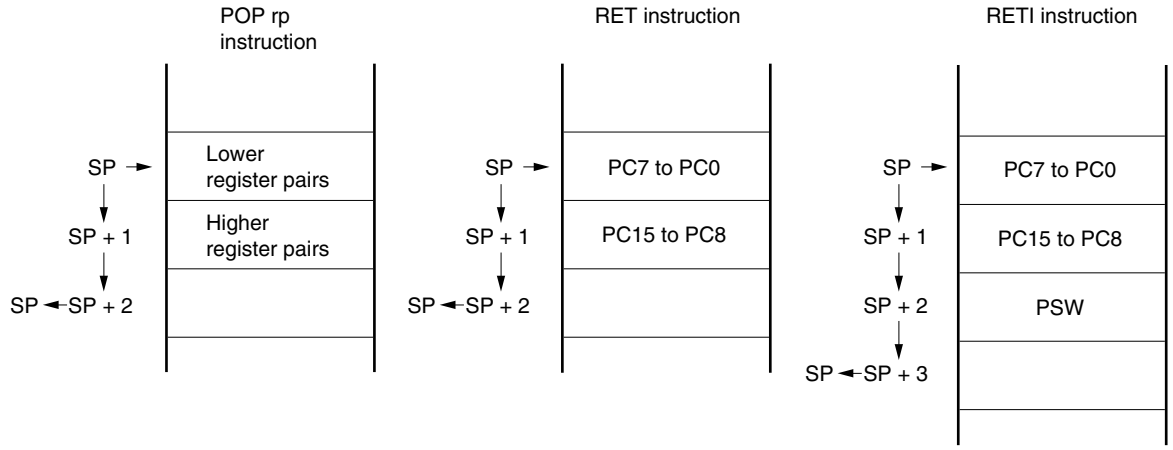


Figure 3-13. Data Restored from Stack Memory



3.2.2 General-purpose registers

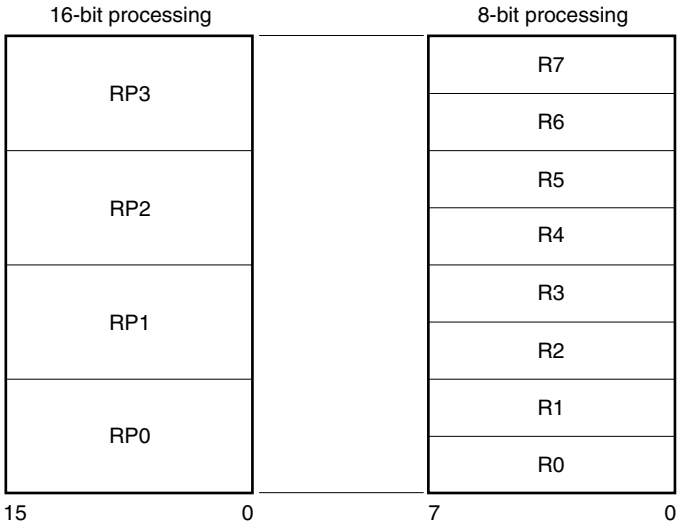
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

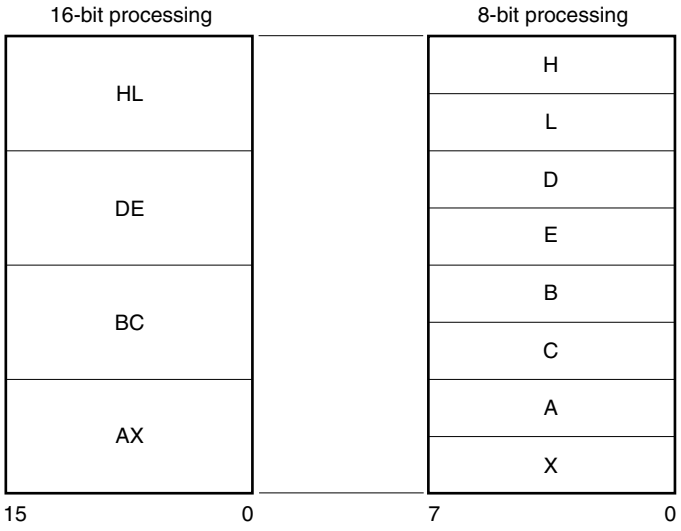
General-purpose registers can be described in terms of functional names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Figure 3-14. General-Purpose Register Configuration

(a) Absolute names



(b) Functional names



3.2.3 Special function registers (SFR)

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

SFRs are allocated in the 256-byte area FF00H to FFFFH.

A special function register can be manipulated, like a general-purpose register, using operation, transfer, and bit manipulation instructions. The manipulatable bit unit (1, 8, or 16) differs depending on the special function register type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describes a symbol reserved by assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified by an address.

- 8-bit manipulation
  Describes a symbol reserved by assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified by an address.

- 16-bit manipulation
  Describes a symbol reserved by assembler for the 16-bit manipulation instruction operand. When addressing an address, describe an even address.

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

- Symbol
  Indicates the address of the special function register. The symbols shown in this column are reserved words in the assembler, and have been defined in the header file named “sfrbit.h” in the C compiler. Therefore, these symbols can be used as instruction operands if an assembler or integrated debugger is 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

- Manipulatable bit unit
  Indicates the bit units (1, 8, 16) in which the special function register in question can be manipulated.

- After reset
  Indicates the status of the special function register when the $\overline{\text{RESET}}$ signal is input.

Table 3-3. Special Function Register List (1/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W	√	√	–	00H
FF02H	Port 2	P2			√	√	–	
FF04H	Port 4	P4			√	√	–	
FF05H	Port 5	P5			√	√	–	
FF06H	Port 6	P6		R	√	√	–	
FF08H	Port 8	P8		R/W	√	√	–	
FF09H	Port 9	P9			√	√	–	
FF10H	Transmit shift register 00	TXS00	SIO00	W	–	√	–	FFH
	Receive buffer register 00	RXB00		R	–	√	–	Undefined
FF14H	A/D conversion result register 0	ADCR0			–	√Note 1	√Note 2	
FF15H								
FF16H	16-bit compare register 50	CR50L	CR50	W	–	–	√Notes 2, 3	FFFFH
FF17H		CR50H						
FF18H	16-bit timer counter 50	TM50L	TM50	R	–	–	√Notes 2, 3	0000H
FF19H		TM50H						
FF1AH	16-bit capture register 50	TCP50L	TCP50		–	–	√Notes 2, 3	Undefined
FF1BH		TCP50H						
FF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FF22H	Port mode register 2	PM2			√	√	–	
FF24H	Port mode register 4	PM4			√	√	–	
FF25H	Port mode register 5	PM5			√	√	–	
FF28H	Port mode register 8	PM8			√	√	–	
FF29H	Port mode register 9	PM9			√	√	–	
FF42H	Timer clock selection register 2	TCL2			–	√	–	00H
FF48H	16-bit timer mode control register 50	TMC50			√	√	–	
FF4AH	Watch timer mode control register	WTM			√	√	–	
FF4EH	Comparator mode register 0	CMPRM0			√	√	–	

Notes 1. If the A/D conversion result register is used for the 8-bit A/D converter (μPD789407A Subseries), it can be accessed only in 8-bit units. In this case, it is considered to have been mapped at address FF15H. If the register is used for the 10-bit A/D converter (μPD789417A Subseries), it can be accessed only in 16-bit units. If the μPD78F9418A is used as the flash memory version of the μPD789405A, μPD789406A, or μPD789407A, 8-bit access is also possible, provided that the object file has been assembled using the μPD789405A, μPD789406A, or μPD789407A.

2. 16-bit access is possible only in short direct addressing.

3. Although CR50, TM50, and TCP50 are 16-bit access dedicated registers, an 8-bit access is also possible. When performing an 8-bit access, use direct addressing.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF50H	8-bit compare register 00	CR00	W	–	√	–	Undefined
FF51H	8-bit timer counter 00	TM00	R	–	√	–	00H
FF53H	8-bit timer mode control register 00	TMC00	R/W	√	√	–	
FF54H	8-bit compare register 01	CR01	W	–	√	–	Undefined
FF55H	8-bit timer counter 01	TM01	R	–	√	–	00H
FF57H	8-bit timer mode control register 01	TMC01	R/W	√	√	–	
FF58H	8-bit compare register 02	CR02	W	–	√	–	Undefined
FF59H	8-bit timer counter 02	TM02	R	–	√	–	00H
FF5BH	8-bit timer mode control register 02	TMC02	R/W	√	√	–	
FF70H	Asynchronous serial interface mode register 00	ASIM00		√	√	–	
FF71H	Asynchronous serial interface status register 00	ASIS00	R	√	√	–	
FF72H	Serial operation mode register 00	CSIM00	R/W	√	√	–	
FF73H	Baud rate generator control register 00	BRGC00		–	√	–	
FF80H	A/D converter mode register 0	ADM0		√	√	–	
FF84H	A/D input selection register 0	ADS0		√	√	–	
FFB0H	LCD display mode register 0	LCDM0		√	√	–	
FFB1H	LCD port selector 0	LPS0		√	√	–	
FFB2H	LCD clock control register 0	LCDC0		√	√	–	
FFE0H	Interrupt request flag register 0	IF0		√	√	–	
FFE1H	Interrupt request flag register 1	IF1		√	√	–	
FFE4H	Interrupt mask flag register 0	MK0		√	√	–	FFH
FFE5H	Interrupt mask flag register 1	MK1		√	√	–	
FFECH	External interrupt mode register 0	INTM0		–	√	–	00H
FFEDH	External interrupt mode register 1	INTM1		–	√	–	
FFF0H	Suboscillation mode register	SCKM		√	√	–	
FFF2H	Subclock control register	CSS		√	√	–	
FFF3H	Pull-up resistor option register 1	PU1		√	√	–	
FFF4H	Pull-up resistor option register 2	PU2		√	√	–	
FFF5H	Key return mode register 00	KRM00		√	√	–	
FFF7H	Pull-up resistor option register 0	PU0		√	√	–	
FFF9H	Watchdog timer mode register	WDTM		√	√	–	
FFFAH	Oscillation stabilization time selection register	OSTS		–	√	–	04H
FFFBH	Processor clock control register	PCC		√	√	–	02H

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents. PC contents are normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing (for details of each instruction, refer to the **78K/0S Series Instructions User's Manual (U11047E)**).

3.3.1 Relative addressing

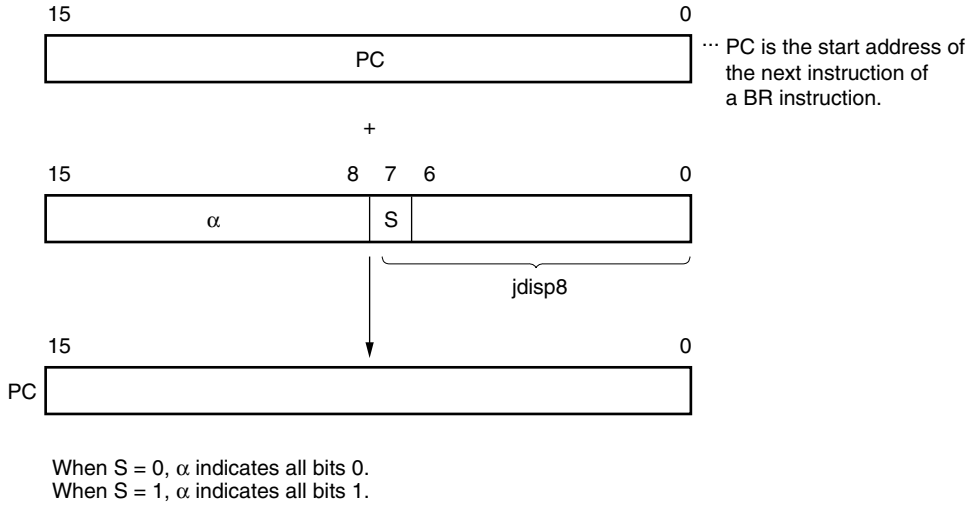
[Function]

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

This means that information is relatively branched to a location between −128 and +127, from the start address of the next instruction when relative addressing is used.

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

[Illustration]



When S = 0, α indicates all bits 0.
When S = 1, α indicates all bits 1.

3.3.2 Immediate addressing

[Function]

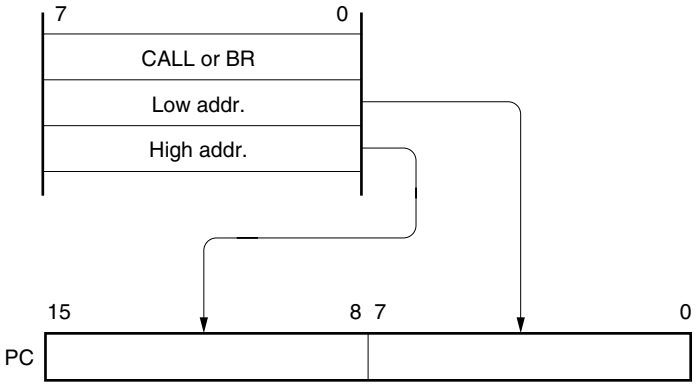
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

The CALL !addr16 and BR !addr16 instructions can be branched to any location in the memory space.

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



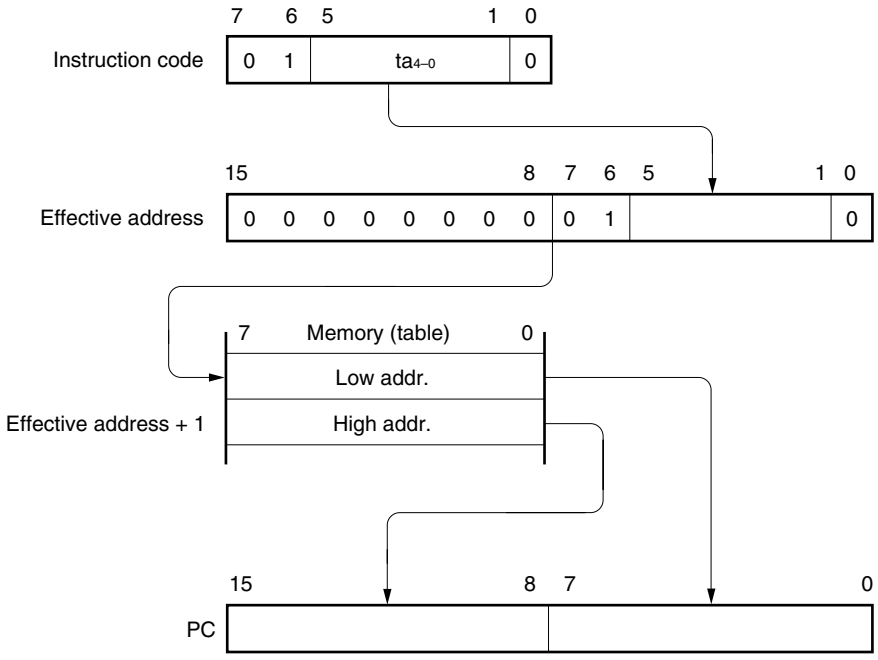
3.3.3 Table indirect addressing

[Function]

The table contents (branch destination address) of the particular location to be addressed by the lower 5-bit immediate data of an instruction code from bit 1 to bit 5 are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed. The instruction enables a branch to any location in the memory space by referring to the addresses stored in the memory table at 40H to 7FH.

[Illustration]



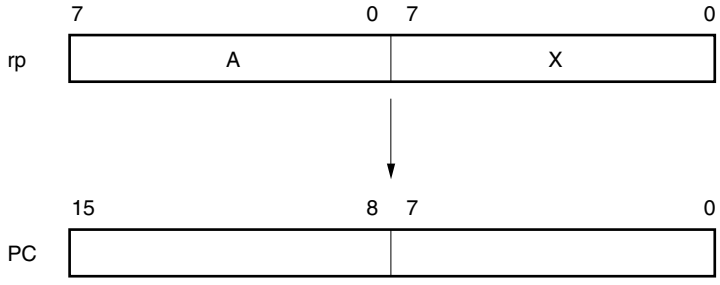
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Direct addressing

[Function]

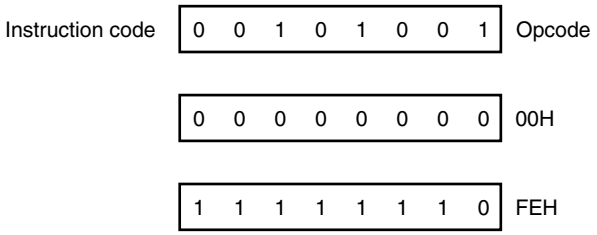
The memory indicated with immediate data in an instruction word is directly addressed.

[Operand format]

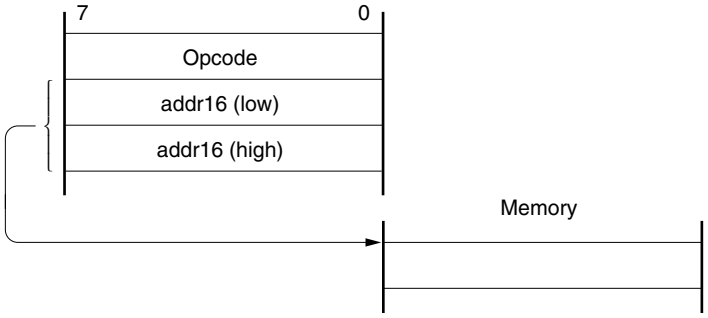
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !FE00H; When setting !addr16 to FE00H



[Illustration]



3.4.2 Short direct addressing

[Function]

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

The fixed space is the 256-byte space FE20H to FF1FH where the addressing is applied. An internal high-speed RAM and special function registers (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

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

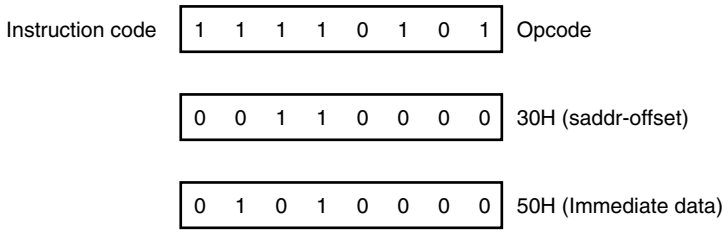
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

[Operand format]

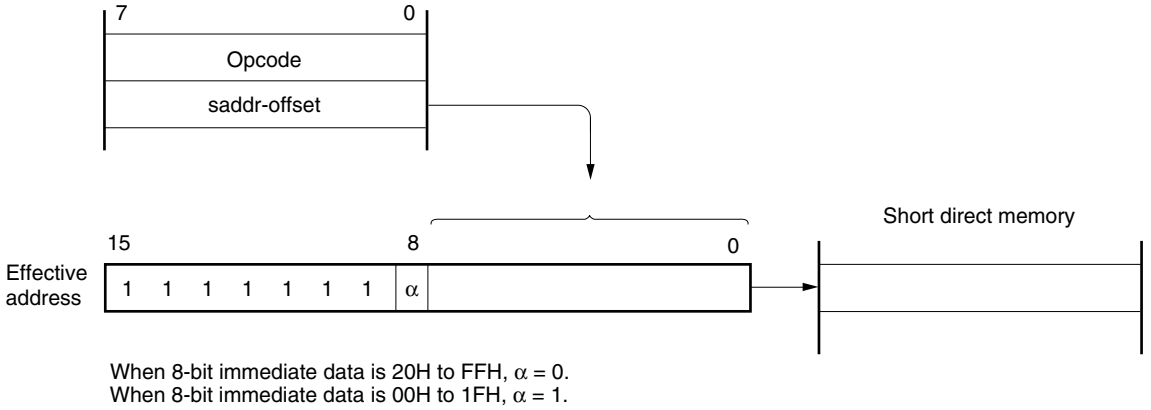
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

MOV FE30H, #50H; When setting saddr to FE30H and the immediate data to 50H



[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$.
When 8-bit immediate data is 00H to 1FH, $\alpha = 1$.

3.4.3 Special function register (SFR) addressing

[Function]

A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word.

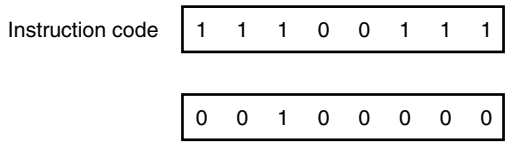
This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed using short direct addressing.

[Operand format]

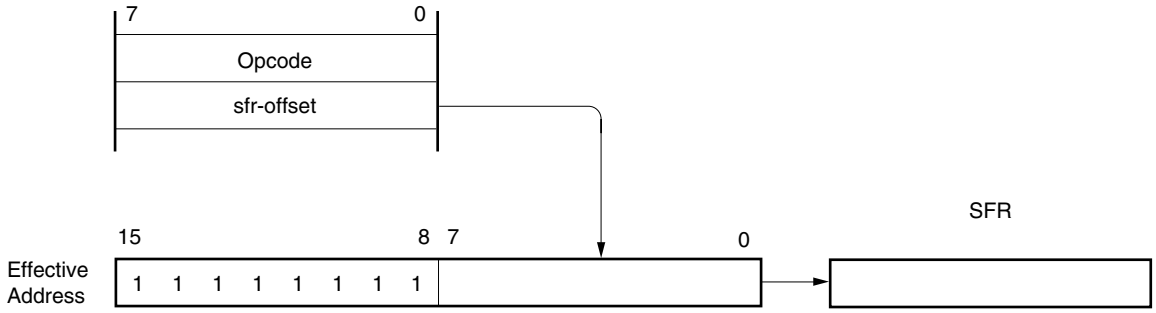
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr



[Illustration]



3.4.4 Register addressing

[Function]

In the register addressing mode, general-purpose registers are accessed as operands. The general-purpose register to be accessed is specified by the register specification code or functional name in the instruction code. Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the instruction code.

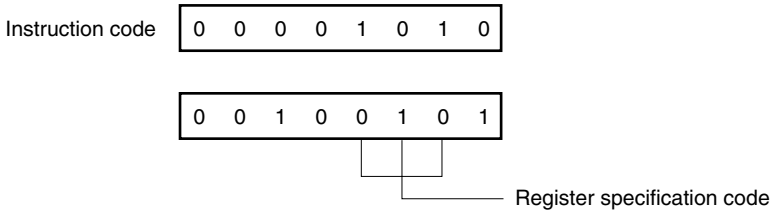
[Operand format]

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

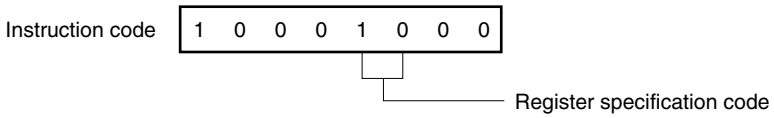
r and rp can be described using absolute names (R0 to R7 and RP0 to RP3) as well as functional names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; When selecting the C register for r



INCW DE; When selecting the DE register pair for rp



3.4.5 Register indirect addressing

[Function]

In the register indirect addressing mode, memory is manipulated according to the contents of a register pair specified as an operand. The register pair to be accessed is specified by the register pair specification code in an instruction code.

This addressing can be carried out for all the memory spaces.

[Operand format]

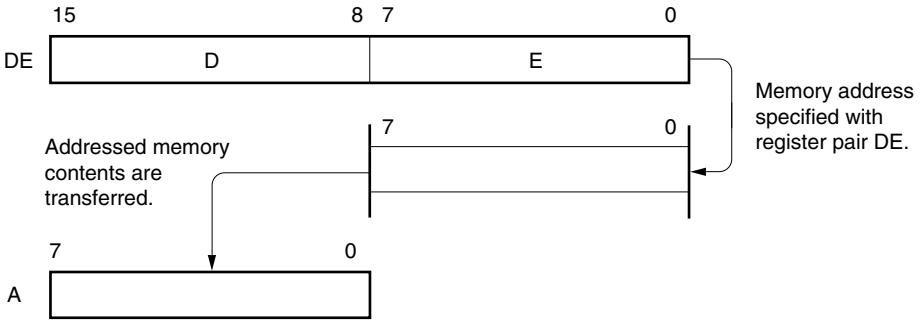
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; When selecting register pair [DE]

Instruction code | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

[Illustration]



3.4.6 Based addressing

[Function]

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

[Operand format]

Identifier	Description
–	[HL+byte]

[Description example]

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

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

3.4.7 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Instruction code	1	0	1	0	1	0	1	0

CHAPTER 4 PORT FUNCTIONS

4.1 Function of Port

The μPD789407A and μPD789417A Subseries are provided with the ports shown in Figure 4-1, enabling various methods of control.

Numerous other functions are provided that can be used in addition to the digital I/O port function. For more information on these additional functions, see **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Types

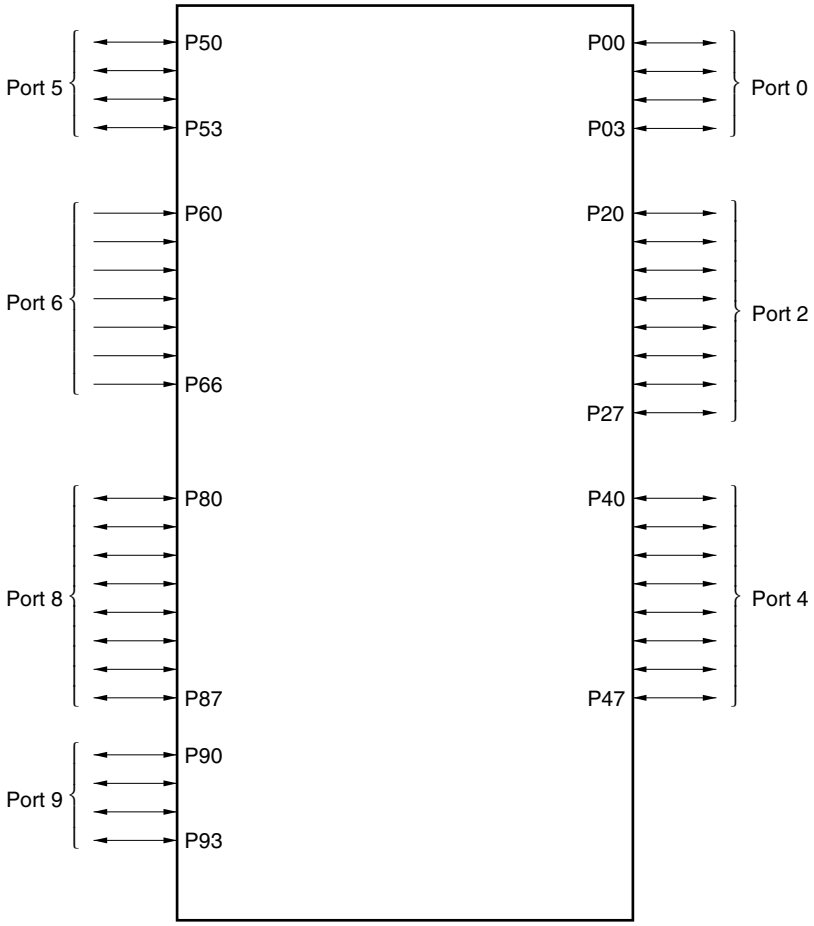


Table 4-1. Port Functions

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 1 (PU1).	Input	$\overline{\text{SCK}}$/ASCK
P21				SO/TxD
P22				SI/RxD
P23				CMPTOUT0/TO2
P24				INTP0/TI0
P25				INTP1/TI1
P26				INTP2/TO5
P27				INTP3/CPT5
P40 to P45	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	KR0 to KR5
P46, P47				–
P50 to P53	I/O	Port 5. 4-bit N-ch open-drain I/O port. Input/output can be specified in 1-bit units. For a mask ROM version, use of an on-chip pull-up resistor can be specified by the mask option.	Input	–
P60	Input	Port 6. 7-bit input port.	Input	ANI0/CMPIN0
P61				ANI1/CMPREF0
P62 to P66				ANI2 to ANI6
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 2 (PU2).	Input	S27 to S20
P90 to P93	I/O	Port 9. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 2 (PU2).	Input	S19 to S16

4.2 Configuration of Ports

The ports consist of the following hardware.

Table 4-2. Configuration of Port

Item	Configuration
Control registers	Port mode registers (PMm: m = 0, 2, 4, 5, 8, 9) Pull-up resistor option registers (PUm: m = 0 to 2)
Ports	Total: 43 (input: 7, I/O: 36)
Pull-up resistors	• Mask ROM version Total: 36 (software control: 32, mask option control: 4) • Flash memory version Total: 32 (software control only)

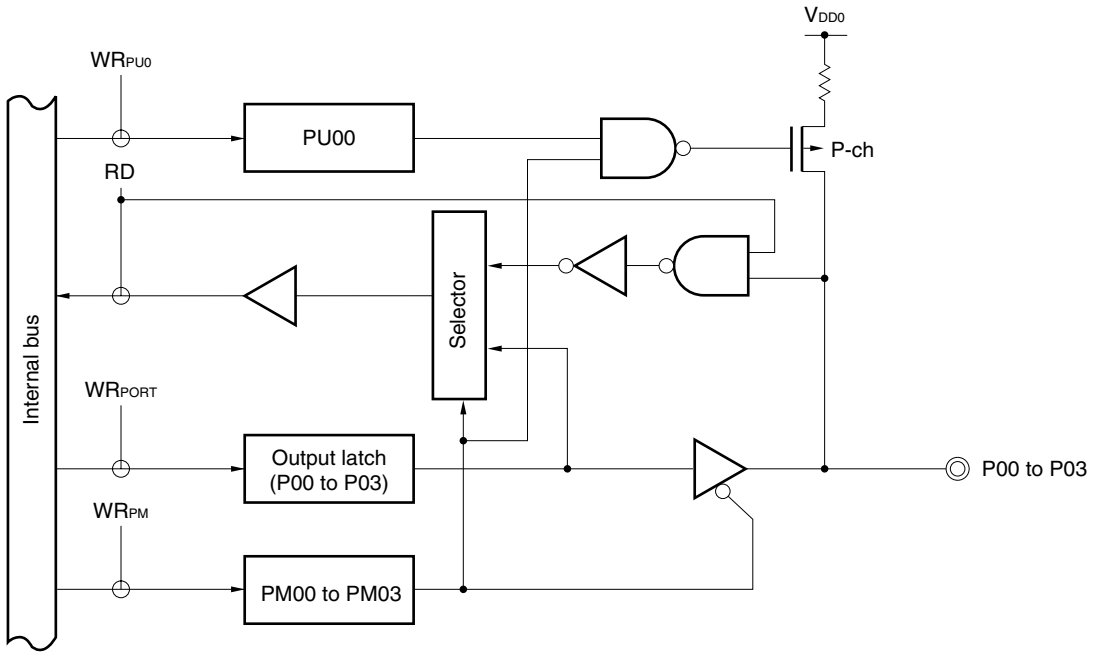
4.2.1 Port 0

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

Port 0 is set to input mode when the $\overline{\text{RESET}}$ signal is input.

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

Figure 4-2. Block Diagram of P00 to P03



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

4.2.2 Port 2

This is an 8-bit I/O port with an output latch. Port 2 can be specified as input or output in 1-bit units by using port mode register 2 (PM2). When using the P20 to P27 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by setting pull-up resistor option register 1 (PU1).

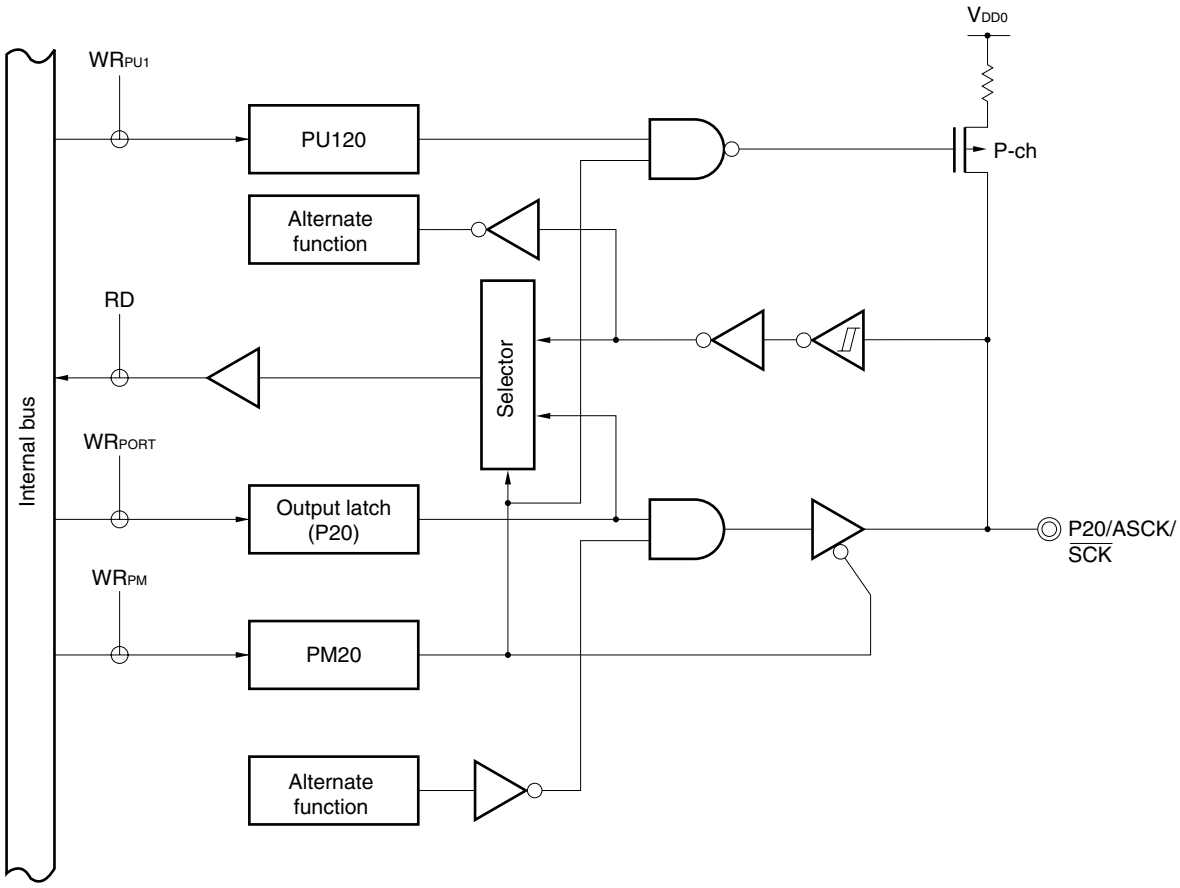
Port 2 is also used as a data I/O and clock I/O to and from the serial interface, timer I/O, and external interrupt.

Port 2 is set to input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 4-3 through 4-7 show block diagrams of port 2.

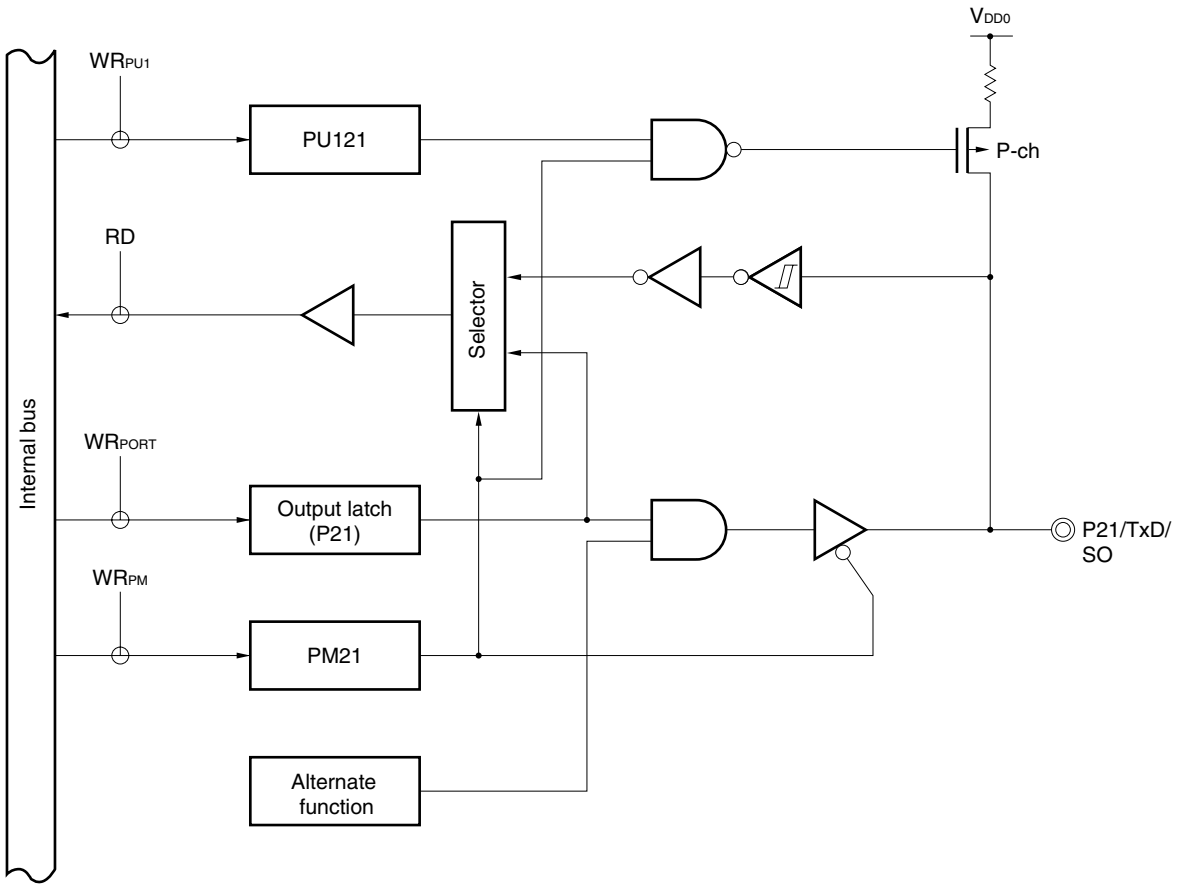
Caution When using the pins of port 2 for the serial interface, the I/O or output latch must be set according to the function to be used. For how to set the latches, see Table 13-2 Operation Mode Settings of Serial Interface 00.

Figure 4-3. Block Diagram of P20



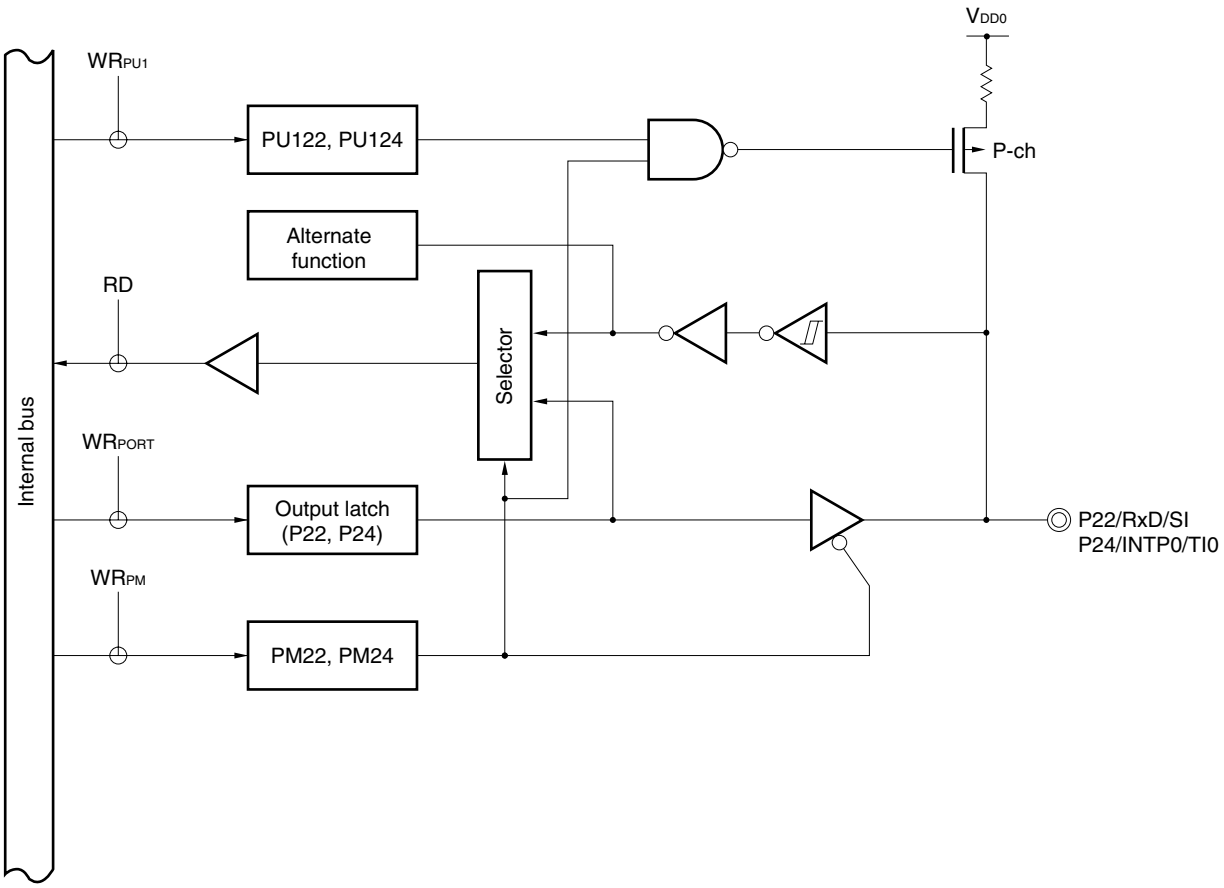
PU1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-4. Block Diagram of P21



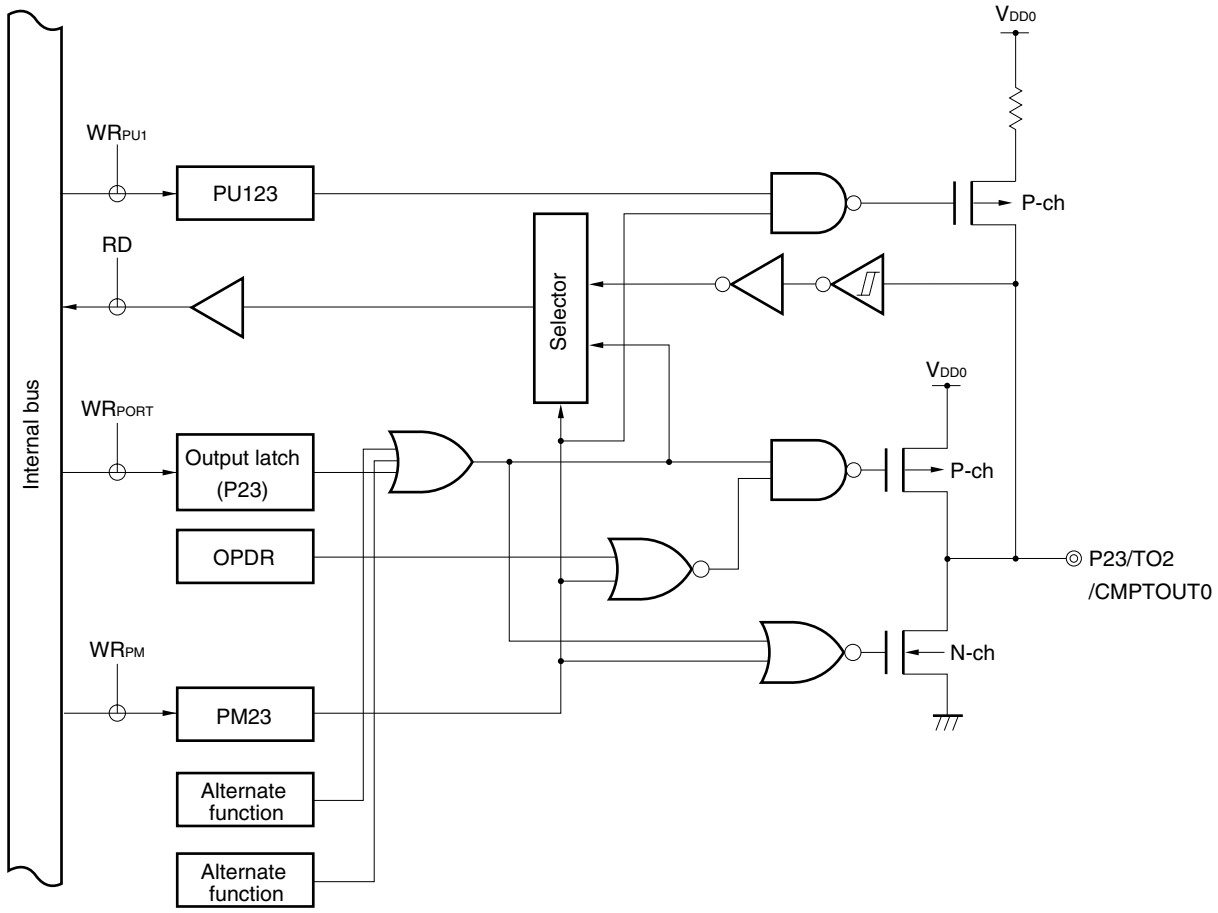
PU1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-5. Block Diagram of P22 and P24



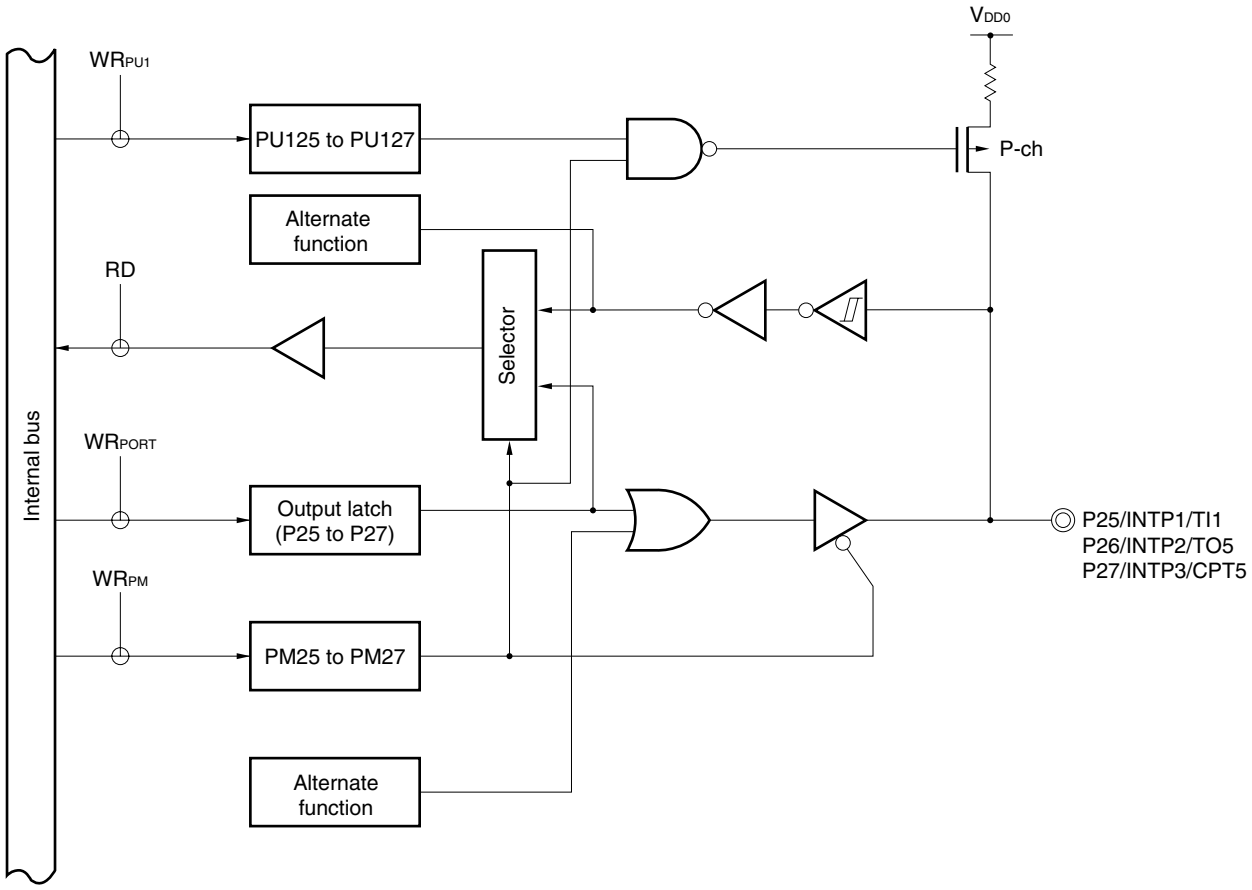
PU1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-6. Block Diagram of P23



OPDR: Bit 1 of comparator mode register 0, selection of N-ch open-drain output
PU1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-7. Block Diagram of P25 to P27



PU1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

4.2.3 Port 4

This is an 8-bit I/O port with an output latch. Port 4 can be specified as input or output in 1-bit units by using port mode register 4 (PM4). When using the P40 to P47 pins as input port pins, on-chip pull-up resistors can be connected in 8-bit units by setting pull-up resistor option register 0 (PU0).

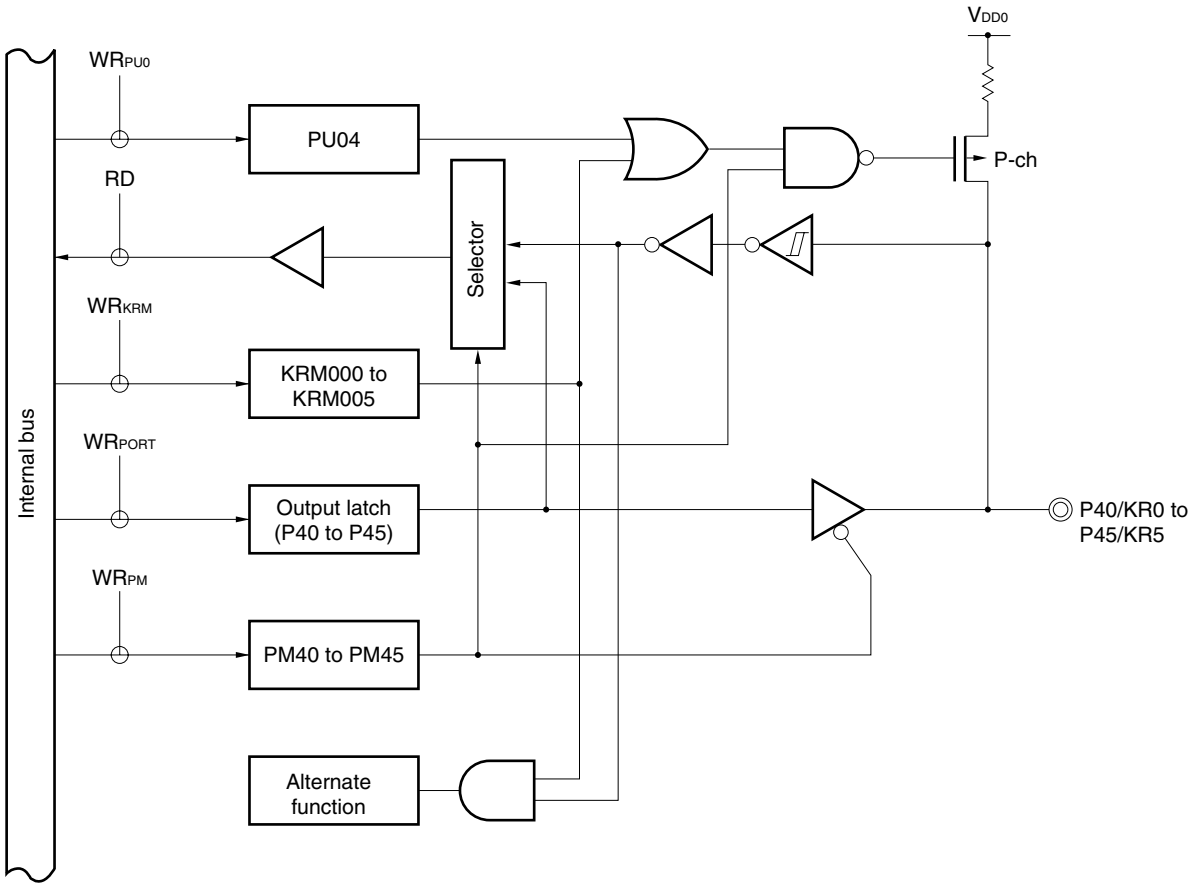
Port 4 is also used as a key return input.

Port 4 is set to input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 4-8 and 4-9 show block diagrams of port 4.

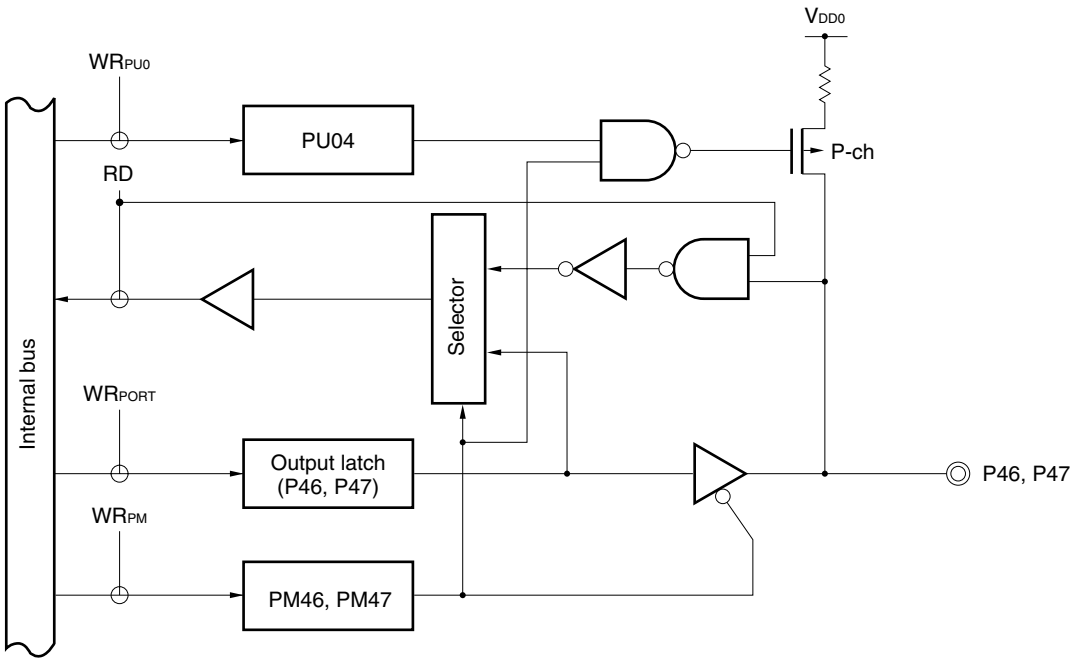
Caution When using the pins of port 4 as the key return, the key return mode register must be set according to the function to be used. For how to set the registers, see 15.3 (6) Key return mode register 00 (KRM00).

Figure 4-8. Block Diagram of P40 to P45



KRM00: Key return mode register 00
PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 4 read signal
WR: Port 4 write signal

Figure 4-9. Block Diagram of P46 and P47



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 4 read signal
WR: Port 4 write signal

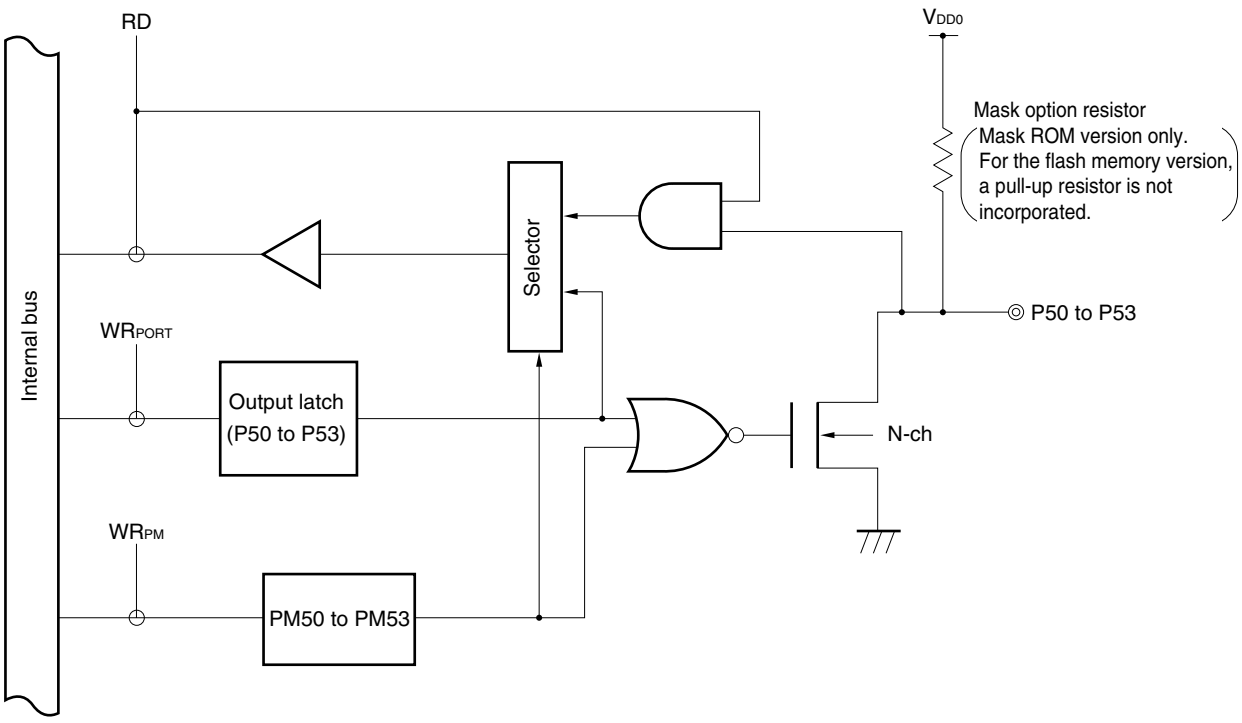
4.2.4 Port 5

This is a 4-bit N-ch open-drain I/O port with an output latch. Port 5 can be specified as input or output in 1-bit units by using port mode register 5 (PM5). For a mask ROM version, whether a pull-up resistor is to be incorporated can be specified by a mask option.

Port 5 is set to input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-10 shows a block diagram of port 5.

Figure 4-10. Block Diagram of P50 to P53



PM: Port mode register
RD: Port 5 read signal
WR: Port 5 write signal

4.2.5 Port 6

This is a 7-bit input port.

Port 6 is also used as an analog input to the A/D converter or comparator input.

Port 6 is set to input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 4-11 and 4-12 show block diagrams of port 6.

Figure 4-11. Block Diagram of P60 and P61

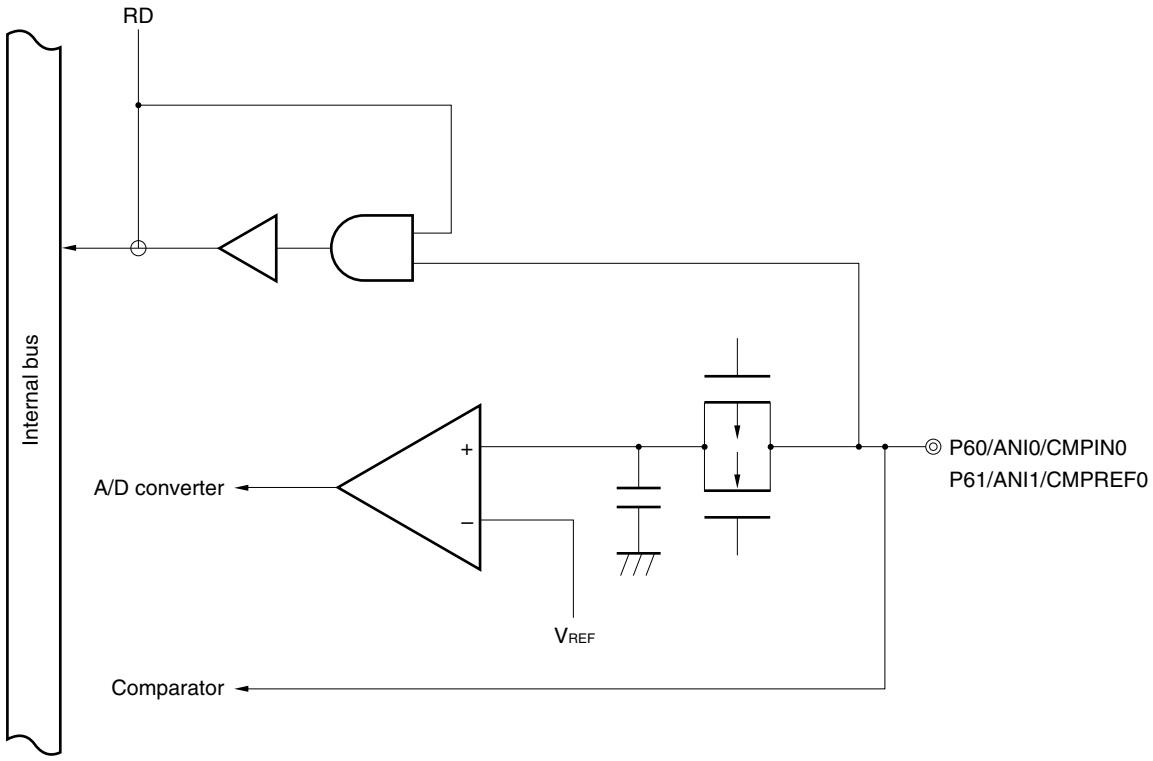
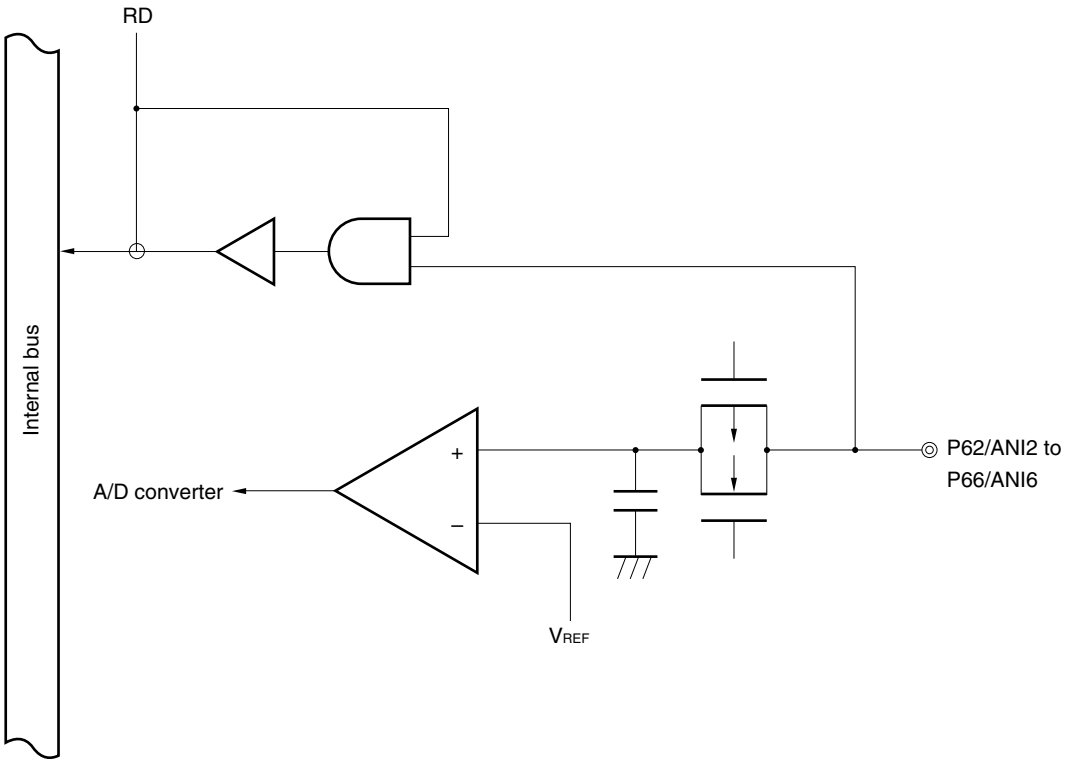


Figure 4-12. Block Diagram of P62 to P66



4.2.6 Port 8

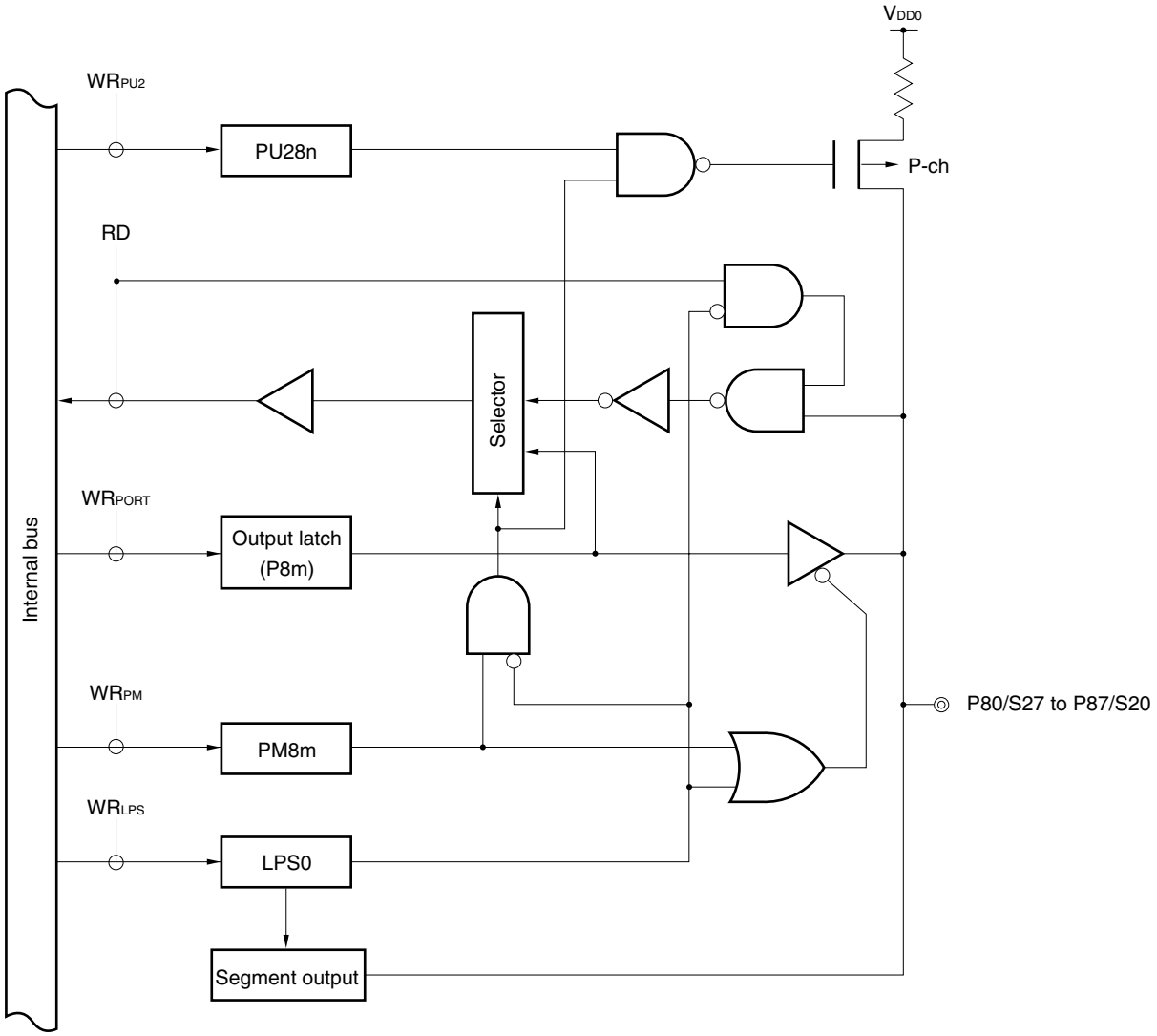
This is an 8-bit I/O port with an output latch. Port 8 can be specified as input or output in 1-bit units by using port mode register 8 (PM8). When using the P80 to P87 pins as input port pins, internal pull-up resistors can be connected in 2-bit units by using pull-up resistor option register 2 (PU2).

Port 8 is also used to output segment signals for the LCD controller/driver.

Port 8 is set to input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-13 shows a block diagram of port 8.

Figure 4-13. Block Diagram of P80 to P87



PU2: Pull-up resistor option register 2
PM: Port mode register
RD: Port 8 read signal
WR: Port 8 write signal
LPS0: LCD port selector 0
n = 0, 2, 4, 6, m = 0 to 7

4.2.7 Port 9

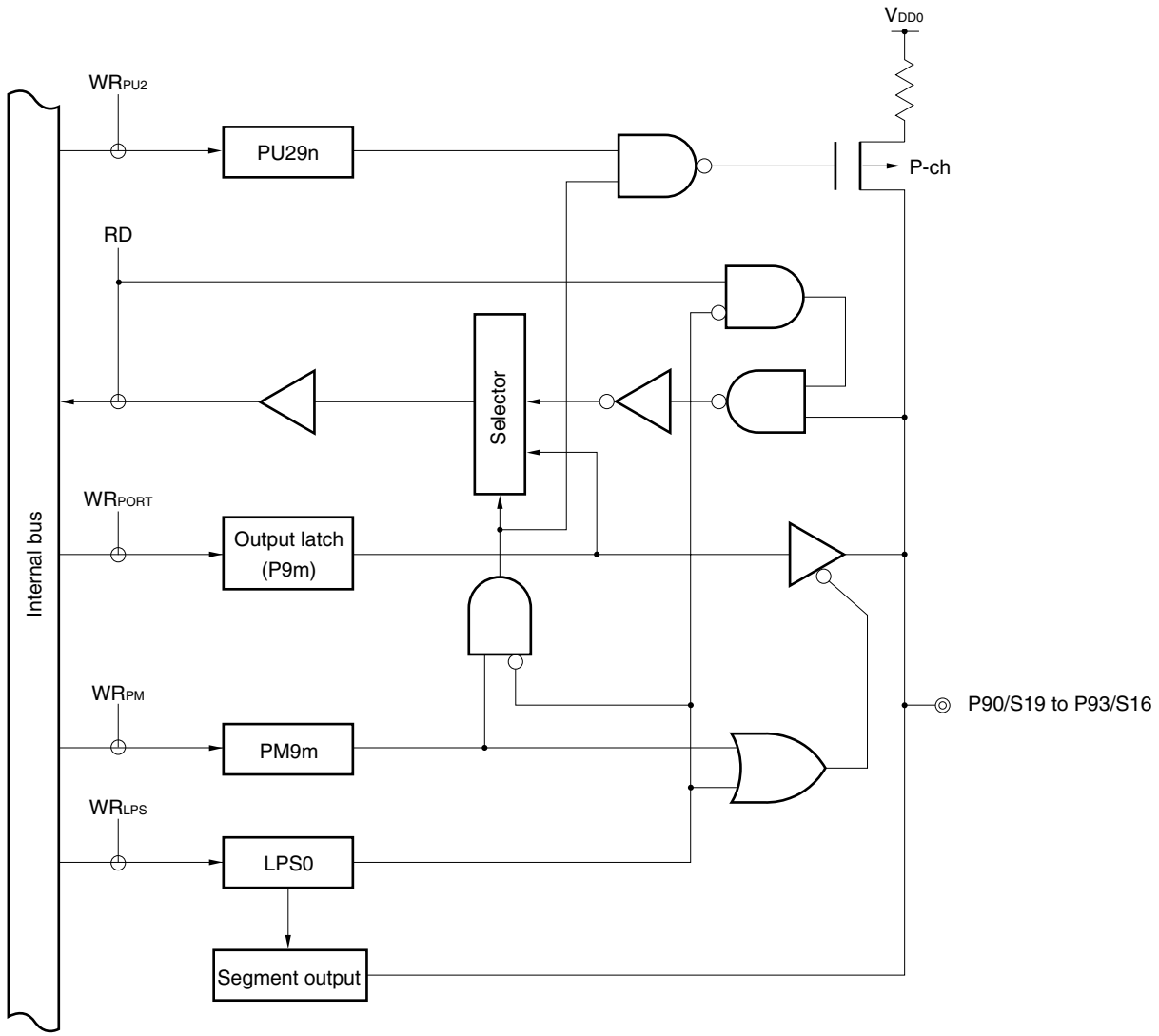
This is a 4-bit I/O port with an output latch. Port 9 can be specified as input or output in 1-bit units by using port mode register 9 (PM9). When using the P90 to P93 pins as input port pins, on-chip pull-up resistors can be connected in 2-bit units by using pull-up resistor option register 2 (PU2).

Port 9 is also used to output segment signals for the LCD controller/driver.

Port 9 is set to input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-14 shows a block diagram of port 9.

Figure 4-14. Block Diagram of P90 to P93



PU2: Pull-up resistor option register 2
PM: Port mode register
RD: Port 9 read signal
WR: Port 9 write signal
LPS0: LCD port selector 0
n = 0, 2, m = 0 to 3

4.3 Registers Controlling Ports

The following two registers control the ports.

- Port mode registers (PM0, PM2, PM4, PM5, PM8, and PM9)
- Pull-up resistor option registers (PU0 to PU2)

(1) Port mode registers (PM0, PM2, PM4, PM5, PM8, and PM9)

These registers are used to set port input/output in 1-bit units.

The port mode registers are independently set using a 1-bit or 8-bit memory manipulation instruction.

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

When port pins are used as alternate-function pins, set the port mode register and output latch according to Table 4-3.

Caution As port 2 has an alternate function as the external interrupt input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. When the output mode is used, therefore, the interrupt mask flag should be set to 1 beforehand.

Table 4-3. Port Mode Register and Output Latch Settings When Using Alternate Functions

Pin Name	Alternate Function		PMxx	Pxx
	Name	I/O		
P23	CMPTOUT0	Output	0	0
	TO2	Output	0	0
P24	INTP0	Input	1	x
	TI0	Input	1	x
P25	INTP1	Input	1	x
	TI1	Input	1	x
P26	INTP2	Input	1	x
	TO5	Output	0	0
P27	INTP3	Input	1	x
	CPT5	Input	1	x
P40 to P45[Note]	KR0 to KR5	Input	1	x
P80 to P87	S27 to S20	Output	0	0
P90 to P93	S19 to S16	Output	0	0

Note Set key return mode register 00 (KRM00) to 1 when using the alternate function (see **15.3 (6) Key return mode register 00 (KRM00)**).

Caution When port 2 is used for the serial interface, the I/O or output latch must be set according to the function used. For the setting method, see Table 13-2 Operation Mode Settings of Serial Interface 00.

Remark x: Don't care
PMxx: Port mode register
Pxx: Port output latch

Figure 4-15. 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	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM5	1	1	1	1	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM9	1	1	1	1	PM93	PM92	PM91	PM90	FF29H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 5, 9: n = 0 to 3; m = 2, 4, 8: n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Pull-up resistor option registers (PU0 to PU2)

The pull-up resistor option registers (PU0 to PU2) set whether an on-chip pull-up resistor is used on each port.

On a port specified by PU0 to PU2 to use an on-chip pull-up resistor, the pull-up resistor can be internally used only for the bits set in the input mode. No on-chip pull-up resistors can be used for the bits set in the output mode regardless of the setting of PU0 to PU2. This also applies when using the pins for alternate functions.

PU0 to PU2 are set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PU0 to PU2 to 00H.

Figure 4-16. Format of Pull-Up Resistor Option Register 0

Symbol	7	6	5	<4>	3	2	1	<0>	Address	After reset	R/W
PU0	0	0	0	PU04	0	0	0	PU00	FFF7H	00H	R/W

PU0m	Pm on-chip pull-up resistor selection[Note] (m = 0 or 4)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Note PU0 selects whether on-chip pull-up resistors are to be used in 8-bit units, except for port 0, for which on-chip pull-up resistors can be used only for four bits (P00 to P03).

Caution Bits 1, 2, 3, 5, 6, and 7 must be fixed to 0.

Figure 4-17. Format of Pull-Up Resistor Option Register 1

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PU1	PU127	PU126	PU125	PU124	PU123	PU122	PU121	PU120	FFF3H	00H	R/W

PU12m	P2 on-chip pull-up resistor selection[Note] (m = 0 to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Note PU1 selects whether on-chip pull-up resistors are to be used in 1-bit units.

Figure 4-18. Format of Pull-Up Resistor Option Register 2

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PU2	0	0	PU292	PU290	PU286	PU284	PU282	PU280	FFF4H	00H	R/W

PU2mn	Pm on-chip pull-up resistor selection[Note] (m = 8 or 9; n = 0, 2, 4, or 6)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Note PU2 selects whether on-chip pull-up resistors are to be used in 2-bit units (bit n and bit n+1).

Caution Bits 6 and 7 must be fixed to 0.

4.4 Operation of Ports

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.

Once data is written to the output latch, it 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.

Once data is written to the output latch, it 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 I/O 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.

Once data is written to the output latch, it 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 I/O 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 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following two types of system clock oscillators are used.

- **Main system clock oscillator**
  This circuit oscillates at 1.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).
- **Subsystem clock oscillator**
  This circuit oscillates at 32.768 kHz. Oscillation can be stopped by the suboscillation mode register (SCKM).

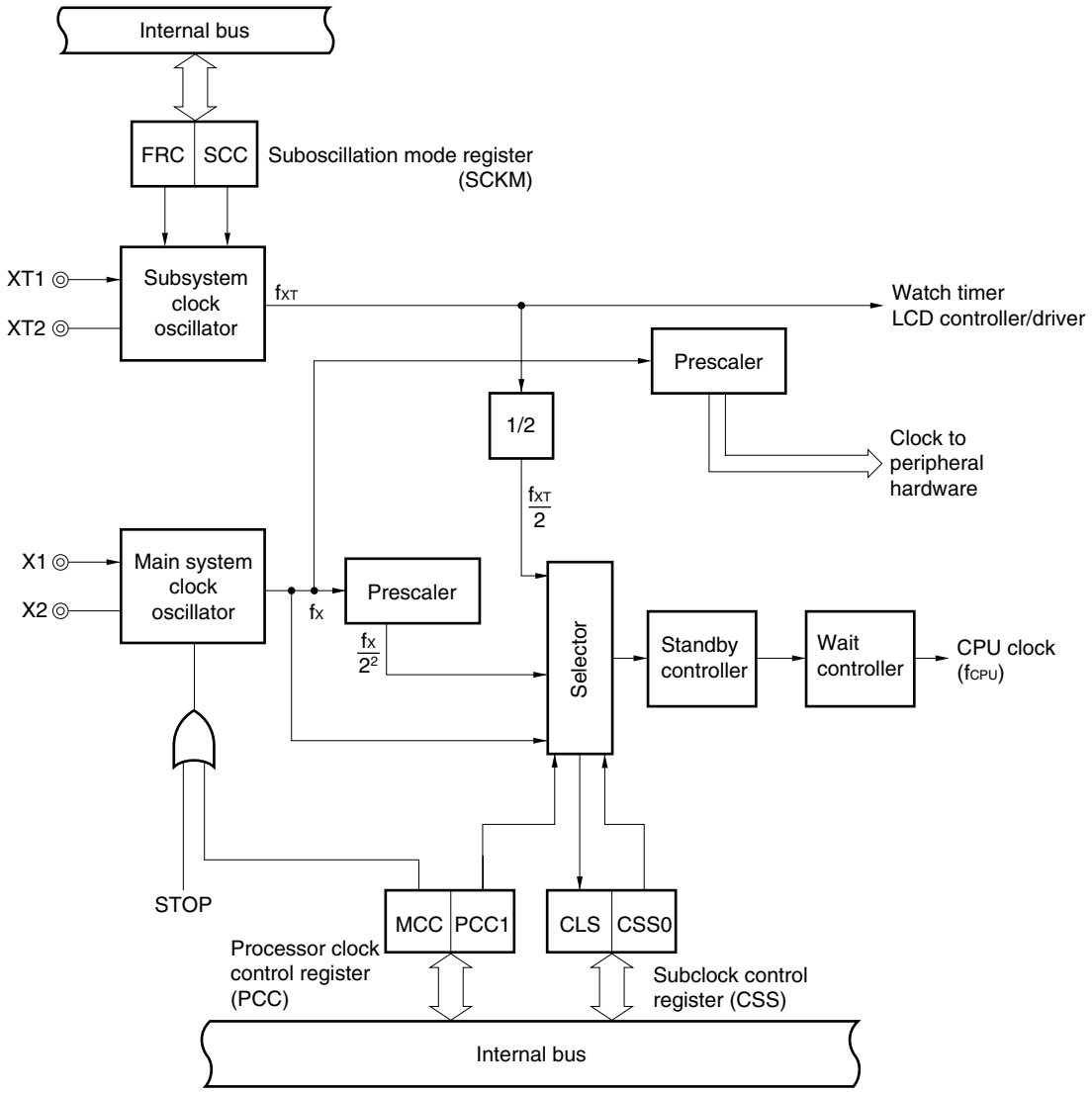
5.2 Configuration of Clock Generator

The clock generator consists of the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Processor clock control register (PCC) Suboscillation mode register (SCKM) Subclock control register (CSS)
Oscillators	Main system clock oscillator Subsystem clock oscillator

Figure 5-1. Block Diagram of Clock Generator



5.3 Registers Controlling Clock Generator

The clock generator is controlled by the following registers.

- Processor clock control register (PCC)
- Suboscillation mode register (SCKM)
- Subclock control register (CSS)

(1) Processor clock control register (PCC)

PCC selects the CPU clock and sets the division ratio.

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

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

Figure 5-2. Format of Processor Clock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PCC	MCC	0	0	0	0	0	PCC1	0	FFFBH	02H	R/W

MCC	Control of main system clock oscillator operation
0	Operation enabled
1	Operation disabled

CSS0	PCC1	Selection of CPU clock (f_{CPU})[Note]	Minimum instruction execution time: $2/f_{CPU}$
			f_X = 5.0 MHz or f_{XT} = 32.768 kHz operation
0	0	f_X	0.4 μs
0	1	$f_X/2^2$	1.6 μs
1	0	$f_{XT}/2$	122 μs
1	1		

Note The CPU clock is selected according to a combination of the PCC1 flag in the processor clock control register (PCC) and the CSS0 flag in the subclock control register (CSS). See **5.3 (3) Subclock control register (CSS)**.

Cautions 1. Bits 0 and 2 to 6 must be fixed to 0.

2. The MCC bit can be set only when the subsystem clock has been selected as the CPU clock.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

(2) Suboscillation mode register (SCKM)

SCKM selects whether a feedback resistor is used for the subsystem clock, and controls the oscillation of the clock.

SCKM is set using a 1-bit or 8-bit memory manipulation instruction.

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

Figure 5-3. Format of Suboscillation Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SCKM	0	0	0	0	0	0	FRC	SCC	FFF0H	00H	R/W

FRC	Feedback resistor selection[Note]
0	On-chip feedback resistor used
1	On-chip feedback resistor not used

SCC	Control of subsystem clock oscillator operation
0	Operation enabled
1	Operation disabled

★ **Note** The feedback resistor is necessary to adjust the bias point of the oscillation waveform to close to the mid point of the supply voltage. Only when the subclock is not used, the power consumption in STOP mode can be further reduced by setting FRC = 1.

Cautions 1. Bits 2 to 7 must be fixed to 0.

2. Do not set the SCC bit when an external clock pulse is input, because the XT2 pin is pulled up to V_{DD0} or V_{DD1}.

(3) Subclock control register (CSS)

CSS specifies whether the main system or subsystem clock oscillator is to be selected. It also specifies how the CPU clock operates.

CSS is set using a 1-bit or 8-bit memory manipulation instruction.

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

Figure 5-4. Format of Subclock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CSS	0	0	CLS	CSS0	0	0	0	0	FFF2H	00H	R/W[Note]

CLS	CPU clock operation status
0	Operation based on the output of the divided main system clock
1	Operation based on the subsystem clock

CSS0	Selection of main system or subsystem clock oscillator
0	Divided output from the main system clock oscillator
1	Output from the subsystem clock oscillator

Note Bit 5 is read only.

Caution Bits 0, 1, 2, 3, 6, and 7 must be fixed to 0.

5.4 System Clock Oscillators

5.4.1 Main system clock oscillator

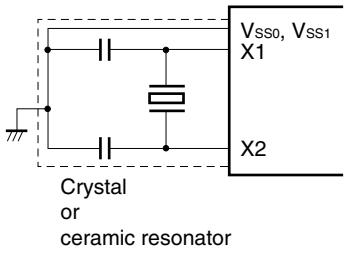
The main system clock oscillator is oscillated by a crystal or ceramic resonator (5.0 MHz TYP.) connected across the X1 and X2 pins.

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.

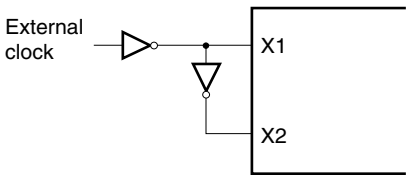
Figure 5-5 shows the external circuit of the main system clock oscillator.

Figure 5-5. External Circuit of Main System Clock Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Caution When using the main system or subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-5 and 5-6 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_{SS0} and V_{SS1}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

5.4.2 Subsystem clock oscillator

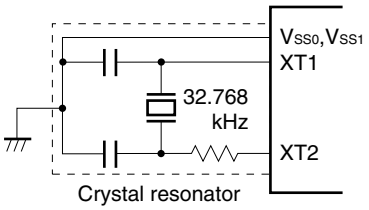
The subsystem clock oscillator is oscillated by a crystal resonator (32.768 kHz TYP.) connected across the XT1 and XT2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the XT1 pin, and input the inverted signal to the XT2 pin.

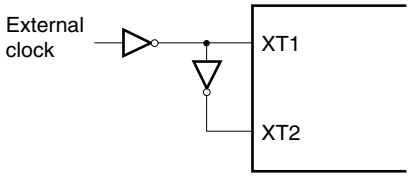
Figure 5-6 shows the external circuit of the subsystem clock oscillator.

Figure 5-6. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



Caution When using the main system or subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-5 and 5-6 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_{SS0} and V_{SS1}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

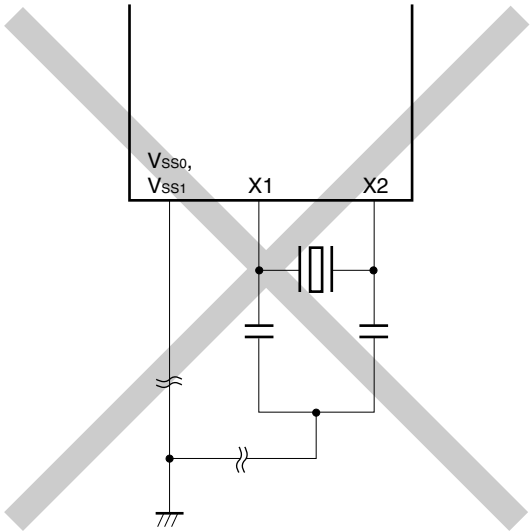
When using the subsystem clock oscillator, pay special attention because the subsystem clock oscillator has low amplification to minimize current consumption.

5.4.3 Examples of incorrect resonator connection

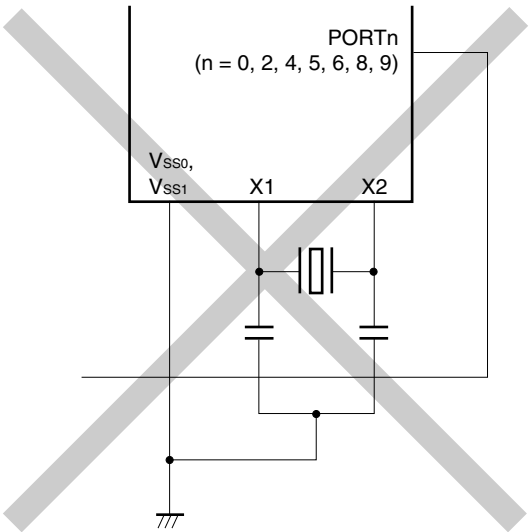
Figure 5-7 shows examples of incorrect resonator connection.

Figure 5-7. Examples of Incorrect Resonator Connection (1/2)

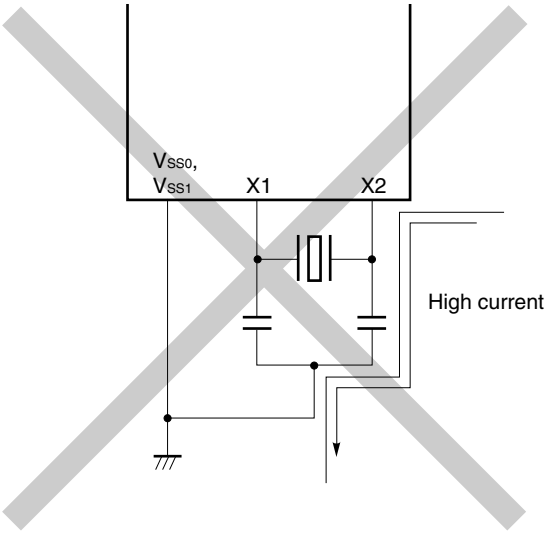
(a) Too long wiring



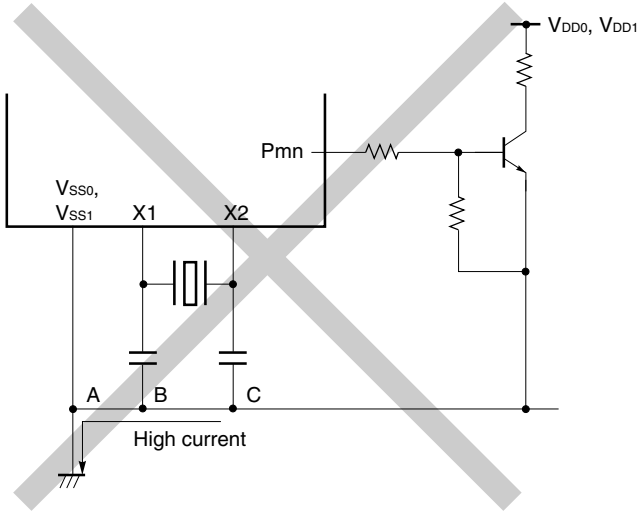
(b) Crossed signal line



(c) Wiring near high fluctuating current



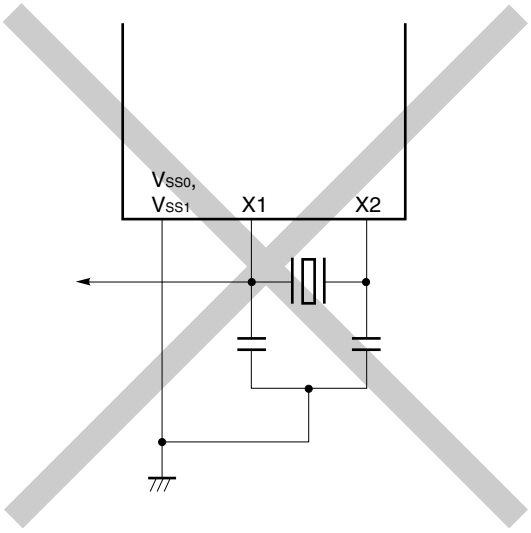
(d) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



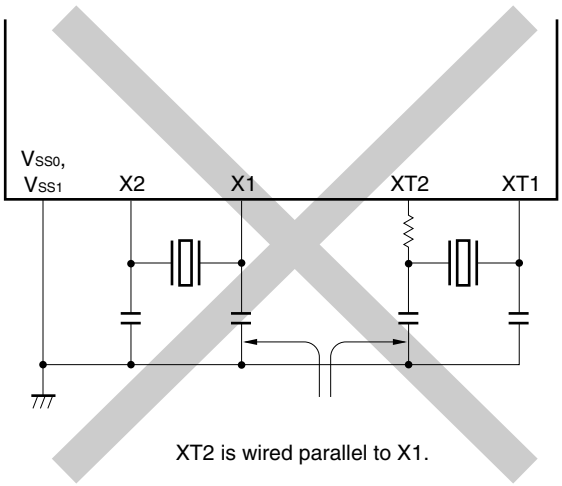
Remark When using the subsystem clock, read X1 and X2 as XT1 and XT2, respectively, and connect resistors to the XT2 side in series.

Figure 5-7. Examples of Incorrect Resonator Connection (2/2)

(e) Signal is fetched



(f) Signal lines of main system clock and subsystem clock are parallel and close together



Remark When using the subsystem clock, read X1 and X2 as XT1 and XT2, respectively, and connect resistors to the XT2 side in series.

Caution If the X1 wire is parallel with the XT2 wire, crosstalk noise may occur between X1 and XT2, resulting in a malfunction.
To avoid this, do not place the X1 and XT2 wires in parallel.

5.4.4 Divider

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

5.4.5 When no subsystem clock is used

If a subsystem clock is not necessary, for example, for low-power consumption operation or clock operation, handle the XT1 and XT2 pins as follows:

XT1: Connect directly to V_{SS0} or V_{SS1}
XT2: Leave open

In this case, however, a small current leaks via the on-chip feedback resistor in the subsystem clock oscillator when the main system clock is stopped. To avoid this, set bit 1 (FRC) of the suboscillation mode register (SCKM) so that the on-chip feedback resistor will not be used. Also in this case, handle the XT1 and XT2 pins as stated above.

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.

- Main system clock f_X
- Subsystem clock f_{XT}
- CPU clock f_{CPU}
- Clock to peripheral hardware

The operation of the clock generator is determined by the processor clock control register (PCC), suboscillation mode register (SCKM), and subclock control register (CSS), as follows.

(a) The slow mode (1.6 μs at 5.0 MHz operation) of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is being input to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Three types of minimum instruction execution time (0.4 μs and 1.6 μs main system clock (at 5.0 MHz operation), 122 μs subsystem clock (at 32.768 kHz operation)) can be selected by the PCC, SCKM, and CSS settings.

(c) Two standby modes, STOP and HALT, can be used with the main system clock selected. In a system where no subsystem clock is used, setting bit 1 (FRC) of SCKM so that the on-chip feedback resistor cannot be used reduces current consumption in the STOP mode. In a system where a subsystem clock is used, setting bit 0 of SCKM to 1 can cause the subsystem clock to stop oscillation.

(d) Bit 4 (CSS0) of CSS can be used to select the subsystem clock so that low current consumption operation is used (at 122 μs, 32.768 kHz operation).

(e) With the subsystem clock selected, it is possible to cause the main system clock to stop oscillating by setting bit 7 (MCC) of PCC. The HALT mode can be used, but the STOP mode cannot.

(f) The clock pulse for the peripheral hardware is generated by dividing the frequency of the main system clock. The subsystem clock pulse is supplied to 8-bit timer 02, the watch timer, and the LCD controller/driver only. As a result, 8-bit timer 02 (when watch timer output is selected for the count clock when the subsystem clock is running) and the watch function can continue running even in the standby mode. The other hardware stops when the main system clock stops, because it runs based on the main system clock (except for external input clock pulses).

5.6 Changing Setting of System Clock and CPU Clock

5.6.1 Time required for switching between system clock and CPU clock

The CPU clock can be selected by using bit 1 (PCC1) of the processor clock control register (PCC) and bit 4 (CSS0) of the subclock control register (CSS).

Actually, the specified clock is not selected immediately after the setting of PCC has been changed; the old clock is used for the duration of several instructions after that (see **Table 5-2**).

Table 5-2. Maximum Time Required for Switching CPU Clock

Set Value Before Switching		Set Value After Switching					
CSS0	PCC1	CSS0	PCC1	CSS0	PCC1	CSS0	PCC1
		0	0	0	1	1	x
0	0			4 clocks		$2f_X/f_{XT}$ clocks (306 clocks)	
	1	2 clocks				$f_X/2f_{XT}$ clocks (76 clocks)	
1	x	2 clocks		2 clocks			

Remarks 1. Two clocks is the minimum instruction execution time of the CPU clock before switching.

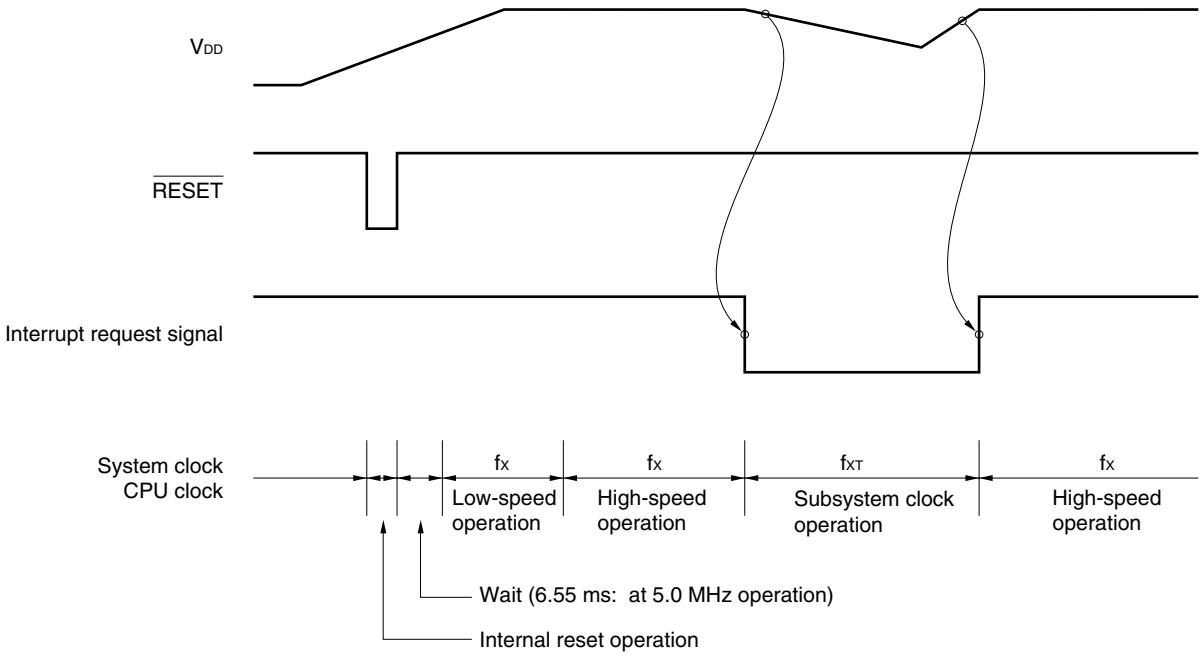
2. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

3. x: Don't care

5.6.2 Switching between system clock and CPU clock

The following figure illustrates how the CPU clock and system clock are switched.

Figure 5-8. Switching Between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. Reset is released when the $\overline{\text{RESET}}$ pin is later made high, and the main system clock starts oscillating. At this time, the oscillation stabilization time (2^{15}/fx) is automatically secured.
After that, the CPU starts instruction execution at the low speed of the main system clock (1.6 μs at 5.0 MHz operation).

<2> After the time required for the V_{DD} voltage to rise to the level at which the CPU can operate at the high speed has elapsed, bit 1 (PCC1) of the processor clock control register (PCC) and bit 4 (CSS0) of the subclock control register (CSS) are rewritten so that the high-speed operation can be selected.

<3> A drop of the V_{DD} voltage is detected by an interrupt request signal. The clock is switched to the subsystem clock (at this moment, the subsystem clock must be in the stable oscillation status).

<4> Recovery of the V_{DD} voltage is detected by an interrupt request signal. Bit 7 (MCC) of PCC is set to 0, and the main system clock starts oscillating. After the time required for the oscillation to stabilize has elapsed, PCC1 and CSS0 are rewritten so that high-speed operation can be selected again.

Caution When the main system clock is stopped and the device is operating on the subsystem clock, wait until the oscillation stabilization time has been secured by the program before switching back to the main system clock.

CHAPTER 6 16-BIT TIMER 50

16-bit timer 50 references the free-running counter and provides functions such as timer interrupt and timer output. In addition, the count value can be captured by a trigger pin.

6.1 Function of 16-Bit Timer 50

16-bit timer 50 has the following functions.

- Timer interrupt
- Timer output
- Count value capture

(1) Timer interrupt

An interrupt is generated when the count value and compare value match.

(2) Timer output

Timer output control is possible when the count value and compare value match.

(3) Count value capture

The count value of 16-bit timer counter 50 (TM50) is latched to the capture register in synchronization with the capture trigger and retained.

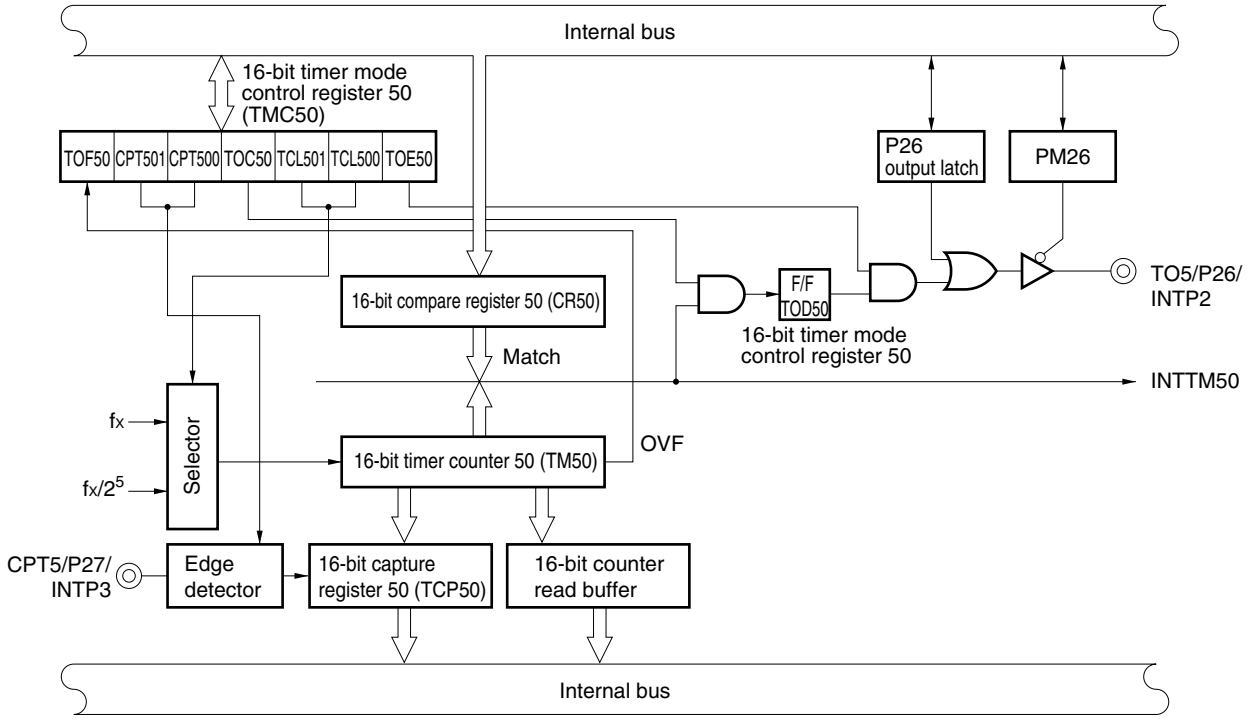
6.2 Configuration of 16-Bit Timer 50

16-bit timer 50 consists of the following hardware.

Table 6-1. Configuration of 16-Bit Timer 50

Item	Configuration
Timer counter	16 bits × 1 (TM50)
Registers	Compare register: 16 bits × 1 (CR50) Capture register: 16 bits × 1 (TCP50)
Timer outputs	1 (TO5)
Control registers	16-bit timer mode control register 50 (TMC50) Port mode register 2 (PM2)

Figure 6-1. Block Diagram of 16-Bit Timer 50



(1) 16-bit compare register 50 (CR50)

This register compares the value set to CR50 with the count value of 16-bit timer counter 50 (TM50), and when they match, generates an interrupt request (INTTM50).

CR50 is set using a 16-bit memory manipulation instruction. Values from 0000H to FFFFH can be set.

$\overline{\text{RESET}}$ input sets CR50 to FFFFH.

Cautions 1. Although this register is manipulated by a 16-bit memory manipulation instruction, an 8-bit memory manipulation instruction can also be used. When manipulated by an 8-bit memory manipulation instruction, the accessing method should be direct addressing.

2. When rewriting CR50 during a count operation, preset CR50 to interrupt disabled using interrupt mask flag register 1 (MK1). Also, set the timer output data to inversion disabled using 16-bit timer mode control register 50 (TMC50).

If CR50 is rewritten while interrupts are enabled, an interrupt request may be generated at the time of the rewrite.

(2) 16-bit timer counter 50 (TM50)

This is a 16-bit register that counts count pulses.

TM50 is read using a 16-bit memory manipulation instruction.

TM50 is in free-running mode during count clock input.

$\overline{\text{RESET}}$ input sets TM50 to 0000H, after which it enters free-running mode again.

Cautions 1. The count value after releasing stop becomes undefined because the count operation is executed during the oscillation stabilization time.

2. Although this register is manipulated by a 16-bit memory manipulation instruction, an 8-bit memory manipulation instruction can also be used. When manipulated by an 8-bit memory manipulation instruction, the accessing method should be direct addressing.

3. When manipulated by an 8-bit memory manipulation instruction, readout should be performed in order from lower byte to higher byte and must be in pairs.

(3) 16-bit capture register 50 (TCP50)

This is a 16-bit register that captures the contents of 16-bit timer counter 50 (TM50).

TCP50 is set using a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes TCP50 undefined.

Caution Although this register is manipulated by a 16-bit memory manipulation instruction, an 8-bit memory manipulation instruction can also be used. When manipulated by an 8-bit memory manipulation instruction, the accessing method should be direct addressing.

(4) 16-bit counter read buffer

This buffer latches the counter value of 16-bit timer counter 50 (TM50) and retains the count value.

6.3 Registers Controlling 16-Bit Timer 50

The following two registers are used to control 16-bit timer 50.

- 16-bit timer mode control register 50 (TMC50)
- Port mode register 2 (PM2)

(1) 16-bit timer mode control register 50 (TMC50)

16-bit timer mode control register 50 (TMC50) controls the setting of the count clock, capture edge, etc.

TMC50 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC50 to 00H.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 50

Symbol	7	<6>	5	4	3	2	1	<0>	Address	After reset	R/W
TMC50	TOD50	TOF50	CPT501	CPT500	TOC50	TCL501	TCL500	TOE50	FF48H	00H	R/W Note 1

TOD50	Timer output data
0	Timer output is “0”
1	Timer output is “1”

TOF50	Overflow flag set
0	Clear by reset and software
1	Set by overflow of 16-bit timer

CPT501	CPT500	Capture edge selection
0	0	Capture operation disabled
0	1	Rising edge of CPT5
1	0	Falling edge of CPT5
1	1	Both edges of CPT5

TOC50	Timer output data inverse control
0	Inverse disabled
1	Inverse enabled

TCL501	TCL500	16-bit timer 50 count clock selection
0	0	f_X (5.0 MHz) Note 2
0	1	$f_X/2^5$ (156.3 kHz) Note 3
Other than above		Setting prohibited

TOE50	16-bit timer 50 output control
0	Output disabled (port mode)
1	Output enabled

Notes 1. Bit 7 is read-only.

2. If the count clock is set to f_X (TCL501 = 0, TCL500 = 0), the capture function cannot be used. When reading, set the CPU clock to the main system clock high-speed mode (PCC1 = 0, CSS0 = 0) (see **Figure 5-2**).

3. When reading, specify the main system clock as the CPU clock (PCC1 = 0, CSS0 = 0 or PCC1 = 1, CSS0 = 0) (see **Figure 5-2**).

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Port mode register 2 (PM2)

This register sets input/output of port 2 in 1-bit units.

To use the P26/INTP2/TO5 pin for timer output, set PM26 and the output latch of P26 to 0.

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

$\overline{\text{RESET}}$ input sets PM2 to FFH.

Figure 6-3. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM26	P26 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Operation of 16-Bit Timer 50

6.4.1 Operation as timer interrupt

In the timer interrupt function, interrupts are repeatedly generated at the count value set to 16-bit compare register 50 (CR50) in advance at the interval set in TCL501 and TCL500.

To operate the 16-bit timer as a timer interrupt, the following settings are required.

- Set the count value to CR50
- Set 16-bit timer mode control register 50 (TMC50) as shown in Figure 6-4.

Figure 6-4. Settings of 16-Bit Timer Mode Control Register 50 for Timer Interrupt Operation

	TOD50	TOF50	CPT501	CPT500	TOC50	TCL501	TCL500	TOE50
TMC50	–	0/1	0/1	0/1	0/1	0	0/1	0/1

Setting of count clock (see **Table 6-2**)

Caution If both the CPT501 flag and CPT500 flag are set to 0, the capture edge becomes operation prohibited.

When the count value of 16-bit timer counter 50 (TM50) matches the value set to CR50, counting of TM50 continues and an interrupt request signal (INTTM50) is generated.

Table 6-2 shows the interval time, and Figure 6-5 shows the timing of the timer interrupt operation.

Caution Be sure to process as follows when rewriting CR50 during a count operation.

<1> Set interrupts to disabled (TMMK50 (bit 4 of interrupt mask flag register 1 (MK1)) = 1)

<2> Set the inversion control of timer output data to disabled (TOC50 = 0)

If CR50 is rewritten while interrupts are enabled, an interrupt request may be generated at the time of rewrite.

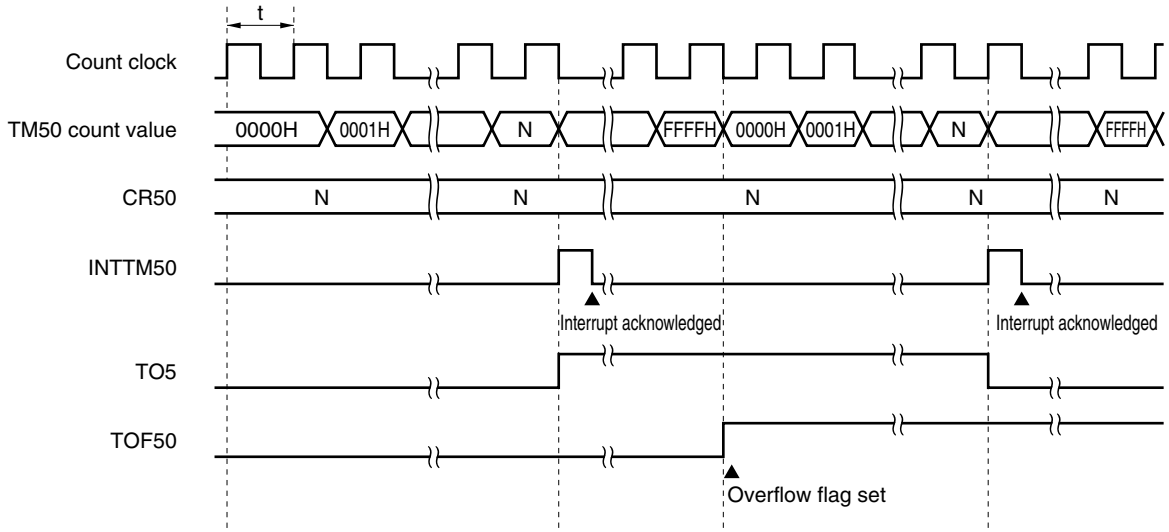
Table 6-2. Interval Time of 16-Bit Timer 50

TCL501	TCL500	Count Clock	Interval Time
0	0	$1/f_X$ (0.2 μs)	$2^{16}/f_X$ (13.1 ms)
0	1	$2^5/f_X$ (6.4 μs)	$2^{21}/f_X$ (419.4 ms)
Other than above		Setting prohibited	

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 6-5. Timing of Timer Interrupt Operation



Remark N = 0000H to FFFFH

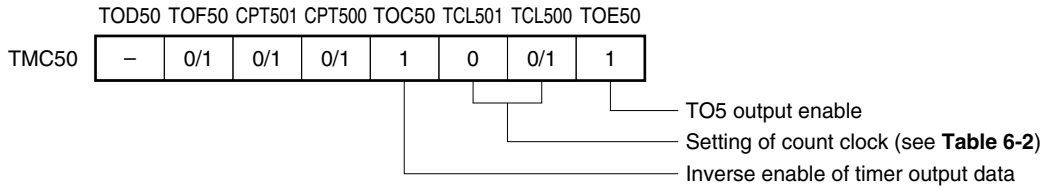
6.4.2 Operation as timer output

Timer outputs are repeatedly generated at the count value set to 16-bit compare register 50 (CR50) in advance at the interval set in TCL501 and TCL500.

To operate 16-bit timer as a timer output, the following settings are required.

- Set P26 to output mode (PM26 = 0)
- Set the output latch of P26 to 0
- Set the count value to CR50
- Set 16-bit timer mode control register 50 (TMC50) as shown in Figure 6-6

Figure 6-6. Settings of 16-Bit Timer Mode Control Register 50 for Timer Output Operation

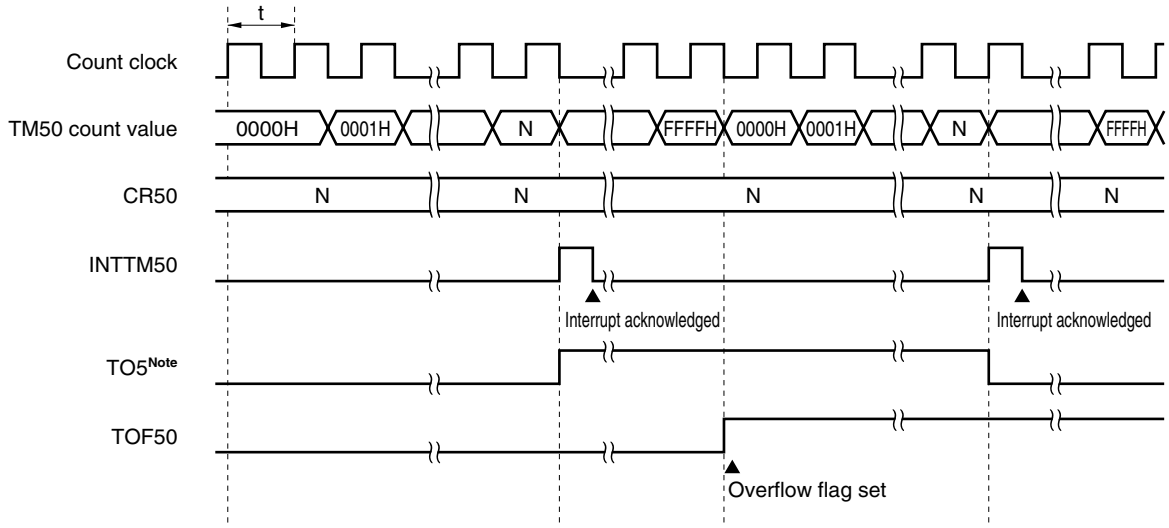


Caution If both the CPT501 flag and CPT500 flag are set to 0, the capture edge becomes operation prohibited.

When the count value of 16-bit timer counter 50 (TM50) matches the value set in CR50, the output status of the TO5/INTP2/P26 pin is inverted. This enables timer output. At that time, TM50 counting continues and an interrupt request signal (INTTM50) is generated.

Figure 6-7 shows the timing of timer output (see **Table 6-2** for the interval time of 16-bit timer 50).

Figure 6-7. Timer Output Timing



Note The TO5 initial value becomes low level when output is enabled (TOE50 = 1).

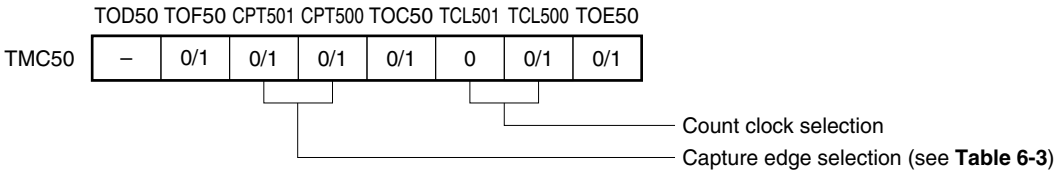
Remark N = 0000H to FFFFH

6.4.3 Capture operation

In a capture operation, the count value of 16-bit timer counter 50 (TM50) is captured and latched to the capture register in synchronization with a capture trigger.

Set as shown in Figure 6-8 to allow the 16-bit timer to start a capture operation.

Figure 6-8. Settings of 16-Bit Timer Mode Control Register 50 for Capture Operation



16-bit capture register 50 (TCP50) starts a capture operation after the CPT5 capture trigger edge is defected, and latches and retains the count value of 16-bit timer counter 50 (TM50). TCP50 fetches the count value within 2 clocks and retains the count value until the next capture edge detection.

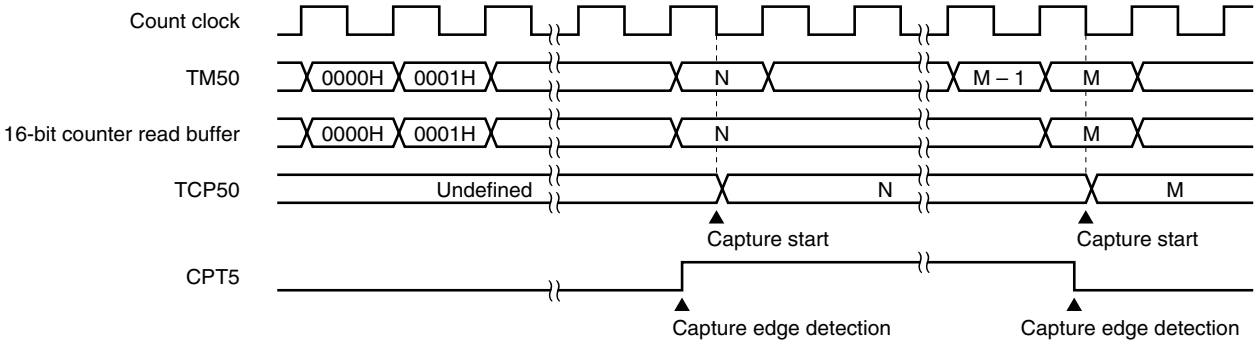
Table 6-3 and Figure 6-9 shows the settings of the capture edge and the capture operation timing, respectively.

Table 6-3. Settings of Capture Edge

CPT501	CPT500	Capture Edge Selection
0	0	Capture operation prohibited
0	1	CPT5 pin rising edge
1	0	CPT5 pin falling edge
1	1	CPT5 pin both edges

Caution Because TCP50 is rewritten when a capture trigger edge is detected during TCP50 read, disable capture trigger edge detection during TCP50 read.

Figure 6-9. Capture Operation Timing (Both Edges of CPT5 Pin Are Specified)



6.4.4 16-bit timer counter 50 readout

The count value of 16-bit timer counter 50 (TM50) is read out by a 16-bit manipulation instruction.

TM50 readout is performed via a 16-bit counter read buffer. The 16-bit counter read buffer latches the TM50 count value, the buffer operation is held pending at the CPU clock falling edge after the read signal of the TM50 lower byte rises, and the count value is retained. The 16-bit counter read buffer value in the retention state can be read out as the count value.

Cancellation of pending is performed at the CPU clock falling edge after the read signal of the TM50 higher byte falls.

$\overline{\text{RESET}}$ input sets TM50 to 0000H and then to free-running mode again.

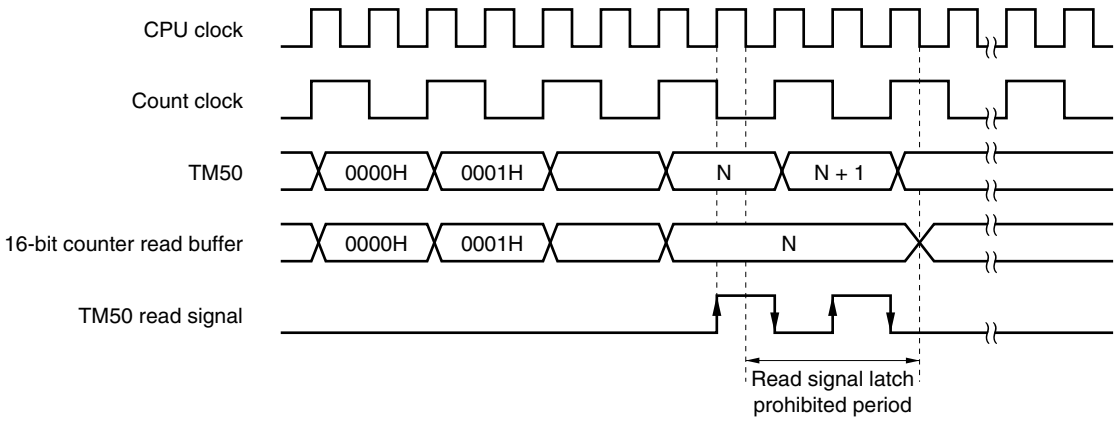
Figure 6-10 shows the timing of 16-bit timer counter 50 readout.

Cautions 1. The count value after releasing stop becomes undefined because the count operation is executed during the oscillation stabilization time.

2. Although TM50 is manipulated by a 16-bit transfer instruction, 8-bit transfer instruction can also be used.
When using an 8-bit transfer instruction, execute by direct addressing.

3. When using an 8-bit transfer instruction, execute in order from lower byte to higher byte in pairs. If the only lower byte is read, the pending state of the 16-bit counter read buffer is not canceled, and if the only higher byte is read, an undefined count value is read.

Figure 6-10. Readout Timing of 16-Bit Timer Counter 50



★ 6.5 Cautions on Using 16-Bit Timer 50

6.5.1 Restrictions when rewriting 16-bit compare register 50

(1) Disable interrupts (TMMK50 = 1) and the inversion control of timer output (TOC50 = 0) before rewriting the compare register (CR50).
If CR50 is rewritten with interrupts enabled, an interrupt request may be generated immediately.

(2) Depending on the timing of rewriting the compare register (CR50), the interval time may become twice as long as the intended time. Similarly, a shorter waveform or twice-longer waveform than the intended timer output waveform may be output.
To avoid this problem, rewrite the compare register using either of the following procedures.

<Countermeasure A> When rewriting using 8-bit access

<1> Disable interrupts (TMMK50 = 1) and the inversion control of timer output (TOC50 = 0).
<2> First rewrite the higher 1 byte of CR50 (16 bits).
<3> Then rewrite the lower 1 byte of CR50 (16 bits).
<4> Clear the interrupt request flag (TMIF50).
<5> Enable timer interrupts/timer output inversion after half a cycle or more of the count clock has elapsed from the beginning of the interrupt.

<Program example A> (count clock = 32/fx, CPU clock = fx)

```
TM50_VCT: SET1 TMMK50      ;  Disable timer interrupts (6 clocks)
          CLR1 TMC50.3     ;  Disable timer output inversion (6 clocks)
          MOV  A,#xxH      ;  Set the rewrite value of higher byte (6 clocks)
          MOV  !0FF17H,A   ;  Rewrite CR50 higher byte (8 clocks)
          MOV  A,#yyH      ;  Set the rewrite value of lower byte (6 clocks)
          MOV  !0FF16H,A   ;  Rewrite CR50 lower byte (8 clocks)
          CLR1 TMIF50      ;  Clear interrupt request flag (6 clocks)
          CLR1 TMMK50      ;  Enable timer interrupts (6 clocks)
          SET1 TMC50.3     ;  Enable timer output inversion
```

Total: 16 clocks or more[Note]

Note Because the INTTM50 signal becomes high level for half a cycle of the count clock after an interrupt is generated, the output is inverted if TOC50 is set to 1 during this period.

<Countermeasure B> When rewriting using 16-bit access

<1> Disable interrupts (TMMK50 = 1) and the inversion control of timer output (TOC50 = 0).

<2> Rewrite CR50 (16 bits).

<3> Wait for one cycle or more of the count clock.

<4> Clear the interrupt request flag (TMIF50).

<5> Enable timer interrupts/timer output inversion.

<Program example B> (count clock = 32/fx, CPU clock = fx)

```
TM50_VCT   SET1 TMMK50      ;  Disable timer interrupts
           CLR1 TMC50.3     ;  Disable timer output inversion
           MOVW AX,#xxyyH   ;  Set the rewrite value of CR50
           MOVW CR50,AX     ;  Rewrite CR50
           NOP              }
           NOP              }
            :               }  ;  16 NOP instructions (wait for 32/fx)Note
           NOP              }
           NOP              }
           CLR1 TMIF50      ;  Clear interrupt request flag
           CLR1 TMMK50      ;  Enable timer interrupts
           SET1 TMC50.3     ;  Enable timer output inversion
```

Note Clear the interrupt request flag (TMIF50) after waiting for one cycle or more of the count clock from the instruction rewriting CR50 (MOVW CR50, AX).

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 00 TO 02

7.1 Function of 8-Bit Timer/Event Counters 00 to 02

8-bit timer/event counters 00 to 02 have the following functions.

- Interval timer (timer 00, timer 01, and timer 02)
- External event counter (timer 00 and timer 01 only)
- Square-wave output (timer 02 only)

The μPD789407A and μPD789417A Subseries are provided with two 8-bit timer/event counter channels (timer 00 and timer 01) and one 8-bit timer channel (timer 02). When reading the description of timer 02, timer/event counter should be read as a timer.

(1) 8-bit interval timer

When the 8-bit timer/event counter is used as an interval timer, it generates an interrupt at an arbitrary time interval set in advance.

Table 7-1. Interval Time of 8-Bit Timer/Event Counter 00

Minimum Interval Time	Maximum Interval Time	Resolution
2^6/fx (12.8 μs)	2^{14}/fx (3.28 ms)	2^6/fx (12.8 μs)
2^9/fx (102.4 μs)	2^{17}/fx (26.2 ms)	2^9/fx (102.4 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 7-2. Interval Time of 8-Bit Timer/Event Counter 01

Minimum Interval Time	Maximum Interval Time	Resolution
2^4/fx (3.2 μs)	2^{12}/fx (819.2 μs)	2^4/fx (3.2 μs)
2^8/fx (51.2 μs)	2^{16}/fx (13.1 ms)	2^8/fx (51.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 7-3. Interval Time of 8-Bit Timer 02

Minimum Interval Time	Maximum Interval Time	Resolution
2^3/fx (1.6 μs)	2^{11}/fx (409.6 μs)	2^3/fx (1.6 μs)
2^7/fx (25.6 μs)	2^{15}/fx (6.55 ms)	2^7/fx (25.6 μs)
1/fXT (30.5 μs)	2^8/fXT (7.81 ms)	1/fXT (30.5 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. fXT: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at fx = 5.0 MHz or fXT = 32.768 kHz.

(2) External event counter

The number of pulses of an externally input signal can be measured.

(3) Square-wave output

A square wave of any frequency can be output.

Table 7-4. Square-Wave Output Range of 8-Bit Timer 02

Minimum Pulse Width	Maximum Pulse Width	Resolution
$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)
$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)
$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

7.2 Configuration of 8-Bit Timer/Event Counters 00 to 02

8-bit timer/event counters 00 to 02 consist of the following hardware.

Table 7-5. Configuration of 8-Bit Timer/Event Counters 00 to 02

Item	Configuration
Timer counter	8 bits × 3 (TM00, TM01, and TM02)
Register	Compare register: 8 bits × 3 (CR00, CR01, and CR02)
Timer output	1 (TO2)
Control registers	8-bit timer mode control registers 00, 01, and 02 (TMC00, TMC01, and TMC02) Port mode register 2 (PM2)

Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 00

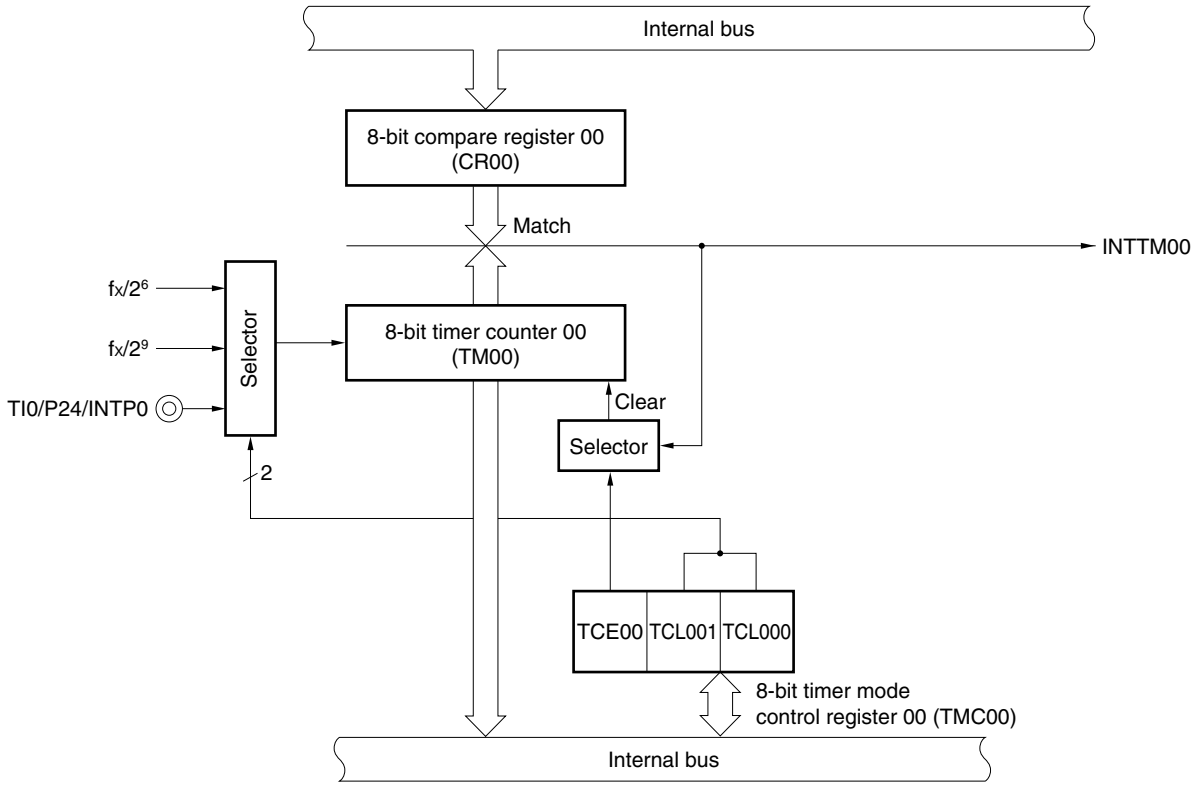


Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 01

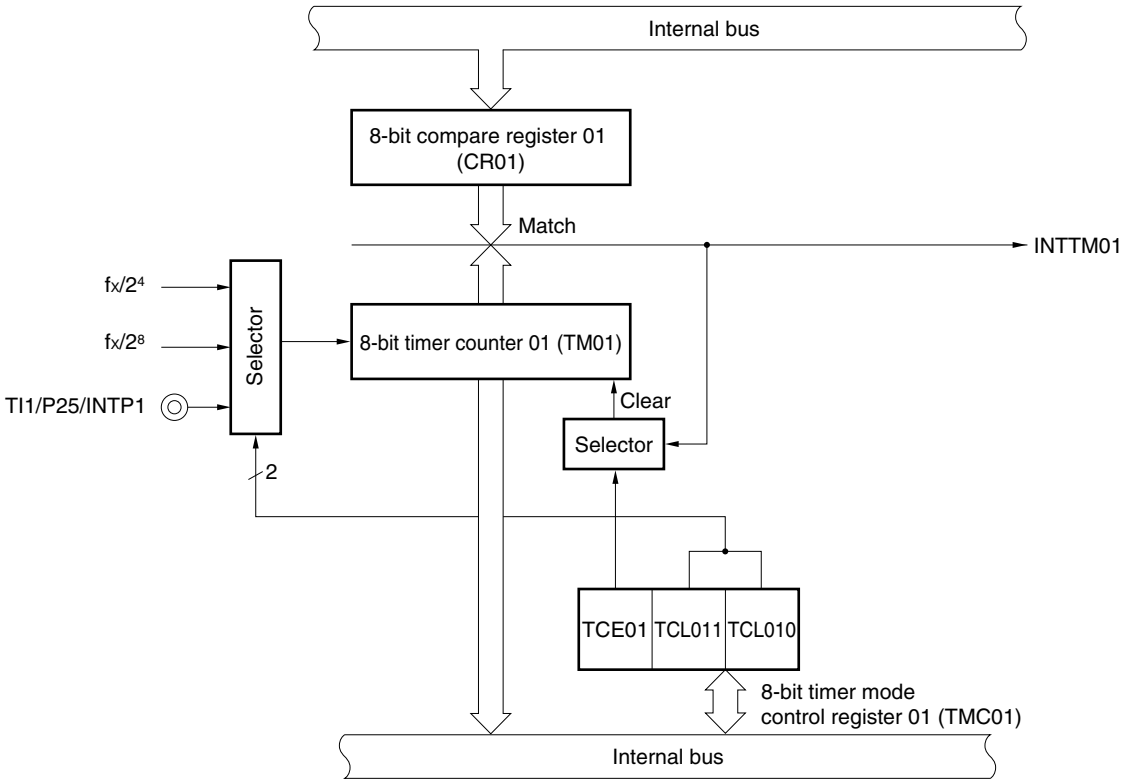
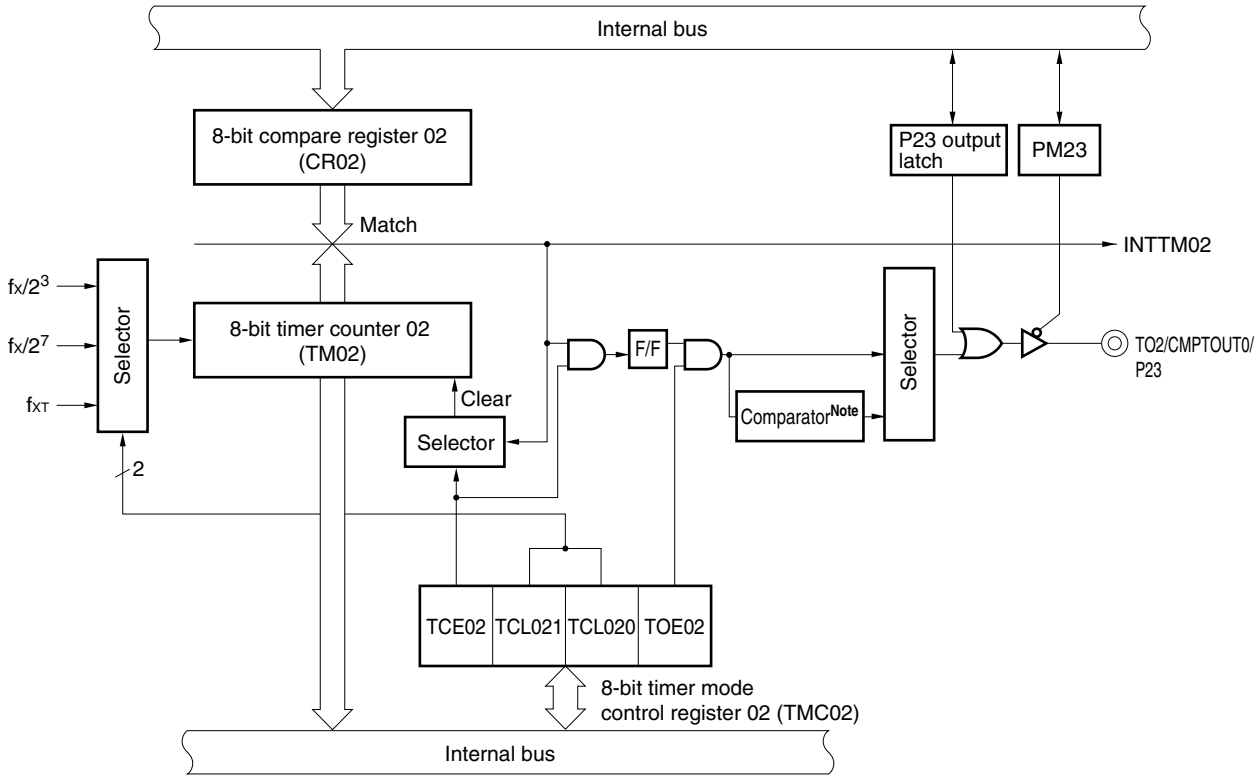


Figure 7-3. Block Diagram of 8-Bit Timer 02



Note See **CHAPTER 12 COMPARATOR** for details of the comparator.

(1) 8-bit compare register 0n (CR0n)

This is an 8-bit register that compares the value set to CR0n with the 8-bit timer counter 0n (TM0n) count value, and if they match, an interrupt request (INTTM0n) is generated.

CR0n is set using an 8-bit memory manipulation instruction. Values from 00H to FFH can be set.

$\overline{\text{RESET}}$ input makes CR0n undefined.

Caution Be sure to stop the operation of the timer before rewriting CR0n. If CR0n is rewritten while the timer is operation-enabled, an interrupt request match signal may be generated at the time of the rewrite.

Remark n = 0 to 2

(2) 8-bit timer counter 0n (TM0n)

This is an 8-bit register that counts pulses.

TM0n is read using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TM0n to 00H.

Remark n = 0 to 2

7.3 Registers Controlling 8-Bit Timer/Event Counters 00 to 02

The following two registers are used to control 8-bit timer/event counters 00 to 02.

- 8-bit timer mode control registers 00, 01, and 02 (TMC00, TMC01, and TMC02)
- Port mode register 2 (PM2)

(1) 8-bit timer mode control register 00 (TMC00)

TMC00 enables/stops operation of 8-bit timer counter 00 (TM00) and sets the count clock of TM00.

TMC00 is set using a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-4. Format of 8-Bit Timer Mode Control Register 00

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC00	TCE00	0	0	0	0	TCL001	TCL000	0	FF53H	00H	R/W

TCE00	Operation control of 8-bit timer counter 00
0	Operation stopped (TM00 is cleared to 00H)
1	Operation enabled

TCL001	TCL000	Count clock selection of 8-bit timer/event counter 00
0	0	$f_X/2^6$ (78.1 kHz)
0	1	$f_X/2^9$ (9.76 kHz)
1	0	Rising edge of TI0
1	1	Falling edge of TI0

Caution Be sure to stop the operation of the timer before setting TMC00.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) 8-bit timer mode control register 01 (TMC01)

TMC01 determines whether to enable or stop operation of 8-bit timer counter 01 (TM01) and specifies the count clock for 8-bit timer/event counter 01.

TMC01 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC01 to 00H.

Figure 7-5. Format of 8-Bit Timer Mode Control Register 01

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC01	TCE01	0	0	0	0	TCL011	TCL010	0	FF57H	00H	R/W

TCE01	Operation control of 8-bit timer counter 01
0	Operation stopped (TM01 is cleared to 00H)
1	Operation enabled

TCL011	TCL010	Count clock selection of 8-bit timer/event counter 01
0	0	$f_X/2^4$ (312.5 kHz)
0	1	$f_X/2^8$ (19.5 kHz)
1	0	Rising edge of TI1
1	1	Falling edge of TI1

Caution Be sure to stop the operation of the timer before setting TMC01.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(3) 8-bit timer mode control register 02 (TMC02)

TMC02 determines whether to enable or stop operation of 8-bit timer counter 02 (TM02) and specifies the count clock for 8-bit timer 02. It also controls the operation of the output controller.

TMC02 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC02 to 00H.

Figure 7-6. Format of 8-Bit Timer Mode Control Register 02

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC02	TCE02	0	0	0	0	TCL021	TCL020	TOE02	FF5BH	00H	R/W

TCE02	Operation control of 8-bit timer counter 02
0	Operation stopped (TM02 is cleared to 00H)
1	Operation enabled

TCL021	TCL020	Count clock selection of 8-bit timer 02
0	0	$f_X/2^3$ (625 kHz)
0	1	$f_X/2^7$ (39.1 kHz)
1	0	f_{XT} (32.768 kHz)
1	1	Setting prohibited

TOE02	Output control of 8-bit timer 02
0	Output disabled (port mode)
1	Output enabled

Caution Be sure to stop the operation of the timer before setting TMC02.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(4) Port mode register 2 (PM2)

This register sets port 2 to input/output in 1-bit units.

When using the P23/COMPTOUT0/TO2 pin for timer output, set PM23 and the output latch of P23 to 0.

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

$\overline{\text{RESET}}$ input sets PM2 to FFH.

Figure 7-7. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM23	P23 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operation of 8-Bit Timer/Event Counters 00 to 02

7.4.1 Operation as interval timer

The interval timer repeatedly generates an interrupt at time intervals specified by the count value set to 8-bit compare registers 00, 01, and 02 (CR00, CR01, and CR02) in advance.

To operate the 8-bit timer/event counter as an interval timer, make the settings in the following order.

<1> Set 8-bit timer counter 0n (TM0n) to operation-disabled (TCE0n (bit 7 of 8-bit timer mode control register 0n (TMC0n)) = 0)
<2> Select the count clock of the 8-bit timer/event counter (see **Tables 7-6** to **7-8**)
<3> Set the count value to CR0n
<4> Set TM0n to operation-enabled (TCE0n = 1)

When the count value of 8-bit timer counter 0n (TM0n) matches the value set to CR0n, the value of TM0n is cleared to 00H and TM0n continues counting. At the same time, an interrupt request signal (INTTM0n) is generated.

Tables 7-6 through 7-8 show the interval time, and Figures 7-8 and 7-9 show the timing of interval timer operation.

Caution When the setting of the count clock using TMC0n and the setting of the TM0n to operation-enable using an 8-bit memory manipulation instruction are performed at the same time, an error of one clock or more may occur in the first cycle after the timer is started. Because of this, when the 8-bit timer/event counter operates as an interval timer, be sure to make the settings in the order described above.

Remark n = 0 to 2

Table 7-6. Interval Time of 8-Bit Timer/Event Counter 00

TCL001	TCL000	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	2^6/fx (12.8 μs)	2^{14}/fx (3.28 ms)	2^6/fx (12.8 μs)
0	1	2^9/fx (102.4 μs)	2^{17}/fx (26.2 ms)	2^9/fx (102.4 μs)
1	0	TI0 input cycle	$2^8 \times$ TI0 input cycle	TI0 input edge cycle
1	1	TI0 input cycle	$2^8 \times$ TI0 input cycle	TI0 input edge cycle

Remarks 1. fx: Main system clock oscillation frequency
2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 7-7. Interval Time of 8-Bit Timer/Event Counter 01

TCL011	TCL010	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	2^4/fx (3.2 μs)	2^{12}/fx (819.2 μs)	2^4/fx (3.2 μs)
0	1	2^8/fx (51.2 μs)	2^{16}/fx (13.1 ms)	2^8/fx (51.2 μs)
1	0	TI1 input cycle	$2^8 \times$ TI1 input cycle	TI1 input edge cycle
1	1	TI1 input cycle	$2^8 \times$ TI1 input cycle	TI1 input edge cycle

Remarks 1. fx: Main system clock oscillation frequency
2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 7-8. Interval Time of 8-Bit Timer 02

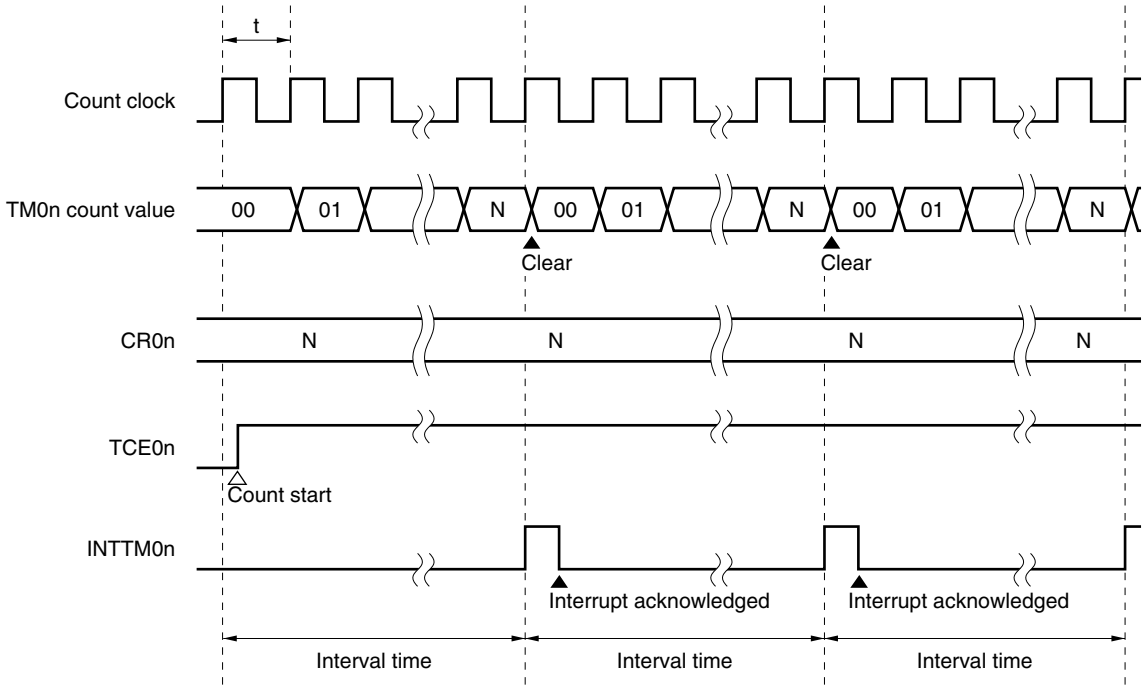
TCL021	TCL020	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)
0	1	$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	1	Setting prohibited		

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

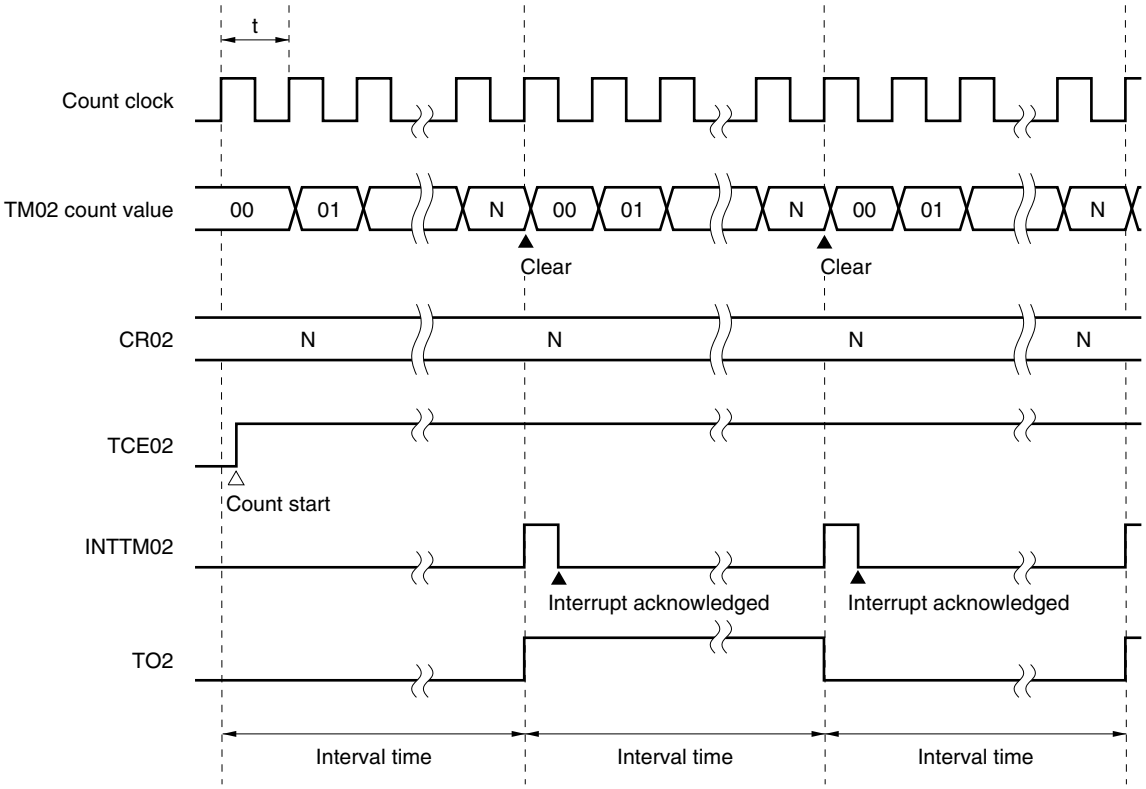
Figure 7-8. Interval Timer Operation Timing of Timer 00 and Timer 01



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

2. n = 0, 1

Figure 7-9. Interval Timer Operation Timing of Timer 02



Remark Interval time = (N + 1) × t where N = 00H to FFH

7.4.2 Operation as external event counter (timer 00 and timer 01 only)

The external event counter counts the number of external clock pulses input to the TI0/P24/INTP0 and TI1/P25/INTP1 pins by using 8-bit timer counters 00 and 01 (TM00 and TM01).

To operate 8-bit timer/event counters 00 and 01 as an external event counter, make the settings in the following order.

<1> Set P24 and P25 to input mode (PM24 = 1, PM25 = 1)
<2> Set 8-bit timer counter 0n (TM0n) to operation-disabled (TCE0n (bit 7 of 8-bit timer mode control register 0n (TMC0n)) = 0)
<3> Specify the rising edge/falling edge of TIn (see **Tables 7-6** and **7-7**)
<4> Set the count value to CR0n
<5> Set TM0n to operation-enabled (TCE0n = 1)

Each time the valid edge specified by bit 1 (TCL0n0) of TMC0n is input, the value of 8-bit timer counter 0n (TM0n) is incremented.

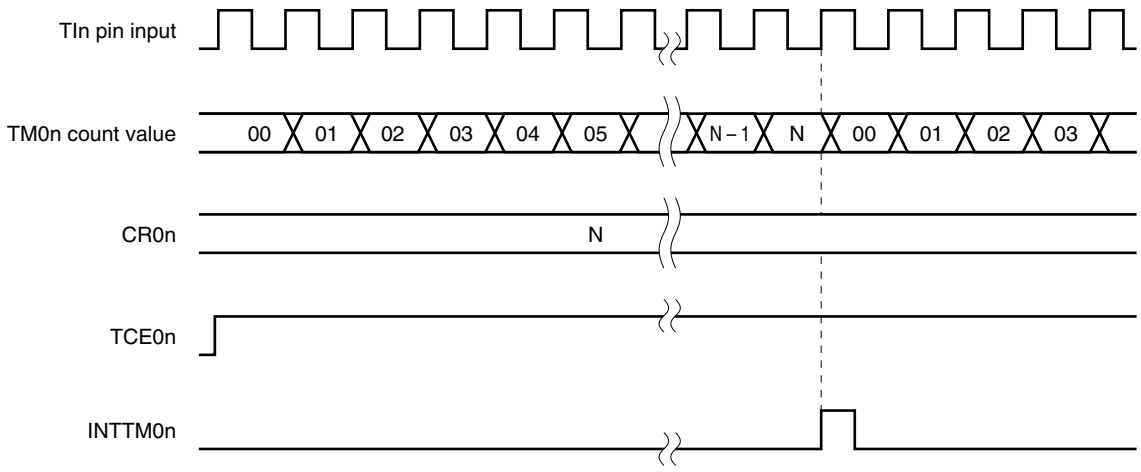
When the count value of TM0n matches the value set to CR0n, the value of TM0n is cleared to 00H and TM0n continues counting. At the same time, an interrupt request signal (INTTM0n) is generated.

Figure 7-10 shows the timing of external event counter operation (with rising edge specified).

Caution When the setting of the count clock using TMC0n and the setting of the TM0n to operation-enable using an 8-bit memory manipulation instruction are performed at the same time, an error of one clock or more may occur in the first cycle after the timer is started. Because of this, when the 8-bit timer/event counter operates as an external event counter, be sure to make the settings in the order described above.

Remark n = 0, 1

Figure 7-10. External Event Counter Operation Timing (with Rising Edge Specified)



Remarks 1. N = 00H to FFH
2. n = 0, 1

7.4.3 Operation as square-wave output (timer 02 only)

The 8-bit timer can generate a square-wave output of any frequency at intervals specified by the count value preset to 8-bit compare register 02 (CR02).

To operate 8-bit timer 02 as a square-wave output, make the settings in the following order.

<1> Set P23 to output mode (PM23 = 0), and set the output latch of P23 to 0
<2> Disable 8-bit timer counter 02 (TM02) operation (TCE02 (bit 0 of 8-bit timer mode control register 02 (TMC02)) = 1)
<3> Set the count clock of 8-bit timer 02 (see **Table 7-9**), and enable TO2 to output (TOE02 (bit 0 of TMC02) = 1)
<4> Set the count value to CR02
<5> Enable TM02 operation (TCE02 = 1)

When the count value of 8-bit timer counter 02 (TM02) matches the value set in CR02, the TO2/P23/CMPTOUT0 pin output is inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, the TM02 value is cleared to 00H, then counting continues count and an interrupt request signal (INTTM02) is generated.

Setting bit 7 of TMC02 (TCE02) to 0 clears the square-wave output to 0.

Table 7-9 lists the square-wave output range, and Figure 7-11 shows the timing of square-wave output.

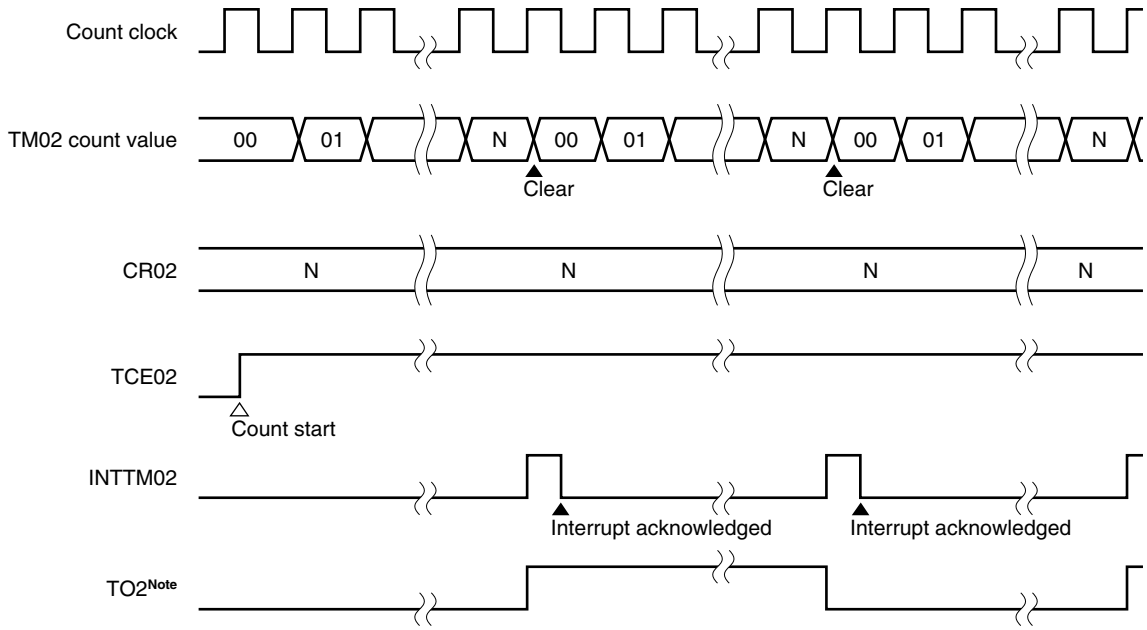
Caution When the setting of the count clock using TMC02 and the setting of the TM02 to operation-enable using an 8-bit memory manipulation instruction are performed at the same time, an error of one clock or more may occur in the first cycle after the timer is started. Because of this, when the 8-bit timer operates as a square-wave output, be sure to make the settings in the order described above.

Table 7-9. Square-Wave Output Range of 8-Bit Timer 02

TCL021	TCL020	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)
0	1	$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	1	Setting prohibited		

Remarks 1. f_X: Main system clock oscillation frequency
2. f_{XT}: Subsystem clock oscillation frequency
3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

Figure 7-11. Square-Wave Output Timing



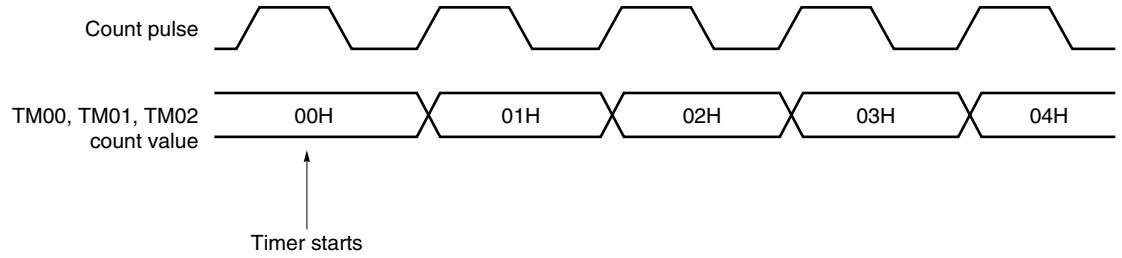
Note The initial value of TO2 when output is enabled (TOE02 = 1) becomes low level.

7.5 Cautions on Using 8-Bit Timer/Event Counters 00 to 02

(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 counters 00, 01, and 02 (TM00, TM01, and TM02) are started asynchronous to the count pulse.

Figure 7-12. Start Timing of 8-Bit Timer Counters 00, 01, and 02

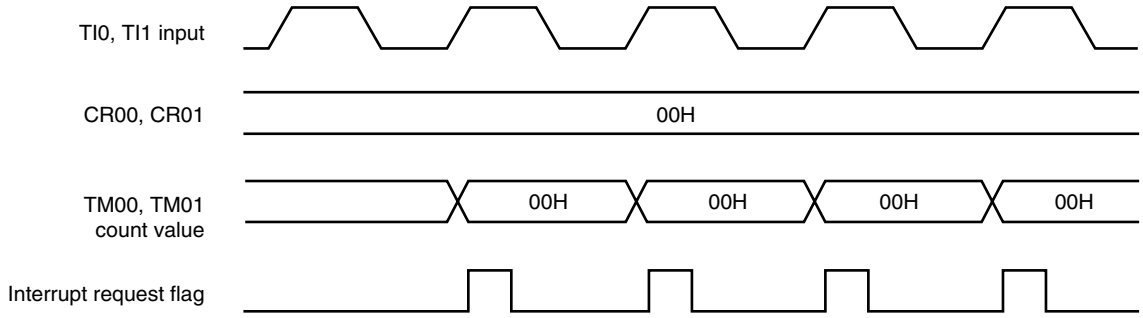


(2) Setting of 8-bit compare register

8-bit compare registers 00, 01, and 02 (CR00, CR01, and CR02) can be set to 00H.

Therefore, one pulse can be counted when an 8-bit timer/event counter operates as an event counter.

Figure 7-13. External Event Counter Operation Timing



CHAPTER 8 WATCH TIMER

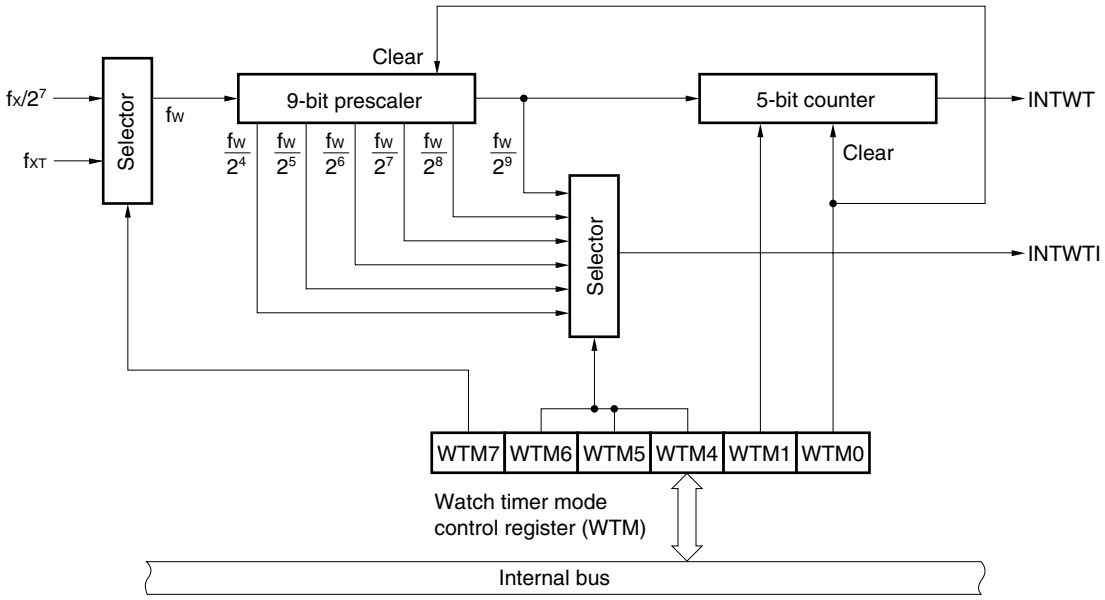
8.1 Functions of Watch Timer

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch and interval timers can be used at the same time.
Figure 8-1 is a block diagram of the watch timer.

Figure 8-1. Block Diagram of Watch Timer



(1) Watch timer

The 4.19 MHz main system clock or 32.768 kHz subsystem clock is used to issue an interrupt request (INTWT) at 0.5-second intervals.

Caution When the main system clock is operating at 5.0 MHz, it cannot be used to generate a 0.5-second interval. In this case, the subsystem clock, which operates at 32.768 kHz, should be used instead.

(2) Interval timer

The interval timer is used to generate an interrupt request (INTWT) at specified intervals.

Table 8-1. Interval Time of Interval Timer

Interval	Operation at f_X = 5.0 MHz	Operation at f_X = 4.19 MHz	Operation at f_{XT} = 32.768 kHz
2^4 1/f_W	409.6 s	489 s	488 s
2^5 1/f_W	819.2 s	978 s	977 s
2^6 1/f_W	1.64 ms	1.96 ms	1.95 ms
2^7 1/f_W	3.28 ms	3.91 ms	3.91 ms
2^8 1/f_W	6.55 ms	7.82 ms	7.81 ms
2^9 1/f_W	13.1 ms	15.6 ms	15.6 ms

Remark f_W: Watch timer clock frequency ($f_X/2^7$ or f_{XT})
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

8.2 Configuration of Watch Timer

The watch timer consists of the following hardware.

Table 8-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits 1
Prescaler	9 bits 1
Control register	Watch timer mode control register (WTM)

8.3 Register Controlling Watch Timer

The watch timer mode control register (WTM) is used to control the watch timer.

- Watch timer mode control register (WTM)
  WTM selects a count clock for the watch timer and specifies whether to enable operation of the timer. It also specifies the prescaler interval and how the 5-bit counter is controlled.
  WTM is set using a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input sets WTM to 00H.

Figure 8-2. Format of Watch Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
WTM	WTM7	WTM6	WTM5	WTM4	0	0	WTM1	WTM0	FF4AH	00H	R/W

WTM7	Watch timer count clock selection
0	$f_X/2^7$ (39.1 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler interval selection
0	0	0	$2^4/f_W$ (488 s)
0	0	1	$2^5/f_W$ (977 s)
0	1	0	$2^6/f_W$ (1.95 ms)
0	1	1	$2^7/f_W$ (3.91 ms)
1	0	0	$2^8/f_W$ (7.81 ms)
1	0	1	$2^9/f_W$ (15.6 ms)
Other than above			Setting prohibited

WTM1	Control of 5-bit counter operation
0	Cleared after stop
1	Started

WTM0	Watch timer operation
0	Operation stopped (both prescaler and timer cleared)
1	Operation enabled

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$ or f_{XT})

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

4. The parenthesized values apply to operation at f_W = 32.768 kHz.

8.4 Operation of Watch Timer

8.4.1 Operation as watch timer

The main system clock (4.19 MHz) or subsystem clock (32.768 kHz) is used as a watch timer that generates interrupts at 0.5-second intervals.

By setting bits 0 and 1 (WTM0 and WTM1) of the watch timer mode control register (WTM) to 1, the watch timer starts counting. By setting them to 0, the 5-bit counter is cleared and the watch timer stops counting.

When the interval timer also operates at the same time by setting WTM1 to 0, only the watch timer can be started from 0 seconds. However, an error of up to 2^9 1/fw seconds may occur for the first overflow of the watch timer (INTWT) after a 0-second start, because the 9-bit prescaler is not cleared in this case.

8.4.2 Operation as interval timer

The interval timer is used to repeatedly generate an interrupt request at the interval specified by a preset count value.

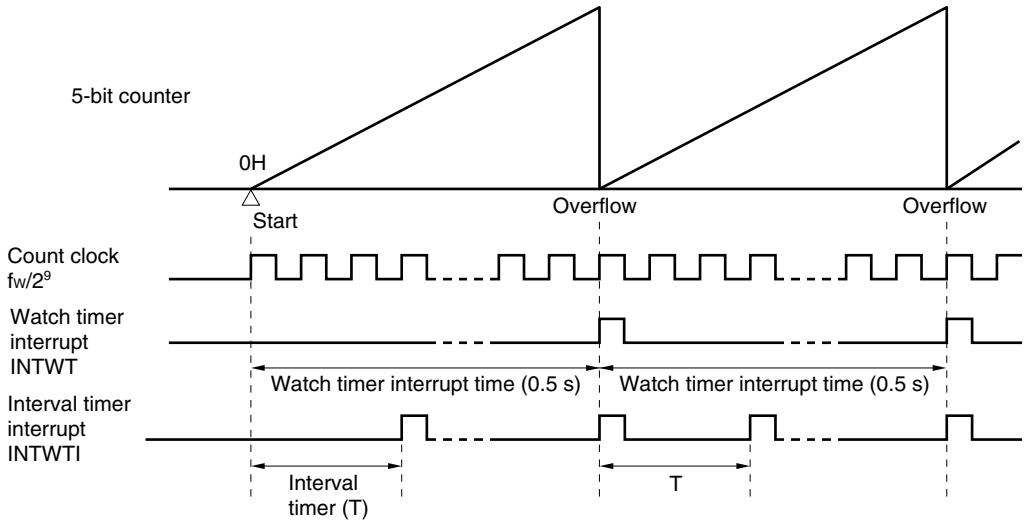
The interval time can be selected by bits 4 to 6 (WTM4 to WTM6) of the watch timer mode control register (WTM).

Table 8-3. Interval Time of Interval Timer

WTM6	WTM5	WTM4	Interval	Operation at f_X = 5.0 MHz	Operation at f_X = 4.19 MHz	Operation at f_{XT} = 32.768 kHz
0	0	0	2^4 1/fw	409.6 s	489 s	488 s
0	0	1	2^5 1/fw	819.2 s	978 s	977 s
0	1	0	2^6 1/fw	1.64 ms	1.96 ms	1.95 ms
0	1	1	2^7 1/fw	3.28 ms	3.91 ms	3.91 ms
1	0	0	2^8 1/fw	6.55 ms	7.82 ms	7.81 ms
1	0	1	2^9 1/fw	13.1 ms	15.6 ms	15.6 ms
Other than above			Setting prohibited			

Remark f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency
fw: Watch timer clock frequency

Figure 8-3. Watch Timer/Interval Timer Operation Timing



Remark fw: Watch timer clock frequency

The parenthesized values apply to operation at fw = 32.768 kHz.

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

Caution Select the watchdog timer mode or interval timer mode by using the watchdog timer mode register (WDTM).

(1) Watchdog timer

The watchdog timer is used to detect an inadvertent program loop. When the program loop is detected, a non-maskable interrupt or the $\overline{\text{RESET}}$ signal can be generated.

Table 9-1. Program Loop Detection Time of Watchdog Timer

Program Loop Detection Time	Operation at f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs
$2^{13} \times 1/f_X$	1.64 ms
$2^{15} \times 1/f_X$	6.55 ms
$2^{17} \times 1/f_X$	26.2 ms

f_X: Main system clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at any intervals set in advance.

Table 9-2. Interval Time

Interval Time	Operation at f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs
$2^{13} \times 1/f_X$	1.64 ms
$2^{15} \times 1/f_X$	6.55 ms
$2^{17} \times 1/f_X$	26.2 ms

f_X: Main system clock oscillation frequency

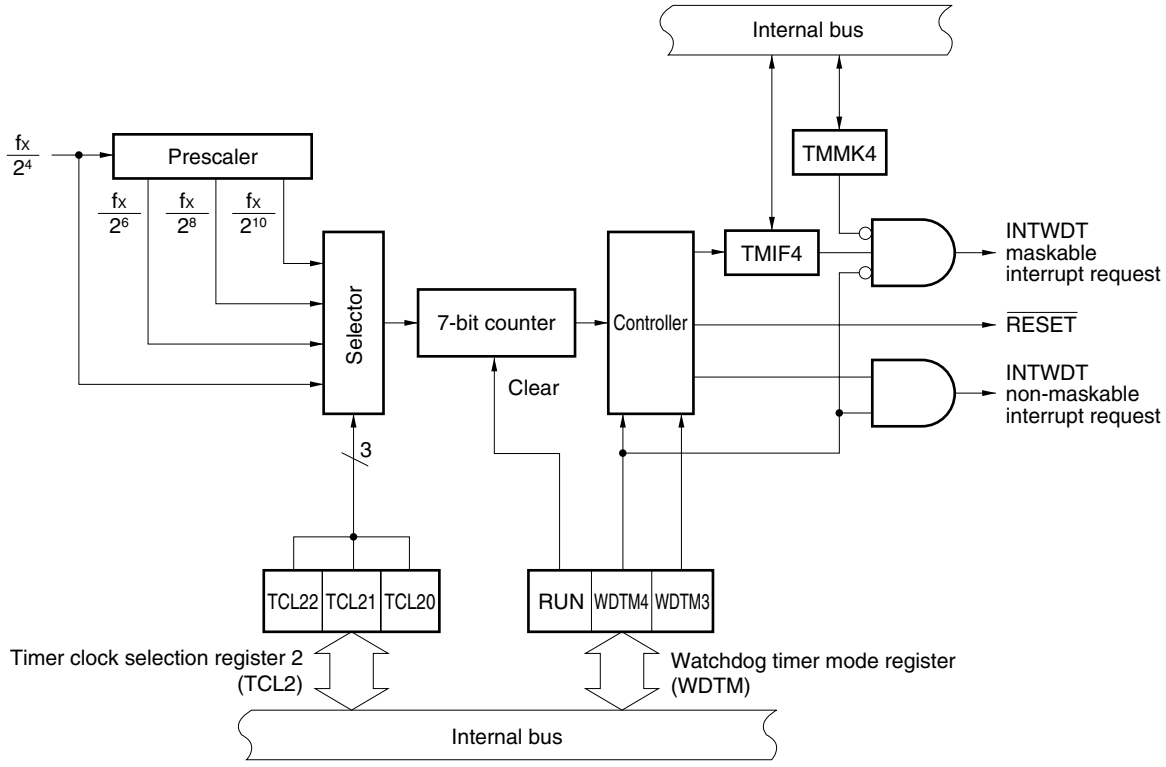
9.2 Configuration of Watchdog Timer

The watchdog timer consists of the following hardware.

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Timer clock selection register 2 (TCL2) Watchdog timer mode register (WDTM)

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Registers Controlling Watchdog Timer

The following two registers are used to control the watchdog timer.

- Timer clock selection register 2 (TCL2)
- Watchdog timer mode register (WDTM)

(1) Timer clock selection register 2 (TCL2)

This register sets the watchdog timer count clock.

TCL2 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL2 to 00H.

Figure 9-2. Format of Timer Clock Selection Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	0	0	0	0	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Watchdog timer count clock selection	Interval time
0	0	0	$f_X/2^4$ (312.5 kHz)	$2^{11}/f_X$ (410 μs)
0	1	0	$f_X/2^6$ (78.1 kHz)	$2^{13}/f_X$ (1.64 ms)
1	0	0	$f_X/2^8$ (19.5 kHz)	$2^{15}/f_X$ (6.55 ms)
1	1	0	$f_X/2^{10}$ (4.88 kHz)	$2^{17}/f_X$ (26.2 ms)
Other than above			Setting prohibited	

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(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 using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets WDTM 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

RUN	Selection of operation of watchdog timer[Note 1]
0	Stop counting
1	Clear counter and start counting

WDTM4	WDTM3	Selection of operation mode of watchdog timer[Note 2]
0	0	Operation stopped
0	1	Interval timer mode (overflow and maskable interrupt occur)[Note 3]
1	0	Watchdog timer mode 1 (overflow and non-maskable interrupt occur)
1	1	Watchdog timer mode 2 (overflow occurs and reset operation started)

Notes 1. 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.

2. Once WDTM3 and WDTM4 have been set to (1), they cannot be cleared to (0) by software.

3. The watchdog timer starts operation as an interval timer when RUN is set to 1.

Cautions 1. When the watchdog timer is cleared by setting RUN to 1, the actual overflow time is up to 0.8% shorter than the time set by timer clock selection register 2 (TCL2).

2. In watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming that TMIF4 (bit 0 of interrupt request flag register 0 (IF0)) is set to 0. While TMIF4 is 1, a non-maskable interrupt is generated upon write completion if watchdog timer mode 1 or 2 is selected.

9.4 Operation of Watchdog Timer

9.4.1 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 count clock (program loop detection time interval) of the watchdog timer can be selected by bits 0 to 2 (TCL20 to TCL22) of timer clock selection register 2 (TCL2). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set program loop detection time interval after the watchdog timer has been started. By setting RUN to 1, the watchdog timer can be cleared and start counting. If RUN is not set to 1, and the program loop detection time is exceeded, the system is reset or a non-maskable interrupt 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 program loop detection time may be up to 0.8% shorter than the set time.

2. When the subsystem clock is selected as the CPU clock, the watchdog timer stops counting.

Table 9-4. Program Loop Detection Time of Watchdog Timer

TCL22	TCL21	TCL20	Program Loop Detection Time	Operation at f_X = 5.0 MHz
0	0	0	2^{11} 1/f_X	410 s
0	1	0	2^{13} 1/f_X	1.64 ms
1	0	0	2^{15} 1/f_X	6.55 ms
1	1	0	2^{17} 1/f_X	26.2 ms

f_X: Main system clock oscillation frequency

9.4.2 Operation as interval timer

When bit 4 (WDTM4) and bit 3 (WDTM3) of the watchdog timer mode register (WDTM) are set to 0 and 1, respectively, the watchdog timer also operates as an interval timer that repeatedly generates an interrupt at time intervals specified by a preset count value.

Select the count clock (or interval time) by setting bits 0 to 2 (TCL20 to TCL22) of timer clock selection register 2 (TCL2). The watchdog timer starts operation as an interval timer when the RUN bit (bit 7 of WDTM) is set to 1.

In the interval timer mode, the interrupt mask flag (TMMK4) is valid, and a maskable interrupt (INTWDT) can be generated. The priority of INTWDT is set as the highest of all the maskable interrupts.

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 is 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 the setting by WDTM may be up to 0.8% shorter than the set time.

Table 9-5. Interval Time of Interval Timer

TCL22	TCL21	TCL20	Interval Time	Operation at f_X = 5.0 MHz
0	0	0	2^{11} 1/f_X	410 s
0	1	0	2^{13} 1/f_X	1.64 ms
1	0	0	2^{15} 1/f_X	6.55 ms
1	1	0	2^{17} 1/f_X	26.2 ms

f_X: Main system clock oscillation frequency

CHAPTER 10 8-BIT A/D CONVERTER (μPD789407A SUBSERIES)

10.1 Function of 8-Bit A/D Converter

The 8-bit A/D converter converts input analog voltages to digital signals with an 8-bit resolution. It can control up to seven analog input channels (ANI0 to ANI6).

A/D conversion can be started only by software.

One of analog inputs ANI0 to ANI6 is selected for A/D conversion. A/D conversion is performed repeatedly, with an interrupt request (INTAD0) being issued each time an A/D conversion is completed.

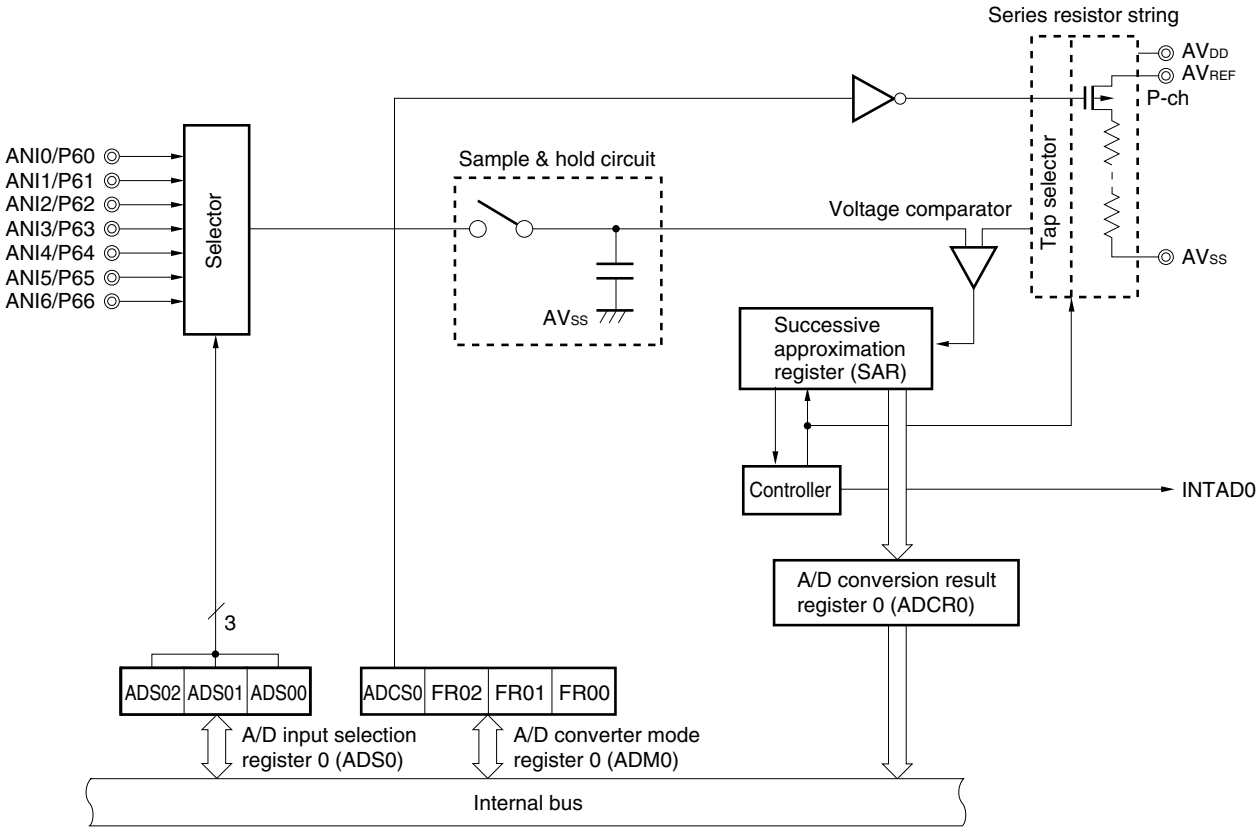
10.2 Configuration of 8-Bit A/D Converter

The 8-bit A/D converter consists of the following hardware.

Table 10-1. Configuration of 8-Bit A/D Converter

Item	Configuration
Analog inputs	7 channels (ANI0 to ANI6)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) A/D input selection register 0 (ADS0)

Figure 10-1. Block Diagram of 8-Bit A/D Converter



(1) Successive approximation register (SAR)

The SAR receives the result of comparing an analog input voltage and a voltage at a voltage tap (comparison voltage), received from the series resistor string, starting from the most significant bit (MSB).

Upon receiving all the bits, down to the least significant bit (LSB), that is, upon the completion of A/D conversion, the SAR sends its contents to A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

ADCR0 holds the result of A/D conversion. Each time A/D conversion ends, the conversion result received from the successive approximation register is loaded into ADCR0, which is an 8-bit register that holds the result of A/D conversion.

ADCR0 is read using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR0 undefined.

(3) Sample & hold circuit

The sample & hold circuit samples consecutive analog inputs from the input circuit, one by one, and sends them to the voltage comparator. The sampled analog input voltage is held during A/D conversion.

(4) Voltage comparator

The voltage comparator compares an analog input with the voltage output by the series resistor string.

(5) Series resistor string

The series resistor string is configured between AV_{REF} and AV_{SS}. It generates the reference voltages against which analog inputs are compared.

(6) ANI0 to ANI6 pins

The ANI0 to ANI6 pins are analog input pins for the seven-channel A/D converter. They are used to receive the analog signals to be subject to A/D conversion.

Caution Do not supply the ANI0 to ANI6 pins with voltages that fall outside the rated range. If a voltage greater than AV_{REF} or less than AV_{SS} (even if within the absolute maximum rating) is supplied to any of these pins, the conversion value for the corresponding channel will be undefined. Furthermore, the conversion values for the other channels may also be affected.

(7) AV_{REF} pin

The AV_{REF} pin is a reference voltage pin for the A/D converter.

Signals received at the ANI0 to ANI6 pins are converted to digital signals based on the voltage across the AV_{REF} and AV_{SS} pins.

(8) AV_{SS} pin

The AV_{SS} pin is a ground potential pin for the A/D converter. This pin must be held at the same potential as the V_{SS0} pin, even while the A/D converter is not being used.

(9) AV_{DD} pin

The AV_{DD} pin is an analog power supply pin for the A/D converter. This pin must be held at the same potential as the V_{DD0} pin, even while the A/D converter is not being used.

10.3 Registers Controlling 8-Bit A/D Converter

The following two registers are used to control the 8-bit A/D converter.

- A/D converter mode register 0 (ADM0)
- A/D input selection register 0 (ADS0)

(1) A/D converter mode register 0 (ADM0)

ADM0 specifies the conversion time for analog inputs. It also specifies whether to enable conversion.

ADM0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADM0 to 00H.

Figure 10-2. Format of A/D Converter Mode Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
ADM0	ADCS0	0	FR02	FR01	FR00	0	0	0	FF80H	00H	R/W

ADCS0	A/D conversion control
0	Conversion stopped
1	Conversion enabled

FR02	FR01	FR00	A/D conversion time selection[Note 1]
0	0	0	144/fx (28.8 μs)
0	0	1	120/fx (24 μs)
0	1	0	96/fx (19.2 μs)
1	0	0	72/fx (14.4 μs)
1	0	1	60/fx (Setting prohibited[Note 2])
1	1	0	48/fx (Setting prohibited[Note 2])
Other than above			Setting prohibited

Notes 1. The specifications of FR02, FR01, and FR00 must be such that the A/D conversion time is at least 14 μs.

2. These bit combinations must not be used, as the A/D conversion time will fall below 14 μs.

Cautions 1. The result of conversion performed immediately after bit 7 (ADCS0) is set is undefined.

2. The result of conversion performed after ADCS0 is cleared may be undefined (see 10.5 (5) Timing that makes the A/D conversion result undefined for details).

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

(2) A/D input selection register 0 (ADS0)

ADS0 register specifies the port used to input the analog voltages to be converted to a digital signal.

ADS0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADS0 to 00H.

Figure 10-3. Format of A/D Input Selection 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	FF84H	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	Setting prohibited

Caution Bits 3 to 7 must be fixed to 0.

10.4 Operation of 8-Bit A/D Converter

10.4.1 Basic operation of 8-bit A/D converter

<1> Select a channel for A/D conversion, using A/D input selection register 0 (ADS0).

<2> The voltage supplied to the selected analog input channel is sampled using the sample & hold circuit.

<3> After sampling continues for a certain period of time, the sample & hold circuit is put on hold to keep the input analog voltage until A/D conversion is completed.

<4> Bit 7 of the successive approximation register (SAR) is set. The series resistor string voltage tap at the tap selector is set to half of AV_{REF}.

<5> The series resistor string tap voltage is compared with the analog input voltage using the voltage comparator. If the analog input voltage is higher than half of AV_{REF}, the MSB of the SAR remains set. If it is lower than half of AV_{REF}, the MSB is reset.

<6> Bit 6 of the SAR is set automatically, and comparison shifts to the next stage. The next voltage tap of the series resistor string is selected according to bit 7, which reflects the previous comparison result, as follows:

- Bit 7 = 1: Three quarters of AV_{REF}
- Bit 7 = 0: One quarter of AV_{REF}

The tap voltage is compared with the analog input voltage. Bit 6 is set or reset according to the result of comparison.

- Analog input voltage ≥ tap voltage: Bit 6 = 1
- Analog input voltage < tap voltage: Bit 6 = 0

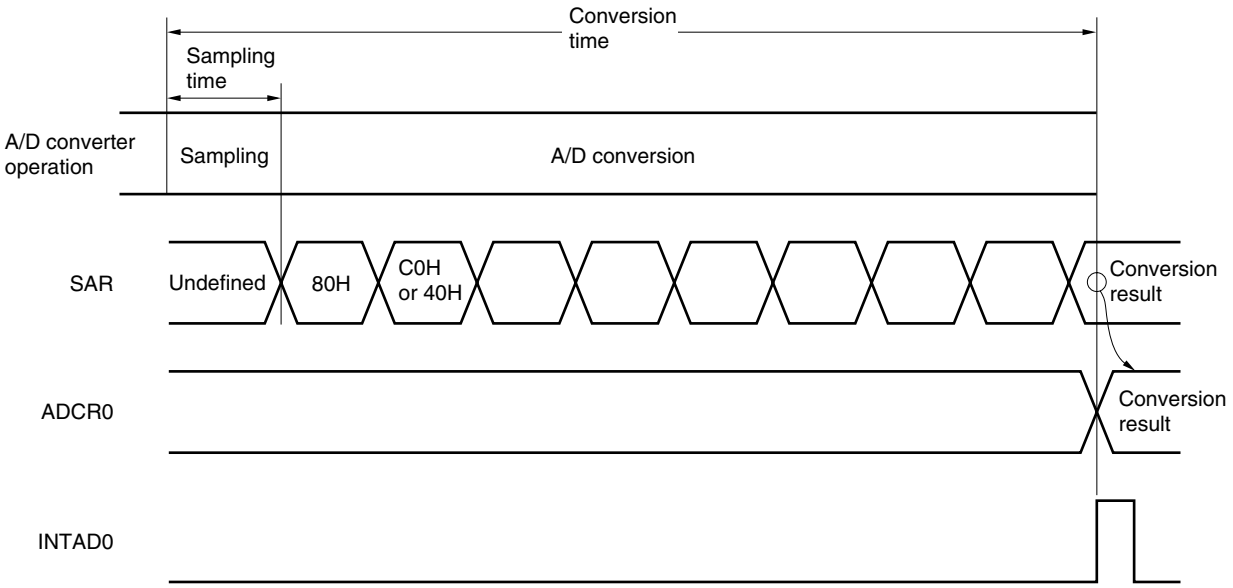
<7> Comparison is repeated until bit 0 of the SAR is reached.

<8> When comparison is completed for all of the 8 bits, a significant digital result is left in the SAR. This value is sent to and latched in A/D conversion result register 0 (ADCR0). At the same time, it is possible to generate an A/D conversion end interrupt request (INTAD0).

Cautions 1. The first A/D conversion value immediately following the start of A/D conversion may be undefined.

2. When the A/D converter enters the standby mode, it stops operating.

Figure 10-4. Basic Operation of 8-Bit A/D Converter



A/D conversion continues until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset (0) by software.

If an attempt is made to write to ADM0 or A/D input selection register 0 (ADS0) during A/D conversion, the current A/D conversion is canceled. In this case, A/D conversion is restarted from the beginning, if the ADCS0 bit is set (1).

$\overline{\text{RESET}}$ makes A/D conversion result register 0 (ADCR0) undefined.

10.4.2 Input voltage and conversion result

The relationship between the analog input voltage at the analog input pins (ANI0 to ANI6) and the A/D conversion result (A/D conversion result register 0 (ADCR0)) is represented by:

$$ADCR0 = INT\left(\frac{V_{IN}}{AV_{REF}} \times 256 + 0.5\right)$$

or

$$(ADCR0 - 0.5) \times \frac{AV_{REF}}{256} \leq V_{IN} < (ADCR0 + 0.5) \times \frac{AV_{REF}}{256}$$

INT( ): Function that returns the integer part of a parenthesized value

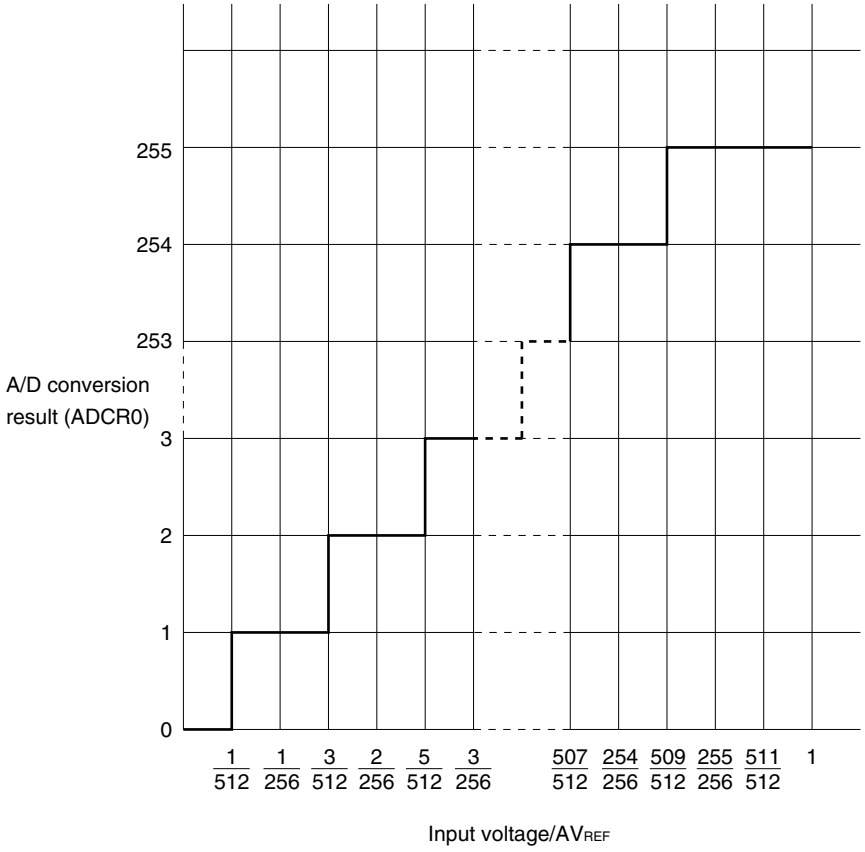
V_{IN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

ADCR0: Value in A/D conversion result register 0 (ADCR0)

Figure 10-5 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-5. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 Operation mode of 8-bit A/D converter

The 8-bit A/D converter is initially in the select mode. In this mode, A/D input selection register 0 (ADS0) is used to select an analog input channel from ANI0 to ANI6 for A/D conversion.

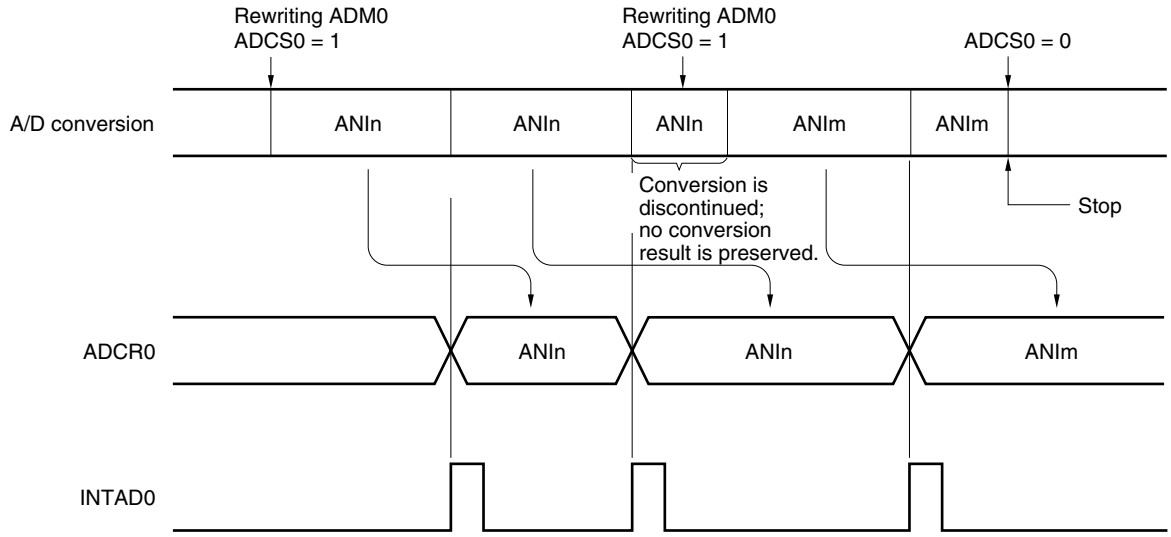
A/D conversion can be started only by software, that is, by setting A/D converter mode register 0 (ADM0).

The A/D conversion result is saved to A/D conversion result register 0 (ADCR0). At the same time, an interrupt request signal (INTAD0) is generated.

- **Software-started A/D conversion**

  Setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) triggers A/D conversion for a voltage applied to the analog input pin specified in A/D input selection register 0 (ADS0). Upon completion of A/D conversion, the conversion result is saved to A/D conversion result register 0 (ADCR0). At the same time, an interrupt request signal (INTAD0) is generated. Once A/D conversion is activated, and completed, another session of A/D conversion is started. A/D conversion is repeated until new data is written to ADM0. If data where the ADCS0 bit is 1 is written to ADM0 again during A/D conversion, the current session of A/D conversion is discontinued, and a new session of A/D conversion begins for the new data. If data where the ADCS0 bit is 0 is written to ADM0 again during A/D conversion, A/D conversion is stopped immediately.

Figure 10-6. Software-Started A/D Conversion



Remarks 1. n = 0, 1, ..., 6

2. m = 0, 1, ..., 6

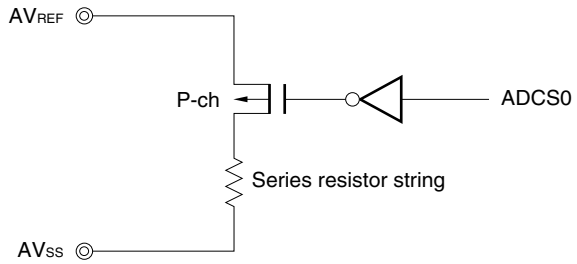
10.5 Cautions on Using 8-Bit A/D Converter

(1) Current consumption in the standby mode

When the A/D converter enters the standby mode, it stops operating. Stopping conversion (bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) = 0) can reduce the current consumption.

Figure 10-7 shows how to reduce the current consumption in the standby mode.

Figure 10-7. How to Reduce Current Consumption in Standby Mode



(2) Input range for the ANI0 to ANI6 pins

Be sure to keep the input voltage at ANI0 to ANI6 within the rated range. If a voltage greater than AV_{REF} or less than AV_{SS} (even within the absolute maximum rating) is input to a conversion channel, the conversion output of the channel becomes undefined, and the conversion output of the other channels may also be affected.

(3) Conflict

<1> Conflict between writing to A/D conversion result register 0 (ADCR0) at the end of conversion and reading from the ADCR0 bit

Reading from the ADCR0 bit takes precedence. After reading, the new conversion result is written to the ADCR0 bit.

<2> Conflict between writing to the ADCR0 bit at the end of conversion and writing to A/D converter mode register 0 (ADM0) or A/D input selection register 0 (ADS0)

Writing to ADM0 or ADS0 takes precedence. A request to write to the ADCR0 bit is ignored. No A/D conversion end interrupt request signal (INTAD0) is generated.

(4) Conversion results immediately following start of A/D conversion

The first A/D conversion value immediately following the start of A/D conversion may be undefined. Be sure to poll the A/D conversion end interrupt request (INTAD0) and perform processing such as discarding the first conversion result.

(5) Timing that makes the A/D conversion result undefined

If the timing of the end of A/D conversion and the timing of the stop of operation of the A/D converter conflict, the A/D conversion value may be undefined. Because of this, be sure to read out the A/D conversion result while the A/D converter is in operation. Furthermore, when reading out an A/D conversion result after A/D conversion has stopped, be sure to have done so by the time the next conversion result is complete.

The conversion result readout timing is shown in Figures 10-8 and 10-9.

Figure 10-8. Conversion Result Readout Timing (When Conversion Result Is Undefined Value)

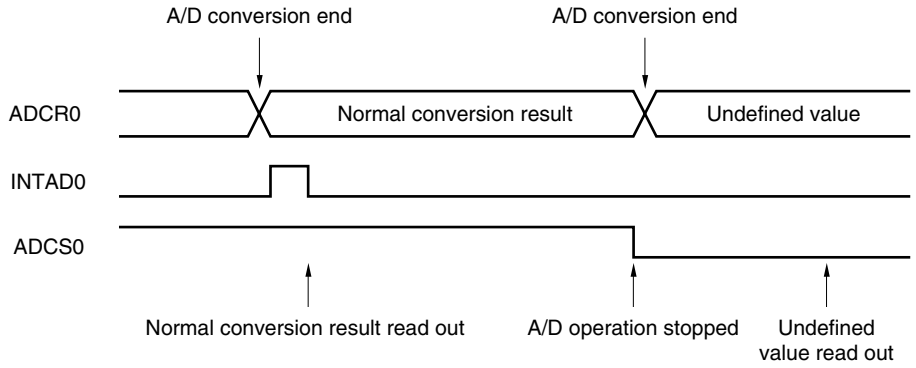
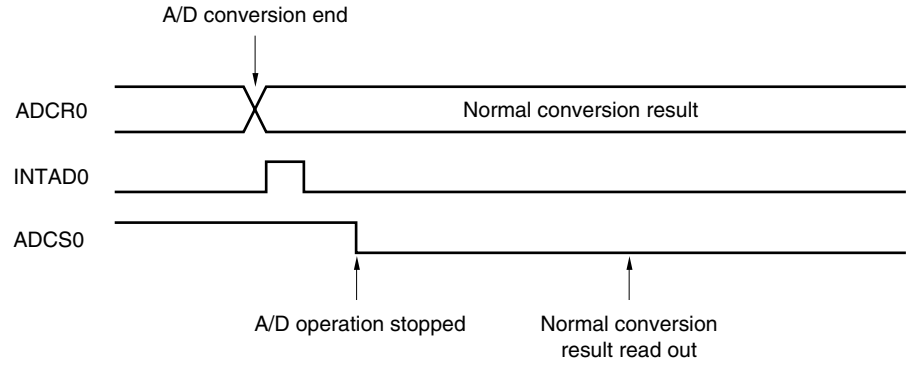


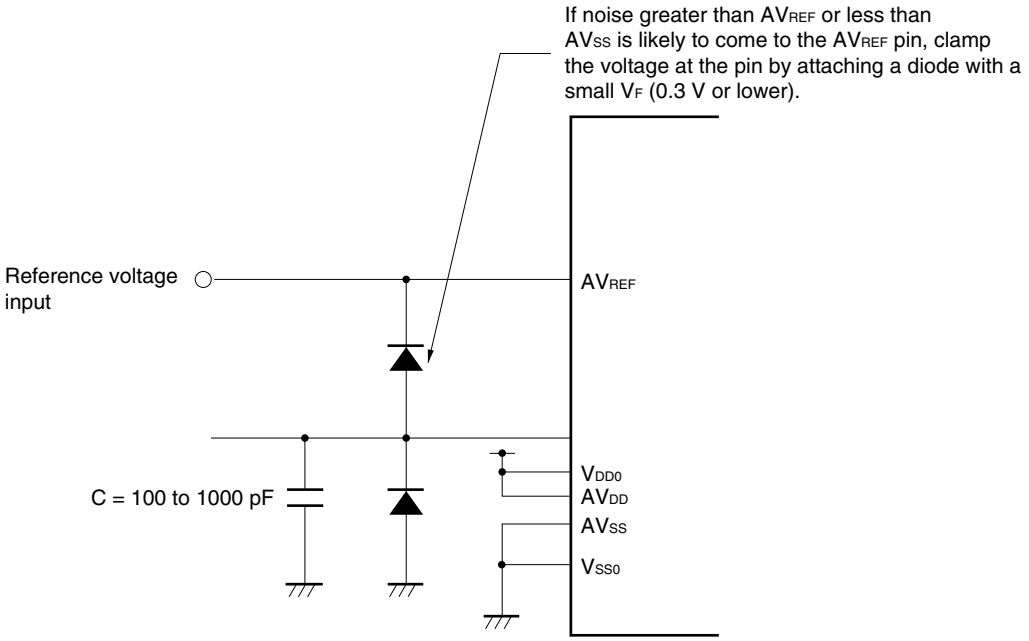
Figure 10-9. Conversion Result Readout Timing (When Conversion Result Is Normal Value)



(6) Noise elimination

To maintain a resolution of 8 bits, it is necessary to avoid noise at the AV_{REF} and ANI0 to ANI6 pins. The higher the output impedance of the analog input source, the larger the effect by noise. To eliminate noise, attach an external capacitor to the relevant pins as shown in Figure 10-10.

Figure 10-10. Analog Input Pin Processing



(7) ANI0 to ANI6

The analog input pins (ANI0 to ANI6) are alternate-function pins. They are also used as port pins (P60 to P66).

If any of ANI0 to ANI6 has been selected for A/D conversion, do not execute input instructions for the ports. Otherwise, the conversion resolution may become lower.

If a digital pulse is applied to a pin adjacent to the analog input pins during A/D conversion, coupling noise may occur which prevents an A/D conversion result from being attained as expected. Avoid applying a digital pulse to pins adjacent to the analog input pins during A/D conversion.

★ **(8) Input impedance of ANI0 to ANI6 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 ANI6 pins (see **Figure 10-10**).

(9) Input impedance of the AVREF pin

A series resistor string of several tens of kΩ is connected across the AVREF and AVSS pins.

If the output impedance of the reference voltage source is high, this high impedance is eventually connected in parallel with the series resistor string across the AVREF and AVSS pins, leading to a higher reference voltage error.

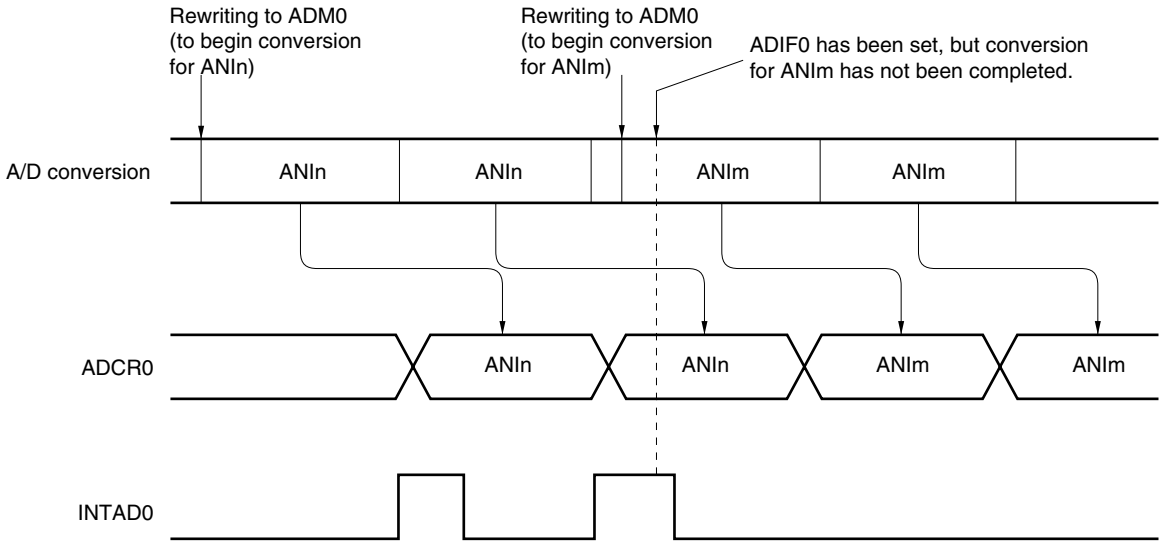
(10) Interrupt request flag (ADIF0)

Changing the contents of A/D converter mode register 0 (ADM0) does not clear the interrupt request flag (ADIF0).

If the voltage at the analog input pins is changed during A/D conversion, therefore, the A/D conversion result and the conversion end interrupt request flag may reflect the previous analog input just before writing to ADM0. In this case, the ADIF0 may appear to be set if it is read-accessed just after ADM0 is write-accessed, even when A/D conversion has not been completed for the new analog input.

In addition, ADIF0 must be cleared before A/D conversion is restarted.

Figure 10-11. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 6

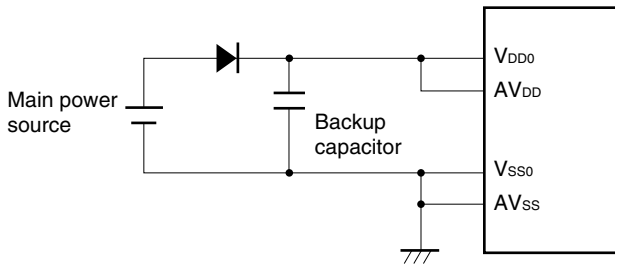
2. m = 0, 1, ..., 6

(11) AV_{DD} pin

The AV_{DD} pin is used to supply power to the analog circuit. It is also used to supply power to the ANI0 to ANI6 input circuit.

If your application is designed to be switched to backup power, the AV_{DD} pin must be supplied with the same voltage level as for the V_{DD0} pin, as shown in Figure 10-12.

Figure 10-12. AV_{DD} Pin Processing



CHAPTER 11 10-BIT A/D CONVERTER (μPD789417A SUBSERIES)

11.1 Function of 10-Bit A/D Converter

The 10-bit A/D converter converts input analog voltages to digital signals with a 10-bit resolution. It can control up to seven analog input channels (ANI0 to ANI6).

A/D conversion can be started only by software.

One of analog inputs ANI0 to ANI6 is selected for A/D conversion. A/D conversion is performed repeatedly, with an interrupt request (INTAD0) being issued each time an A/D conversion is completed.

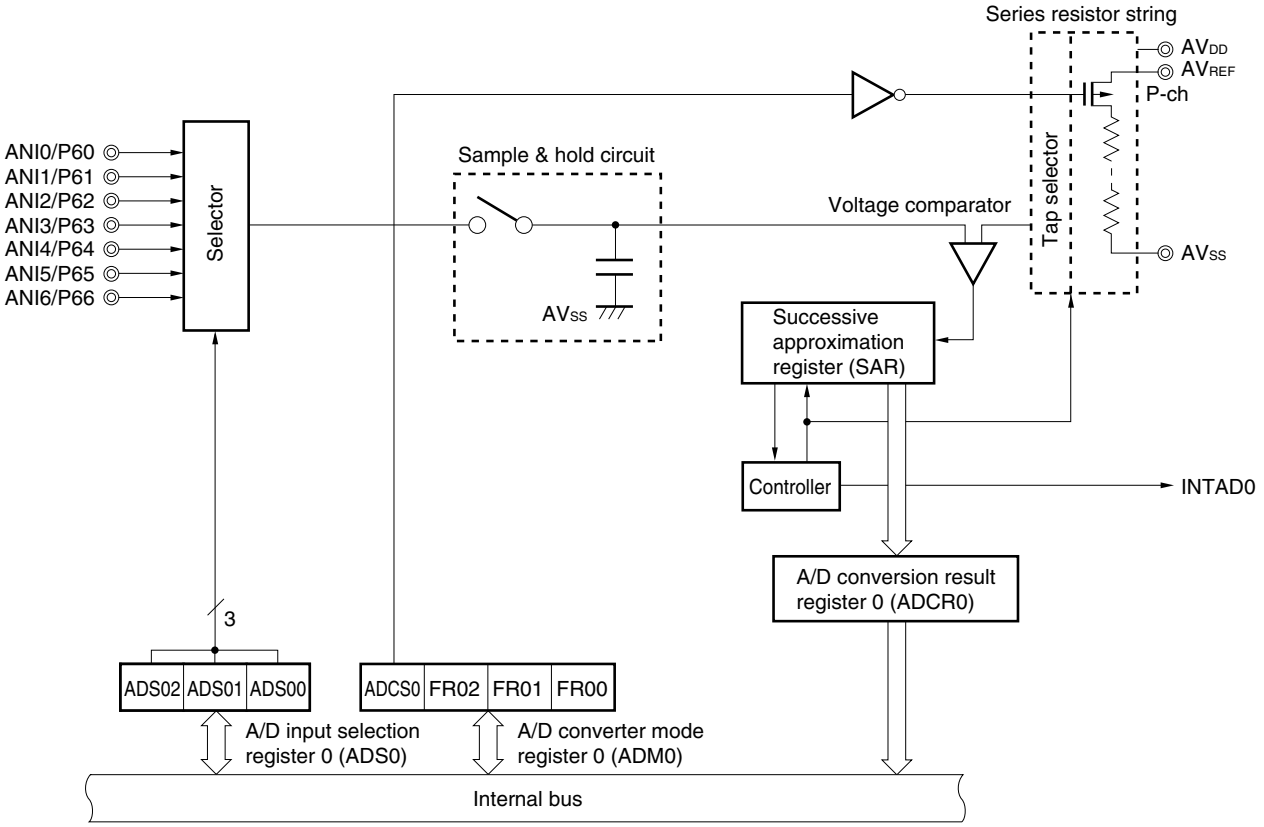
11.2 Configuration of 10-Bit A/D Converter

The A/D converter consists of the following hardware.

Table 11-1. Configuration of 10-Bit A/D Converter

Item	Configuration
Analog inputs	7 channels (ANI0 to ANI6)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) A/D input selection register 0 (ADS0)

Figure 11-1. Block Diagram of 10-Bit A/D Converter



(1) Successive approximation register (SAR)

The SAR receives the result of comparing an analog input voltage and a voltage at a voltage tap (comparison voltage), received from the series resistor string, starting from the most significant bit (MSB).

Upon receiving all the bits, down to the least significant bit (LSB), that is, upon the completion of A/D conversion, the SAR sends its contents to A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

★ ADCR0 is a 16-bit register that holds the result of A/D conversion. Lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result in the successive approximation register is loaded into ADCR0. The conversion results are stored in ADCR0 starting from the most significant bit (MSB). The higher 8 bits of the conversion results are stored in FF15H and the lower 2 bits of the conversion results are stored in FF14H.

ADCR0 is read using a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR0 undefined.

Symbol	FF15H								FF14H								Address	After reset	R/W
ADCR0											0	0	0	0	0	0	FF14H, FF15H	Undefined	R

Caution When the μPD78F9418A is used as the flash memory version of the μPD789405A, 789406A, and 789407A, 8-bit access is possible, providing an object file has been assembled in the μPD789405A, 789406A, and 789407A.

(3) Sample & hold circuit

The sample & hold circuit samples consecutive analog inputs from the input circuit, one by one, and sends them to the voltage comparator. The sampled analog input voltage is held during A/D conversion.

(4) Voltage comparator

The voltage comparator compares an analog input with the voltage output by the series resistor string.

(5) Series resistor string

The series resistor string is configured between AV_{REF} and AV_{SS}. It generates the reference voltages against which analog inputs are compared.

(6) ANI0 to ANI6 pins

The ANI0 to ANI6 pins are analog input pins for the seven-channel A/D converter. They are used to receive the analog signals to be subject to A/D conversion.

Caution Do not supply the ANI0 to ANI6 pins with voltages that fall outside the rated range. If a voltage greater than AV_{REF} or less than AV_{SS} (even if within the absolute maximum rating) is supplied to any of these pins, the conversion value for the corresponding channel will be undefined. Furthermore, the conversion values for the other channels may also be affected.

(7) AV_{REF} pin

The AV_{REF} pin is a reference voltage pin for the A/D converter.

Signals received at the ANI0 to ANI6 pins are converted to digital signals based on the voltage across the AV_{REF} and AV_{SS} pins.

(8) AV_{SS} pin

The AV_{SS} pin is a ground potential pin for the A/D converter. This pin must be held at the same potential as the V_{SS0} pin, even while the A/D converter is not being used.

(9) AV_{DD} pin

The AV_{DD} pin is an analog power supply pin for the A/D converter. This pin must be held at the same potential as the V_{DD0} pin, even while the A/D converter is not being used.

11.3 Registers Controlling 10-Bit A/D Converter

The following two registers are used to control the 10-bit A/D converter.

- A/D converter mode register 0 (ADM0)
- A/D input selection register 0 (ADS0)

(1) A/D converter mode register 0 (ADM0)

ADM0 specifies the conversion time for analog inputs. It also specifies whether to enable conversion.

ADM0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADM0 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	0	FR02	FR01	FR00	0	0	0	FF80H	00H	R/W

ADCS0	A/D conversion control
0	Conversion stopped
1	Conversion enabled

FR02	FR01	FR00	A/D conversion time selection[Note 1]
0	0	0	144/fx (28.8 μs)
0	0	1	120/fx (24 μs)
0	1	0	96/fx (19.2 μs)
1	0	0	72/fx (14.4 μs)
1	0	1	60/fx (Setting prohibited[Note 2])
1	1	0	48/fx (Setting prohibited[Note 2])
Other than above			Setting prohibited

Notes 1. The specifications of FR02, FR01, and FR00 must be such that the A/D conversion time is at least 14 μs.

2. These bit combinations must not be used, as the A/D conversion time will fall below 14 μs.

Cautions 1. The result of conversion performed immediately after bit 7 (ADCS0) is set is undefined.

2. The result of conversion performed after ADCS0 is cleared may be undefined (see 11.5 (5) Timing that makes the A/D conversion result undefined for details).

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

(2) A/D input selection register 0 (ADS0)

ADS0 register specifies the port used to input the analog voltages to be converted to a digital signal.

ADS0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADS0 to 00H.

Figure 11-3. Format of A/D Input Selection 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	FF84H	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	Setting prohibited

Caution Bits 3 to 7 must be fixed to 0.

11.4 Operation of 10-Bit A/D Converter

11.4.1 Basic operation of 10-bit A/D converter

<1> Select a channel for A/D conversion, using A/D input selection register 0 (ADS0).

<2> The voltage supplied to the selected analog input channel is sampled using the sample & hold circuit.

<3> After sampling continues for a certain period of time, the sample & hold circuit is put on hold to keep the input analog voltage until A/D conversion is completed.

<4> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap at the tap selector is set to half of AV_{REF}.

<5> The series resistor string tap voltage is compared with the analog input voltage using the voltage comparator. If the analog input voltage is higher than half of AV_{REF}, the MSB of the SAR remains set. If it is lower than half of AV_{REF}, the MSB is reset.

<6> Bit 8 of the SAR is set automatically, and comparison shifts to the next stage. The next voltage tap of the series resistor string is selected according to bit 9, which reflects the previous comparison result, as follows:

- Bit 9 = 1: Three quarters of AV_{REF}
- Bit 9 = 0: One quarter of AV_{REF}

The tap voltage is compared with the analog input voltage. Bit 8 is set or reset according to the result of comparison.

- Analog input voltage ≥ tap voltage: Bit 8 = 1
- Analog input voltage < tap voltage: Bit 8 = 0

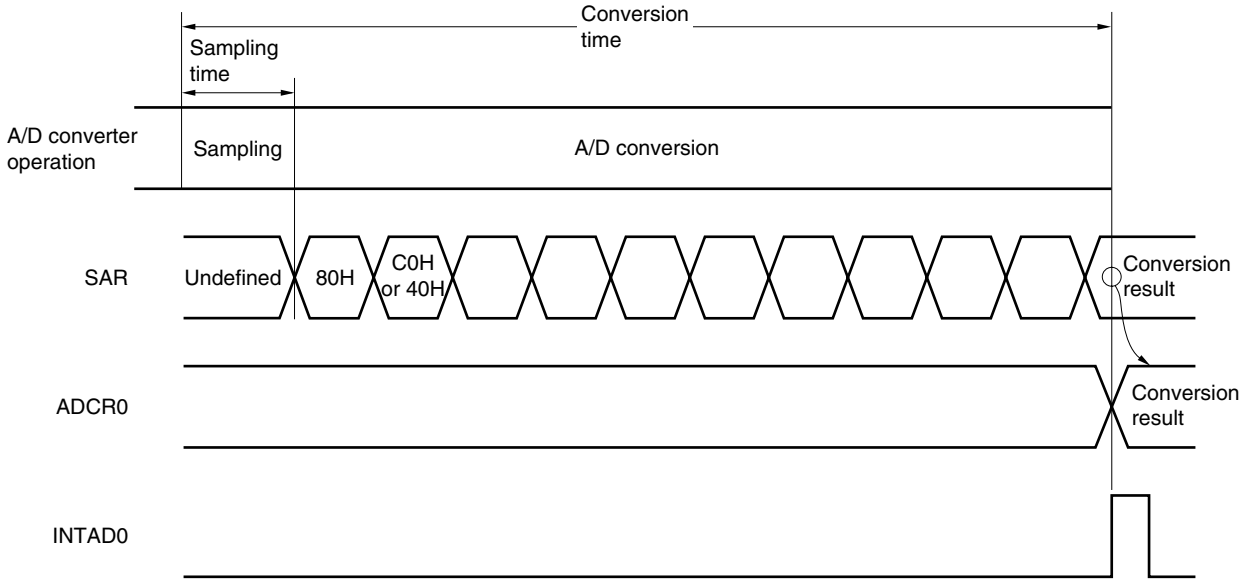
<7> Comparison is repeated until bit 0 of the SAR is reached.

<8> When comparison is completed for all of the 10 bits, a significant digital result is left in the SAR. This value is sent to and latched in A/D conversion result register 0 (ADCR0). At the same time, it is possible to generate an A/D conversion end interrupt request (INTAD0).

Cautions 1. The first A/D conversion value immediately following the start of A/D conversion may be undefined.

2. When the A/D converter enters the standby mode, it stops operating.

Figure 11-4. Basic Operation of 10-Bit A/D Converter



A/D conversion continues until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset (0) by software.

If an attempt is made to write to ADM0 or A/D input selection register 0 (ADS0) during A/D conversion, the current A/D conversion is canceled. In this case, A/D conversion is restarted from the beginning, if the ADCS0 bit is set (1).

$\overline{\text{RESET}}$ makes A/D conversion result register 0 (ADCR0) undefined.

11.4.2 Input voltage and conversion result

The relationship between the analog input voltage at the analog input pins (ANI0 to ANI6) and the A/D conversion result (A/D conversion result register 0 (ADCR0)) is represented by:

$$ADCR0 = INT\left(\frac{V_{IN}}{AV_{REF}} \times 1024 + 0.5\right)$$

or

$$(ADCR0 - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{IN} < (ADCR0 + 0.5) \times \frac{AV_{REF}}{1024}$$

INT( ): Function that returns the integer part of a parenthesized value

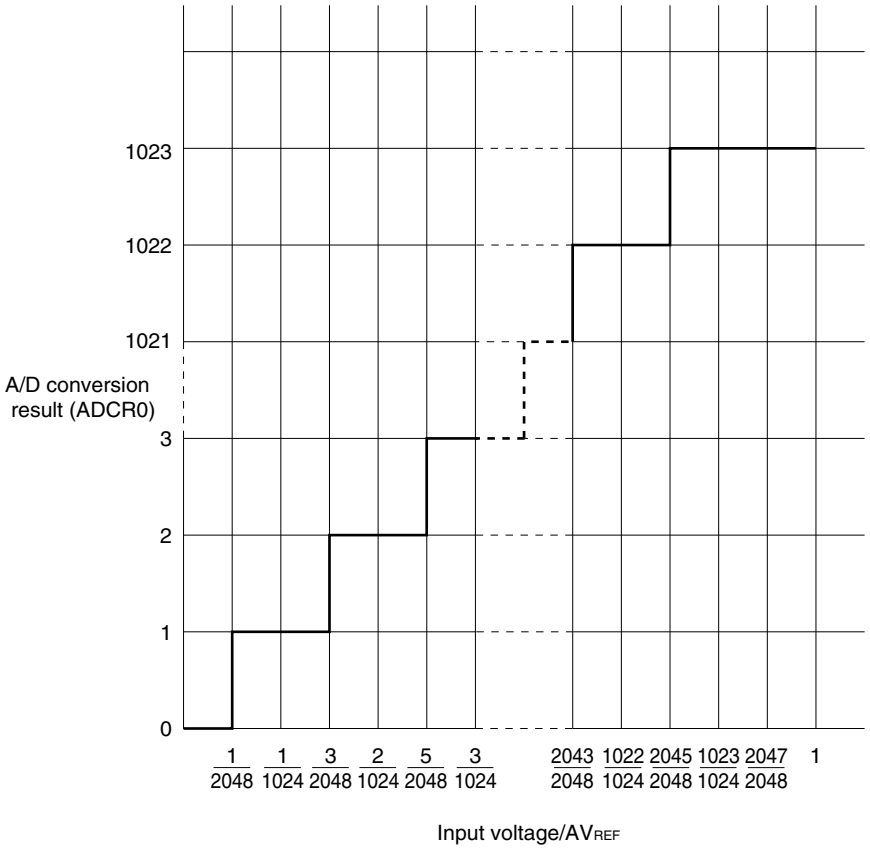
V_{IN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

ADCR0: Value in A/D conversion result register 0 (ADCR0)

Figure 11-5 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-5. Relationship Between Analog Input Voltage and A/D Conversion Result



11.4.3 Operation mode of 10-bit A/D converter

The 10-bit A/D converter is initially in the select mode. In this mode, A/D input selection register 0 (ADS0) is used to select an analog input channel from ANI0 to ANI6 for A/D conversion.

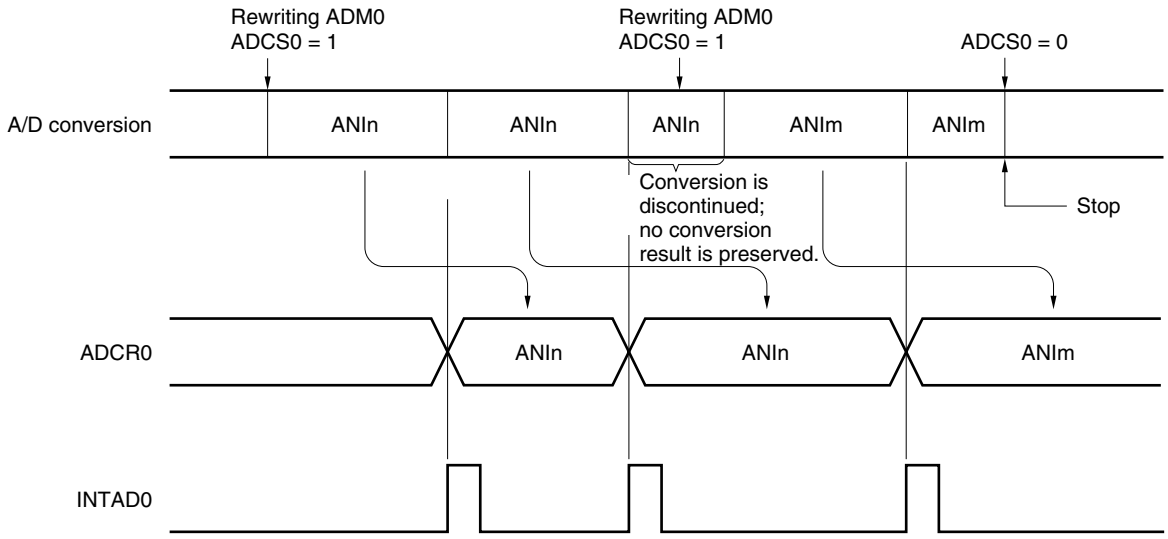
A/D conversion can be started only by software, that is, by setting A/D converter mode register 0 (ADM0).

The A/D conversion result is saved to A/D conversion result register 0 (ADCR0). At the same time, an interrupt request signal (INTAD0) is generated.

- **Software-started A/D conversion**

  Setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) triggers A/D conversion for a voltage applied to the analog input pin specified in A/D input selection register 0 (ADS0). Upon completion of A/D conversion, the conversion result is saved to A/D conversion result register 0 (ADCR0). At the same time, an interrupt request signal (INTAD0) is generated. Once A/D conversion is activated, and completed, another session of A/D conversion is started. A/D conversion is repeated until new data is written to ADM0. If data where the ADCS0 bit is 1 is written to ADM0 again during A/D conversion, the current session of A/D conversion is discontinued, and a new session of A/D conversion begins for the new data. If data where the ADCS0 bit is 0 is written to ADM0 again during A/D conversion, A/D conversion is stopped immediately.

Figure 11-6. Software-Started A/D Conversion



Remarks 1. n = 0, 1, ..., 6

2. m = 0, 1, ..., 6

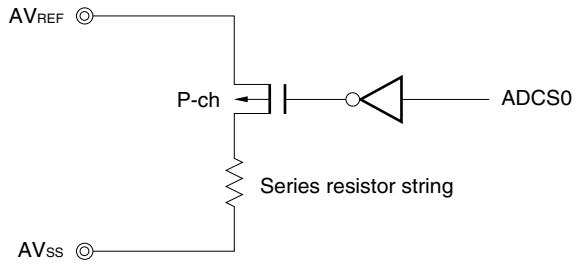
11.5 Cautions on Using 10-Bit A/D Converter

(1) Current consumption in the standby mode

When the A/D converter enters the standby mode, it stops operating. Setting the bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) = 0 can reduce the current consumption.

Figure 11-7 shows how to reduce the current consumption in the standby mode.

Figure 11-7. How to Reduce Current Consumption in Standby Mode



(2) Input range for the ANI0 to ANI6 pins

Be sure to keep the input voltage at ANI0 to ANI6 within the rated range. If a voltage greater than AV_{REF} or less than AV_{SS} (even within the absolute maximum rating) is input a conversion channel, the conversion output of the channel becomes undefined, and the conversion output of the other channels may be affected.

(3) Conflict

<1> Conflict between writing to A/D conversion result register 0 (ADCR0) at the end of conversion and reading from the ADCR0 bit

Reading from the ADCR0 bit takes precedence. After reading, the new conversion result is written to ADCR0 bit.

<2> Conflict between writing to the ADCR0 bit at the end of conversion and writing to A/D converter mode register 0 (ADM0) or A/D input selection register 0 (ADS0)

Writing to ADM0 or ADS0 takes precedence. A request to write to the ADCR0 bit is ignored. No A/D conversion end interrupt request signal (INTAD0) is generated.

(4) Conversion results immediately following start of A/D conversion

The first A/D conversion value immediately following the start of A/D conversion may be undefined. Be sure to poll the A/D conversion end interrupt request (INTAD0) and perform processing such as discarding the first conversion result.

(5) Timing that makes the A/D conversion result undefined

If the timing of the end of A/D conversion and the timing of the stop of operation of the A/D converter conflict, the A/D conversion value may be undefined. Because of this, be sure to read out the A/D conversion result while the A/D converter is in operation. Furthermore, when reading out an A/D conversion result after A/D conversion has stopped, be sure to have done so by the time the next conversion result is complete.

The conversion result readout timing is shown in Figures 11-8 and 11-9.

Figure 11-8. Conversion Result Readout Timing (When Conversion Result Is Undefined Value)

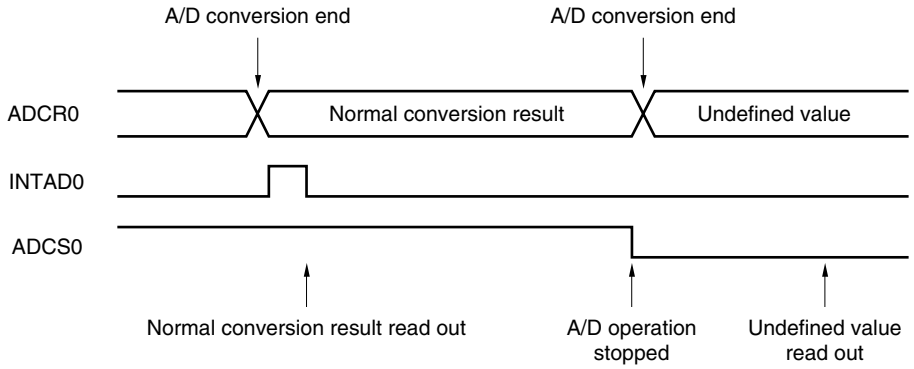
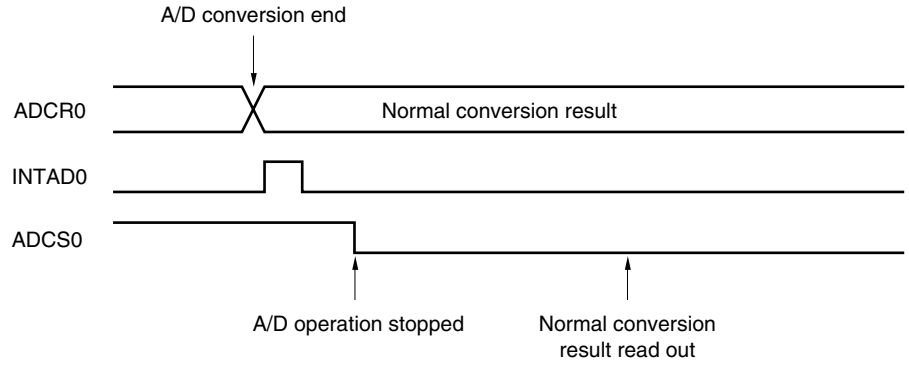


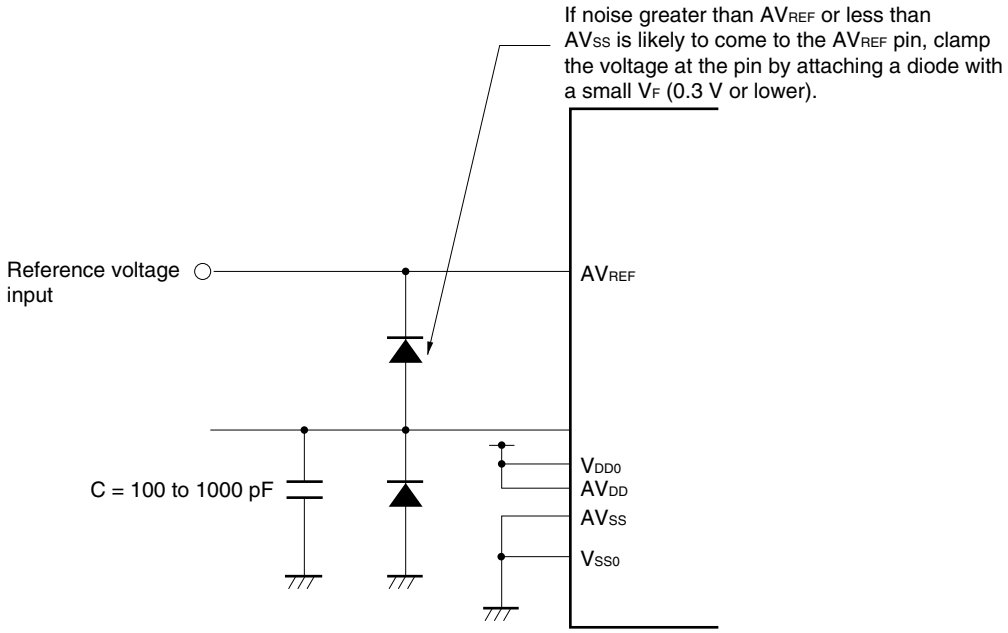
Figure 11-9. Conversion Result Readout Timing (When Conversion Result Is Normal Value)



(6) Noise elimination

To maintain a resolution of 10 bits, it is necessary to avoid for noise at the AV_{REF} and ANI0 to ANI6 pins. The higher the output impedance of the analog input source, the larger the effect by noise. To eliminate noise, attach an external capacitor to the relevant pins as shown in Figure 11-10.

Figure 11-10. Analog Input Pin Processing



(7) ANI0 to ANI6

The analog input pins (ANI0 to ANI6) are alternate-function pins. They are also used as port pins (P60 to P66).

If any of ANI0 to ANI6 has been selected for A/D conversion, do not execute input instructions for the ports. Otherwise, the conversion resolution may become lower.

If a digital pulse is applied to a pin adjacent to the analog input pins during A/D conversion, coupling noise may occur which prevents an A/D conversion result from being attained as expected. Avoid applying a digital pulse to pins adjacent to the analog input pins during A/D conversion.

★ **(8) Input impedance of ANI0 to ANI6 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 ANI6 pins (see **Figure 11-10**).

(9) Input impedance of the AVREF pin

A series resistor string of 10 kΩ is connected across the AVREF and AVSS pins.

If the output impedance of the reference voltage source is high, this high impedance is eventually connected in parallel with the series resistor string across the AVREF and AVSS pins, leading to a higher reference voltage error.

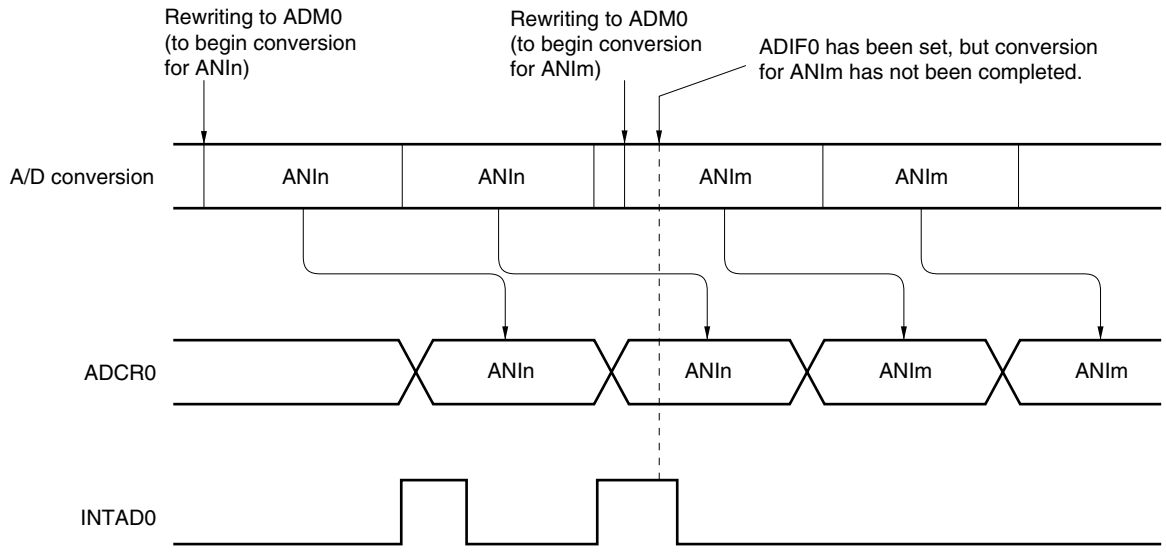
(10) Interrupt request flag (ADIF0)

Changing the contents of A/D converter mode register 0 (ADM0) does not clear the interrupt request flag (ADIF0).

If the voltage at the analog input pins is changed during A/D conversion, therefore, the A/D conversion result and the conversion end interrupt request flag may reflect the previous analog input just before writing to ADM0. In this case, the ADIF0 may appear to be set if it is read-accessed just after ADM0 is write-accessed, even when A/D conversion has not been completed for the new analog input.

In addition, ADIF0 must be cleared before A/D conversion is restarted.

Figure 11-11. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 6

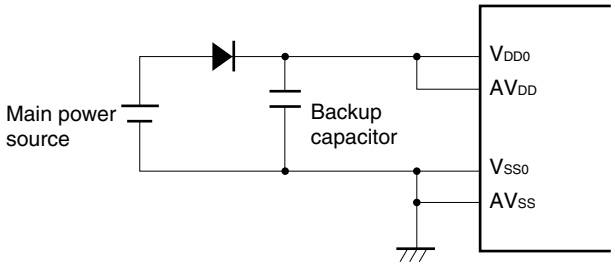
2. m = 0, 1, ..., 6

(11) AV_{DD} pin

The AV_{DD} pin is used to supply power to the analog circuit. It is also used to supply power to the ANI0 to ANI6 input circuit.

If your application is designed to be changed to backup power, the AV_{DD} pin must be supplied with the same voltage level as for the V_{DD0} pin, as shown in Figure 11-12.

Figure 11-12. AV_{DD} Pin Processing



CHAPTER 12 COMPARATOR

12.1 Functions of Comparator

The comparator has the following functions.

(1) Input voltage comparison by comparator

The comparator compares an input voltage at the reference voltage input pin (CMPREF0) with an input voltage at the comparator input pin (CMPIN0). The comparison result can be read using memory manipulation instructions.

(2) Interrupt generation by comparator output

The comparator output is used to generate an interrupt request signal[Note] (INTCMP0).

Note The rising edge, falling edge, or both rising and falling edges can be specified by setting external interrupt mode register 1 (INTM1).

(3) Clock output

When CMPREF0 > CMPIN0, the output of 8-bit timer counter 02 (TM02) is directed to the CMPTOUT0 pin.

(4) Open-drain output selection

Comparator mode register 0 (CMPRM0) is used to specify a port as an N-ch open-drain output.

12.2 Configuration of Comparator

The comparator consists of the following hardware.

(1) CMPIN0

This is the comparator input pin.

(2) CMPTOUT0

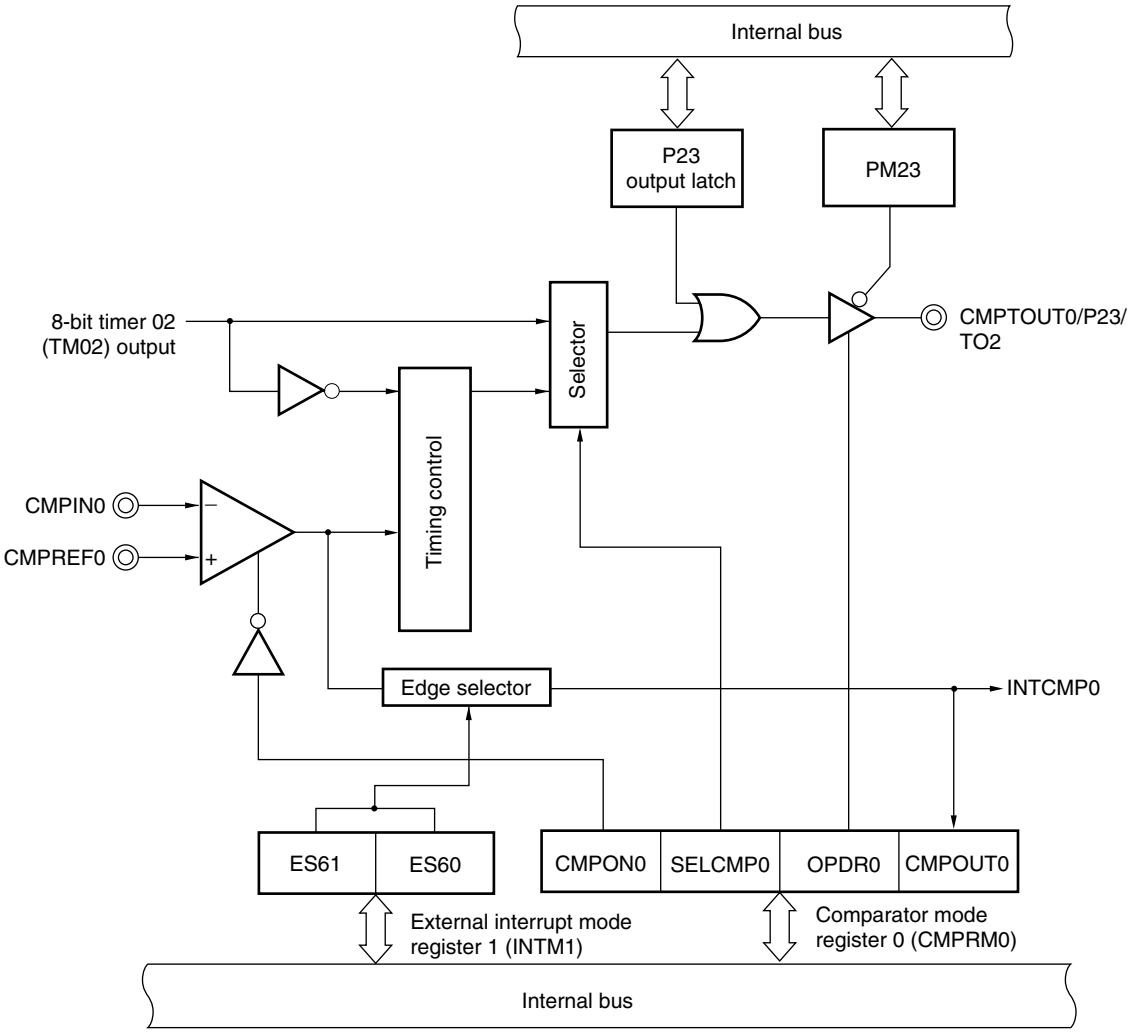
This is the comparator output pin.

(3) CMPREF0

This is the comparator reference voltage input pin.

Figure 12-1 is a block diagram of the comparator.

Figure 12-1. Block Diagram of Comparator



12.3 Register Controlling Comparator

The comparator is controlled by the following register.

(1) Comparator mode register 0 (CMPRM0)

CMPRM0 controls the power supply and clock output of the comparator. It also selects an open-drain output for the comparator.

CMPRM0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CMPRM0 to 00H.

Figure 12-2. Format of Comparator Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CMPRM0	0	0	0	0	CMPON0	SELCMP0	OPDR0	CMPOUT0	FF4EH	00H	R/W[Note]

CMPON0	Comparator power supply on/off control
0	Comparator power supply off
1	Comparator power supply on

SELCMP0	Clock output control
0	8-bit timer 02 (TM02) output
1	8-bit timer counter 02 (TM02) output if CMPREF0 > CMPIN0

OPDR0	Open-drain output selection
0	CMOS output
1	N-ch open-drain output

CMPOUT0	The comparator output is read.

Note Bit 0 is read-only.

Cautions 1. Bits 4 to 7 must be fixed to 0.

2. If the comparator is enabled (CMPON0 = 1), noise may be induced. If it is necessary to generate an interrupt request signal (INTCMP0) from the output of the comparator, enable the comparator (CMPON0 = 1), then clear the interrupt request flag (CMPIF0) to 0, before enabling interrupts.

3. Similarly, if it is necessary to direct the output of the comparator to the port, enable the comparator (CMPON0 = 1) in advance.

12.4 Operation of Comparator

The output of 8-bit timer 02 (TM02) can be controlled and directed to the CMPTOUT0/P23/TO2 pin via the comparator.

To run the comparator, set as follows:

- Set P23 to output mode (PM23 = 0).
- Set comparator mode register 0 (CMPRM0) as shown in Figure 12-3.
- Set external interrupt mode register 1 (INTM1) as shown in Figure 12-4 and select the valid edge of INTCMP0.

Figure 12-3. Settings of Comparator Mode Register 0 for Comparator Operation

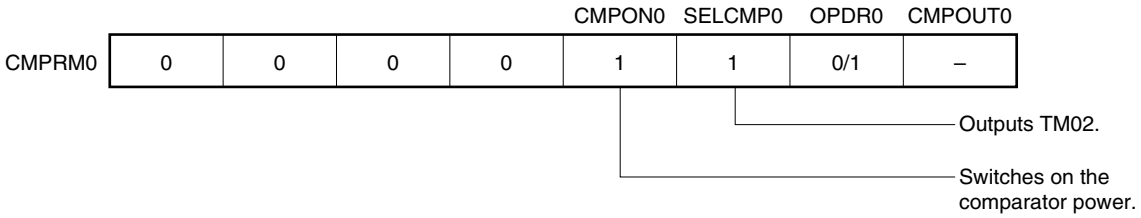


Figure 12-4. Settings of External Interrupt Mode Register 1 at INTCMP0 Occurrence

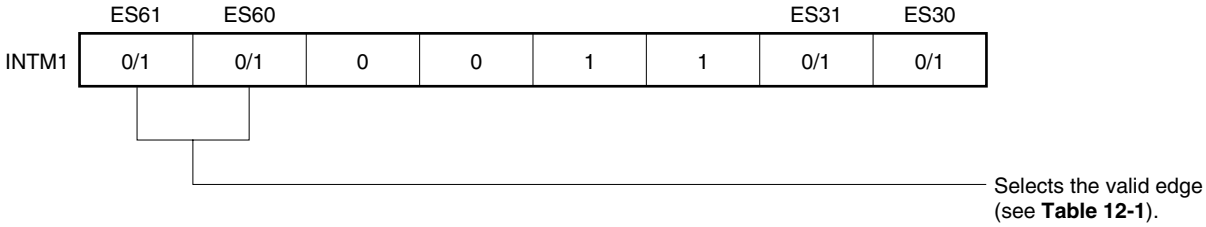
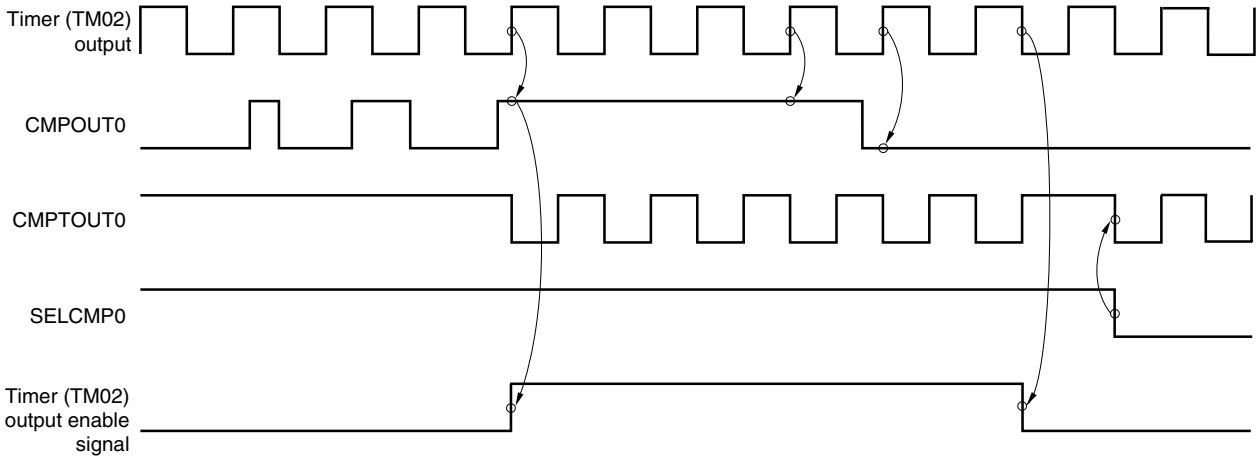


Table 12-1 lists the selection of INTCMP0 valid edges, and Figure 12-5 shows the timing chart of the comparator.

Table 12-1. INTCMP0 Valid Edges

ES61	ES60	INTCMP0 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

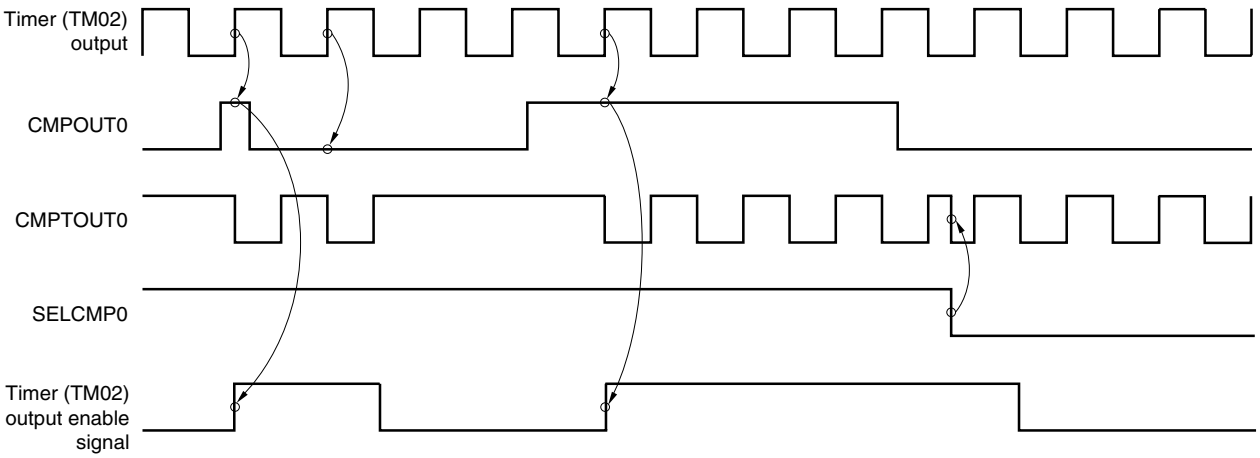
Figure 12-5. Comparator Operation Timing (1/2)



<1> CMPOUT0 is latched on the rising edge of the TM02 output to generate a signal to enable output to the CMPTOUT0/P23/TO2 pin. If CMPOUT0 is high, the TM02 output waveform is output to the CMPTOUT0/P23/TO2 pin on the rising edge of the TM02 output. If CMPOUT0 is low, CMPTOUT0 is not output.

<2> If SELCMP0 is low, the TM02 output is sent to the CMPTOUT0/P23/TO2 pin no matter which level CMPOUT0 is on.

Figure 12-5. Comparator Operation Timing (2/2)



<3> If the high level of CMPOUT0 is latched on the rising edge of the TM02 output, CMPTOUT0 is output to the CMPTOUT0/P23/TO2 pin for at least two clock pulses even if it falls immediately.

<4> Switching SELCMP0 from high to low during CMPTOUT0 output may disturb the output waveform of CMPTOUT0.

CHAPTER 13 SERIAL INTERFACE 00

13.1 Functions of Serial Interface 00

Serial interface 00 has the following three modes.

- Operation stopped mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

(1) Operation stopped mode

This mode is used to reduce power consumption when serial transfer is not carried out.

(2) Asynchronous serial interface (UART) mode

In this mode, one byte of data following the start bit is transmitted/received, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by dividing the clock input to the ASCK pin.

(3) 3-wire serial I/O mode (MSB/LSB start bit switchable)

In this mode, 8-bit data transfer is carried out using three lines, one for the serial clock ($\overline{\text{SCK}}$) and two for serial data (SI, SO).

The 3-wire serial I/O mode supports simultaneous transmit and receive operations, reducing data transfer processing time.

It is possible to switch the start bit of 8-bit data to be transmitted between the MSB and the LSB, thus allowing connection to devices with either start bit.

The 3-wire serial I/O mode is effective for connecting display controllers and peripheral I/Os such as the 75XL Series, 78K Series, and 17K Series, which have conventional clock synchronous serial interfaces.

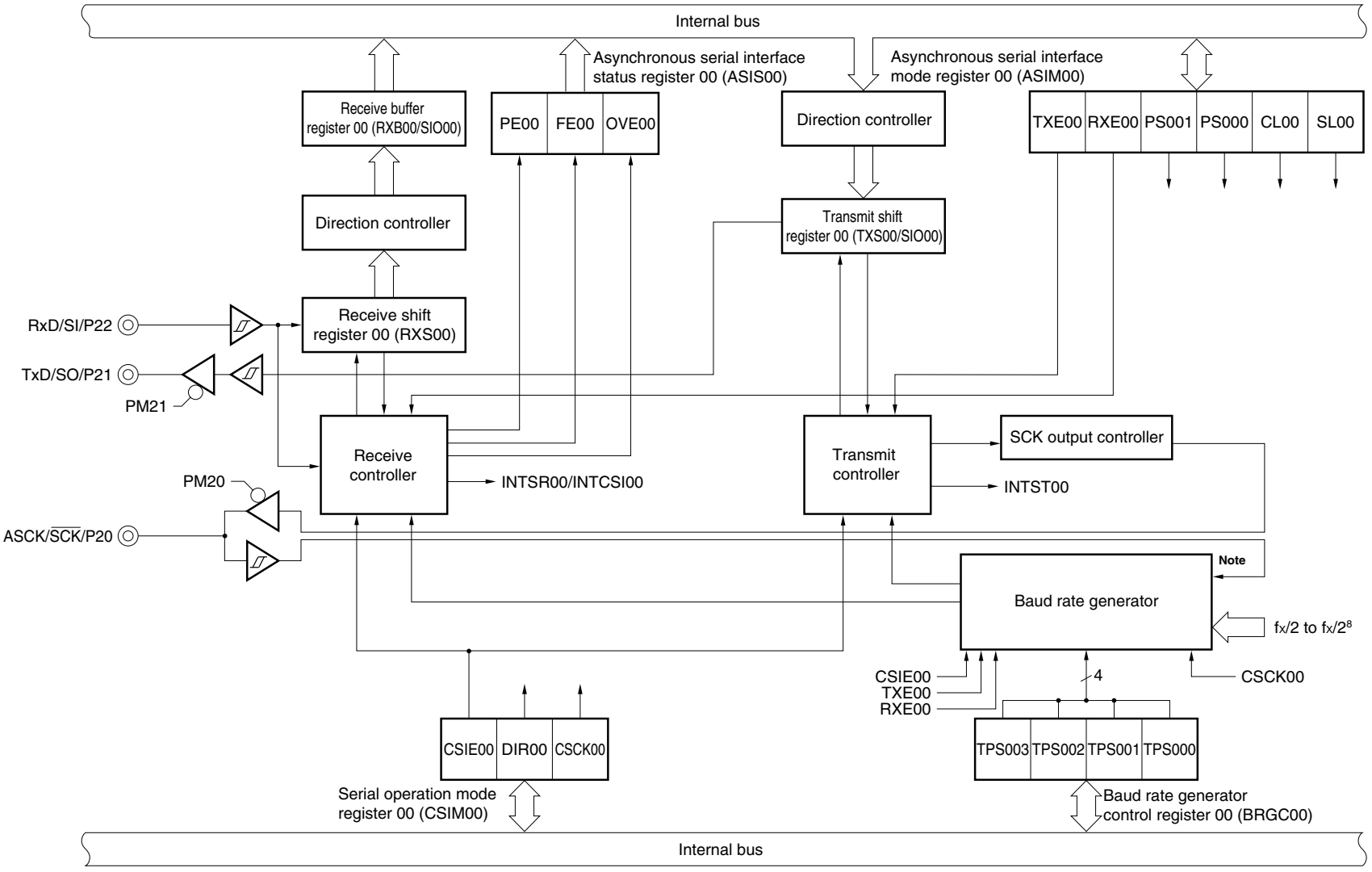
13.2 Configuration of Serial Interface 00

Serial interface 00 consists of the following hardware.

Table 13-1. Configuration of Serial Interface 00

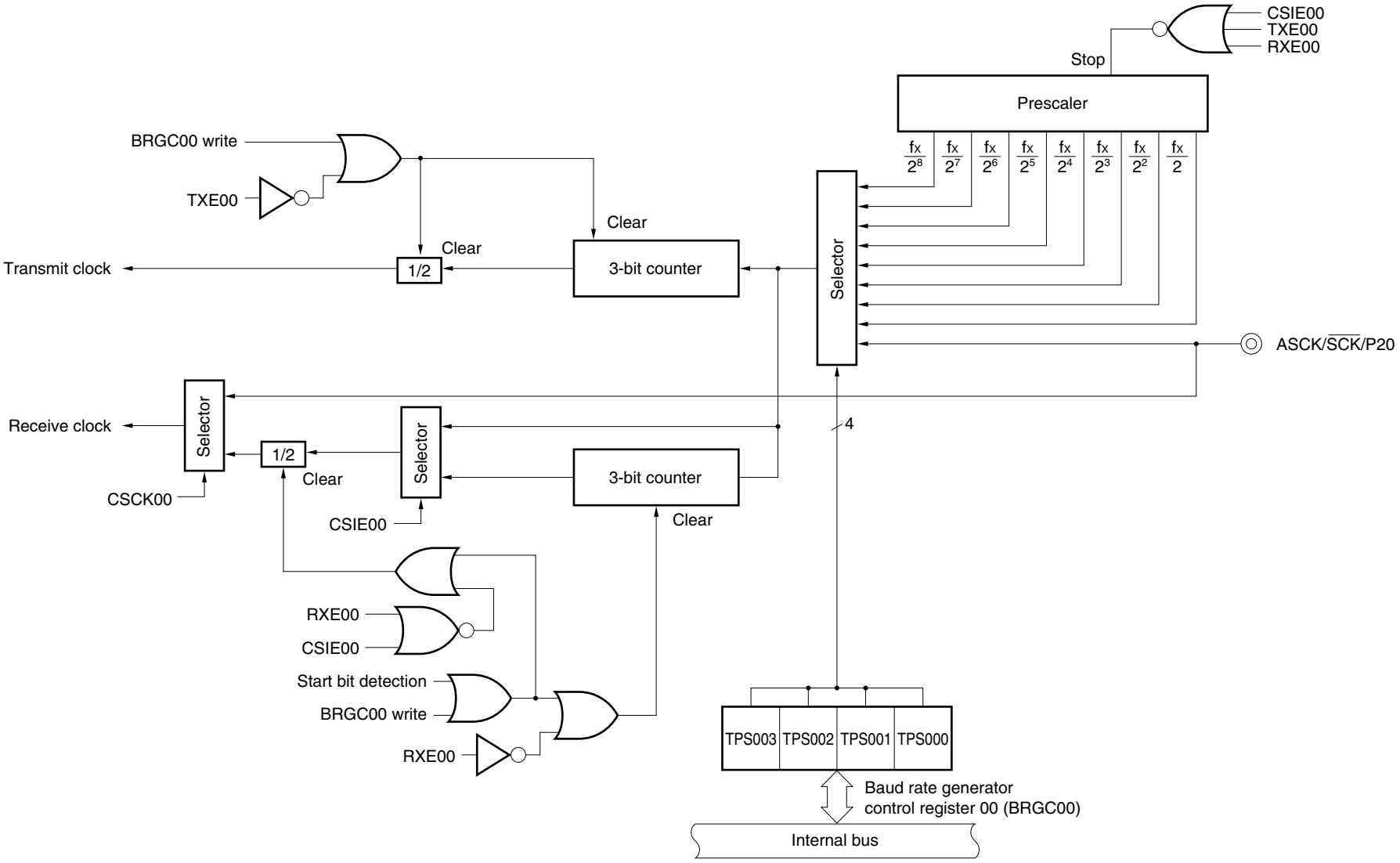
Item	Configuration
Registers	Transmit shift register 00 (TXS00) Receive shift register 00 (RXS00) Receive buffer register 00 (RXB00)
Control registers	Serial operation mode register 00 (CSIM00) Asynchronous serial interface mode register 00 (ASIM00) Asynchronous serial interface status register 00 (ASIS00) Baud rate generator control register 00 (BRGC00)

Figure 13-1. Block Diagram of Serial Interface 00



Note For the baud rate generator configuration, see Figure 13-2.

Figure 13-2. Block Diagram of Baud Rate Generator



(1) Transmit shift register 00 (TXS00)

This register is used to specify data to be transmitted. Data written to TXS00 is transmitted as serial data.

If the data length is specified as 7 bits, bits 0 to 6 of the data written to TXS00 are transferred as the transmit data. The transmit operation is started by writing data to TXS00.

TXS00 is written to using an 8-bit memory manipulation instruction. It cannot be read.

$\overline{\text{RESET}}$ input sets TXS00 to FFH.

Caution Do not write to TXS00 during a transmit operation.
TXS00 and receive buffer register 00 (RXB00) are allocated to the same address, and when reading is performed, RXB00 values are read.

(2) Receive shift register 00 (RXS00)

This register is used to convert serial data input to the RxD pin into parallel data. Each time one byte of data is received, it is transferred to receive buffer register 00 (RXB00).

RXS00 cannot be manipulated directly by program.

(3) Receive buffer register 00 (RXB00)

This register is used to hold received data. Each time one byte of data is received, a new byte of data is transferred from receive shift register 00 (RXS00).

If the data length is specified as 7 bits, receive data is transferred to bits 0 to 6 of RXB00, and the MSB of RXB00 always becomes 0.

RXB00 can be read using an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input makes RXB00 undefined.

Caution RXB00 and transmit shift register 00 (TXS00) are allocated to the same address, and when writing is performed, the values are written to TXS00.

(4) Transmit controller

This circuit controls transmit operations by adding a start bit, parity bit, and stop bit to data written to transmit shift register 00 (TXS00), according to the data set to asynchronous serial interface mode register 00 (ASIM00).

(5) Receive controller

This circuit controls receive operations according to the data set to asynchronous serial interface mode register 00 (ASIM00). It also performs a parity error check, etc., during receive operations, and when an error is detected, it sets a value to asynchronous serial interface status register 00 (ASIS00) in accordance with the nature of the error.

13.3 Registers Controlling Serial Interface 00

The following four registers are used to control serial interface 00.

- Serial operation mode register 00 (CSIM00)
- Asynchronous serial interface mode register 00 (ASIM00)
- Asynchronous serial interface status register 00 (ASIS00)
- Baud rate generator control register 00 (BRGC00)

(1) Serial operation mode register 00 (CSIM00)

This register is set when using serial interface 00 in the 3-wire serial I/O mode.

CSIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM00 to 00H.

Figure 13-3. Format of Serial Operation Mode Register 00

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM00	CSIE00	0	0	0	0	DIR00	CSCK00	0	FF72H	00H	R/W

CSIE00	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

DIR00	Start bit specification
0	MSB
1	LSB

CSCK00	Clock selection in 3-wire serial I/O mode
0	Clock input to $\overline{\text{SCK}}$ pin from external
1	Dedicated baud rate generator output

Cautions 1. Bits 0 and 3 to 6 must be fixed to 0.

2. Set CSIM00 to 00H in the UART mode.

(2) Asynchronous serial interface mode register 00 (ASIM00)

This register is set when using serial interface 00 in the asynchronous serial interface mode.

ASIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM00 to 00H.

Figure 13-4. Format of Asynchronous Serial Interface Mode Register 00

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	0	0	FF70H	00H	R/W

TXE00	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE00	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS001	PS000	Parity bit specification
0	0	No parity
0	1	0 parity always added at transmission Parity check is not performed at reception (no parity error occurs)
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

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. Set ASIM00 to 00H in the 3-wire serial I/O mode.

3. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

Table 13-2. Operation Mode Settings of Serial Interface 00

(1) Operation stopped mode

ASIM00		CSIM00			PM22	P22	PM21	P21	PM20	P20	Start Bit	Shift Clock	P22/SI/RxD Pin Function	P21/SO/TxD Pin Function	P20/$\overline{SCK}$/ASCK Pin Function
TXE00	RXE00	CSIE00	DIR00	CSCK00											
0	0	0	x	x	x Note 1	x Note 1	x Note 1	x Note 1	x Note 1	x Note 1	—	—	P22	P21	P20
Other than above											Setting prohibited				

(2) Asynchronous serial interface mode

ASIM00		CSIM00			PM22	P22	PM21	P21	PM20	P20	Start Bit	Shift Clock	P22/SI/RxD Pin Function	P21/SO/TxD Pin Function	P20/$\overline{SCK}$/ASCK Pin Function
TXE00	RXE00	CSIE00	DIR00	CSCK00											
1	0	0	0	0	x Note 1	x Note 1	0	1	1	x	LSB	External clock	P22	TxD (CMOS output)	ASCK input
									x Note 1	x Note 1		Internal clock			P20
0	1	0	0	0	1	x	x Note 1	x Note 1	1	x		External clock	RxD	P21	ASCK input
									x Note 1	x Note 1		Internal clock			P20
1	1	0	0	0	1	x	0	1	1	x		External clock		TxD (CMOS output)	ASCK input
									x Note 1	x Note 1		Internal clock			P20
Other than above											Setting prohibited				

(3) 3-wire serial I/O mode

ASIM00		CSIM00			PM22	P22	PM21	P21	PM20	P20	Start Bit	Shift Clock	P22/SI/RxD Pin Function	P21/SO/TxD Pin Function	P20/$\overline{SCK}$/ASCK Pin Function
TXE00	RXE00	CSIE00	DIR00	CSCK00											
0	0	1	0	0	1 Note 2	x Note 2	0	1	1	x	MSB	External clock	SI Note 2	SO (CMOS output)	$\overline{SCK}$ input
				1					0	1		Internal clock			$\overline{SCK}$ output
		1	1	0					1	x	LSB	External clock			$\overline{SCK}$ input
				1					0	1		Internal clock			$\overline{SCK}$ output
Other than above											Setting prohibited				

Notes 1. Can be used as port function.

2. If used only for transmission, can be used as P22 (CMOS I/O).

Remark x: Don't care

(3) Asynchronous serial interface status register 00 (ASIS00)

This register indicates the type of error when a reception error occurs in the asynchronous serial interface mode.

ASIS00 is read using a 1-bit or 8-bit memory manipulation instruction.

The contents of ASIS00 become undefined in the 3-wire serial I/O mode.

$\overline{\text{RESET}}$ input sets ASIS00 to 00H.

Figure 13-5. Format of Asynchronous Serial Interface Status Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS00	0	0	0	0	0	PE00	FE00	OVE00	FF71H	00H	R

PE00	Parity error flag
0	Parity error did not occur
1	Parity error occurred (when the transmit parity and receive parity did not match)

FE00	Framing error flag
0	Framing error did not occur
1	Framing error occurred (when stop bit was not detected)[Note 1]

OVE00	Overrun error flag
0	Overrun error did not occur
1	Overrun error occurred[Note 2] (when the next receive operation was completed before the data was read from receive buffer register 00)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL00) of asynchronous serial interface mode register 00 (ASIM00), only one stop bit is detected during reception.

2. Be sure to read receive buffer register 00 (RXB00) when an overrun error occurs. If not, an overrun error will occur every time the data is received.

(4) Baud rate generator control register 00 (BRGC00)

This register is used to set the serial clock of serial interface 00.

BRGC00 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC00 to 00H.

Figure 13-6. Format of Baud Rate Generator Control Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC00	TPS003	TPS002	TPS001	TPS000	0	0	0	0	FF73H	00H	R/W

TPS003	TPS002	TPS001	TPS000	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
1	0	0	0	Clock input from external to ASCK pin[Note]	–
Other than above				Setting prohibited	

Note Only used in the UART mode.

Cautions 1. When BRGC00 is written during a communication operation, the output of the baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during a communication operation.

2. Do not select n = 1 during f_X = 5.0 MHz operation because the baud rate rating is exceeded.

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Value determined in the settings of TPS000 to TPS003 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

The baud rate transmit/receive clock to be generated is either a signal divided from the main system clock, or a signal divided from the clock input from the ASCK pin.

(a) Generation of baud rate transmit/receive clock from main system clock

The transmit/receive clock is generated by dividing the main system clock. The baud rate generated from the main system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} [\text{Hz}]$$

fx: Main system clock oscillation frequency

n: Value in Figure 13-6 that is determined in the settings of TPS000 to TPS003 ($2 \leq n \leq 8$)

Table 13-3. Example of Relationship Between Main System Clock and Baud Rate

Baud Rate (bps)	BRGC00 Set Value	Error (%)	
		fx = 5.0 MHz	fx = 4.9152 MHz
1200	70H	1.73	0
2400	60H		
4800	50H		
9600	40H		
19200	30H		
38400	20H		
76800	10H		

(b) Generation of baud rate transmit/receive clock from external clock of ASCK pin

The transmit/receive clock is generated by dividing the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16}[\text{Hz}]$$

f_{ASCK}: Frequency of clock input to the ASCK pin

Table 13-4. Relationship Between ASCK Pin Input Frequency and Baud Rate (When BRGC00 Is Set to 80H)

Baud Rate (bps)	ASCK Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1200	19.2
2400	38.4
4800	76.8
9600	153.6
19200	307.2
31250	500.0
38400	614.4

13.4 Operation of Serial Interface 00

Serial interface 00 has the following three modes.

- Operation stopped mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

13.4.1 Operation stopped mode

Serial transfer is not executed in the operation stopped mode, therefore the power consumption can be reduced. The P20/$\overline{SCK}$/ASCK, P21/SO/TxD, and P22/SI/RxD pins can be used as normal I/O port pins.

(1) Register setting

Operation stopped mode is set by serial operation mode register 00 (CSIM00) and asynchronous serial interface mode register 00 (ASIM00).

(a) Serial operation mode register 00 (CSIM00)

CSIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM00 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM00	CSIE00	0	0	0	0	DIR00	CSCK00	0	FF72H	00H	R/W

CSIE00	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

Caution Bits 0 and 3 to 6 must be fixed to 0.

(b) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM00 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	0	0	FF70H	00H	R/W

TXE00	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE00	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

Caution Bits 0 and 1 must be fixed to 0.

13.4.2 Asynchronous serial interface (UART) mode

In this mode, the one-byte data following the start bit is transmitted/received and thus full-duplex communications are possible.

This device incorporates a UART-dedicated baud rate generator, enabling communication at the desired baud rate. In addition, the baud rate can also be defined by dividing the clock input to the ASCK pin.

The UART-dedicated baud rate generator can also output a 31.25 kbps baud rate, which complies with the MIDI standard.

(1) Register setting

UART mode is set by serial operation mode register 00 (CSIM00), asynchronous serial interface mode register 00 (ASIM00), asynchronous serial interface status register 00 (ASIS00), and baud rate generator control register 00 (BRGC00).

(a) Serial operation mode register 00 (CSIM00)

CSIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM00 to 00H.

Set CSIM00 to 00H in the UART mode.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM00	CSIE00	0	0	0	0	DIR00	CSCK00	0	FF72H	00H	R/W

CSIE00	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

DIR00	Start bit specification
0	MSB
1	LSB

CSCK00	Clock selection in 3-wire serial I/O mode
0	Clock input to $\overline{\text{SCK}}$ pin from external
1	Dedicated baud rate generator output

Caution Bits 0 and 3 to 6 must be fixed to 0.

(b) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM00 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	0	0	FF70H	00H	R/W

TXE00	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE00	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS001	PS000	Parity bit specification
0	0	No parity
0	1	0 parity always added at transmission Parity check is not performed at reception (no parity error occurs)
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

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

(c) Asynchronous serial interface status register 00 (ASIS00)

ASIS00 is read using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS00 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS00	0	0	0	0	0	PE00	FE00	OVE00	FF71H	00H	R

PE00	Parity error flag
0	Parity error did not occur
1	Parity error occurred (when the transmit parity and receive parity did not match)

FE00	Framing error flag
0	Framing error did not occur
1	Framing error occurred (when stop bit was not detected)Note 1

OVE00	Overrun error flag
0	Overrun error did not occur
1	Overrun error occurredNote 2 (when the next receive operation was completed before the data was read from receive buffer register 00)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL00) of asynchronous serial interface mode register 00 (ASIM00), only one stop bit will be detected during reception.

2. Be sure to read receive buffer register 00 (RXB00) when an overrun error occurs. If not, every time the data is received an overrun error occurs.

(d) Baud rate generator control register 00 (BRGC00)

BRGC00 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC00 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC00	TPS003	TPS002	TPS001	TPS000	0	0	0	0	FF73H	00H	R/W

TPS003	TPS002	TPS001	TPS000	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
1	0	0	0	Clock input from external to ASCK pin	–
Other than above				Setting prohibited	

Cautions 1. When BRGC00 is written during a communication operation, the output of the baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during a communication operation.

2. Do not select n = 1 during f_X = 5.0 MHz operation because the baud rate rating is exceeded.

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Value determined in the settings of TPS000 to TPS003 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

The baud rate transmit/receive clock to be generated is either a signal divided from the main system clock, or a signal divided from the clock input from the ASCK pin.

(i) Generation of baud rate transmit/receive clock from main system clock

The transmit/receive clock is generated by dividing the main system clock. The baud rate generated from the main system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} [\text{Hz}]$$

f_X: Main system clock oscillation frequency

n: Value in the above table that is determined in the settings of TPS000 to TPS003 ($2 \leq n \leq 8$)

Table 13-5. Example of Relationship Between Main System Clock and Baud Rate

Baud Rate (bps)	BRGC00 Set Value	Error (%)	
		f_X = 5.0 MHz	f_X = 4.9152 MHz
1200	70H	1.73	0
2400	60H		
4800	50H		
9600	40H		
19200	30H		
38400	20H		
76800	10H		

(ii) Generation of baud rate transmit/receive clock from external clock of ASCK pin

The transmit/receive clock is generated by dividing the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16}[\text{Hz}]$$

f_{ASCK}: Frequency of clock input to the ASCK pin

Table 13-6. Relationship Between ASCK Pin Input Frequency and Baud Rate (When BRGC00 Is Set to 80H)

Baud Rate (bps)	ASCK Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1200	19.2
2400	38.4
4800	76.8
9600	153.6
19200	307.2
31250	500.0
38400	614.4

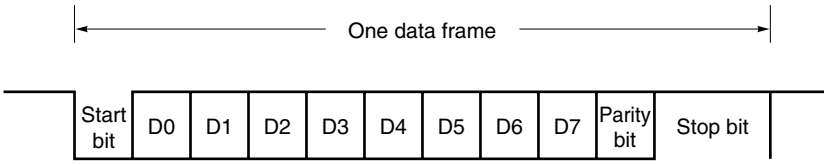
(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 13-7. One data frame consists of a start bit, character bits, parity bit and stop bit(s).

The specification of character bit length, parity selection, and specification of stop bit length for each data frame is carried out using asynchronous serial interface mode register 00 (ASIM00).

Figure 13-7. Format of Asynchronous Serial Interface Transmit/Receive Data



- Start bit ..... 1 bit
- Character bits..... 7 bits/8 bits
- Parity bits ..... Even parity/odd parity/0 parity/no parity
- Stop bit(s)..... 1 bit/2 bits

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; the most significant bit (bit 7) is ignored in transmission, and the most significant bit (bit 7) is always 0 in reception.

The serial transfer rate is selected using ASIM00 and baud rate generator control register 00 (BRGC00).

If a serial data receive error occurs, the receive error contents can be determined by reading the status of asynchronous serial interface status register 00 (ASIS00).

(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 “1” bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **At transmission**

  The transmission operation is controlled so that the number of bits with a value of 1 in the transmit data including the parity bit may be even. The parity bit value should be as follows.

  The number of bits with a value of 1 is an odd number in transmit data: 1
  The number of bits with a value of 1 is an even number in transmit data: 0

- **At reception**

  The number of bits with a value of 1 in the receive data including the parity bit is counted, and if the number is odd, a parity error occurs.

(ii) Odd parity

- **At transmission**

  Conversely to even parity, the transmission operation is controlled so that the number of bits with a value of 1 in the transmit data including the parity bit may be odd. The parity bit value should be as follows.

  The number of bits with a value of 1 is an odd number in transmit data: 0
  The number of bits with a value of 1 is an even number in transmit data: 1

- **At reception**

  The number of bits with a value of 1 in the receive data including the parity bit is counted, and if the number 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 does not occur, irrespective of whether the parity bit is set to 0 or 1.

(iv) No parity

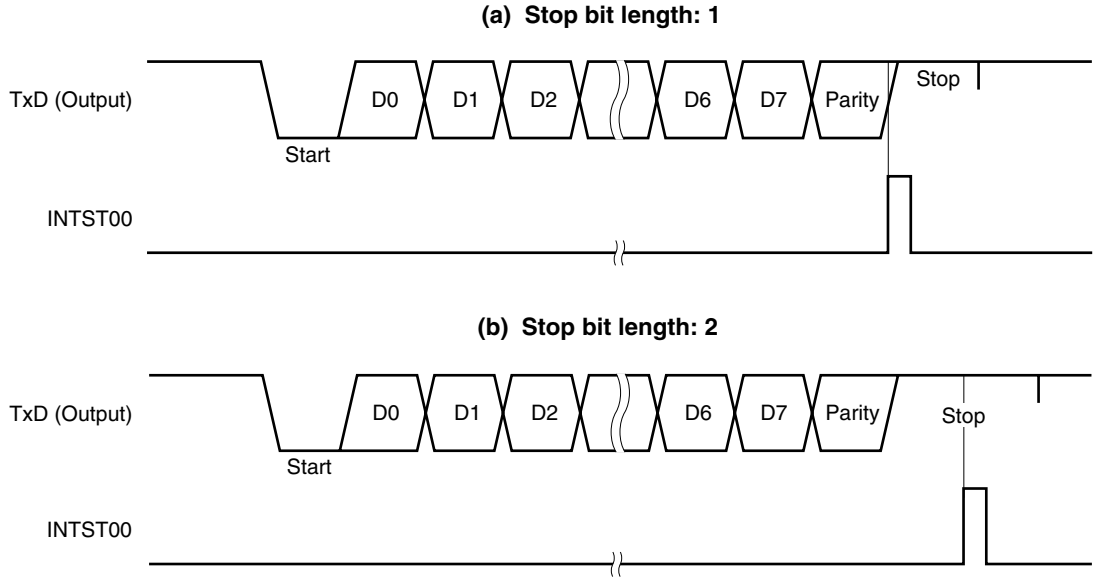
A parity bit is not added to the transmit data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error does not occur.

(c) Transmission

A transmit operation is started by writing transmit data to transmit shift register 00 (TXS00). The start bit, parity bit and stop bit(s) are added automatically.

When the transmit operation starts, the data in TXS00 is shifted out, and when TXS00 is empty, a transmission completion interrupt (INTST00) is generated.

Figure 13-8. Asynchronous Serial Interface Transmission Completion Interrupt Timing



Caution Do not rewrite asynchronous serial interface mode register 00 (ASIM00) during a transmit operation. If ASIM00 is rewritten during transmission, subsequent transmission may not operate correctly (the normal state is restored by $\overline{\text{RESET}}$ input). Whether transmission is in progress or not can be judged by software using a transmission completion interrupt (INTST00) or the interrupt request flag (STIF00) set by INTST00.

(d) Reception

When bit 6 (RXE00) of asynchronous serial interface mode register 00 (ASIM00) is set (1), a receive operation is enabled and sampling of the RxD pin input is performed.

RxD pin input sampling is performed using the serial clock specified by ASIM00.

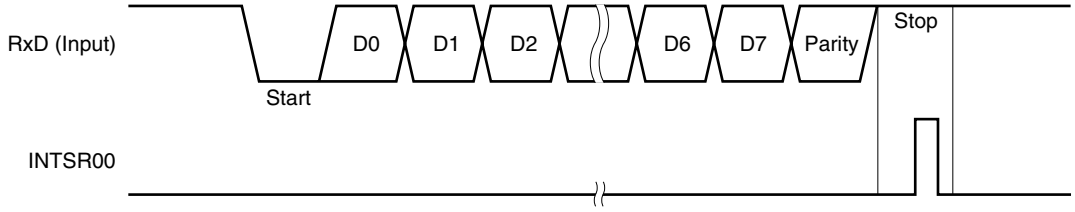
When the RxD pin input becomes low, the 3-bit counter starts counting, and when half the time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 3-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 00 (RXB00), and a reception completion interrupt (INTSR00) is generated.

If an error occurs, the receive data in which the error occurred is still transferred to RXB00, and INTSR00 is generated.

If the RXE00 bit is reset (0) during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB00 and asynchronous serial interface status register 00 (ASIS00) are not changed, and INTSR00 is not generated.

Figure 13-9. Asynchronous Serial Interface Reception Completion Interrupt Timing



Caution Be sure to read receive buffer register 00 (RXB00) even if a receive error occurs. If RXB00 is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

The following three errors may occur during a receive operation: a parity error, framing error, or overrun error. The data reception result error flag is set in asynchronous serial interface status register 00 (ASIS00). Receive error causes are shown in Table 13-7.

What kind of error occurred during reception can be judged by reading the contents of ASIS00 in the receive error interrupt servicing (see **Figures 13-9** and **13-10**).

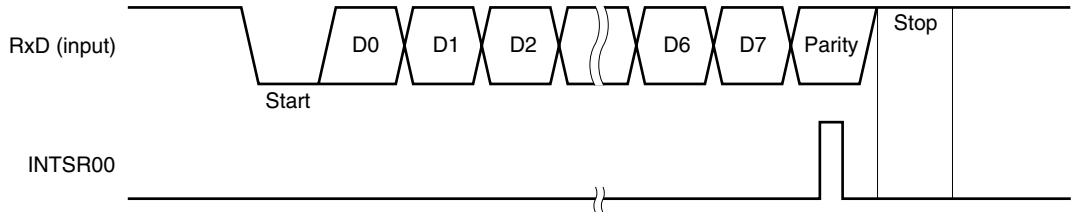
The contents of ASIS00 are reset (0) by reading receive buffer register 00 (RXB00) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 13-7. Receive Error Causes

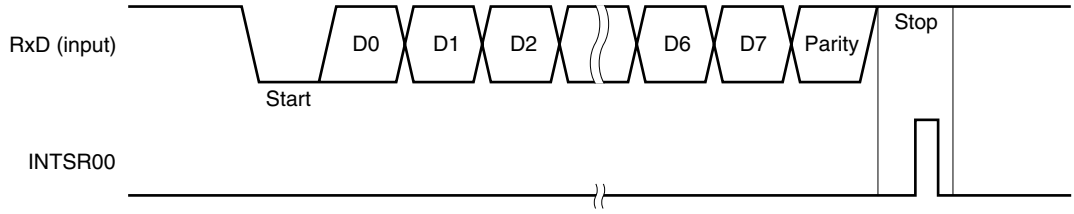
Receive Errors	Cause
Parity error	The parity specified at transmission and the reception data parity do not match.
Framing error	A stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from the receive buffer register.

Figure 13-10. Receive Error Timing

(a) Parity error occurs



(b) Framing error or overrun error occurs



Cautions 1. The contents of the ASIS00 register are reset (0) by reading receive buffer register 00 (RXB00) or receiving the next data. To ascertain the error contents, read ASIS00 before reading RXB00.

2. Be sure to read receive buffer register 00 (RXB00) even if a receive error occurs. If RXB00 is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

★ **(f) Reading receive data**

When the reception completion interrupt (INTSR00) is generated, receive data can be read by reading the value of receive buffer register 00 (RXB00).

To read the receive data stored in receive buffer register 00 (RXB00), read while reception is enabled (RXE00 = 1).

Remark However, if it is necessary to read receive data after reception has stopped (RXE00 = 0), read using either of the following methods.

(a) Read after setting RXE00 = 0 after waiting for one cycle or more of the source clock selected by BRGC00.

(b) Read after bit 2 (DIR00) of serial operation mode register 00 (CSIM00) is set (1).

Program example of (a) (BRGC00 = 00H (source clock = fx/2))

```
INTRXE:                 ;<Reception completion interrupt routine>
      NOP               ;2 clocks
      CLR1 RXE00        ;Reception stopped
      MOV  A, RXB00     ;Read receive data
```

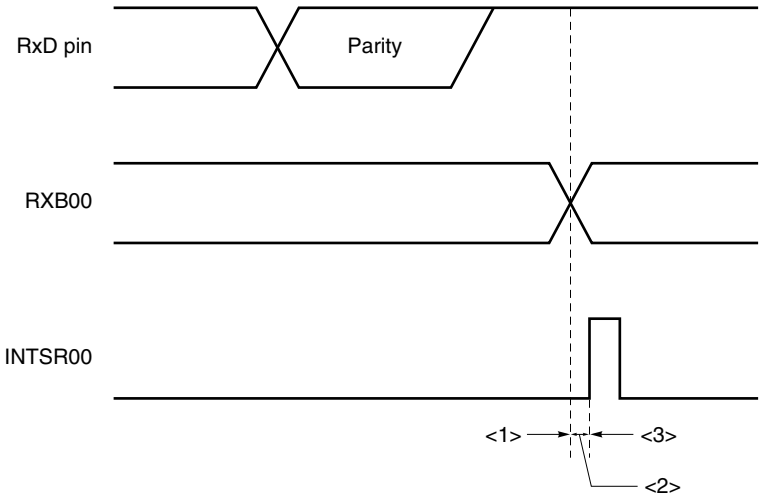
Program example of (b)

```
INTRXE:                 ;<Reception completion interrupt routine>
      SET1 CSIM00.2     ;DIR00 flag is set to LSB first
      CLR1 RXE00        ;Reception stopped
      MOV  A, RXB00     ;Read receive data
```

(3) Cautions on UART mode

(a) When bit 7 (TXE00) of asynchronous serial interface mode register 00 (ASIM00) is cleared during transmission, be sure to set transmit shift register 00 (TXS00) to FFH, then set the TXE00 bit to 1 before executing the next transmission.

(b) When bit 6 (RXE00) of asynchronous serial interface mode register 00 (ASIM00) is cleared during reception, receive buffer register 00 (RXB00) and the reception completion interrupt (INTSR00) are as follows.



When RXE00 is set to 0 at the timing indicated by <1>, RXB00 holds the previous data and does not generate INTSR00.
When RXE00 is set to 0 at the timing indicated by <2>, RXB00 renews the data and does not generate INTSR00.
When RXE00 is set to 0 at the timing indicated by <3>, RXB00 renews the data and generates INTSR00.

13.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous serial interface, such as the 75XL Series, 78K Series, and 17K Series.

Communication is performed using three lines: the serial clock ($\overline{SCK}$), serial output (SO), and serial input (SI).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 00 (CSIM00), asynchronous serial interface mode register 00 (ASIM00), and baud rate generator control register 00 (BRGC00).

(a) Serial operation mode register 00 (CSIM00)

CSIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM00 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM00	CSIE00	0	0	0	0	DIR00	CSCK00	0	FF72H	00H	R/W

CSIE00	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

DIR00	Start bit specification
0	MSB
1	LSB

CSCK00	Clock selection in 3-wire serial I/O mode
0	Clock input to $\overline{SCK}$ pin from external
1	Dedicated baud rate generator output

Caution Bits 0 and 3 to 6 must be fixed to 0.

(b) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM00 to 00H.

ASIM00 must be set to 00H in the 3-wire serial I/O mode.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	0	0	FF70H	00H	R/W

TXE00	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE00	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS001	PS000	Parity bit specification
0	0	No parity
0	1	0 parity always added at transmission Parity check is not performed at reception (no parity error occurs.)
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

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

(c) Baud rate generator control register 00 (BRGC00)

BRGC00 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC00 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC00	TPS003	TPS002	TPS001	TPS000	0	0	0	0	FF73H	00H	R/W

TPS003	TPS002	TPS001	TPS000	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
Other than above				Setting prohibited	

Cautions 1. When BRGC00 is written during a communication operation, the output of the baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during a communication operation.

2. Do not select n = 1 during f_X = 5.0 MHz operation because the baud rate rating is exceeded.

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Value in the above table that is determined in the settings of TPS000 to TPS003 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

If the internal clock is used as the serial clock for the 3-wire serial I/O mode, set the TPS000 to TPS003 bits to set the frequency of the serial clock. To obtain the frequency to be set, use the following formula. When the serial clock is input from off-chip, setting BRGC00 is unnecessary.

$$\text{Serial clock frequency} = \frac{f_X}{2^{n+1}} \text{ [Hz]}$$

f_X: Main system clock oscillation frequency

n: Value in the above table that is determined in the settings of TPS000 to TPS003 ($1 \leq n \leq 8$)

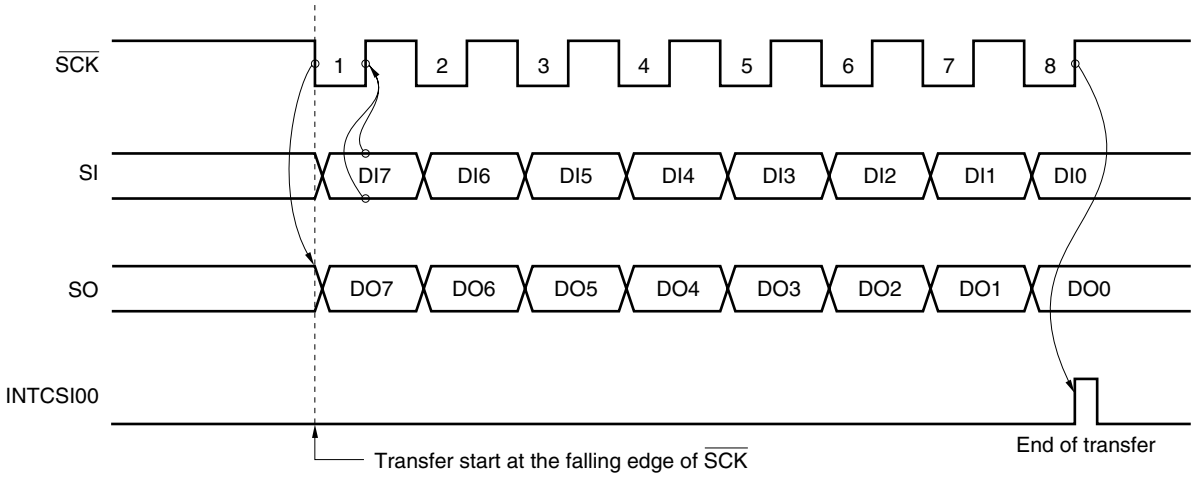
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmit shift register 00 (TXS00/SIO00) and receive shift register 00 (RXS00) shift operations are performed in synchronization with the fall of the serial clock ($\overline{SCK}$). Then transmit data is held in the SO latch and output from the SO pin. Also, receive data input to the SI pin is latched in receive buffer register 00 (RXB00/SIO00) on the rise of $\overline{SCK}$.

At the end of an 8-bit transfer, the operation of TXS00/SIO00 or RXS00 stops automatically, and the interrupt request signal (INTCSI00) is generated.

Figure 13-11. 3-Wire Serial I/O Mode Timing



(3) Transfer start

Serial transfer is started by setting transfer data to transmit shift register 00 (TXS00/SIO00) when the following two conditions are satisfied.

- Bit 7 (CSIE00) of serial operation mode register 00 (CSIM00) = 1
- Internal serial clock is stopped or $\overline{SCK}$ is a high level after 8-bit serial transfer.

Caution If CSIE00 is set to 1 after data is written to TXS00/SIO00, transfer does not start.

Termination of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI00).

CHAPTER 14 LCD CONTROLLER/DRIVER

14.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver of the μPD789407A and 789417A Subseries are as follows.

(1) Automatic output of segment and common signals based on automatic display data memory read
(2) Five different display modes:
- Static
- 1/2 duty (1/2 bias)
- 1/3 duty (1/2 bias)
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(3) Four different frame frequencies, selectable in each display mode
(4) Up to 28 segment signal outputs (S0 to S27) and four common signal outputs (COM0 to COM3)
Of these segment signal outputs, 12 outputs can be switched to I/O ports in 2-output units (P80/S27 to P87/S20 and P90/S19 to P93/S16).
(5) Voltage divider resistors (for LCD drive voltage generation) that a port itself can contain if so specified with a mask option
(6) Operation with a subsystem clock

Table 14-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 14-1. Maximum Number of Pixels

Bias Mode	Number of Time Slices	Common Signals Used	Maximum Number of Pixels
–	Static	COM0 (COM1 to COM3)	28 (28 segment signals, 1 common signal)[Note 1]
1/2	2	COM0, COM1	56 (28 segment signals, 2 common signals)[Note 2]
	3	COM0 to COM2	84 (28 segment signals, 3 common signals)[Note 3]
1/3	3	COM0 to COM2	
	4	COM0 to COM3	112 (28 segment signals, 4 common signals)[Note 4]

Notes 1. Three-digit LCD panel, each digit having an 8-segment 8. configuration.
2. Seven-digit LCD panel, each digit having a 4-segment 8. configuration.
3. Nine-digit LCD panel, each digit having a 3-segment 8. configuration.
4. Fourteen-digit LCD panel, each digit having a 2-segment 8. configuration.

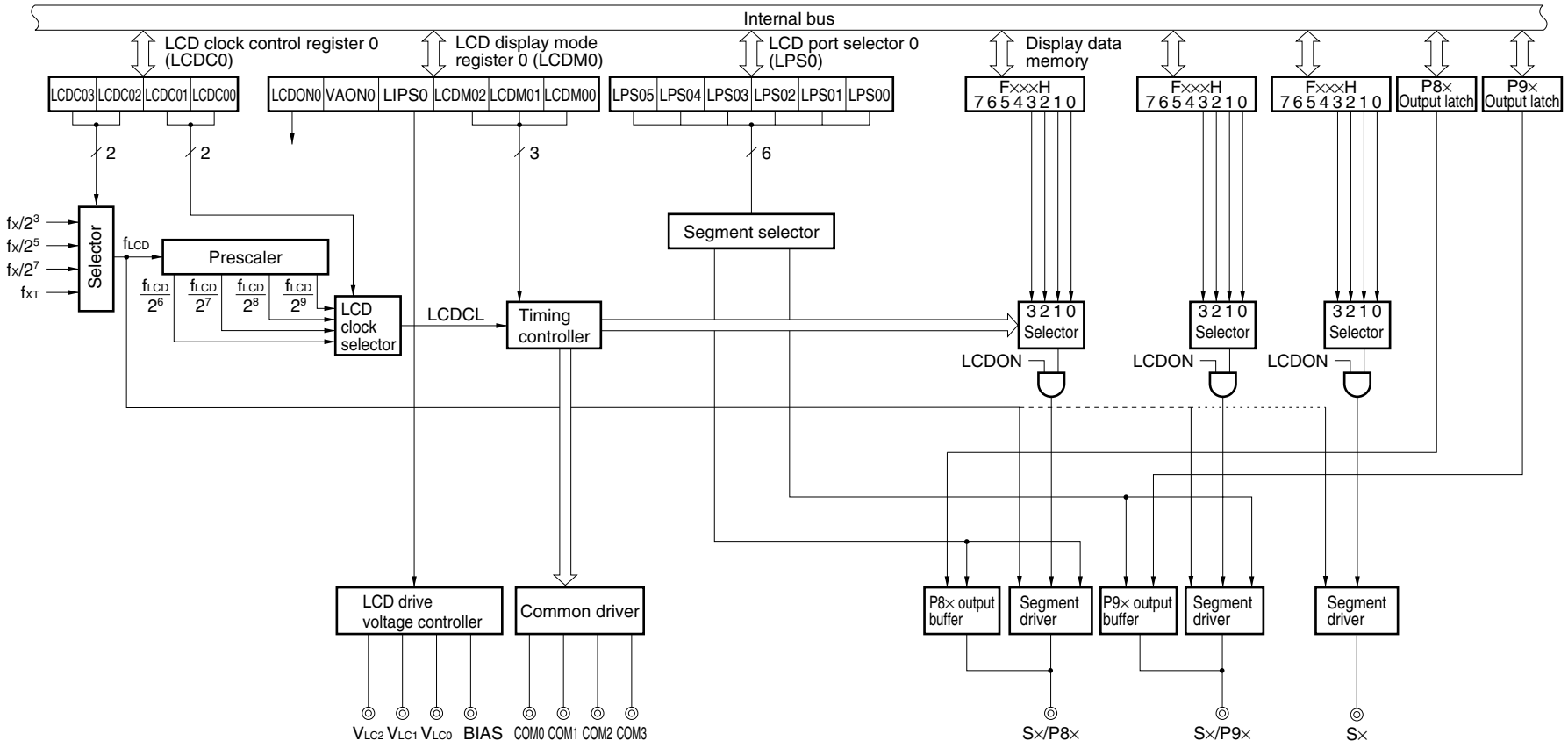
14.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

Table 14-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	28 segment signals (16 dedicated segment signals and 12 segment and I/O port signals) 4 common signals (COM0 to COM3)
Control registers	LCD display mode register 0 (LCDM0) LCD port selector 0 (LPS0) LCD clock control register 0 (LCDC0)

Figure 14-1. Block Diagram of LCD Controller/Driver



14.3 Registers Controlling LCD Controller/Driver

The following three registers are used to control the LCD controller/driver.

- LCD display mode register 0 (LCDM0)
- LCD port selector 0 (LPS0)
- LCD clock control register 0 (LCDC0)

(1) LCD display mode register 0 (LCDM0)

LCDM0 specifies whether to enable display operation. It also specifies the operation mode, LCD drive power supply, and display mode.

LCDM0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDM0 to 00H.

Figure 14-2. Format of LCD Display Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDM0	LCDON0	VAON0	0	LIPS0	0	LCDM02	LCDM01	LCDM00	FFB0H	00H	R/W

LCDON0	Control of LCD display
0	Display off (all segment outputs are deselected.)
1	Display on

VAON0	LCD controller/driver operation mode[Note]
0	Normal operation
1	Low-voltage operation

LIPS0	LCD drive power supply selection
0	LCD drive power is not supplied.
1	LCD drive power is supplied to the BIAS pin.

LCDM02	LCDM01	LCDM00	LCD controller/driver display mode selection	
			Number of time slices	Bias mode
0	0	0	4	1/3
0	0	1	3	1/3
0	1	0	2	1/2
0	1	1	3	1/2
1	0	0	Static	
Other than above			Setting prohibited	

Note When the LCD display panel is not used, VAON0 and LIPS0 must be fixed to 0 to conserve power.

Caution Before attempting to manipulate VAON0, set LIPS0 and LCDON0 to 0 to turn off the LCD.

(2) LCD port selector 0 (LPS0)

LPS0 controls port and segment signal output switching.

LPS0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LPS0 to 00H.

Figure 14-3. Format of LCD Port Selector 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LPS0	0	0	LPS05	LPS04	LPS03	LPS02	LPS01	LPS00	FFB1H	00H	R/W

	LPS05	LPS04	LPS03	LPS02	LPS01	LPS00
	P93/S16, P92/S17	P91/S18, P90/S19	P87/S20, P86/S21	P85/S22, P84/S23	P83/S24, P82/S25	P81/S26, P80/S27
0	Used as ports (Pmn)					
1	Used as segments (S×)					

Cautions 1. Bits 6 and 7 must be fixed to 0.

2. Be sure to use segments in sequence from the smallest segment value (LPS05 → LPS04 → … → LPS00).

Remark m = 8 n = 0 to 7

m = 9 n = 0 to 3

× = 16 to 27

(3) LCD clock control register 0 (LCDC0)

LCDC0 specifies the LCD source clock and LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

LCDC0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDC0 to 00H.

Figure 14-4. Format of LCD Clock Control Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDC0	0	0	0	0	LCDC03	LCDC02	LCDC01	LCDC00	FFB2H	00H	R/W

LCDC03	LCDC02	Selection of LCD source clock frequency (f_{LCD})[Note]
0	0	$f_X/2^7$ (39.1 kHz)
0	1	f_{XT} (32.768 kHz)
1	0	$f_X/2^5$ (156.3 kHz)
1	1	$f_X/2^3$ (625 kHz)

LCDC01	LCDC00	Selection of LCD clock (LCDCL) frequency
0	0	$f_{LCD}/2^6$
0	1	$f_{LCD}/2^7$
1	0	$f_{LCD}/2^8$
1	1	$f_{LCD}/2^9$

Note Specify an LCD source clock (f_{LCD}) frequency of at least 32 kHz.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

For example, Table 14-3 lists the frame frequencies used when f_{XT} (32.768 kHz) is supplied to the LCD source clock (f_{LCD}).

Table 14-3. Frame Frequencies (Hz)

LCD Clock (LCDCL) Frequency / Number of Time Slices	$f_{XT}/2^9$ (64 Hz)	$f_{XT}/2^8$ (128 Hz)	$f_{XT}/2^7$ (256 Hz)	$f_{XT}/2^6$ (512 Hz)
Static	64	128	256	512
2	32	64	128	256
3	21	43	85	171
4	16	32	64	128

14.4 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

<1> Set the initial values in the LCD display data memory (FA00H to FA1BH).
<2> Set the pins to be used for segment output in LCD port selector 0 (LPS0).
<3> Set the display and operation modes in LCD display mode register 0 (LCDM0).
<4> Set the LCD clock in LCD clock control register 0 (LCDC0).

Subsequent to this procedure, set the data to be displayed in the data memory.

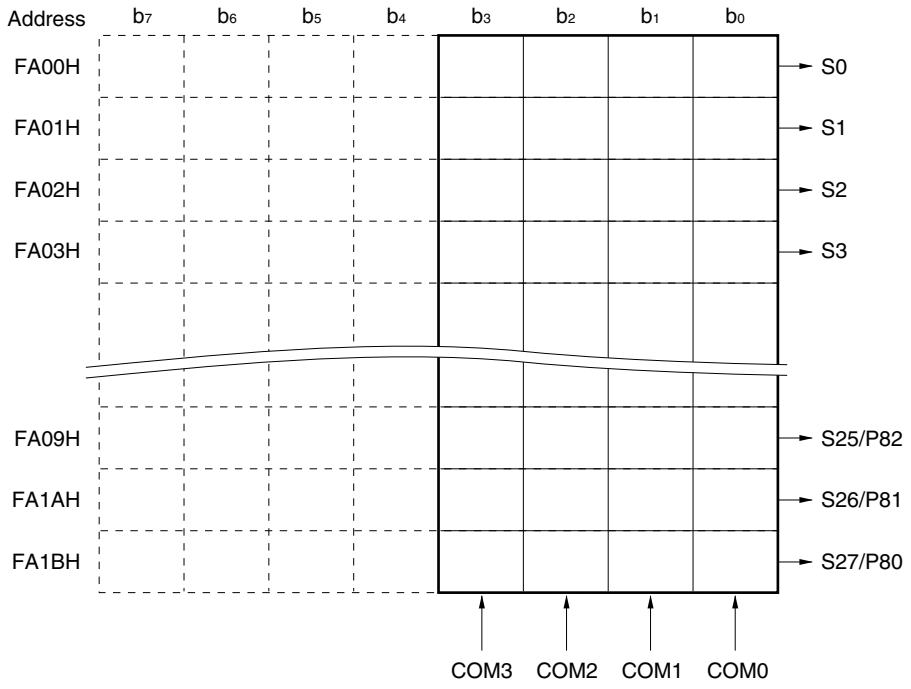
14.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses FA00H to FA1BH. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 14-5 shows the relationship between the contents of the LCD display data memory and the segment/common outputs.

The part of the display data memory not used for display can be used as ordinary RAM.

Figure 14-5. Relationship Between LCD Display Data Memory Contents and Segment/Common Outputs



Caution No memory is allocated to the higher 4 bits of the LCD display data memory. Be sure to fix there bits to 0.

14.6 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). The pixels turn off when the potential difference becomes lower than V_{LCD}.

Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

(1) Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in Table 14-4. In the static display mode, the same signal is output to COM0 to COM3.

In the two-time-slice mode, leave the COM2 and COM3 pins open. In the three-time-slice mode, leave the COM3 pin open.

Table 14-4. COM Signals

COM Signal / Number of Time Slices	COM0	COM1	COM2	COM3
Static display mode				
Two-time-slice mode			Open	Open
Three-time-slice mode				Open
Four-time-slice mode				

(2) Segment signals

The segment signals correspond to 28 bytes of LCD display data memory (FA00H to FA1BH). Bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (S0 to S27). Note that S16 to S27 can also be used as I/O port pins.

Check, with the information given above, what combination of front-surface electrodes (corresponding to the segment signals) and rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data memory, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

LCD display data memory bits 1 and 2, bits 2 and 3, and bit 3 are not used for LCD display in the static display, two-time slot, and three-time slot modes, respectively. So these bits can be used for purposes other than display.

LCD display data memory bits 4 to 7 are fixed to 0.

(3) Output waveforms of common and segment signals

The voltages listed in Table 14-5 are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of $\pm V_{LCD}$ is obtained.

The other combinations of the signals correspond to the display off-voltage.

Table 14-5. LCD Drive Voltage

(a) Static display mode

Segment Signal / Common Signal	Select Signal Level	Deselect Signal Level
	V_{SS0}/V_{LC0}	V_{LC0}/V_{SS0}
V_{LC0}/V_{SS0}	$-V_{LCD}/+V_{LCD}$	0 V/0 V

(b) 1/2 bias method

Segment Signal / Common Signal		Select Signal Level	Deselect Signal Level
		V_{SS0}/V_{LC0}	V_{LC0}/V_{SS0}
Select signal level	V_{LC0}/V_{SS0}	$-V_{LCD}/+V_{LCD}$	0 V/0 V
Deselect signal level	$V_{LC1} = V_{LC2}$	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$

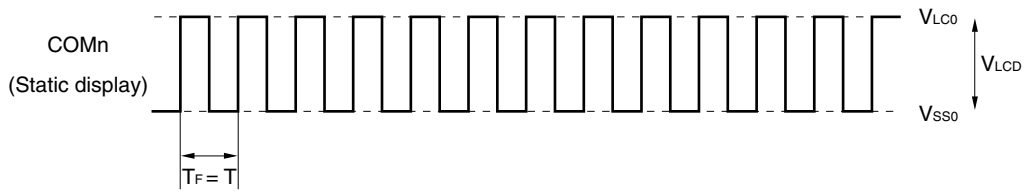
(c) 1/3 bias method

Segment Signal / Common Signal		Select Signal Level	Deselect Signal Level
		V_{SS0}/V_{LC0}	V_{LC1}/V_{LC2}
Select signal level	V_{LC0}/V_{SS0}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$
Deselect signal level	V_{LC2}/V_{LC1}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$

Figure 14-6 shows the common signal waveforms, and Figure 14-7 shows the voltages and phases of the common and segment signals.

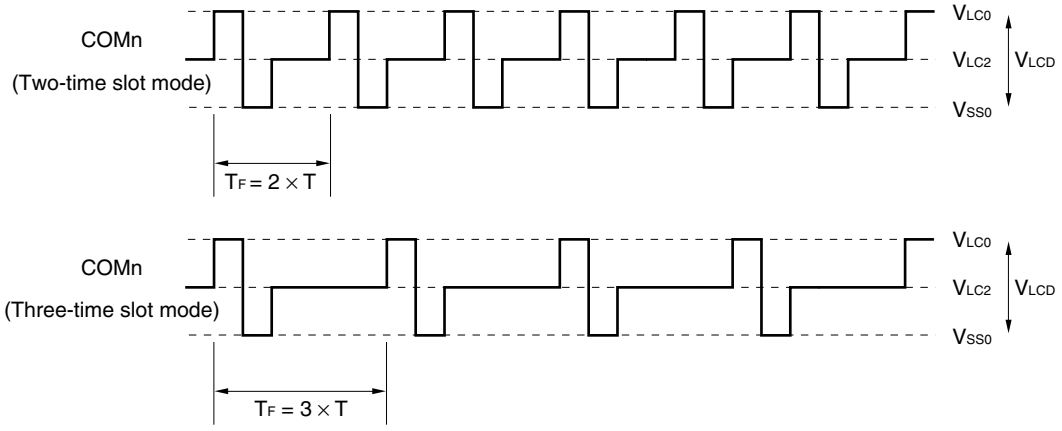
Figure 14-6. Common Signal Waveforms

(a) Static display mode



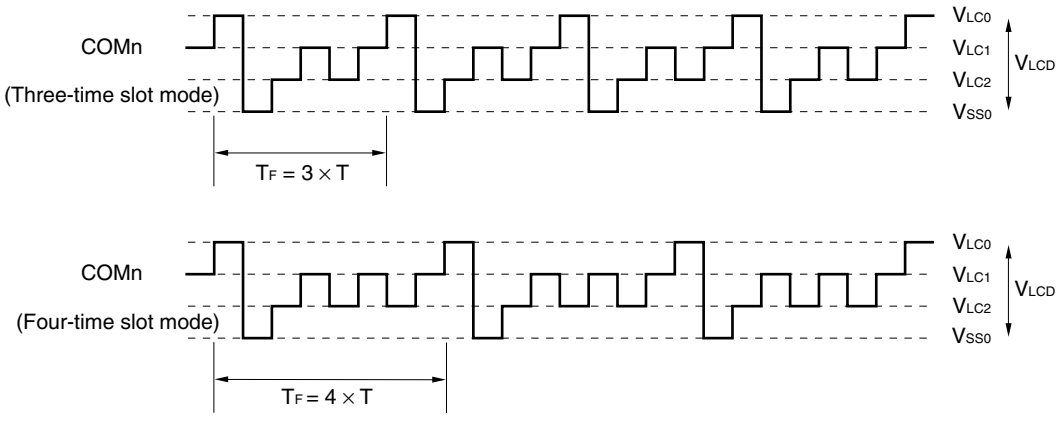
T: One LCD clock period T_F: Frame frequency

(b) 1/2 bias method



T: One LCD clock period T_F: Frame frequency

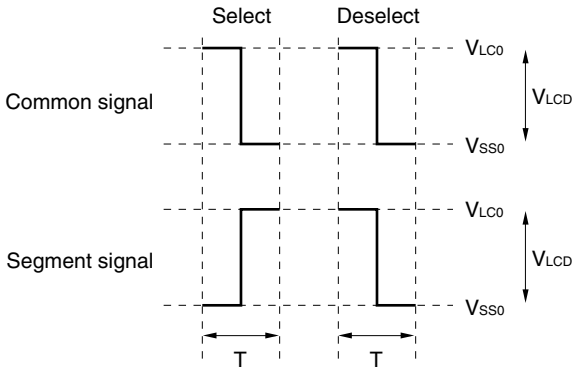
(c) 1/3 bias method



T: One LCD clock period T_F: Frame frequency

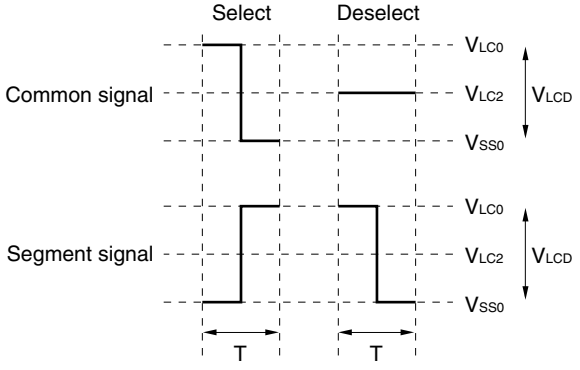
Figure 14-7. Voltages and Phases of Common and Segment Signals

(a) Static display mode



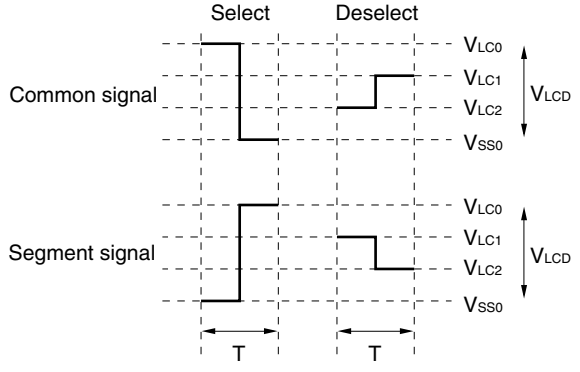
T: One LCD clock period

(b) 1/2 bias method



T: One LCD clock period

(c) 1/3 bias method



T: One LCD clock period

14.7 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, and V_{LC2}

The mask ROM versions (except the μPD78F9418A) of the LCD display can incorporate voltage divider resistors for generating LCD drive power as specified using a mask option. Incorporating voltage divider resistors can generate LCD drive voltages that meet each bias method listed in Table 14-6, without using external voltage divider resistors.

The LCD drive voltage can be supplied to the BIAS pin to support various LCD drive voltage levels.

Table 14-6. LCD Drive Voltages (with On-Chip Voltage Divider Resistors)

Bias Method / LCD Drive Voltage Pin	No Bias (Static)	1/2 Bias Method	1/3 Bias Method
V_{LC0}	V_{LCD}	V_{LCD}	V_{LCD}
V_{LC1}	V_{LCD}	V_{LCD}[Note]	$\frac{2}{3}V_{LCD}$
V_{LC2}	$\frac{2}{3}V_{LCD}$	$\frac{1}{2}$	$\frac{1}{3}V_{LCD}$

Note For the 1/2 bias method, it is necessary to connect the V_{LC1} and V_{LC2} pins externally.

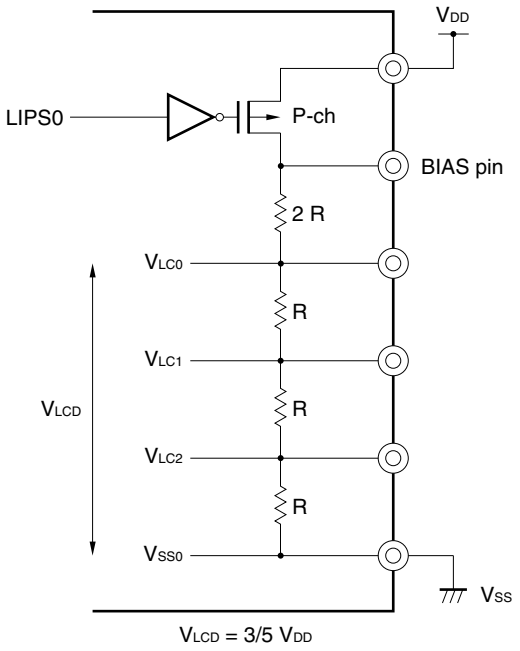
Remarks 1. If the BIAS and V_{LC0} pins are open, $V_{LCD} = \frac{3}{5}V_{DD}$ (if voltage divider resistors are included).

2. If the BIAS and V_{LC0} pins are connected, $V_{LCD} = V_{DD}$.

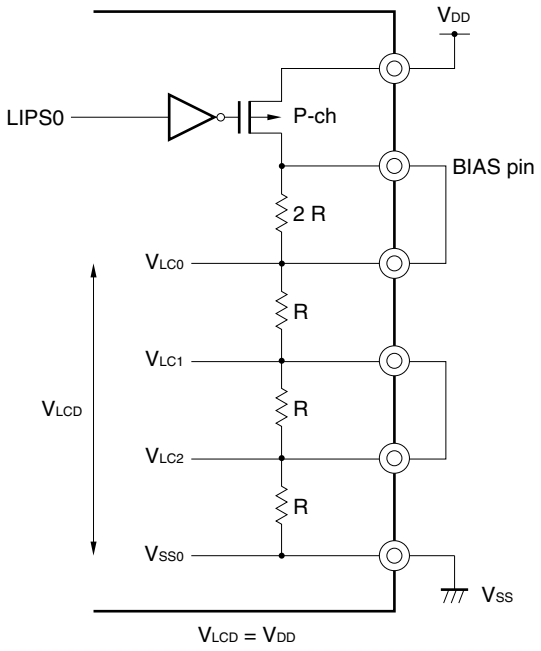
Figure 14-8 shows examples of generating LCD drive voltages internally according to Table 14-6.

Figure 14-8. Examples of LCD Drive Power Connections (with On-Chip Voltage Divider Resistors)

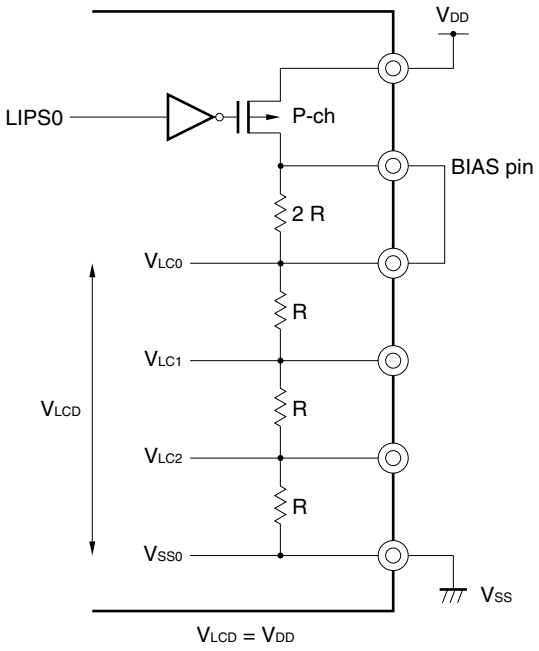
(a) 1/3 bias method and static display mode (V_{DD} = 5 V and V_{LCD} = 3 V)



(b) 1/2 bias method (V_{DD} = 5 V and V_{LCD} = 5 V)



(c) 1/3 bias method and static display mode (V_{DD} = 5 V and V_{LCD} = 5 V)



LIPS0: Bit 4 of LCD display mode register 0 (LCDM0)

14.8 Display Modes

14.8.1 Static display example

Figure 14-10 shows how the three-digit LCD panel having the display pattern shown in Figure 14-9 is connected to the segment signals (S0 to S23) and the common signal (COM0) of the μPD789407A or 789417A Subseries chip. This example displays data "12.3" in the LCD panel. The contents of the display data memory (addresses FA00H to FA17H) correspond to this display.

The following description focuses on numeral "2." (2.) displayed in the second digit. To display "2." in the LCD panel, it is necessary to apply the select or deselect voltage to the S8 to S15 pins according to Table 14-7 at the timing of the common signal COM0; see Figure 14-9 for the relationship between the segment signals and LCD segments.

Table 14-7. Select and Deselect Voltages (COM0)

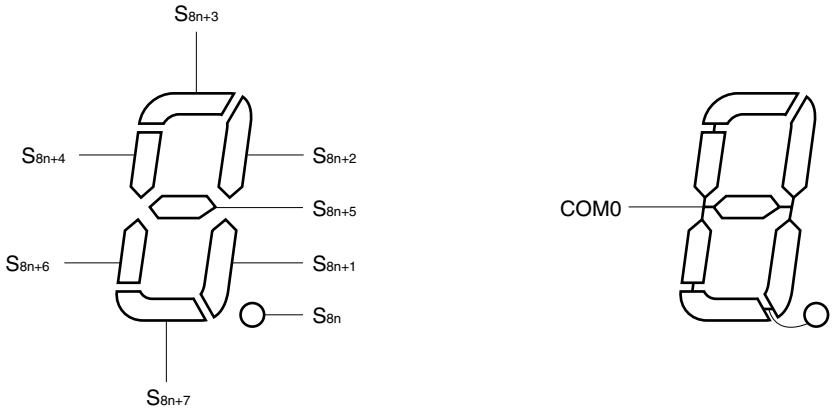
Segment / Common	S8	S9	S10	S11	S12	S13	S14	S15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to Table 14-7, it is determined that the bit-0 pattern of the display data memory locations (FA08H to FA0FH) must be 10110111.

Figure 14-11 shows the LCD drive waveforms of S11 and S12, and COM0. When the select voltage is applied to S11 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

Figure 14-9. Static LCD Display Pattern and Electrode Connections



Remark n = 0 to 2

Figure 14-10. Example of Connecting Static LCD Panel

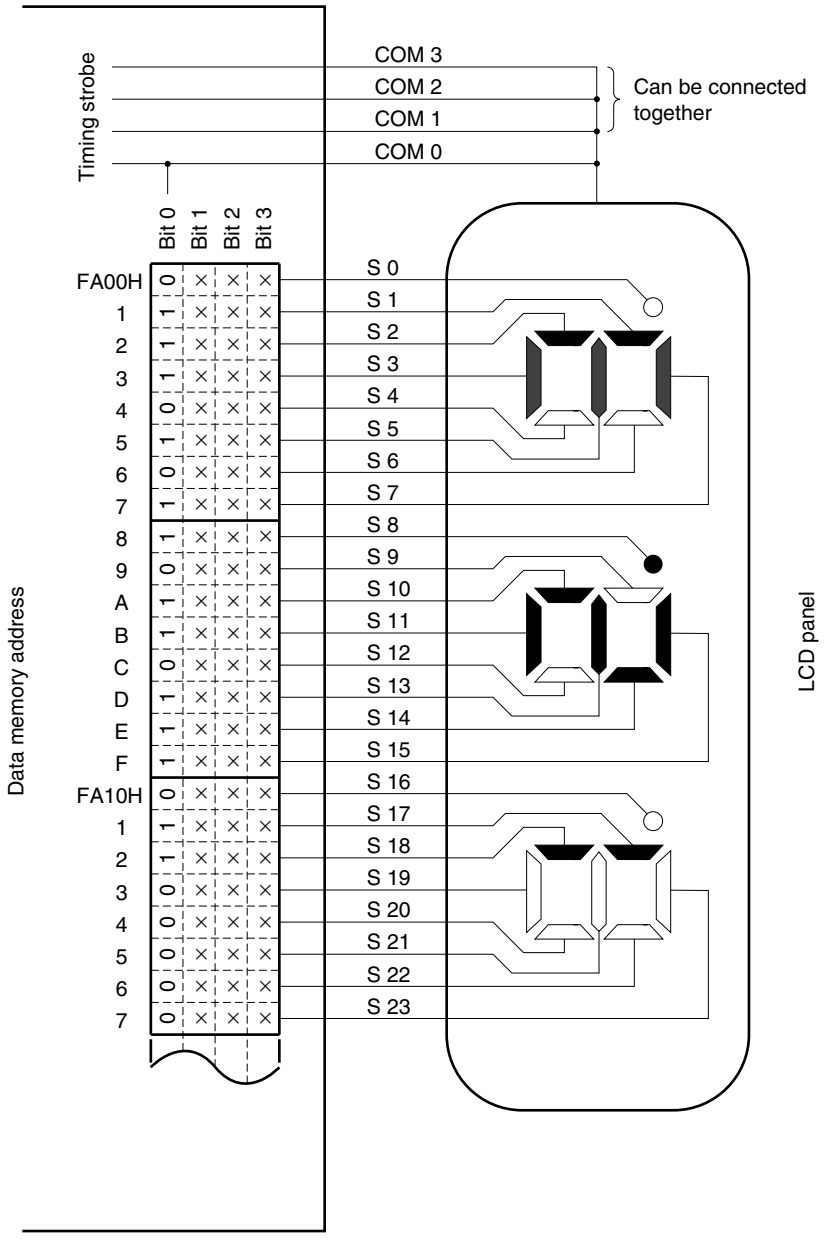
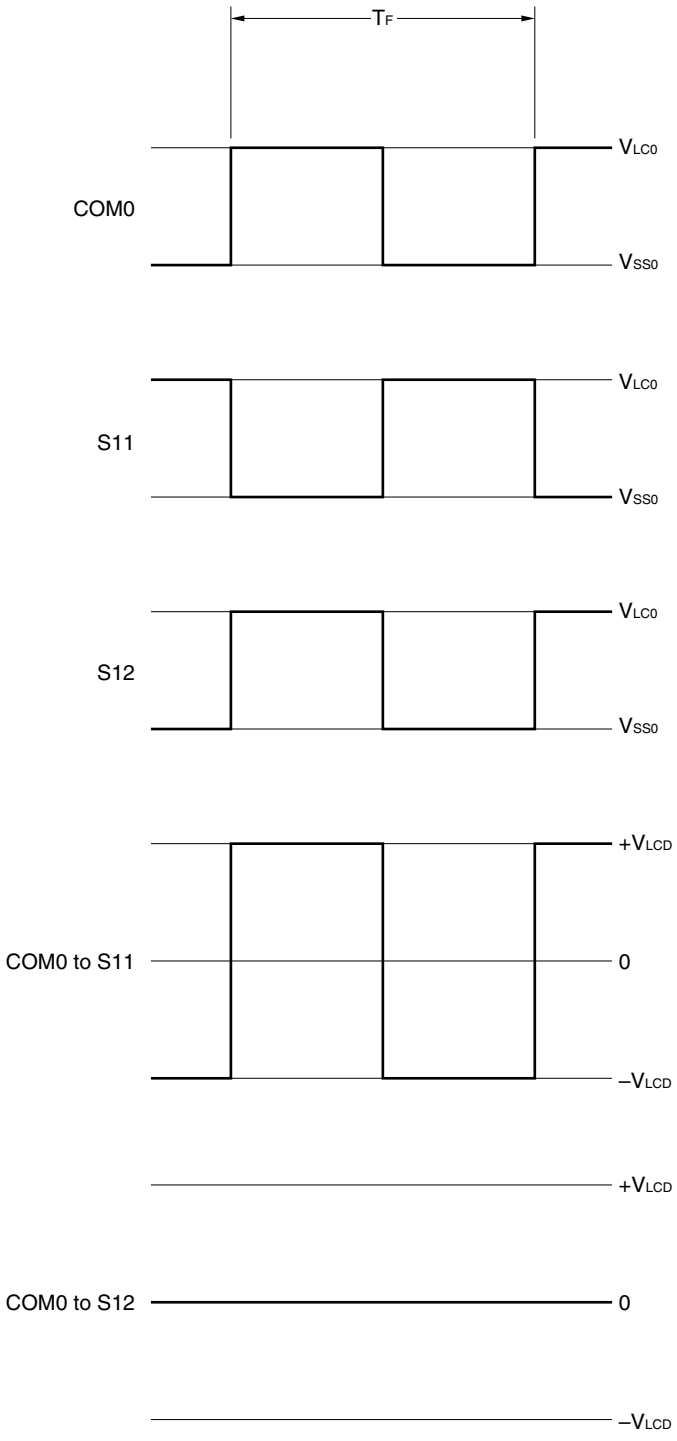


Figure 14-11. Static LCD Drive Waveform Examples



14.8.2 Two-time-slice display example

Figure 14-13 shows how the seven-digit LCD panel having the display pattern shown in Figure 14-12 is connected to the segment signals (S0 to S27) and the common signals (COM0 and COM1) of the μPD789407A or 789417A Subseries chip. This example displays data "123456.7" in the LCD panel. The contents of the display data memory (addresses FA00H to FA1BH) correspond to this display.

The following description focuses on numeral "3" (∃) displayed in the fifth digit. To display "3" in the LCD panel, it is necessary to apply the select or deselect voltage to the S16 to S19 pins according to Table 14-8 at the timing of the common signals COM0 and COM1; see Figure 14-12 for the relationship between the segment signals and LCD segments.

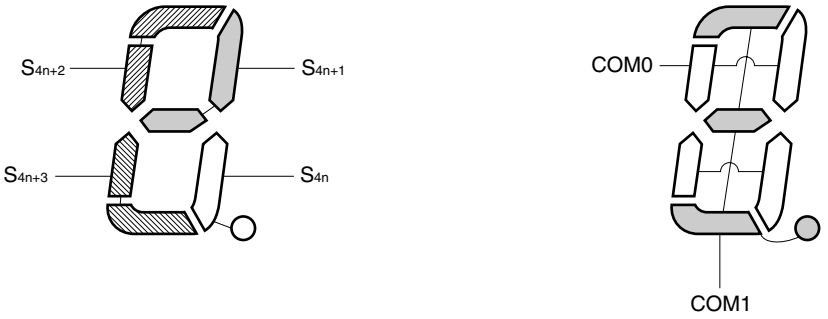
Table 14-8. Select and Deselect Voltages (COM0 and COM1)

Segment / Common	S16	S17	S18	S19
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to Table 14-8, it is determined that the display data memory location (FA13H) that corresponds to S19 must contain xx10.

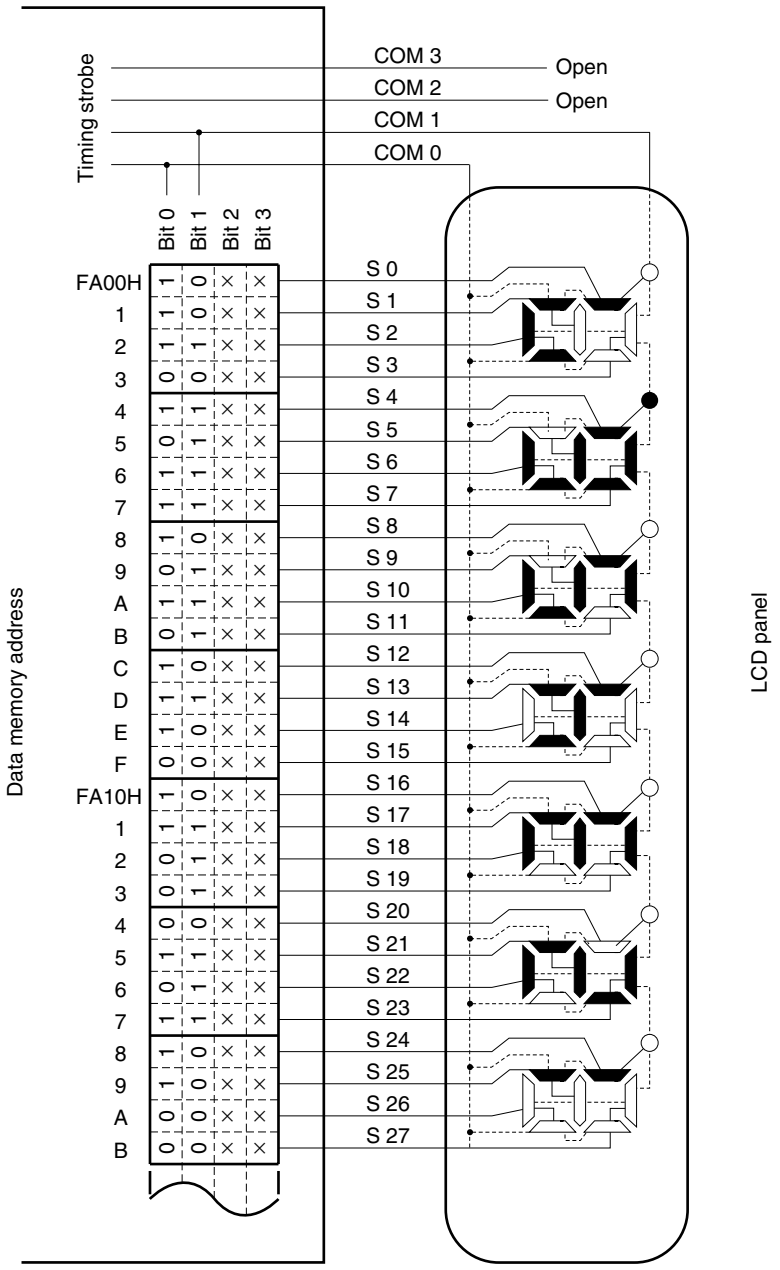
Figure 14-14 shows examples of LCD drive waveforms between the S19 signal and each common signal. When the select voltage is applied to S19 at the timing of COM1, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 14-12. Two-Time-Slice LCD Display Pattern and Electrode Connections



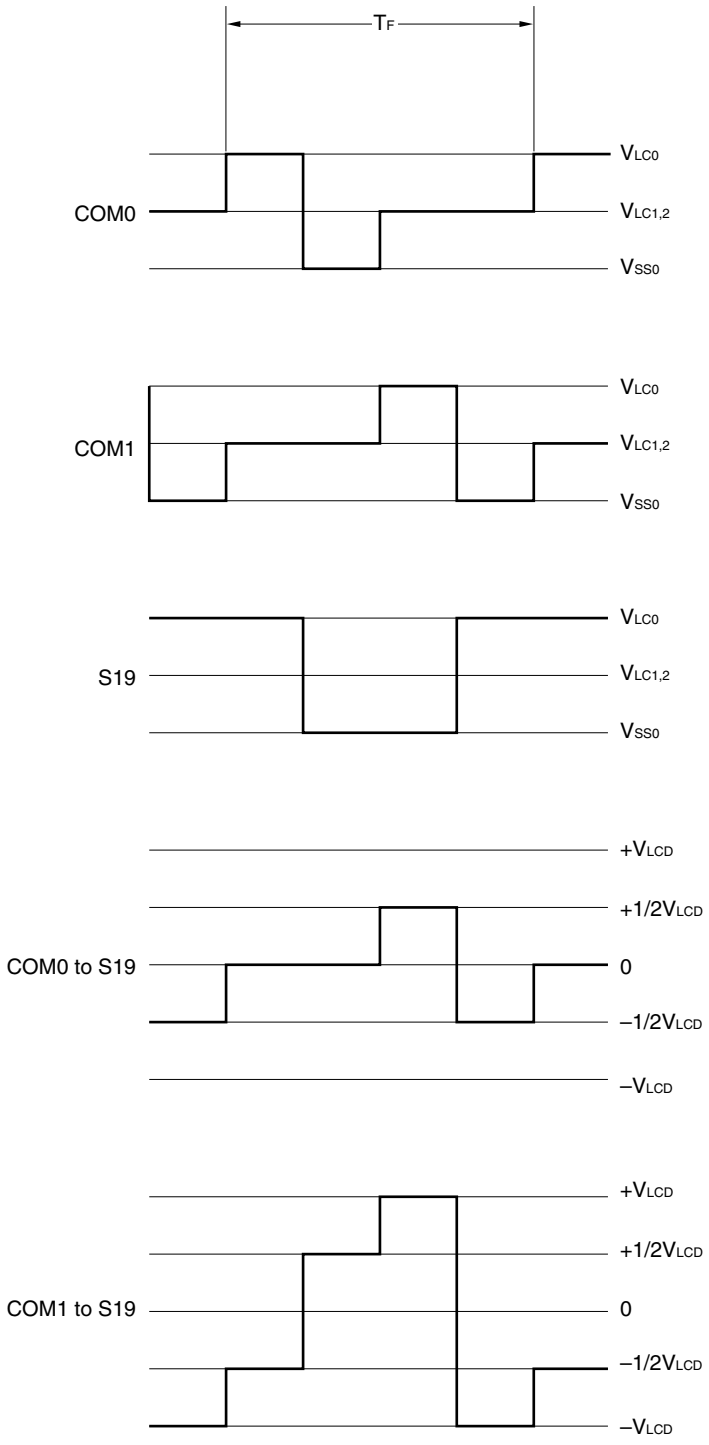
Remark n = 0 to 6

Figure 14-13. Example of Connecting Two-Time-Slice LCD Panel



×: Can always be used to store any data because the two-time-slice mode is being used.

Figure 14-14. Two-Time-Slice LCD Drive Waveform Examples (1/2 Bias Method)



14.8.3 Three-time-slice display example

Figure 14-16 shows how the nine-digit LCD panel having the display pattern shown in Figure 14-15 is connected to the segment signals (S0 to S26) and the common signals (COM0 to COM2) of the μPD789407A or 789417A Subseries chip. This example displays data "123456.789" in the LCD panel. The contents of the display data memory (addresses FA00H to FA1AH) correspond to this display.

The following description focuses on numeral "6." (6.) displayed in the fourth digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the S9 to S11 pins according to Table 14-9 at the timing of the common signals COM0 to COM2; see Figure 14-15 for the relationship between the segment signals and LCD segments.

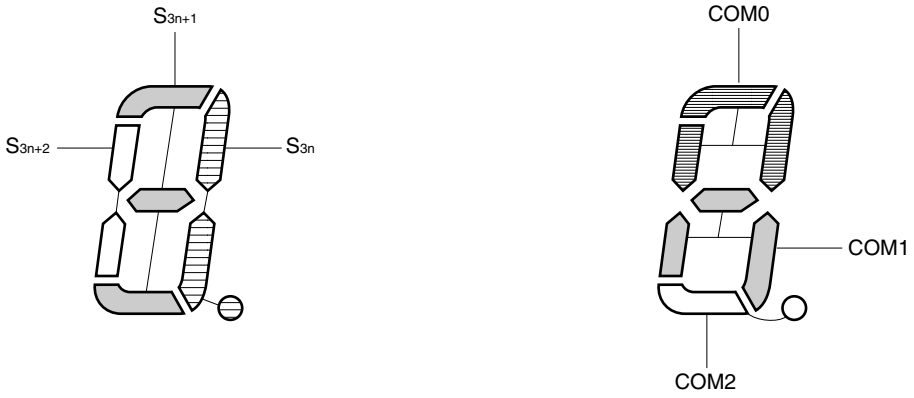
Table 14-9. Select and Deselect Voltages (COM0 to COM2)

Segment / Common	S9	S10	S11
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to Table 14-9, it is determined that the display data memory location (FA09H) that corresponds to S9 must contain x110.

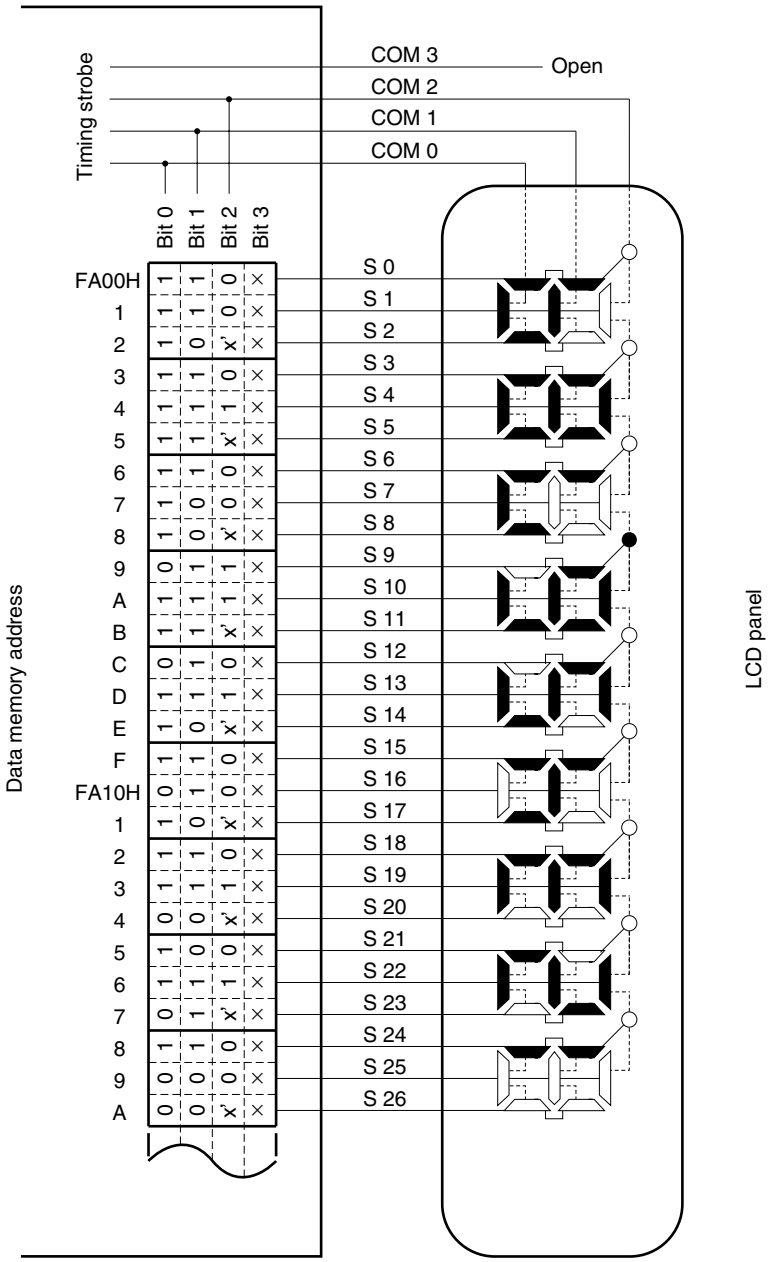
Figures 14-17 and 14-18 show examples of LCD drive waveforms between the S9 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to S9 at the timing of COM1 or COM2, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 14-15. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 8

Figure 14-16. Example of Connecting Three-Time-Slice LCD Panel



x': Can be used to store any data because there is no corresponding segment in the LCD panel.

×: Can always be used to store any data because the three-time-slice mode is being used.

Figure 14-17. Three-Time-Slice LCD Drive Waveform Examples (1/2 Bias Method)

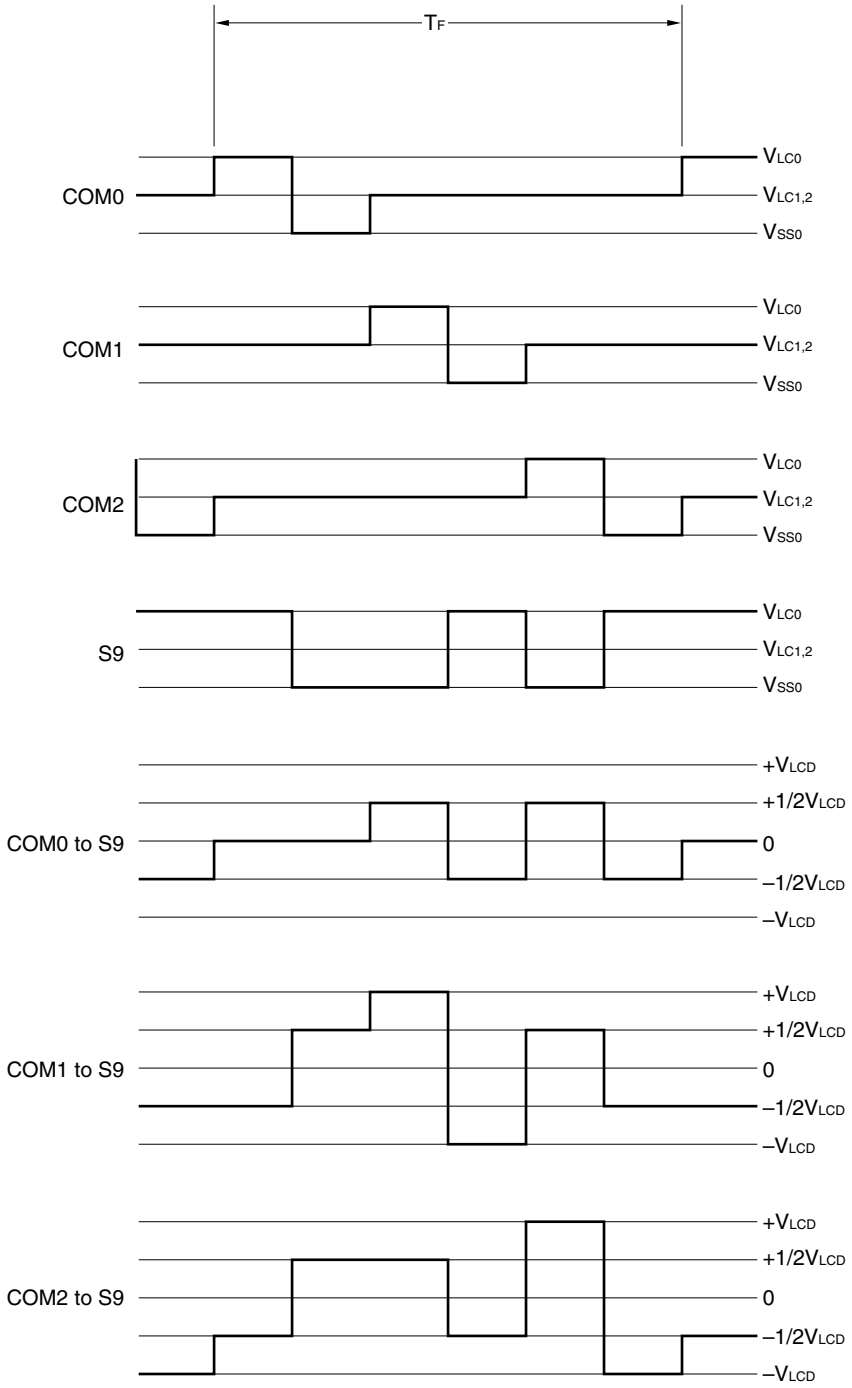
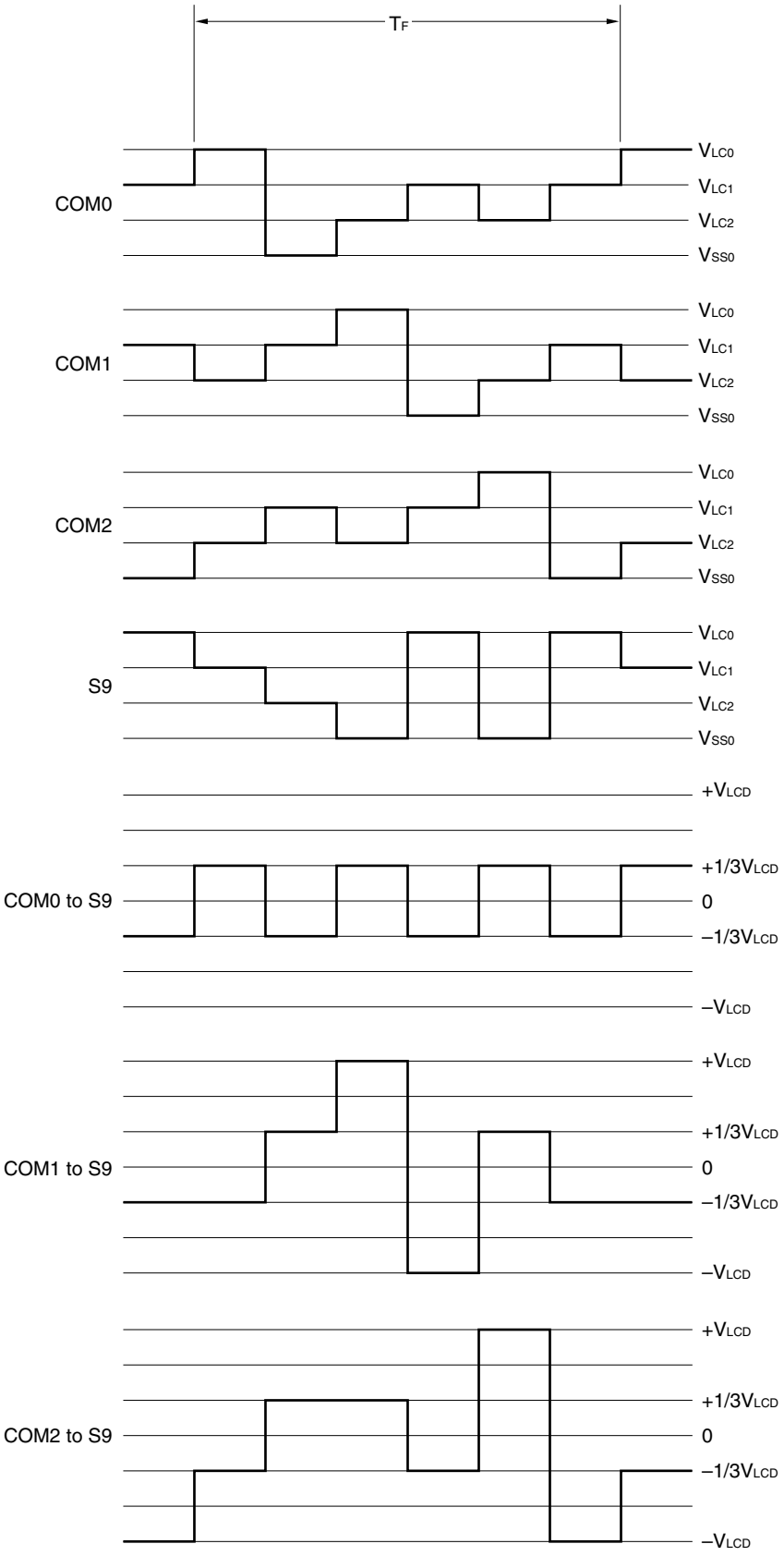


Figure 14-18. Three-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



14.8.4 Four-time-slice display example

Figure 14-20 shows how the 14-digit LCD panel having the display pattern shown in Figure 14-19 is connected to the segment signals (S0 to S27) and the common signals (COM0 to COM3) of the μPD789407A or 789417A Subseries chip. This example displays data "123456.78901234" in the LCD panel. The contents of the display data memory (addresses FA00H to FA1BH) correspond to this display.

The following description focuses on numeral "6." ( ) displayed in the ninth digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the S16 and S17 pins according to Table 14-10 at the timing of the common signals COM0 to COM3; see Figure 14-19 for the relationship between the segment signals and LCD segments.

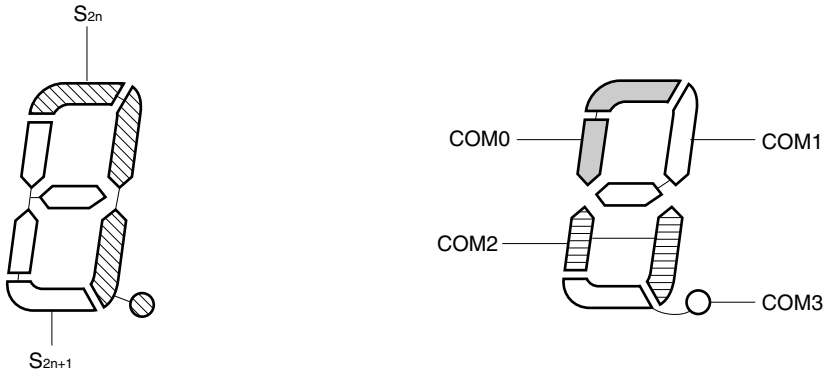
Table 14-10. Select and Deselect Voltages (COM0 to COM3)

Segment / Common	S16	S17
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 14-10, it is determined that the display data memory location (FA16H) that corresponds to S16 must contain 1101.

Figure 14-21 shows examples of LCD drive waveforms between the S16 signal and each common signal. When the select voltage is applied to S16 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 14-19. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 13

Figure 14-20. Example of Connecting Four-Time-Slice LCD Panel

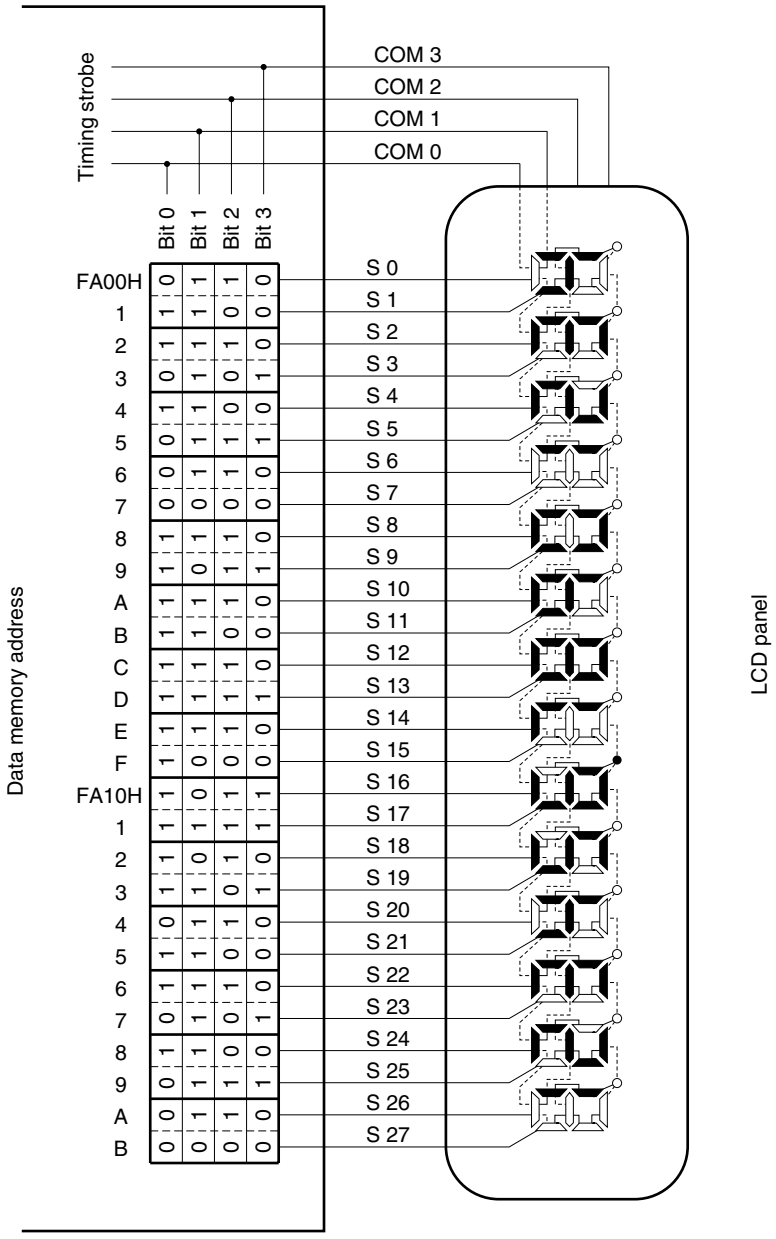
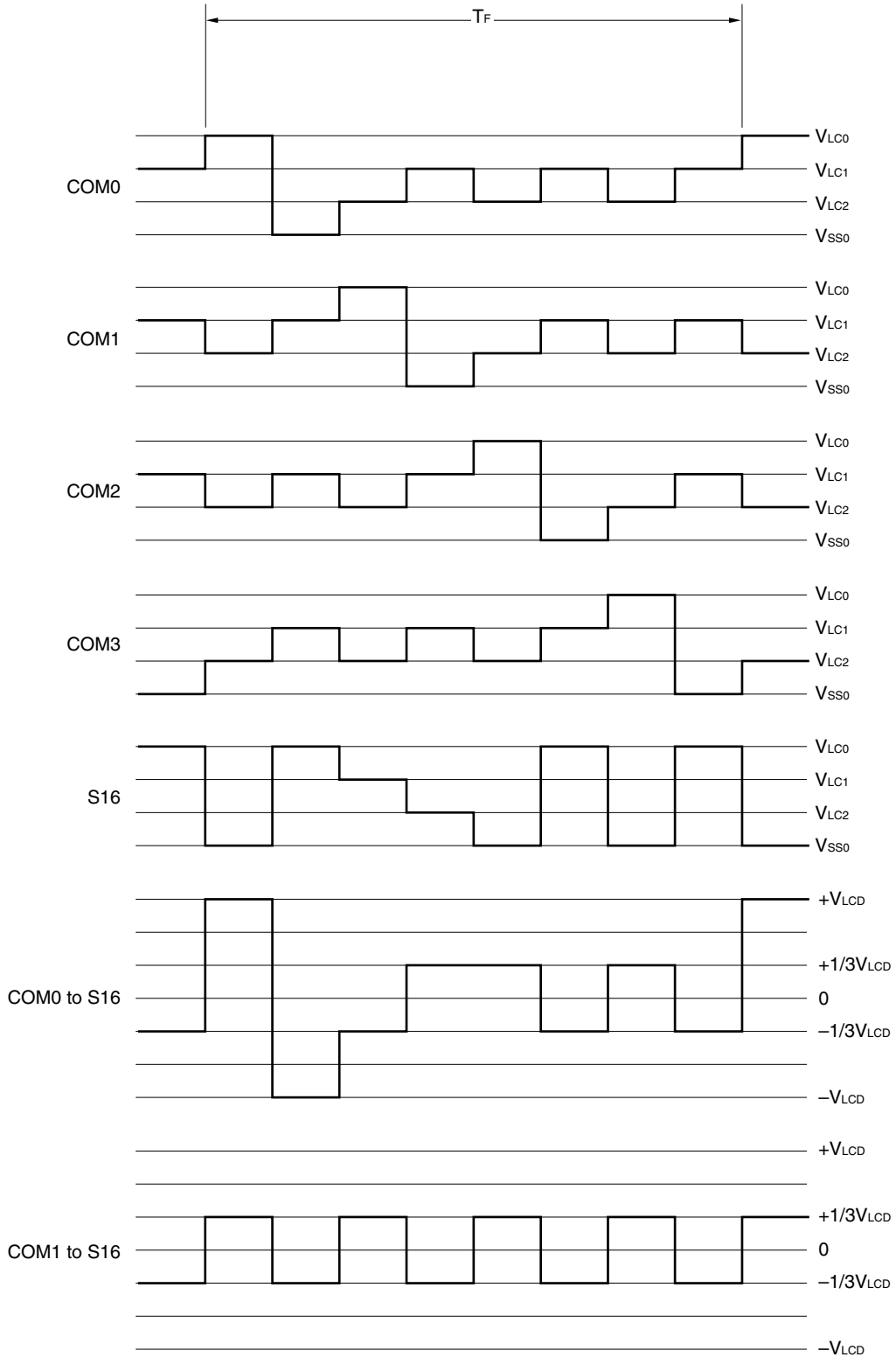


Figure 14-21. Four-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



Remark The waveforms for COM2 to S16 and COM3 to S16 are omitted.

CHAPTER 15 INTERRUPT FUNCTIONS

15.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupt

These interrupts undergo mask control. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority as shown in Table 15-1.

A standby release signal is generated.

Five external interrupt and 11 internal interrupt sources are incorporated as maskable interrupts.

15.2 Interrupt Sources and Configuration

A total of 17 non-maskable and maskable interrupts are incorporated as interrupt sources (see **Table 15-1**).

Table 15-1. Interrupt Source List

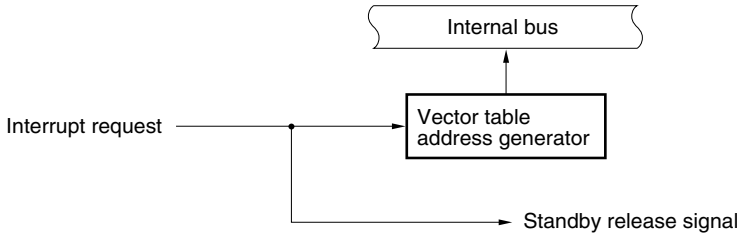
Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	–	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTSR00	End of serial interface 00 UART reception	Internal	000EH	(B)
		INTCSI00	End of serial interface 00 3-wire SIO transfer reception			
	6	INTST00	End of serial interface 00 UART transmission		0010H	
	7	INTWT	Watch timer interrupt		0012H	
	8	INTWTI	Interval timer interrupt		0014H	
	9	INTTM00	Generation of matching signal of 8-bit timer/event counter 00		0016H	
	10	INTTM01	Generation of matching signal of 8-bit timer/event counter 01		0018H	
	11	INTTM02	Generation of matching signal of 8-bit timer 02		001AH	
	12	INTTM50	Generation of matching signal of 16-bit timer 50		001CH	
	13	INTKR00	Key return signal detection	External	001EH	(C)
	14	INTAD0	A/D conversion completion signal	Internal	0020H	(B)
	15	INTCMP0	Comparator signal		0022H	

Notes 1. “Priority” is the priority order when several maskable interrupts are generated at the same time. 0 is the highest and 15 is the lowest.

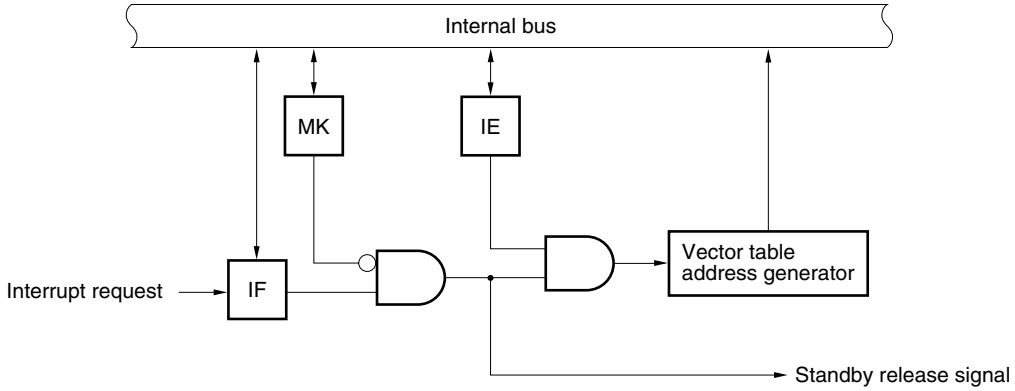
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 15-1.

Figure 15-1. Basic Configuration of Interrupt Function

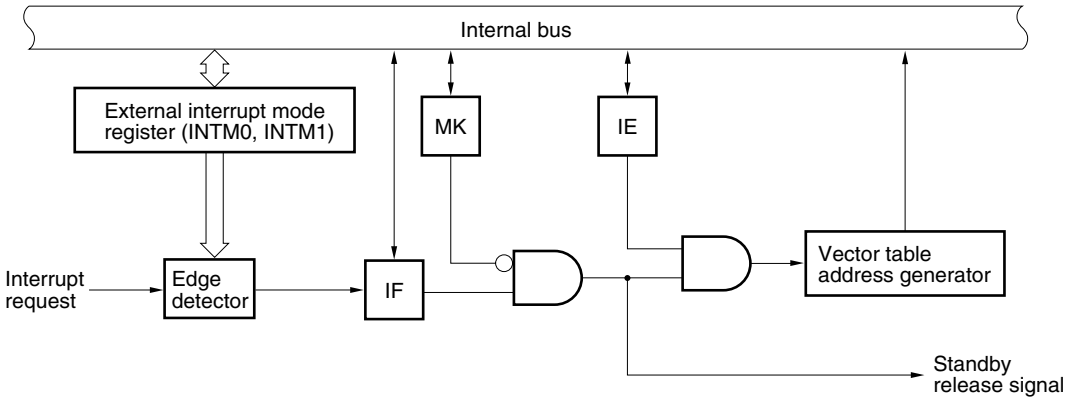
(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
MK: Interrupt mask flag

15.3 Registers Controlling Interrupt Function

The following five registers are used to control the interrupt functions.

- Interrupt request flag registers 0, 1 (IF0 and IF1)
- Interrupt mask flag registers 0, 1 (MK0 and MK1)
- External interrupt mode registers 0, 1 (INTM0 and INTM1)
- Program status word (PSW)
- Key return mode register 00 (KRM00)

Table 15-2 lists the interrupt request flag and interrupt mask flag names corresponding to interrupt requests.

Table 15-2. Flags Corresponding to Interrupt Request Signal Name

Interrupt Request Signal Name	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	TMIF4	TMMK4
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTP2	PIF2	PMK2
INTP3	PIF3	PMK3
INTSR00/INTCSI00	SRIF00	SRMK00
INTST00	STIF00	STMK00
INTWT	WTIF	WTMK
INTWTI	WTIIF	WTIMK
INTTM00	TMIF00	TMMK00
INTTM01	TMIF01	TMMK01
INTTM02	TMIF02	TMMK02
INTTM50	TMIF50	TMMK50
INTKR00	KRIF00	KRMK00
INTAD0	ADIF0	ADMK0
INTCMP0	CMPIF0	CMPMK0

(1) Interrupt request flag registers 0, 1 (IF0 and IF1)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon $\overline{\text{RESET}}$ input.

IF0 and IF1 are set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IF0 and IF1 to 00H.

Figure 15-2. Format of Interrupt Request Flag Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	WTIF	STIF00	SRIF00	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
IF1	CMPIF0	ADIF0	KRIF00	TMIF50	TMIF02	TMIF01	TMIF00	WTIIF	FFE1H	00H	R/W

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated; Interrupt request state

Cautions 1. The TMIF4 flag is R/W enabled only when the watchdog timer is used as an interval timer. If watchdog timer mode 1 or 2 is used, set the TMIF4 flag to 0.

2. Because port 2 has an alternate function as an external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

★ **3. If an interrupt is acknowledged, the interrupt request flag is automatically cleared before the interrupt routine is entered.**

(2) Interrupt mask flag registers 0, 1 (MK0 and MK1)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service.

MK0 and MK1 are set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0 and MK1 to FFH.

Figure 15-3. Format of Interrupt Mask Flag Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	WTMK	STMK00	SRMK00	PMK3	PMK2	PMK1	PMK0	TMMK4	FFE4H	FFH	R/W
MK1	CMPMK0	ADMK0	KRMK00	TMMK50	TMMK02	TMMK01	TMMK00	WTIMK	FFE5H	FFH	R/W

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the TMMK4 flag is read when the watchdog timer is used in watchdog timer mode 1 or 2, its value becomes undefined.

2. Because port 2 has an alternate function as an external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

(3) External interrupt mode register 0 (INTM0)

This register is used to specify a valid edge for INTP0 to INTP2.

INTM0 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM0 to 00H.

Figure 15-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES21	ES20	INTP2 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES11	ES10	INTP1 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES01	ES00	INTP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. Before setting the INTM0 register, be sure to set xxMKx of the relevant interrupt mask flag to 1 to disable interrupts. After that, clear the interrupt mask flag (xxMKx = 0) to enable interrupts after clearing the interrupt request flag (xxIFx = 0).

(4) External interrupt mode register 1 (INTM1)

INTM1 is used to specify a valid edge for INTP3 and INTCMP0.

INTM1 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM1 to 00H.

Figure 15-5. Format of External Interrupt Mode Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM1	ES61	ES60	0	0	0	0	ES31	ES30	FFEDH	00H	R/W

ES61	ES60	INTCMP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	INTP3 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Bits 2 to 5 must be fixed to 0.

2. Before setting INTM1, set the corresponding interrupt mask flag register to 1 to disable interrupts. After that, clear (0) the corresponding interrupt request flag to enable interrupts, then clear the corresponding interrupt mask flag register.

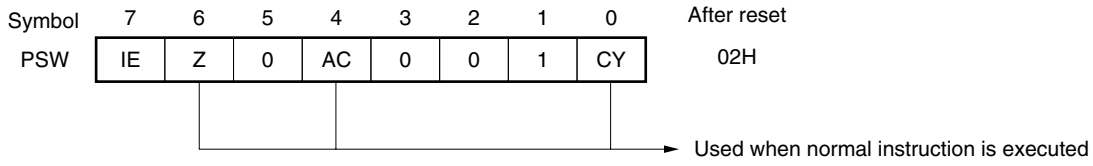
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for interrupt requests. The IE flag to set maskable interrupt enable/disable is mapped to the PSW.

Besides 8-bit unit read/write, this register can carry out operations via bit manipulation instructions and dedicated instructions (EI, DI). When a vectored interrupt is acknowledged, the PSW is automatically saved into a stack, and the IE flag is reset to 0.

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

Figure 15-6. Configuration of Program Status Word



IE	Interrupt acknowledge enable/disable
0	Disabled
1	Enabled

(6) Key return mode register 00 (KRM00)

This register sets the pin that detects a key return signal (falling edge of port 4).

KRM00 is set using a 1-bit or 8-bit memory manipulation instruction.

Bit 0 (KRM000) is set in 4-bit units for KR0/P40 to KR3/P43 pins. Bits 4 and 5 (KRM004 and KRM005) are set in 1-bit units for KR4/P44 and KR5/P45 pins, respectively.

$\overline{\text{RESET}}$ input sets KRM00 to 00H.

Figures 15-7 and 15-8 show the format of key return mode register 00 and the block diagram of the falling edge detector, respectively.

Figure 15-7. Format of Key Return Mode Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM00	0	0	KRM005	KRM004	0	0	0	KRM000	FFF5H	00H	R/W

KRM00n	Key return signal detection selection
0	No detection
1	Detection (detecting falling edge of port 4)

Cautions 1. Bits 1 to 3, 6, and 7 must be fixed to 0.

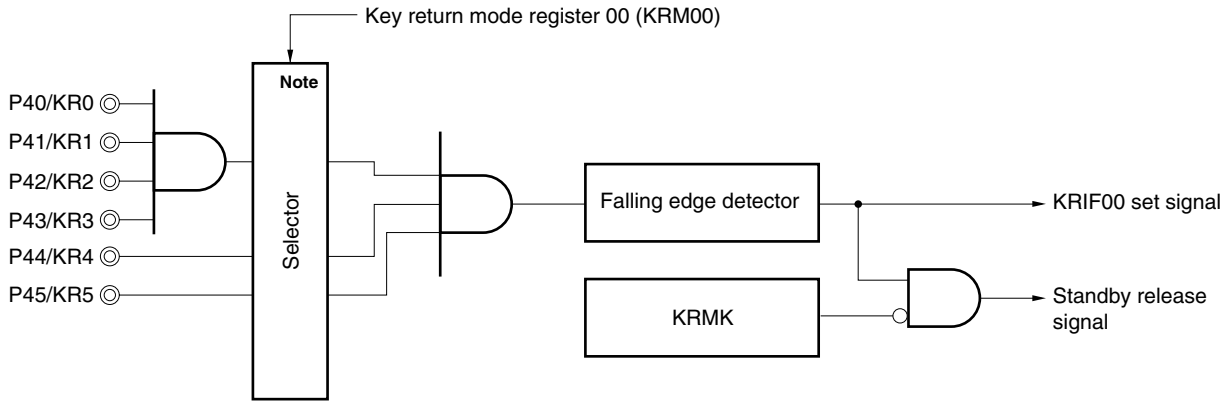
2. When the KRM00 register is set to 1, a pull-up resistor is connected automatically. However, the pull-up resistor is cut if the pin is in output mode.

3. Before setting KRM00, always set bit 5 of MK1 (KRMK00 = 1) to disable interrupts in advance. After setting KRM00, clear bit 5 of MK1 (KRMK00 = 0) after clearing bit 5 of IF1 (KRIF00 = 0) to enable interrupts.

★ **4. The key return signal cannot be detected while even one of the pins that specify detection of the key return signal is low, even if a falling edge is generated at other key return pins.**

Remark n = 0, 4, 5

Figure 15-8. Block Diagram of Falling Edge Detector



Note Selector that selects the pin used for falling edge input

15.4 Operation of Interrupt Servicing

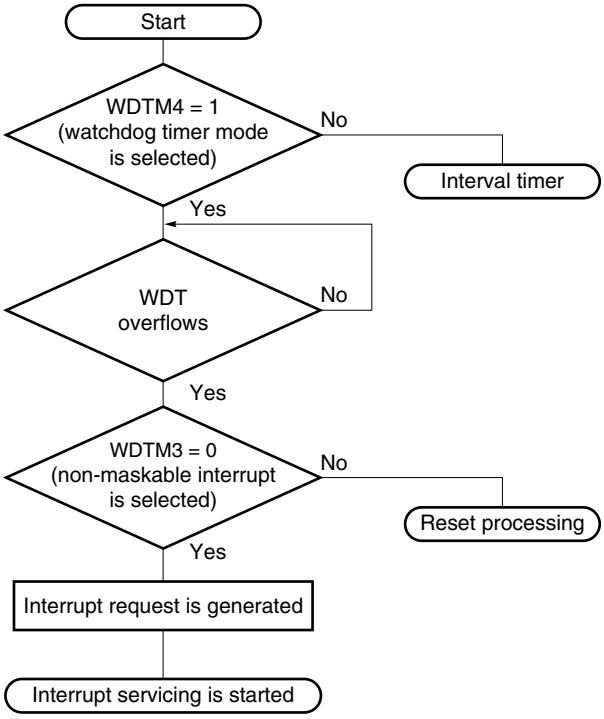
15.4.1 Non-maskable interrupt acknowledgment operation

The non-maskable interrupt is unconditionally acknowledged even when interrupts 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 PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

Caution During non-maskable interrupt servicing program execution, do not input another non-maskable interrupt request; if it is input, the servicing program will be interrupted and the new non-maskable interrupt request will be acknowledged.

Figure 15-9. Flowchart of Non-Maskable Interrupt Request Acknowledgment



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 15-10. Timing of Non-Maskable Interrupt Request Acknowledgment

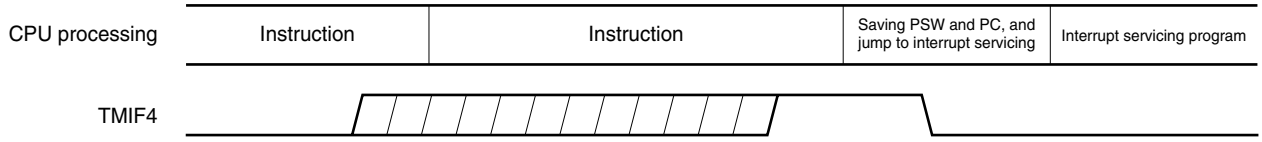
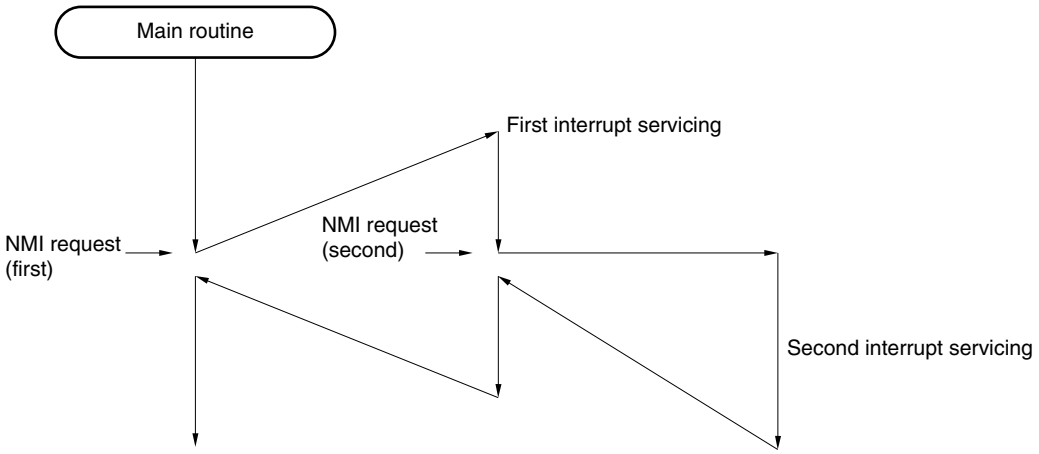


Figure 15-11. Non-Maskable Interrupt Request Acknowledgment



15.4.2 Maskable interrupt acknowledgment operation

A maskable interrupt can be acknowledged when the interrupt request flag is set to 1 and the corresponding interrupt mask flag is cleared to 0. A vectored interrupt is acknowledged in the interrupt enabled status (when the IE flag is set to 1).

The time required to start the interrupt servicing after a maskable interrupt request has been generated is as follows:

Table 15-3. Time from Generation of Maskable Interrupt Request to Servicing

Minimum Time	Maximum Time[Note]
9 clocks	19 clocks

Note The wait time is maximum when an interrupt request is generated immediately before the BT or BF 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.

An interrupt held pending is acknowledged when the status in which it can be acknowledged is set.

Figure 15-12 shows the algorithm of acknowledging interrupts.

When a maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, and the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To restore from interrupt servicing, use the RETI instruction.

Figure 15-12. Interrupt Acknowledgment Program Algorithm

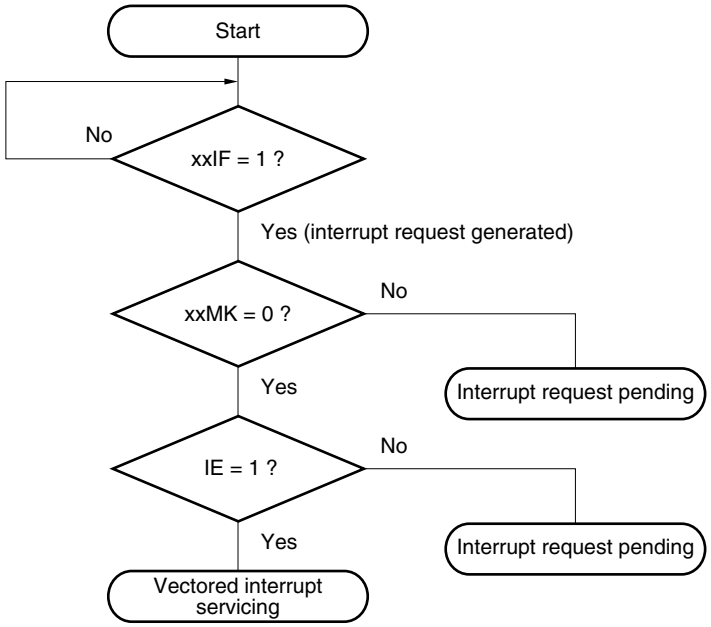
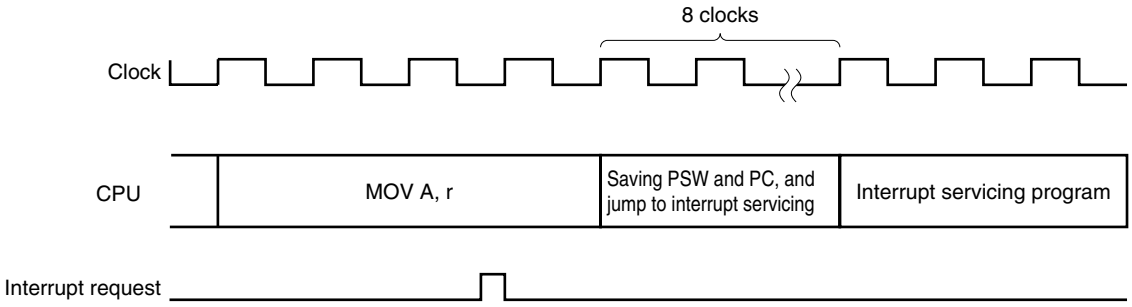
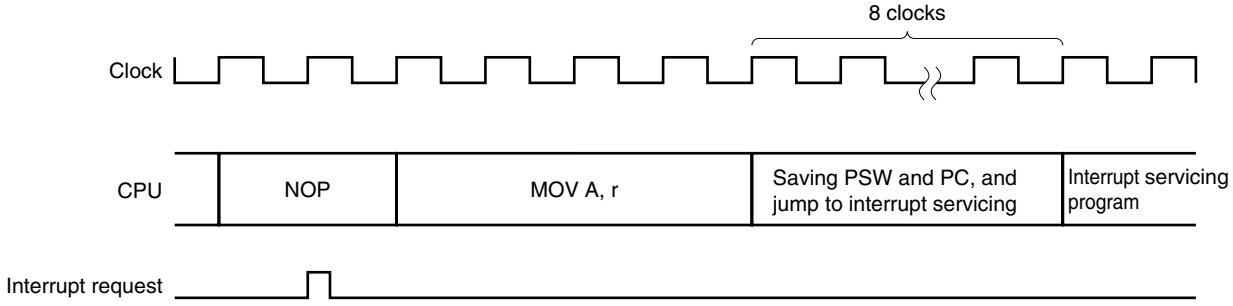


Figure 15-13. Interrupt Request Acknowledgment Timing (Example: MOV A, r)



If the interrupt request has generated an interrupt request flag (XXIF) by the time the instruction clocks under execution, n clocks (n = 4 to 10), are n − 1, interrupt request acknowledgment processing will start following the completion of the instruction under execution. Figure 15-13 shows an example using the 8-bit data transfer instruction MOV A, r. Because this instruction is executed in 4 clocks, if an interrupt request is generated between the start of execution and the 3rd clock, interrupt request acknowledgment processing will take place following the completion of MOV A, r.

Figure 15-14. Interrupt Request Acknowledgment Timing
(When Interrupt Request Flag Is Generated in Final Clock Under Execution)



If the interrupt request flag (XXIF) is generated in the final clock of the instruction, interrupt request acknowledgment processing will begin after execution of the next instruction is complete.

Figure 15-14 shows an example whereby an interrupt request was generated in the 2nd clock of NOP (a 2-clock instruction). In this case, the interrupt request will be processed after execution of MOV A, r, which follows NOP, is complete.

Caution When interrupt request flag registers 0 and 1 (IF0 and IF1) or interrupt mask flag registers 0 and 1 (MK0 and MK1) are being accessed, interrupt requests will be held pending.

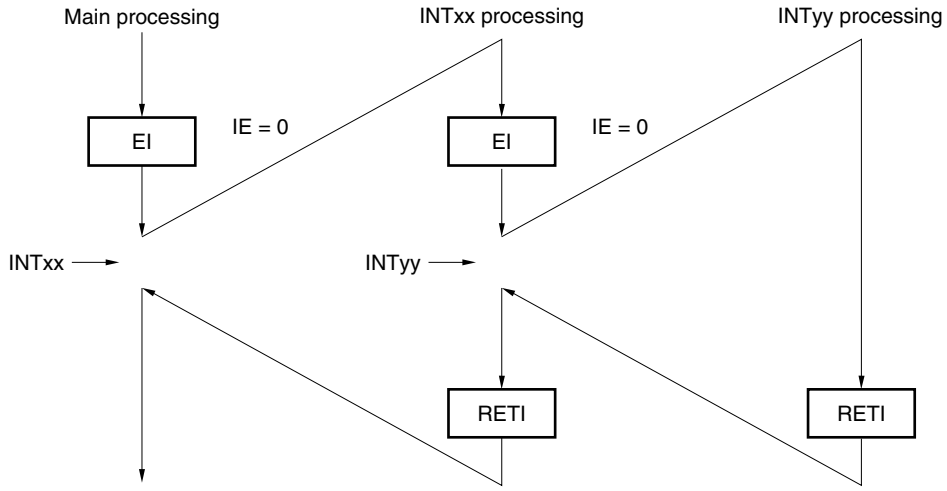
15.4.3 Multiple interrupt servicing

Processing in which another interrupt request is acknowledged while an interrupt request is serviced is called multiple interrupt servicing.

Multiple interrupts are not performed unless an interrupt request is enabled (IE = 1) (except non-maskable interrupt request). The other interrupt request is disabled (IE = 0) at the time when an interrupt request is acknowledged. Therefore, it is necessary to set (1) the IE flag to realize the interrupt enable state using an EI instruction during interrupt request servicing in order to enable multiple interrupt servicing.

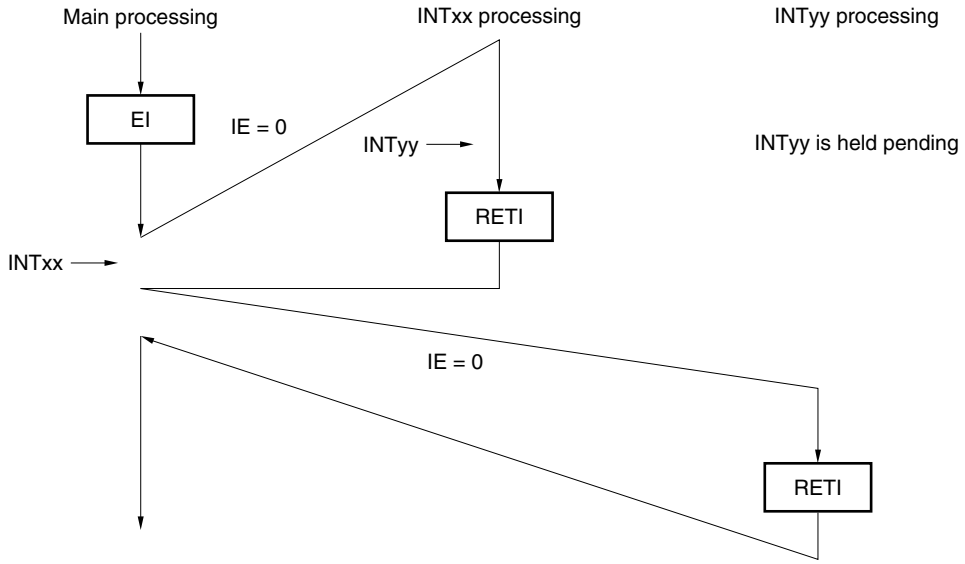
Figure 15-15. Example of Multiple Interrupt

Example 1. Acknowledging multiple interrupts



The interrupt request INTyy is acknowledged and multiple interrupts are performed during the interrupt INTxx processing. Before each interrupt request is acknowledged, the EI instruction is issued and the interrupt request is enabled.

Example 2. Multiple interrupts are not performed because interrupts are disabled



Interrupt requests are disabled (the EI instruction is not issued) in the interrupt INTxx processing. The interrupt request INTyy is not acknowledged and multiple interrupts are not performed. INTyy is held pending and is acknowledged after INTxx servicing is completed.

IE = 0: Interrupt request disabled

15.4.4 Putting interrupt requests on hold

If an interrupt (such as a maskable, non-maskable, or external interrupt) is requested when a certain type of instruction is being executed, the interrupt request will not be acknowledged until the instruction is completed. Such instructions include:

- Instructions that manipulate interrupt request flag registers 0, 1 (IF0 and IF1)
- Instructions that manipulate interrupt mask flag registers 0, 1 (MK0 and MK1)

CHAPTER 16 STANDBY FUNCTION

16.1 Standby Function and Configuration

16.1.1 Standby function

The standby function is used to reduce the power 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 power consumption as much as the STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

(2) STOP mode

This mode is set when the STOP instruction is executed. The STOP mode stops the main system clock oscillator and stops the entire system. The power consumption of the CPU can be substantially reduced in this mode.

The data memory can be retained at the low voltage (V_{DD} = 1.8 V). 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 that this mode can be used for intermittent operation. However, some time is required until the system clock oscillator stabilizes after the STOP mode has been 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.

Caution To set the STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

16.1.2 Standby function control register

The wait time after the STOP mode is released upon interrupt request until oscillation stabilizes is controlled by the oscillation stabilization time selection register (OSTS).

OSTS is set using an 8-bit memory manipulation instruction.

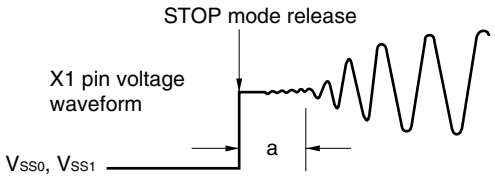
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, it takes 2^{15}/fx, not 2^{17}/fx, until the STOP mode is released by $\overline{\text{RESET}}$ input.

Figure 16-1. Format of Oscillation Stabilization Time Selection 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	Oscillation stabilization time selection
0	0	0	2^{12}/fx (819 μs)
0	1	0	2^{15}/fx (6.55 ms)
1	0	0	2^{17}/fx (26.2 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of release by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

16.2 Operation of Standby Function

16.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The operation status in the HALT mode is shown in the following table.

Table 16-1. HALT Mode Operating Status

Item	HALT Mode Operation Status While Main System Clock Is Running		HALT Mode Operation Status While Subsystem Clock Is Running	
	While the subsystem clock is running	While the subsystem clock is not running	While the main system clock is running	While the main system clock is not running
Main system clock generator	Oscillation enabled			Does not run.
CPU	Operation stopped			
Port (output latch)	Remains in the state existing before the selection of HALT mode.			
16-bit timer (TM50)	Operation enabled			Operation stopped
8-bit timer/event counters (TM00 and TM01)	Operation enabled			Operation enabled[Note 1]
8-bit timer (TM02)	Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
Watch timer	Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
Watchdog timer	Operation enabled			Operation stopped
Serial interface	Operation enabled			Operation enabled[Note 4]
A/D converter	Operation stopped			
LCD controller/driver	Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
Comparator	Operation enabled[Note 5]			
External interrupt	Operation enabled[Note 6]			

Notes 1. Operation is enabled only when TI0 or TI1 is selected as the count clock.

2. Operation is enabled while the main system clock is selected.

3. Operation is enabled while the subsystem clock is selected.

4. Operation is enabled in both 3-wire serial I/O and UART modes while an external clock is being used.

5. Operation is enabled while TM02 is operating, or as an external interrupt.

6. Maskable interrupt that is not masked

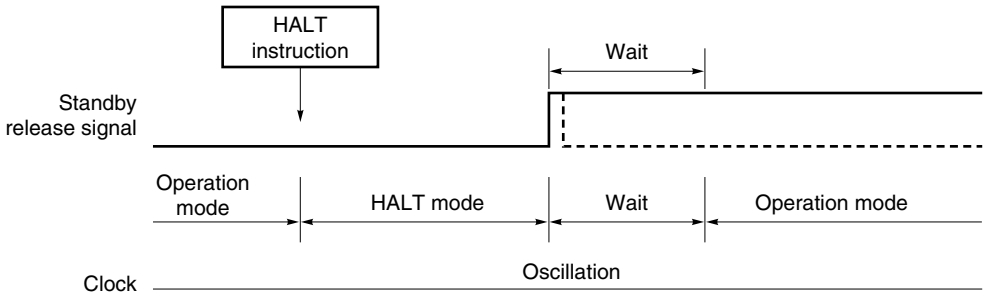
(2) Releasing HALT mode

The HALT mode can be released by the following three types of sources:

(a) Releasing by unmasked interrupt request

The HALT mode is released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed. If interrupts are disabled, the instruction at the next address is executed.

Figure 16-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken lines indicate the case where the interrupt request that has released the standby mode is acknowledged.

2. The wait time is as follows:

- When vectored interrupt servicing is performed: 9 to 10 clocks
- When vectored interrupt servicing is not performed: 1 to 2 clocks

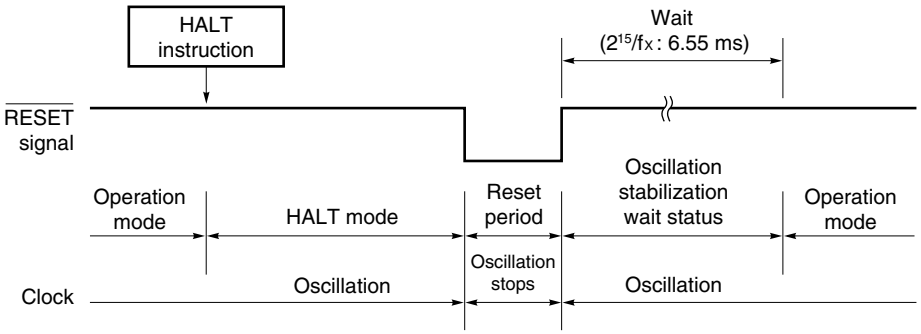
(b) Releasing by non-maskable interrupt request

The HALT mode is released regardless of whether interrupts are enabled or disabled, and vectored interrupt servicing is performed.

(c) Releasing by $\overline{\text{RESET}}$ input

When the HALT mode is released by the $\overline{\text{RESET}}$ signal, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 16-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 16-2. Operation After Release of HALT Mode

Releasing Source	MKxx	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	x	Retains HALT mode
Non-maskable interrupt request	–	x	Executes interrupt servicing
$\overline{\text{RESET}}$ input	–	–	Reset processing

x: 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.

Caution 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 selection register (OSTS) elapses, and then an operation mode is set.

The operation status in the STOP mode is shown in the following table.

Table 16-3. STOP Mode Operating Status

Item	STOP Mode Operation Status While Main System Clock Is Running	
	While the subsystem clock is running	While the subsystem clock is not running
Main system clock generator	Oscillation stopped	
CPU	Operation stopped	
Port (output latch)	Remains in the state existing before the selection of STOP mode.	
16-bit timer (TM50)	Operation stopped	
8-bit timer/event counter (TM00 and TM01)	Operation enabled[Note 1]	
8-bit timer (TM02)	Operation enabled[Note 2]	Operation stopped
Watch timer	Operation enabled[Note 2]	Operation stopped
Watchdog timer	Operation stopped	
Serial interface	Operation enabled[Note 3]	
A/D converter	Operation stopped	
LCD controller/driver	Operation enabled[Note 2]	Operation stopped
Comparator	Operation enabled[Notes 5, 6]	Operation enabled[Note 6]
External interrupt	Operation enabled[Note 4]	

Notes 1. Operation is enabled only when TI0 or TI1 is selected as the count clock.

2. Operation is enabled while the subsystem clock is selected.

3. Operation is enabled in both 3-wire serial I/O and UART modes while an external clock is being used.

4. Maskable interrupt that is not masked

5. Operation is enabled while TM02 is running.

6. Operation is enabled as an external interrupt.

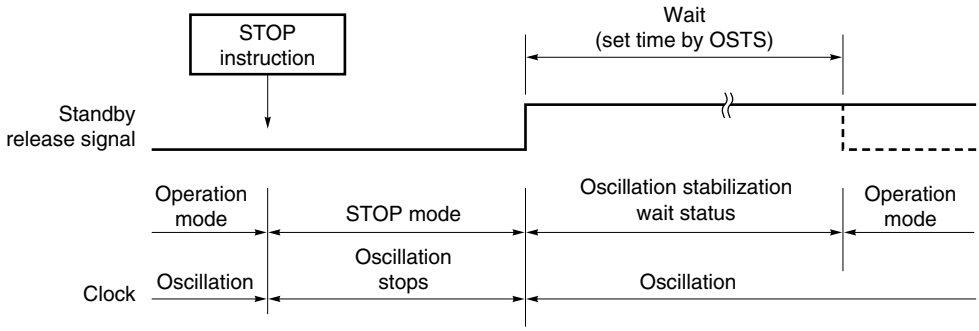
(2) Releasing STOP mode

The STOP mode can be released by the following two types of sources:

(a) Releasing by unmasked interrupt request

The STOP mode can be released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed, after the oscillation stabilization time has elapsed. If interrupts are disabled, the instruction at the next address is executed.

Figure 16-4. Releasing STOP Mode by Interrupt

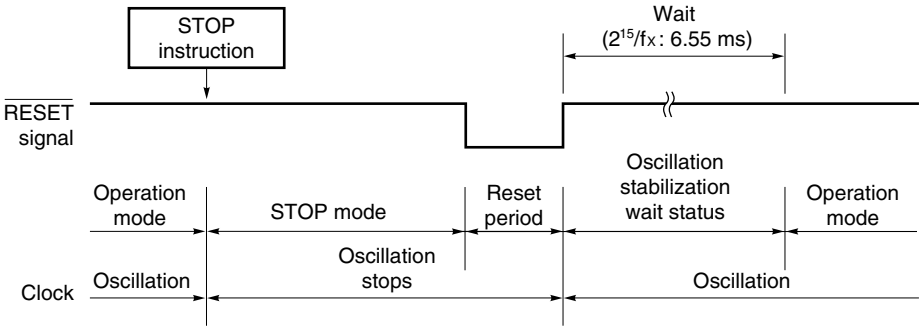


Remark The broken lines indicate the case where the interrupt request that has released the standby mode is acknowledged.

(b) Releasing by $\overline{\text{RESET}}$ input

When the STOP mode is released by the $\overline{\text{RESET}}$ signal, 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: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 16-4. Operation After Release of STOP Mode

Releasing Source	MKxx	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	x	Retains STOP mode
$\overline{\text{RESET}}$ input	–	–	Reset processing

x: Don't care

CHAPTER 17 RESET FUNCTION

The following two operations are available to generate reset signals.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by program loop time detected by the watchdog timer

The external and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware item is set to the status shown in Table 17-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution is started after the oscillation stabilization time (2^{15}/fx) has elapsed. The reset applied by the watchdog timer overflow is automatically released after reset, and program execution is started after the oscillation stabilization time (2^{15}/fx) has elapsed (see **Figures 17-2** through **17-4**).

Cautions 1. For an external reset, input a low level for 10 s or more to the $\overline{\text{RESET}}$ pin.

2. When the STOP mode is released by reset, the STOP mode contents are held during reset input. However, the port pins become high impedance.

Figure 17-1. Block Diagram of Reset Function

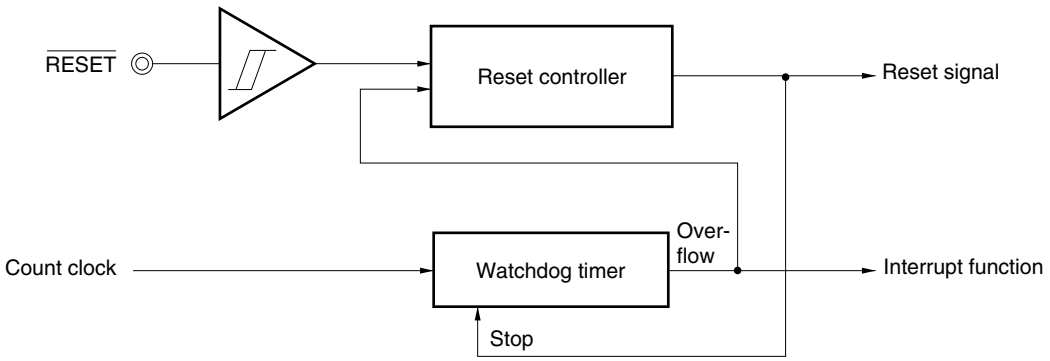


Figure 17-2. Reset Timing by $\overline{\text{RESET}}$ Input

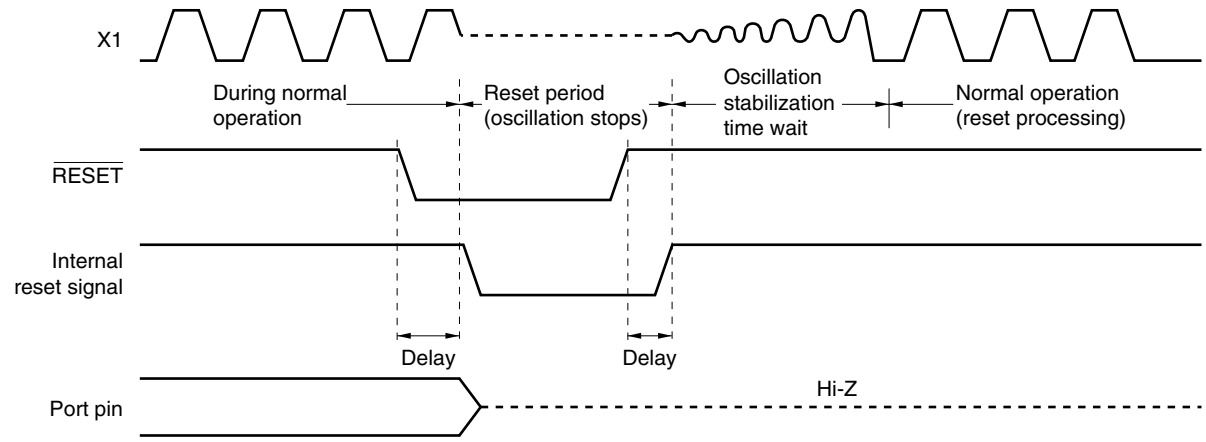


Figure 17-3. Reset Timing by Overflow in Watchdog Timer

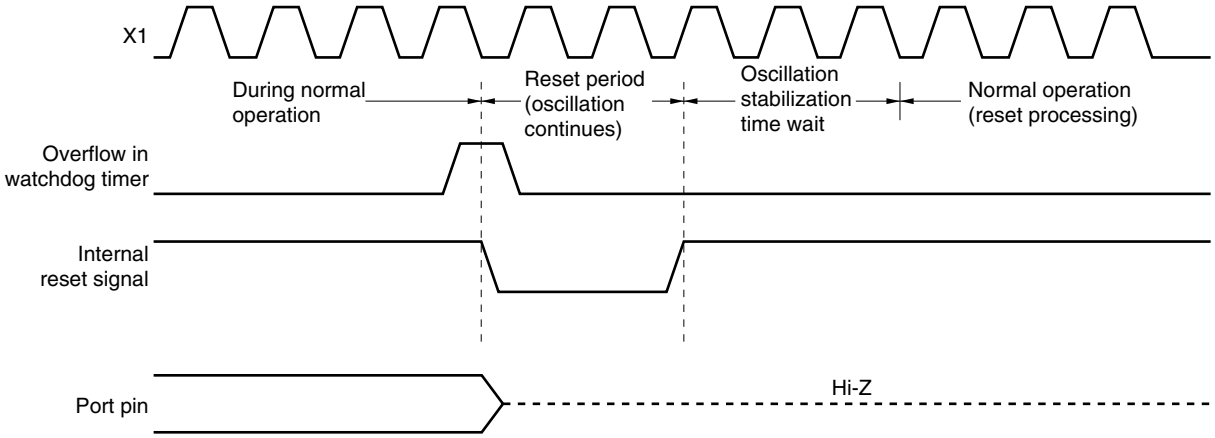


Figure 17-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

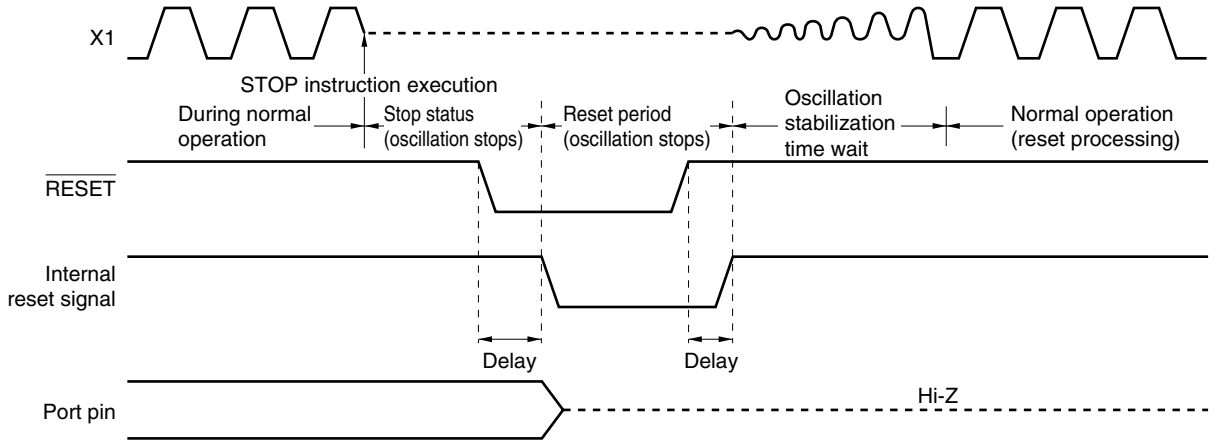


Table 17-1. Hardware Status After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Ports (P0, P2, P4, P5, P8, and P9) (Output latch)		00H
Port mode registers (PM0, PM2, PM4, PM5, PM8, and PM9)		FFH
Pull-up resistor option registers (PU0 to PU2)		00H
Processor clock control register (PCC)		02H
Suboscillation mode register (SCKM)		00H
Subclock control register (CSS)		00H
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer	Timer counter (TM50)	0000H
	Compare register (CR50)	FFFFH
	Capture register (TCP50)	Undefined
	Mode control register (TMC50)	00H
8-bit timer/event counter	Timer counters (TM00, TM01, and TM02)	00H
	Compare registers (CR00, CR01, and CR02)	Undefined
	Mode control registers (TMC00, TMC01, and TMC02)	00H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Timer clock selection register (TCL2)	00H
	Mode register (WDTM)	00H
A/D converter	Mode register (ADM0)	00H
	A/D input selection register (ADS0)	00H
	A/D conversion result register (ADCR0)	Undefined
Comparator	Mode register (CMPRM0)	00H

Notes 1. During reset input and oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined.
All other hardware remains unchanged after reset.

2. The post-reset values are retained in the standby mode.

Table 17-1. Hardware Status After Reset (2/2)

Hardware		Status After Reset
Serial interface	Mode register (CSIM00)	00H
	Asynchronous serial interface mode register (ASIM00)	00H
	Asynchronous serial interface status register (ASIS00)	00H
	Baud rate generator control register (BRGC00)	00H
	Transmit shift register (TXS00)	FFH
	Receive buffer register (RXB00)	Undefined
LCD controller/driver	LCD display mode register (LCDM0)	00H
	LCD port selector (LPS0)	00H
	LCD clock control register (LCDC0)	00H
Interrupts	Request flag registers (IF0 and IF1)	00H
	Mask flag registers (MK0 and MK1)	FFH
	External interrupt mode registers (INTM0 and INTM1)	00H
	Key return mode register (KRM00)	00H

CHAPTER 18 μPD78F9418A

The μPD78F9418A is a version with the internal ROM of the mask ROM version replaced by flash memory. The differences between the μPD78F9418A and the mask ROM versions are shown in Table 18-1.

Table 18-1. Differences Between μPD78F9418A and Mask ROM Versions

Item		Flash Memory Version	Mask ROM Version		
		μPD78F9418A	μPD789405A μPD789415A	μPD789406A μPD789416A	μPD789407A μPD789417A
Internal memory	ROM	32 KB (Flash memory)	12 KB	16 KB	24 KB
	High-speed RAM	512 bytes			
	LCD data RAM	28 bytes			
★ Pull-up resistor		32 (software control only)	36 (software control: 32, mask option control: 4)		
★ Divider resistor for LCD driving		Not provided	Can be specified on-chip by mask option		
IC pin		Not provided	Provided		
V_{PP} pin		Provided	Not provided		
Electrical specifications		Refer to **CHAPTER 21 ELECTRICAL SPECIFICATIONS**.			

Cautions 1. There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

2. When A/D conversion result register 0 (ADCR0) is used as the 8-bit A/D converter (μPD789407A Subseries), ADCR0 will be manipulated by an 8-bit memory manipulation instruction. When used as the 10-bit A/D converter (μPD789417A Subseries), ADCR0 will be manipulated by a 16-bit memory manipulation instruction.

However, when the μPD78F9418A is used as the flash memory version of the μPD789405A, 789406A, and 789407A, ADCR0 can be manipulated by an 8-bit memory manipulation instruction. In this case, use the object file assembled in the μPD789405A, 789406A, and 789407A.

★ 18.1 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 μPD78F9418A mounted on the target system (on-board). A flash memory program adapter (FA adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the program adapter are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

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 small-quantity, varied model production
- Easy data adjustment when starting mass production

18.1.1 Programming environment

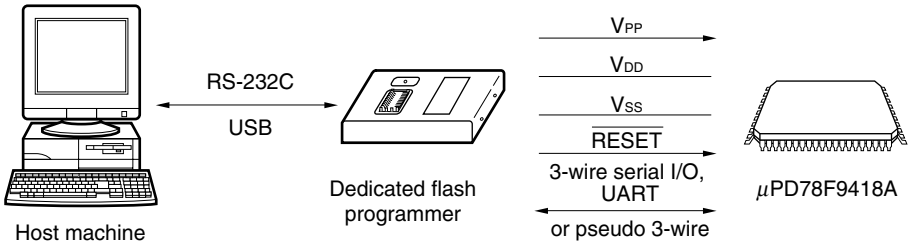
The following shows the environment required for μPD78F9418A flash memory programming.

When Flashpro III (part no. FL-PR3, PG-FP3) or Flashpro IV (part no. FL-PR4, PG-FP4) 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 18-1. Environment for Writing Program to Flash Memory



18.1.2 Communication mode

Use the communication mode shown in Table 18-2 to perform communication between the dedicated flash programmer and μPD78F9418A.

Table 18-2. Communication Mode List

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU Clock		Multiple Rate		
			In Flashpro	On Target Board			
3-wire serial I/O	SIO ch-0 (3-wire, sync.)	100 Hz to 1.25 MHz[Note 2]	1, 2, 4, 5 MHz[Notes 2, 3]	1 to 5 MHz[Note 2]	1.0	SI/RxD/P22 SO/TxD/P21 $\overline{SCK}$/ASCK/P20	0
UART	UART ch-0 (Async.)	4,800 to 76,800 bps [Notes 2, 4]	5 MHz[Note 5]	4.91 or 5 MHz[Note 2]	1.0	RxD/SI/P22 TxD/SO/P21	8
Pseudo 3-wire	Port A (Pseudo-3 wire)	100 Hz to 1 kHz	1, 2, 4, 5 MHz[Notes 2, 3]	1 to 5 MHz[Note 2]	1.0	P01 P02 P00	12
	Port B (Pseudo-3 wire)					P40/KR0 P41/KR1 P42/KR2	13

Notes 1. Selection items for TYPE settings on the dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)).

2. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 21 ELECTRICAL SPECIFICATIONS**.

3. 2 or 4 MHz only for Flashpro III

4. Because signal wave slew also affects UART communication, in addition to the baud rate error, thoroughly evaluate the slew and baud rate error.

5. Only for Flashpro IV. However, when using Flashpro III, be sure to select the clock of the resonator on the board. UART cannot be used with the clock supplied by Flashpro III.

Figure 18-2. Communication Mode Selection Format

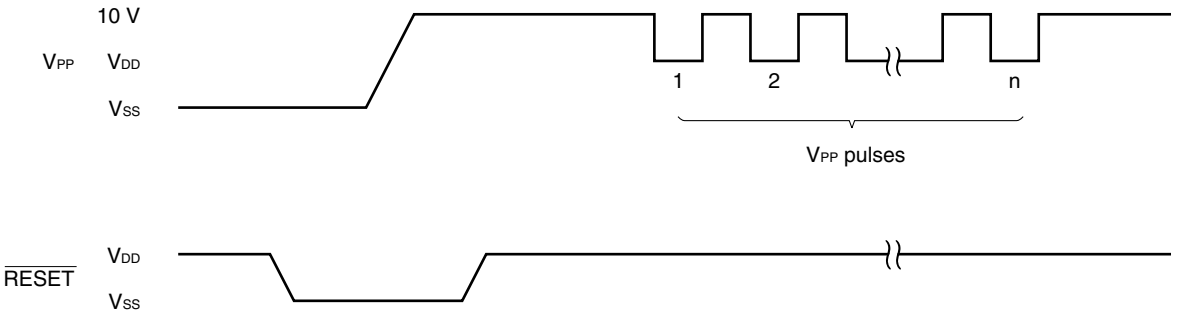
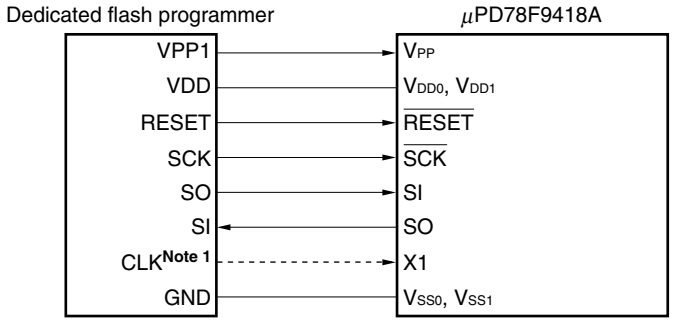
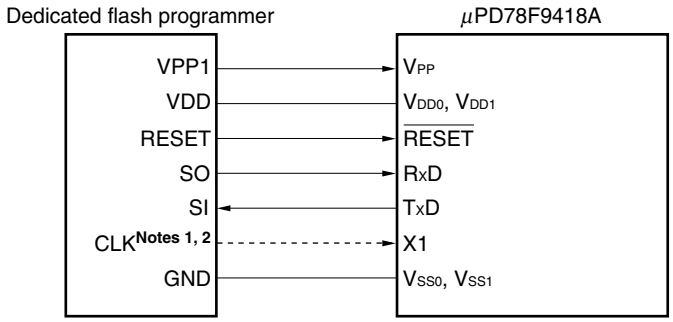


Figure 18-3. Example of Connection with Dedicated Flash Programmer

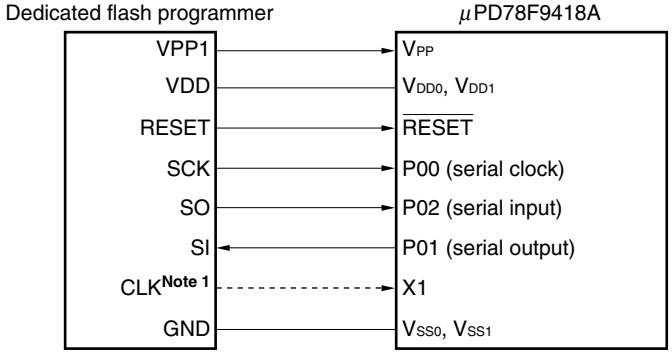
(a) 3-wire serial I/O



(b) UART



(c) Pseudo 3-wire (when P0 is used)



Notes 1. Connect this pin when the system clock is supplied from the dedicated flash programmer. If a resonator is already connected to the X1 pin, do not connect to the CLK pin.

2. When using UART with Flashpro III, the clock of the resonator connected to the X1 pin must be used, so do not connect to the CLK pin.

Caution The V_{DD} pin, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. When using the power supply connected to the V_{DD} pin, supply voltage before starting programming.

If Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, the following signals are generated for the μPD78F9418A. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 18-3. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	3-Wire Serial I/O	UART	Pseudo 3-Wire
VPP1	Output	Write voltage	V_{PP}	◎	◎	◎
VPP2	–	–	–	×	×	×
VDD	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD0}, V_{DD1}	◎Note	◎Note	◎Note
GND	–	Ground	V_{SS0}, V_{SS1}	◎	◎	◎
CLK	Output	Clock output	X1	○	○	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI	Input	Receive signal	SO/TxD/P01/P41	◎	◎	◎
SO	Output	Transmit signal	SI/RxD/P02/P42	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK}}$/P00/P40	◎	×	◎
HS	Input	Handshake signal	–	×	×	×

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 does not need to be connected.
×: Pin does not need to be connected.

18.1.3 On-board pin connections

When programming on the target system, provide a connector on the target system to connect to the dedicated flash programmer.

There may be cases in which an on-board function that switches from the normal operation mode to flash memory programming mode is required.

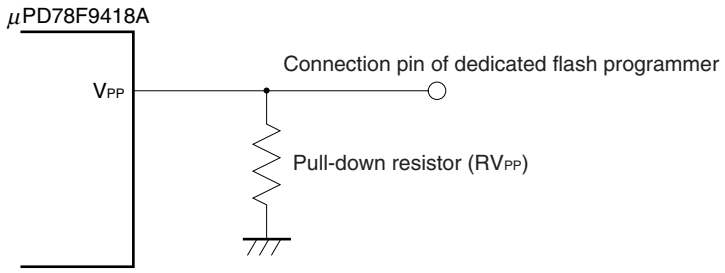
<V_{PP} pin>

Input 0 V to the V_{PP} pin in the normal operation mode. A write voltage of 10.0 V (TYP.) is supplied to the V_{PP} pin in the flash memory programming mode. Therefore, connect the V_{PP} pin using method (1) or (2) below.

(1) Connect a pull-down resistor of RV_{PP} = 10 kΩ to the V_{PP} pin.
(2) Set the jumper on the board to switch the input of V_{PP} pin to the programmer side or directly to GND.

The following shows an example of V_{PP} pin connection.

Figure 18-4. V_{PP} Pin Connection Example



<Serial interface pins>

The following shows the pins used by each serial interface.

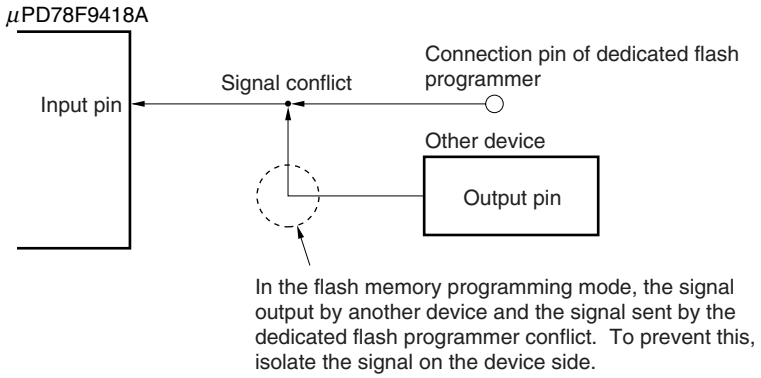
Serial Interface	Pins Used
3-wire serial I/O	SI, SO, $\overline{SCK}$
UART	RxD, TxD
Pseudo 3-wire	P00, P01, P02
	P40, P41, P42

Note that signal conflict or malfunction of other devices may occur when an on-board serial interface pin that is connected to another device is connected to the dedicated flash programmer.

(1) Signal conflict

A signal conflict occurs if the dedicated flash programmer (output) is connected to a serial interface pin (input) connected to another device (output). To prevent this signal conflict, isolate the connection with the other device or put the other device in the output high impedance status.

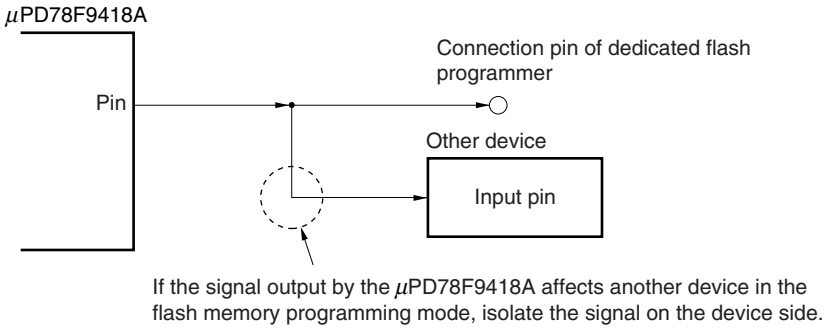
Figure 18-5. Signal Conflict (Serial Interface Input Pin)

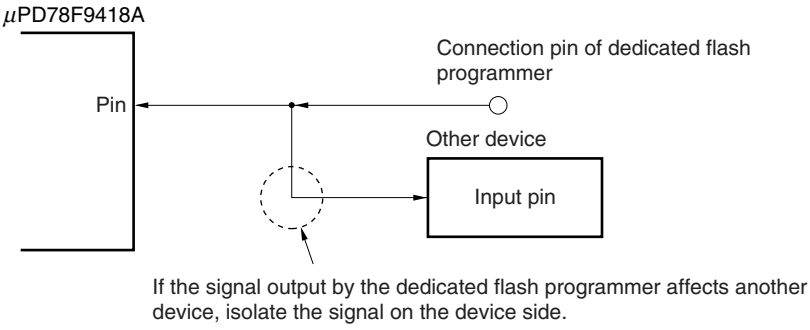


(2) Malfunction of another device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) connected to another device (input), a signal may be output to the device, causing a malfunction. To prevent such malfunction, isolate the connection with other device or set so that the input signal to the device is ignored.

Figure 18-6. Malfunction of Another Device



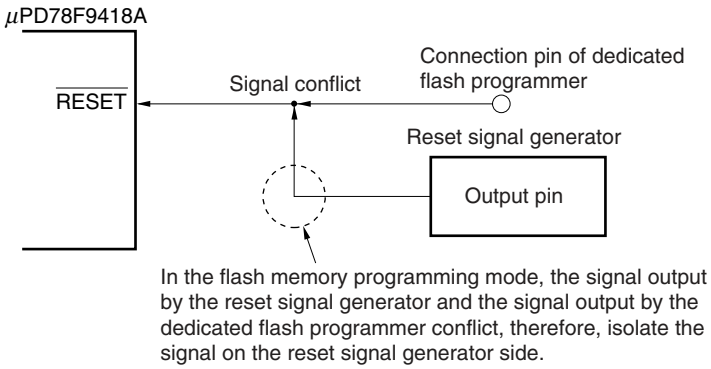


<$\overline{\text{RESET}}$ pin>

When the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin connected to the reset signal generator on the board, a signal conflict occurs. To prevent this signal conflict, isolate the connection with the reset signal generator.

If a reset signal is input from the user system in the flash memory programming mode, a normal programming operation will not be performed. Do not input signals other than reset signals from the dedicated flash programmer during this period.

Figure 18-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)



<Port pins>

Shifting to the flash memory programming mode sets all the pins except those used for flash memory programming communication to the status immediately after reset.

Therefore, if the external device does not acknowledge an initial status such as the output high impedance status, connect the external device to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor.

<Oscillation pins>

When using an on-board clock, connection of X1, X2, XT1, and XT2 must conform to the methods in the normal operation mode.

When using the clock output of the flash programmer, directly connect it to the X1 pin with the on-board main oscillator disconnected, and leave the X2 pin open. For the subclock, connection conforms to that in the normal operation mode.

<Power supply>

To use the power output of 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, connection must conform to that in the normal operation mode. However, because the voltage is monitored by the flash programmer, therefore, VDD of the flash programmer must be connected.

For the other power supply pins (AV_{DD}, AV_{REF}, AV_{SS}), supply the same power supply as in the normal operation mode.

<Other pins>

Handle the other pins (S0 to S15, COM0 to COM3, V_{LC0} to V_{LC2}, BIAS) in the same way as in the normal operation mode.

18.1.4 Connection when using flash memory writing adapter

The following shows an example of the recommended connection when using the flash memory writing adapter.

Figure 18-8. Example of Flash Memory Writing Adapter Connection When Using 3-Wire Serial I/O Mode

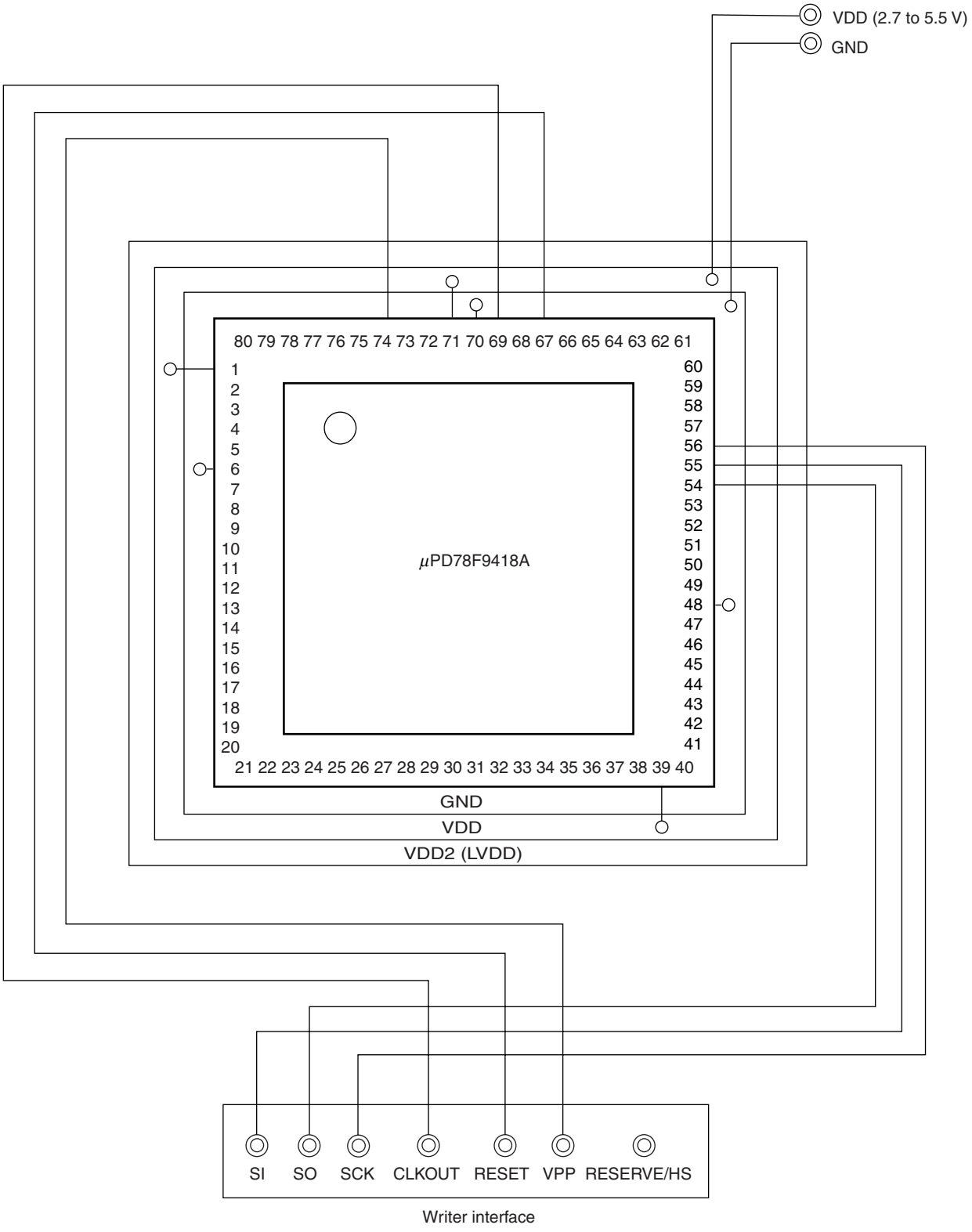


Figure 18-9. Example of Flash Memory Writing Adapter Connection When Using UART Mode

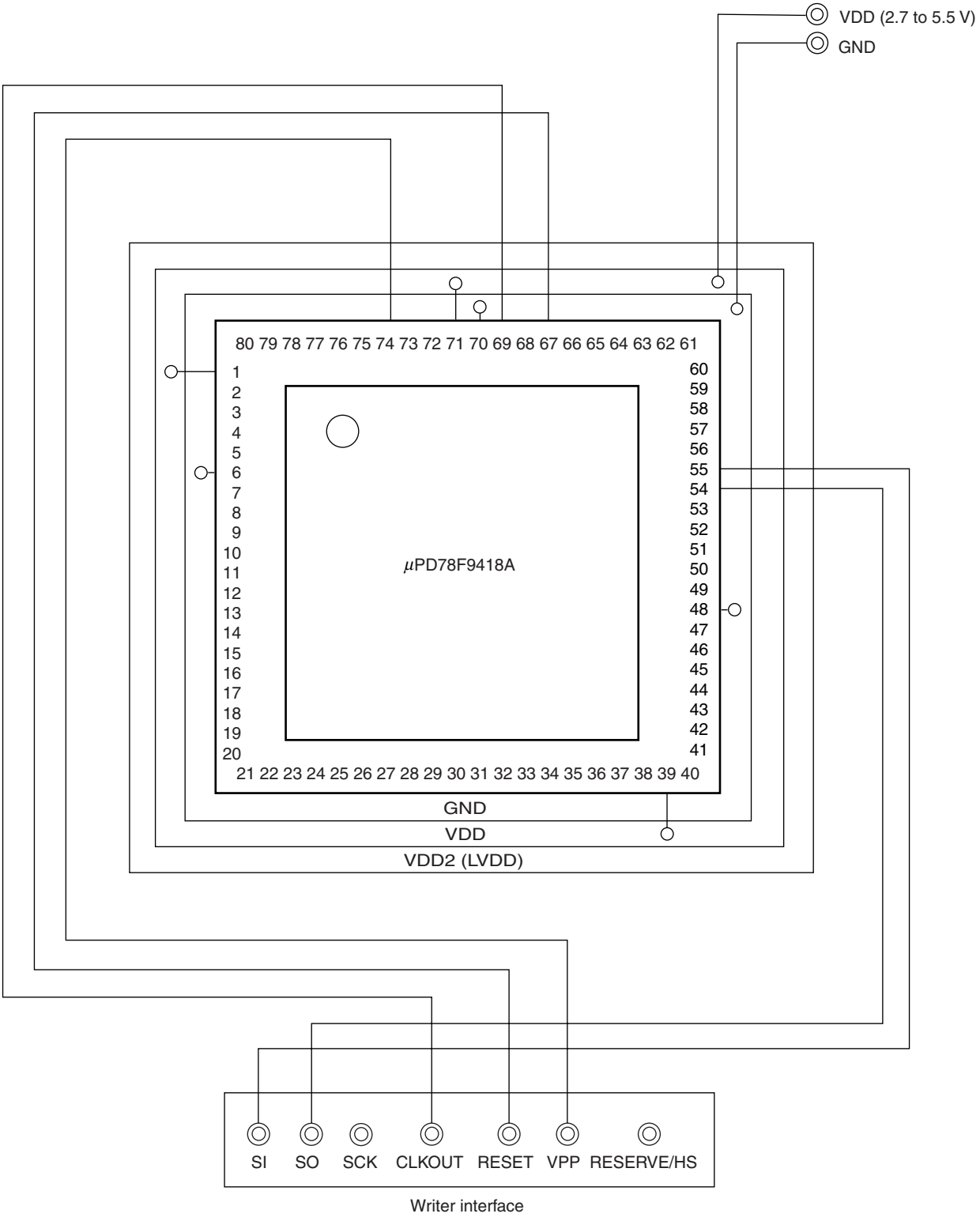
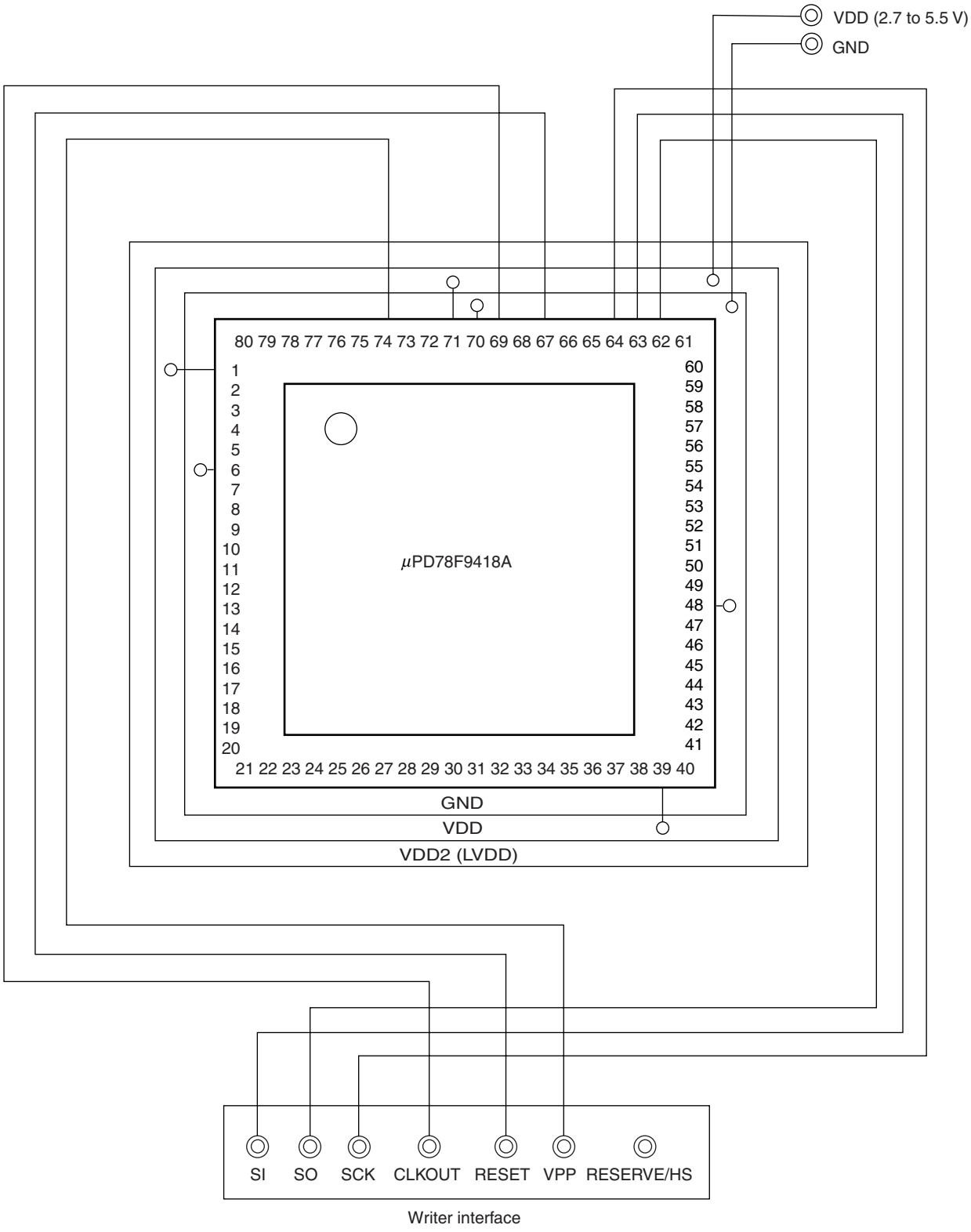


Figure 18-10. Example of Flash Memory Writing Adapter Connection When Using Pseudo 3-Wire Mode (When P0 Is Used)



CHAPTER 19 MASK OPTIONS

The mask ROM versions of the μPD789407A and μPD789417A Subseries have the following mask options.

Caution The flash memory version does not have a mask option.

19.1 Mask Option for Pins

Table 19-1. Selection of Mask Option for Pins

Pin	Mask Option
P50 to P53	Whether a pull-up resistor is to be incorporated can be specified in 1-bit units.

For P50 to P53 (port 5), a mask option is used to specify whether a pull-up resistor is to be incorporated. The mask option is selectable in 1-bit units.

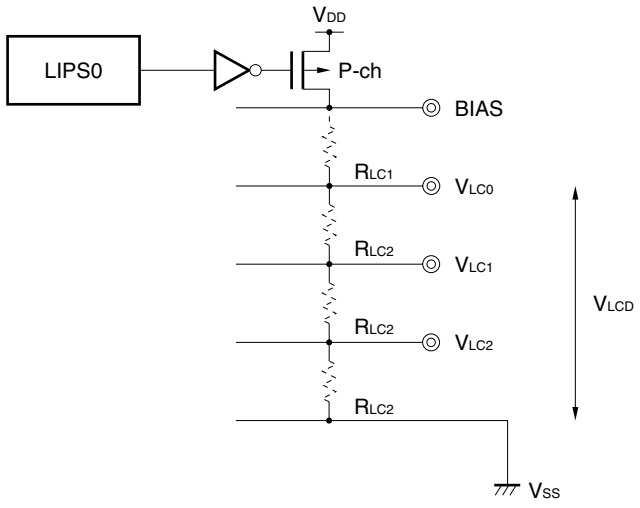
19.2 Mask Option for Voltage Division Resistor for LCD Driver

A mask option is used to specify whether a voltage division resistor is to be incorporated for the LCD driver, as listed below:

Table 19-2. Combination of Selectable Voltage Division Resistor

		R_{LC1} ($2 \times R_{LC2}$)		
		None	20 kΩ	200 kΩ
R_{LC2}	None	○	–	–
	10 kΩ	○	○	–
	100 kΩ	○	–	○

○: Selectable
–: Not selectable



LIPS0: Bit 4 of LCD display mode register 0 (LCDM0)

CHAPTER 20 INSTRUCTION SET

This chapter lists the instruction set of the μPD789407A and 789417A Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**.

20.1 Operation

20.1.1 Operand identifiers and description methods

Operands are described in the Operands column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers r and rp, either functional names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 20-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol
saddr saddrp	FE20H to FF1FH Immediate data or label FE20H to FF1FH Immediate data or label (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or label (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or label (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark See **Table 3-3** for symbols of special function registers.

20.1.2 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
IE:	Interrupt request enable flag
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	Memory contents indicated by address or register contents in parenthesis
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
$\overline{\quad}$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

20.1.3 Description of “Flag” column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
x:	Set/cleared according to the result
R:	Previously saved value is restored

20.2 Operation List

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
MOV	r, #byte	3	6	r ← byte	
	saddr, #byte	3	6	(saddr) ← byte	
	sfr, #byte	3	6	sfr ← byte	
	A, r[Note 1]	2	4	A ← r	
	r, A[Note 1]	2	4	r ← A	
	A, saddr	2	4	A ← (saddr)	
	saddr, A	2	4	(saddr) ← A	
	A, sfr	2	4	A ← sfr	
	sfr, A	2	4	sfr ← A	
	A, !addr16	3	8	A ← (addr16)	
	!addr16, A	3	8	(addr16) ← A	
	PSW, #byte	3	6	PSW ← byte	x x x
	A, PSW	2	4	A ← PSW	
	PSW, A	2	4	PSW ← A	x x x
	A, [DE]	1	6	A ← (DE)	
	[DE], A	1	6	(DE) ← A	
	A, [HL]	1	6	A ← (HL)	
	[HL], A	1	6	(HL) ← A	
	A, [HL+byte]	2	6	A ← (HL + byte)	
	[HL+byte], A	2	6	(HL + byte) ← A	
XCH	A, X	1	4	A ↔ X	
	A, r[Note 2]	2	6	A ↔ r	
	A, saddr	2	6	A ↔ (saddr)	
	A, sfr	2	6	A ↔ sfr	
	A, [DE]	1	8	A ↔ (DE)	
	A, [HL]	1	8	A ↔ (HL)	
	A, [HL+byte]	2	8	A ↔ (HL + byte)	

Notes 1. Except r = A.

2. Except r = A, X.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
MOVW	rp, #word	3	6	rp ← word	
	AX, saddrp	2	6	AX ← (saddrp)	
	saddrp, AX	2	8	(saddrp) ← AX	
	AX, rp[Note]	1	4	AX ← rp	
	rp, AX[Note]	1	4	rp ← AX	
XCHW	AX, rp[Note]	1	8	AX ↔ rp	
ADD	A, #byte	2	4	A, CY ← A + byte	x x x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	x x x
	A, r	2	4	A, CY ← A + r	x x x
	A, saddr	2	4	A, CY ← A + (saddr)	x x x
	A, !addr16	3	8	A, CY ← A + (addr16)	x x x
	A, [HL]	1	6	A, CY ← A + (HL)	x x x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte)	x x x
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	x x x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	x x x
	A, r	2	4	A, CY ← A + r + CY	x x x
	A, saddr	2	4	A, CY ← A + (saddr) + CY	x x x
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	x x x
	A, [HL]	1	6	A, CY ← A + (HL) + CY	x x x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte) + CY	x x x
SUB	A, #byte	2	4	A, CY ← A – byte	x x x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	x x x
	A, r	2	4	A, CY ← A – r	x x x
	A, saddr	2	4	A, CY ← A – (saddr)	x x x
	A, !addr16	3	8	A, CY ← A – (addr16)	x x x
	A, [HL]	1	6	A, CY ← A – (HL)	x x x
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte)	x x x

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	x	x	x
	A, r	2	4	A, CY ← A – r – CY	x	x	x
	A, saddr	2	4	A, CY ← A – (saddr) – CY	x	x	x
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	x	x	x
	A, [HL]	1	6	A, CY ← A – (HL) – CY	x	x	x
	A, [HL+byte]	2	6	A, CY ← A– (HL + byte) – CY	x	x	x
AND	A, #byte	2	4	A ← A ∧ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	x		
	A, r	2	4	A ← A ∧ r	x		
	A, saddr	2	4	A ← A ∧ (saddr)	x		
	A, !addr16	3	8	A ← A ∧ (addr16)	x		
	A, [HL]	1	6	A ← A ∧ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∧ (HL + byte)	x		
OR	A, #byte	2	4	A ← A ∨ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	x		
	A, r	2	4	A ← A ∨ r	x		
	A, saddr	2	4	A ← A ∨ (saddr)	x		
	A, !addr16	3	8	A ← A ∨ (addr16)	x		
	A, [HL]	1	6	A ← A ∨ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∨ (HL + byte)	x		
XOR	A, #byte	2	4	A ← A ∨̄ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨̄ byte	x		
	A, r	2	4	A ← A ∨̄ r	x		
	A, saddr	2	4	A ← A ∨̄ (saddr)	x		
	A, !addr16	3	8	A ← A ∨̄ (addr16)	x		
	A, [HL]	1	6	A ← A ∨̄ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∨̄ (HL + byte)	x		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
CMP	A, #byte	2	4	A – byte	x x x
	saddr, #byte	3	6	(saddr) – byte	x x x
	A, r	2	4	A – r	x x x
	A, saddr	2	4	A – (saddr)	x x x
	A, !addr16	3	8	A – (addr16)	x x x
	A, [HL]	1	6	A – (HL)	x x x
	A, [HL+byte]	2	6	A – (HL + byte)	x x x
ADDW	AX, #word	3	6	AX, CY ← AX + word	x x x
SUBW	AX, #word	3	6	AX, CY ← AX – word	x x x
CMPW	AX, #word	3	6	AX – word	x x x
INC	r	2	4	r ← r + 1	x x
	saddr	2	4	(saddr) ← (saddr) + 1	x x
DEC	r	2	4	r ← r – 1	x x
	saddr	2	4	(saddr) ← (saddr) – 1	x x
INCW	rp	1	4	rp ← rp + 1	
DECW	rp	1	4	rp ← rp – 1	
ROR	A, 1	1	2	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$	x
ROL	A, 1	1	2	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$	x
RORC	A, 1	1	2	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$	x
ROLC	A, 1	1	2	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$	x
SET1	saddr.bit	3	6	(saddr.bit) ← 1	
	sfr.bit	3	6	sfr.bit ← 1	
	A.bit	2	4	A.bit ← 1	
	PSW.bit	3	6	PSW.bit ← 1	x x x
	[HL].bit	2	10	(HL).bit ← 1	
CLR1	saddr.bit	3	6	(saddr.bit) ← 0	
	sfr.bit	3	6	sfr.bit ← 0	
	A.bit	2	4	A.bit ← 0	
	PSW.bit	3	6	PSW.bit ← 0	x x x
	[HL].bit	2	10	(HL).bit ← 0	
SET1	CY	1	2	CY ← 1	1
CLR1	CY	1	2	CY ← 0	0
NOT1	CY	1	2	CY ← CY	x

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
CALL	!addr16	3	6	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$	
CALLT	[addr5]	1	8	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$	
RET		1	6	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
RETI		1	8	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, PSW $\leftarrow (SP+2)$, $SP \leftarrow SP+3$, NMIS $\leftarrow 0$	R R R
PUSH	PSW	1	2	$(SP-1) \leftarrow$ PSW, $SP \leftarrow SP-1$	
	rp	1	4	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$	
POP	PSW	1	4	PSW $\leftarrow (SP)$, $SP \leftarrow SP+1$	R R R
	rp	1	6	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
MOVW	SP, AX	2	8	$SP \leftarrow$ AX	
	AX, SP	2	6	AX $\leftarrow$ SP	
BR	!addr16	3	6	PC $\leftarrow$ addr16	
	$addr16	2	6	PC $\leftarrow$ PC + 2 + jdisp8	
	AX	1	6	$PC_H \leftarrow$ A, $PC_L \leftarrow$ X	
BC	$saddr16	2	6	PC $\leftarrow$ PC + 2 + jdisp8 if CY = 1	
BNC	$saddr16	2	6	PC $\leftarrow$ PC + 2 + jdisp8 if CY = 0	
BZ	$saddr16	2	6	PC $\leftarrow$ PC + 2 + jdisp8 if Z = 1	
BNZ	$saddr16	2	6	PC $\leftarrow$ PC + 2 + jdisp8 if Z = 0	
BT	saddr.bit, $addr16	4	10	PC $\leftarrow$ PC + 4 + jdisp8 if (saddr.bit) = 1	
	sfr.bit, $addr16	4	10	PC $\leftarrow$ PC + 4 + jdisp8 if sfr.bit = 1	
	A.bit, $addr16	3	8	PC $\leftarrow$ PC + 3 + jdisp8 if A.bit = 1	
	PSW.bit, $addr16	4	10	PC $\leftarrow$ PC + 4 + jdisp8 if PSW.bit = 1	
BF	saddr.bit, $addr16	4	10	PC $\leftarrow$ PC + 4 + jdisp8 if (saddr.bit) = 0	
	sfr.bit, $addr16	4	10	PC $\leftarrow$ PC + 4 + jdisp8 if sfr.bit = 0	
	A.bit, $addr16	3	8	PC $\leftarrow$ PC + 3 + jdisp8 if A.bit = 0	
	PSW.bit, $addr16	4	10	PC $\leftarrow$ PC + 4 + jdisp8 if PSW.bit = 0	
DBNZ	B, $addr16	2	6	B $\leftarrow$ B–1, then PC $\leftarrow$ PC + 2 + jdisp8 if B $\neq 0$	
	C, $addr16	2	6	C $\leftarrow$ C–1, then PC $\leftarrow$ PC + 2 + jdisp8 if C $\neq 0$	
	saddr, $addr16	3	8	(saddr) $\leftarrow$ (saddr) – 1, then PC $\leftarrow$ PC + 3 + jdisp8 if (saddr) $\neq 0$	
NOP		1	2	No Operation	
EI		3	6	IE $\leftarrow$ 1 (Enable interrupt)	
DI		3	6	IE $\leftarrow$ 0 (Disable interrupt)	
HALT		1	2	Set HALT mode	
STOP		1	2	Set STOP mode	

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

20.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

1st Operand \ 2nd Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV 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											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											
[HL+byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ 2nd Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
SP		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

1st Operand \ 2nd Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand / 1st Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

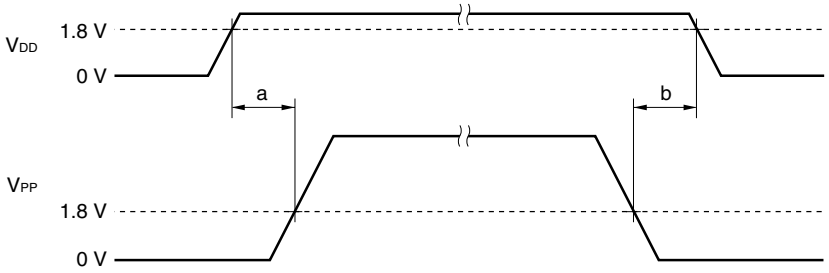
RET, RETI, NOP, EI, DI, HALT, STOP

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}	$AV_{DD} - 0.3\ V \le V_{DD} \le AV_{DD} + 0.3\ V$ $AV_{REF} \le V_{DD} + 0.3\ V$ $AV_{REF} \le AV_{DD} + 0.3\ V$		−0.3 to +6.5	V
	AV_{DD}				
	AV_{REF}				
	V_{PP}	μPD78F9418A only **Note**		−0.3 to +10.5	V
Input voltage	V_{I1}	Pins other than P50 to P53		−0.3 to V_{DD} + 0.3	V
	V_{I2}	P50 to P53	N-ch open drain	−0.3 to +13	V
Output voltage	V_O			−0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	1 pin		−10	mA
		Total for all pins		−30	mA
Output current, low	I_{OL}	1 pin		30	mA
		Total for all pins		160	mA
Operating ambient temperature	T_A	In normal operation mode		−40 to +85	°C
		During flash memory programming		10 to 40	°C
Storage temperature	T_{stg}	Mask ROM version		−65 to +150	°C
		μPD78F9418A		−40 to +125	°C

Note Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

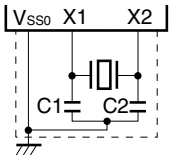
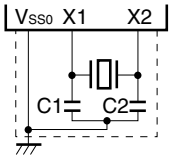
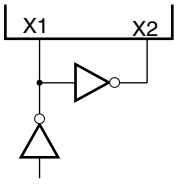
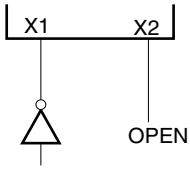
- **When supply voltage rises**
  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (1.8 V) of the operating voltage range (see a in the figure below).
- **When supply voltage drops**
  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (1.8 V) of the operating voltage range of V_{DD} (see b in the figure below).



Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Main System Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)Note 1	V_{DD} = Oscillation voltage range	1.0		5.0	MHz
		Oscillation stabilization timeNote 2	After V_{DD} has reached MIN. of oscillation start voltage			4	ms
Crystal resonator		Oscillation frequency (f_X)Note 1		1.0		5.0	MHz
		Oscillation stabilization timeNote 2	V_{DD} = 4.5 to 5.5 V			10	ms
			V_{DD} = 1.8 to 5.5 V			30	ms
External clock		X1 input frequency (f_X)Note 1		1.0		5.0	MHz
		X1 input high-/low-level widths (t_{XH}, t_{XL})		85		500	ns
		X1 input frequency (f_X)Note 1	V_{DD} = 2.7 to 5.5 V	1.0		5.0	MHz
		X1 input high-/low-level widths (t_{XH}, t_{XL})	V_{DD} = 2.7 to 5.5 V	85		500	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use a resonator whose oscillation is stabilized within the oscillation wait time.

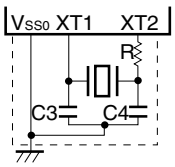
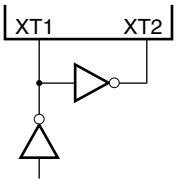
Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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_{SS0}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. When the main system clock is stopped and the device is operating on the subsystem clock, wait until the oscillation stabilization time has been secured by the program before switching back to the main system clock.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Subsystem Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})Note 1		32	32.768	35	kHz
		Oscillation stabilization timeNote 2	V_{DD} = 4.5 to 5.5 V		1.2	2	s
			V_{DD} = 1.8 to 5.5 V			10	s
External clock		XT1 input frequency (f_{XT})Note 1		32		35	kHz
		XT1 input high-/low-level widths (t_{XTH}, t_{XTL})		14.3		15.6	μs

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use a resonator whose oscillation is stabilized within the oscillation wait time.

Cautions 1. When using the subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS0}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (1/3)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin					–1	mA
		Total for all pins					–15	mA
Output current, low	I_{OL}	Per pin					10	mA
		Total for all pins					80	mA
Input voltage, high	V_{IH1}	P00 to P03, P46, P47, P60 to P66, P80 to P87, P90 to P93		V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH2}	P50 to P53	N-ch open drain	V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		12	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		12	V
			On-chip pull-up resistor	V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH3}	$\overline{\text{RESET}}$, P20 to P27, P40 to P45		V_{DD} = 2.7 to 5.5 V	0.8V_{DD}		V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH4}	X1, X2, XT1, XT2		V_{DD} = 1.8 to 5.5 V	V_{DD} – 0.1		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P03, P46, P47, P60 to P66, P80 to P87, P90 to P93		V_{DD} = 2.7 to 5.5 V	0		0.3V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL2}	P50 to P53		V_{DD} = 2.7 to 5.5 V	0		0.3V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL3}	$\overline{\text{RESET}}$, P20 to P27, P40 to P45		V_{DD} = 2.7 to 5.5 V	0		0.2V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL4}	X1, X2, XT1, XT2		V_{DD} = 1.8 to 5.5 V	0		0.1	V
Output voltage, high	V_{OH}	I_{OH} = –1 mA		V_{DD} = 4.5 to 5.5 V	V_{DD} – 1.0			V
		I_{OH} = –100 μA		V_{DD} = 1.8 to 5.5 V	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	Pins other than P50 to P53		V_{DD} = 4.5 to 5.5 V I_{OL} = 10 mA			1.0	V
				V_{DD} = 1.8 to 5.5 V I_{OL} = 400 μA			0.5	V
	V_{OL2}	P50 to P53		V_{DD} = 4.5 to 5.5 V I_{OL} = 10 mA			1.0	V
				V_{DD} = 1.8 to 5.5 V I_{OL} = 1.6 mA			0.4	V
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}		Pins other than P50 to P53 (N-ch open drain), X1, X2, XT1, and XT2			3	μA
	I_{LIH2}			X1, X2, XT1, XT2			20	μA
	I_{LIH3}	V_{IN} = 12 V		P50 to P53 (N-ch open drain)			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V		Pins other than P50 to P53 (N-ch open drain), X1, X2, XT1, and XT2			–3	μA
	I_{LIL2}			X1, X2, XT1, XT2			–20	μA
	I_{LIL3}			P50 to P53 (N-ch open drain)			–3[Note]	μA

Note A low-level input leakage current of –30 μA (MAX.) flows only during the 1-cycle time after a read instruction is executed to P50 to P53 when on-chip pull-up resistors are not connected to P50 to P53 (specified by mask option) and P50 to P53 are set to input mode. At times other than this, a –3 μA (MAX.) current flows.

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} = 1.8 to 5.5 V) (2/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
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_1	V_{IN} = 0 V, pins other than P50 to P53		50	100	200	kΩ
Mask option pull-up resistor[Note 1]	R_2	V_{IN} = 0 V, P50 to P53		15	30	60	kΩ
Supply current (mask ROM version)	I_{DD1}[Note 2]	5.0 MHz crystal oscillation operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%[Note 5]		2.0	4.0	mA
			V_{DD} = 3.0 V ±10%[Note 6]		0.6	1.2	mA
			V_{DD} = 2.0 V ±10%[Note 6]		0.3	0.6	mA
	I_{DD2}[Note 2]	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%[Note 5]		1.1	2.2	mA
			V_{DD} = 3.0 V ±10%[Note 6]		0.4	0.8	mA
			V_{DD} = 2.0 V ±10%[Note 6]		0.2	0.4	mA
	I_{DD3}[Note 2]	32.768 kHz crystal oscillation operating mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	V_{DD} = 5.0 V ±10%		30	90	μA
			V_{DD} = 3.0 V ±10%		9	50	μA
			V_{DD} = 2.0 V ±10%		4	25	μA
	I_{DD4}[Note 2]	32.768 kHz crystal oscillation HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	V_{DD} = 5.0 V ±10%		25	55	μA
			V_{DD} = 3.0 V ±10%		5	25	μA
			V_{DD} = 2.0 V ±10%		2.5	12.5	μA
	I_{DD5}[Note 2]	32.768 kHz crystal oscillation STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	5.0	μA
			T_A = 25°C		0.05	3.0	μA
			V_{DD} = 2.0 V ±10%		0.05	3.0	μA
	I_{DD6}[Note 3]	5.0 MHz crystal oscillation A/D operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%		2.6	6.0	mA
			V_{DD} = 3.0 V ±10%		1.2	3.6	mA
			V_{DD} = 2.0 V ±10%		0.9	2.7	mA

Notes 1. Mask ROM version only

2. The current flowing to AV_{REF} (A/D operation ON (ADCS0 = 1)), AV_{DD} current, and the port current (including the current flowing through the on-chip pull-up resistors) is not included.

3. The current flowing to AV_{REF} (A/D operation ON (ADCS0 = 1)) and the port current (including the current flowing through the on-chip pull-up resistors) is not included. For the current flowing to AV_{REF}, refer to the parameter of "Resistance between AV_{REF} and AV_{SS}" in the **8-Bit A/D Converter Characteristics** and **10-Bit A/D Converter Characteristics**.

4. When the main system clock is stopped

5. High-speed mode operation (when processor clock control register (PCC) is set to 00H)

6. Low-speed mode operation (when PCC is set to 02H)

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} = 1.8 to 5.5 V) (3/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current (μPD78F9418A)	I_{DD1} Note 1	5.0 MHz crystal oscillation operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10% Note 4		5.0	14.0	mA
			V_{DD} = 3.0 V ±10% Note 5		2.0	5.0	mA
			V_{DD} = 2.0 V ±10% Note 5		1.5	3.0	mA
	I_{DD2} Note 1	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10% Note 4		2.0	6.0	mA
			V_{DD} = 3.0 V ±10% Note 5		1.0	3.0	mA
			V_{DD} = 2.0 V ±10% Note 5		0.7	2.0	mA
	I_{DD3} Note 1	32.768 kHz crystal oscillation operating mode Note 3 (C3 = C4 = 22 pF, R1 = 220 kΩ)	V_{DD} = 5.0 V ±10%		200	600	μA
			V_{DD} = 3.0 V ±10%		150	450	μA
			V_{DD} = 2.0 V ±10%		100	300	μA
	I_{DD4} Note 1	32.768 kHz crystal oscillation HALT mode Note 3 (C3 = C4 = 22 pF, R1 = 220 kΩ)	V_{DD} = 5.0 V ±10%		50	150	μA
			V_{DD} = 3.0 V ±10%		30	90	μA
			V_{DD} = 2.0 V ±10%		20	60	μA
	I_{DD5} Note 1	32.768 kHz crystal oscillation STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	5.0	μA
			T_A = 25°C		0.05	3.0	μA
			V_{DD} = 2.0 V ±10%		0.05	3.0	μA
	I_{DD6} Note 2	5.0 MHz crystal oscillation A/D operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10% Note 4		6.0	16.0	mA
			V_{DD} = 3.0 V ±10% Note 5		3.0	7.0	mA
			V_{DD} = 2.0 V ±10% Note 5		2.5	5.0	mA

Notes 1. The current flowing to AV_{REF} (A/D operation ON (ADCS0 = 1)), AV_{DD} current, and the port current (including the current flowing through the on-chip pull-up resistors) is not included.

2. The current flowing to AV_{REF} (A/D operation ON (ADCS0 = 1)) and the port current (including the current flowing through the on-chip pull-up resistors) is not included. For the current flowing to AV_{REF}, refer to the parameter of "Resistance between AV_{REF} and AV_{SS}" in the **8-Bit A/D Converter Characteristics** and **10-Bit A/D Converter Characteristics**.

3. When the main system clock is stopped

4. High-speed mode operation (when processor clock control register (PCC) is set to 00H)

5. Low-speed mode operation (when PCC is set to 02H)

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

LCD Characteristics (T_A = –40 to +85°C, V_{DD} = 2.2 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}	VAON0 = 1		2.2		V_{DD}	V
		VAON0 = 0[Note 1]	At 1/3 bias	2.7		V_{DD}	V
			At 1/2 bias	3.0		V_{DD}	V
LCD divider resistor[Note 2]	R_{LCD}	When selecting 100 kΩ by mask option		100	200	400	kΩ
		When selecting 10 kΩ by mask option		10	20	40	kΩ
LCD output voltage deviation[Note 3] (common)	V_{ODC}	I_O = ±5 μA	$V_{LCD0} = V_{LCD}$ $V_{LCD1} = V_{LCD} \times 2/3$	0		±0.2	V
LCD output voltage deviation[Note 3] (segment)	V_{ODS}	I_O = ±1 μA	2.2 V ≤ V_{LCD} ≤ V_{DD} $V_{LCD2} = V_{LCD} \times 1/3$[Note 1]	0		±0.2	V

Notes 1. T_A = –10 to +85°C in the normal mode (VAON0 = 0)

2. For mask ROM version, 10 kΩ, 100 kΩ, or no divider resistor can be selected by mask option. The μPD78F9418A has no divider resistor.

3. Voltage deviation is the voltage difference between the ideal value of the segment or common output (V_{LCDn}: n = 0 to 2) and the output voltage.

Flash Memory Write/Erase Characteristics (μPD78F9418A only)

(T_A = 10 to 40°C, V_{DD} = 1.8 to 5.5 V, in 5.0 MHz crystal oscillation operating mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write current[Note] (V_{DD} pin)	I_{DDW}	When V_{PP} supply voltage = V_{PP1}			18	mA
Write current[Note] (V_{PP} pin)	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			22.5	mA
Erase current[Note] (V_{DD} pin)	I_{DDE}	When V_{PP} supply voltage = V_{PP1}			18	mA
Erase current[Note] (V_{PP} pin)	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			115	mA
Unit erase time	t_{er}		0.5	1	1	s
Total erase time	t_{era}				20	s
Write count		Erase/write are regarded as 1 cycle			20	Times
V_{PP} supply voltage	V_{PP0}	In normal operation	0		0.2V_{DD}	V
	V_{PP1}	During flash memory programming	9.7	10.0	10.3	V

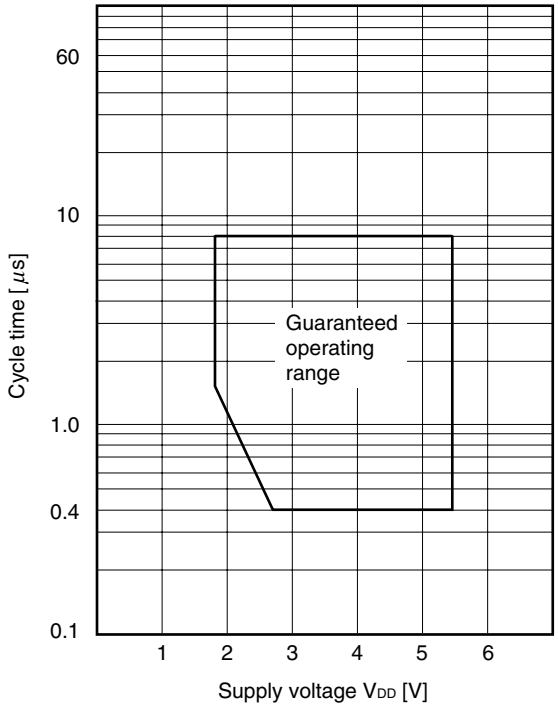
Note The current flowing to the ports (including the current flowing through the on-chip pull-up resistors) is not included.

AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operating with main system clock	V_{DD} = 2.7 to 5.5 V	0.4		8	μs
			V_{DD} = 1.8 to 5.5 V	1.6		8	μs
		Operating with subsystem clock		114	122	125	μs
TI0, TI1 input frequency	f_{TI}	V_{DD} = 2.7 to 5.5 V		0		4	MHz
		V_{DD} = 1.8 to 5.5 V		0		275	kHz
TI0, TI1 input high-/low-level widths	t_{TIH}, t_{TIL}	V_{DD} = 2.7 to 5.5 V		0.1			μs
		V_{DD} = 1.8 to 5.5 V		1.8			μs
Interrupt input high-/low-level widths	t_{INTH}, t_{INTL}	INTP0 to INTP3		10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}			10			μs

T_{CY} vs V_{DD} (Main system clock)



(2) Serial interface (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) 3-wire serial I/O mode ($\overline{SCK}$ ... Internal clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK}$ cycle time	t_{KCY1}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3200			ns
$\overline{SCK}$ high-/low-level widths	t_{KH1}, t_{KL1}	V_{DD} = 2.7 to 5.5 V		t_{KCY1}/2–50			ns
		V_{DD} = 1.8 to 5.5 V		t_{KCY1}/2–150			ns
SI setup time (to $\overline{SCK}\uparrow$)	t_{SIK1}	V_{DD} = 2.7 to 5.5 V		150			ns
		V_{DD} = 1.8 to 5.5 V		500			ns
SI hold time (from $\overline{SCK}\uparrow$)	t_{KSI1}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
SO output delay time from $\overline{SCK}\downarrow$	t_{KSO1}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		250	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO output line.

(b) 3-wire serial I/O mode ($\overline{SCK}$ ... External clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK}$ cycle time	t_{KCY2}	V_{DD} = 2.7 to 5.5 V		900			ns
		V_{DD} = 1.8 to 5.5 V		3500			ns
$\overline{SCK}$ high-/low-level widths	t_{KH2}, t_{KL2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		1600			ns
SI setup time (to $\overline{SCK}\uparrow$)	t_{SIK2}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		150			ns
SI hold time (from $\overline{SCK}\uparrow$)	t_{KSI2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
SO output delay time from $\overline{SCK}\downarrow$	t_{KSO2}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		300	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO output line.

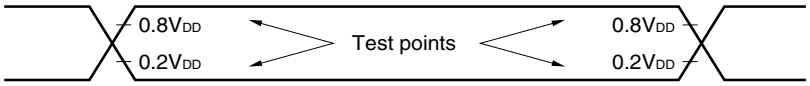
(c) UART mode (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		V_{DD} = 2.7 to 5.5 V			78125	bps
		V_{DD} = 1.8 to 5.5 V			19531	bps

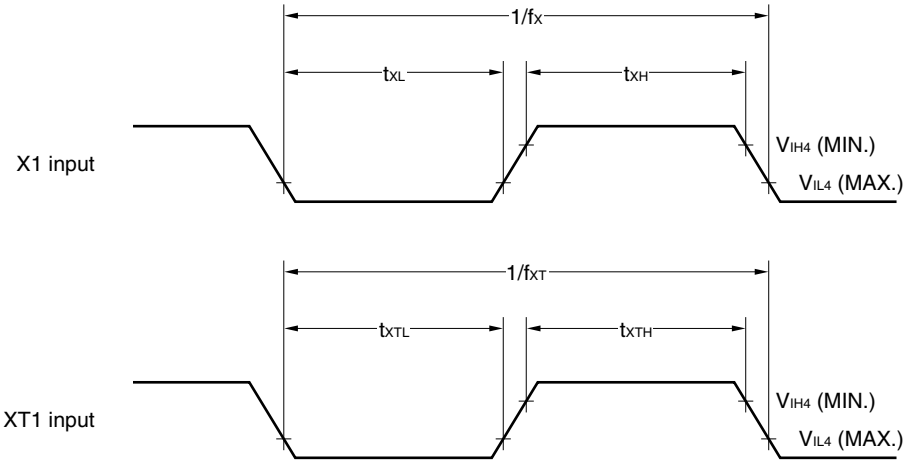
(d) UART mode (external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK cycle time	t_{KCY3}	V_{DD} = 2.7 to 5.5 V	900			ns
		V_{DD} = 1.8 to 5.5 V	3500			ns
ASCK high-/low-level widths	t_{KH3}, t_{KL3}	V_{DD} = 2.7 to 5.5 V	400			ns
		V_{DD} = 1.8 to 5.5 V	1600			ns
Transfer rate		V_{DD} = 2.7 to 5.5 V			39063	bps
		V_{DD} = 1.8 to 5.5 V			9766	bps
ASCK rise/fall times	t_R, t_F				1	μs

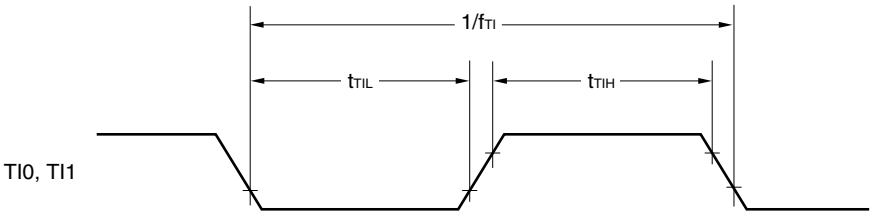
AC Timing Test Points (Excluding X1 and XT1 Inputs)



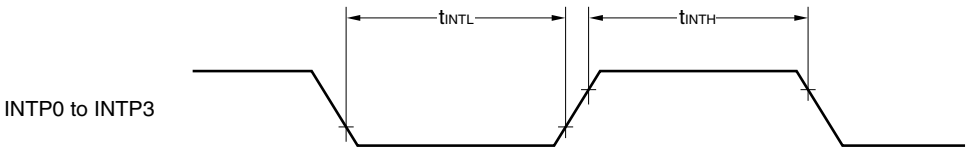
Clock Timing



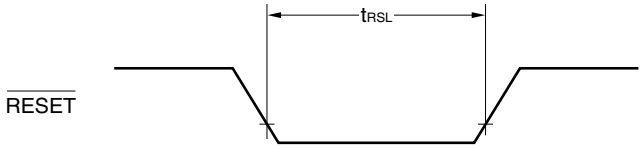
TI Timing



Interrupt Input Timing

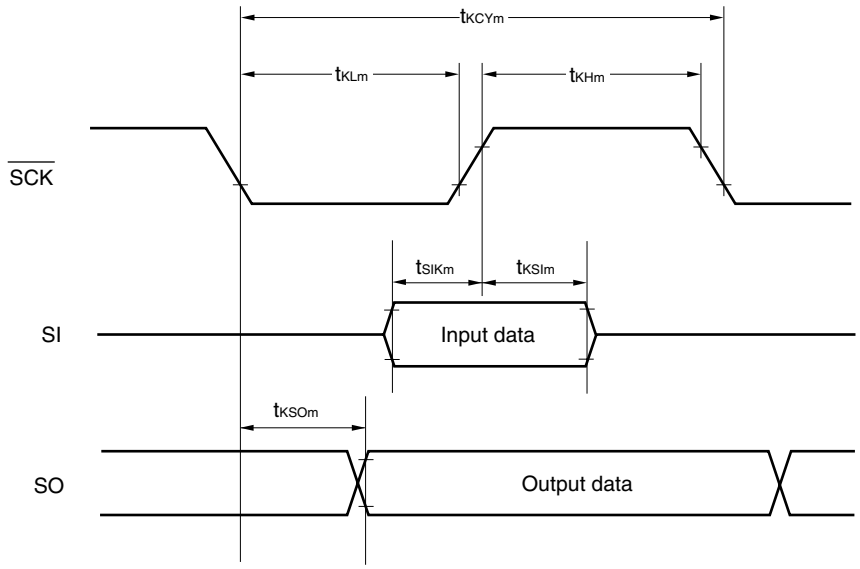


$\overline{\text{RESET}}$ Input Timing



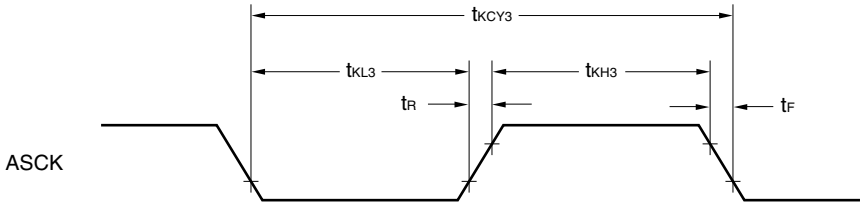
Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1 or 2

UART mode (external clock input):



8-Bit A/D Converter Characteristics (μPD789405A, 789406A, 789407A)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{REF} ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			8	8	8	bit
Overall error[Note]		2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V		±0.4	±0.6	%FSR
				±0.8	±1.2	%FSR
Conversion time	t_{CONV}	2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V	14		100	μs
			28		100	μs
Analog input voltage	V_{IAN}		0		AV_{REF}	V
Reference voltage	AV_{REF}		1.8		AV_{DD}	V
Resistance between AV_{REF} and AV_{SS}	R_{ADREF}		20	40		kΩ

Note Excludes quantization error (±0.2%FSR).

Remark FSR: Full-scale range

10-Bit A/D Converter Characteristics (μPD789415A, 789416A, 789417A, 78F9418A)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{REF} ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note]		4.5 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V		±0.4	±0.6	%FSR
		1.8 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V		±0.8	±1.2	%FSR
Conversion time	t_{CONV}	4.5 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V	14		100	μs
		2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V	14		100	μs
		1.8 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V	28		100	μs
Zero-scale error[Note]	AINL	4.5 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±0.6	%FSR
		1.8 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±1.2	%FSR
Full-scale error[Note]	AINL	4.5 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±0.6	%FSR
		1.8 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±1.2	%FSR
Non-integral linearity[Note]	INL	4.5 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±4.5	LSB
		1.8 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±8.5	LSB
Non-differential linearity[Note]	DNL	4.5 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±2.0	LSB
		1.8 V ≤ AV_{REF} ≤ AV_{DD} ≤ 5.5 V			±3.5	LSB
Analog input voltage	V_{IAN}		0		AV_{REF}	V
Reference voltage	AV_{REF}		1.8		AV_{DD}	V
Resistance between AV_{REF} and AV_{SS}	R_{ADREF}		20	40		kΩ

Note Excludes quantization error (±0.05%FSR).

Remark FSR: Full-scale range

Comparator Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Analog input range	V_{CIN}		0		V_{DD}	V
Reference voltage input range	V_{CREF}	V_{DD} = 2.7 to 5.5 V	1.35	1.6	1.85	V
		V_{DD} = 1.8 to 5.5 V	1.35	1.4	1.45	V
Accuracy					±100	mV

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

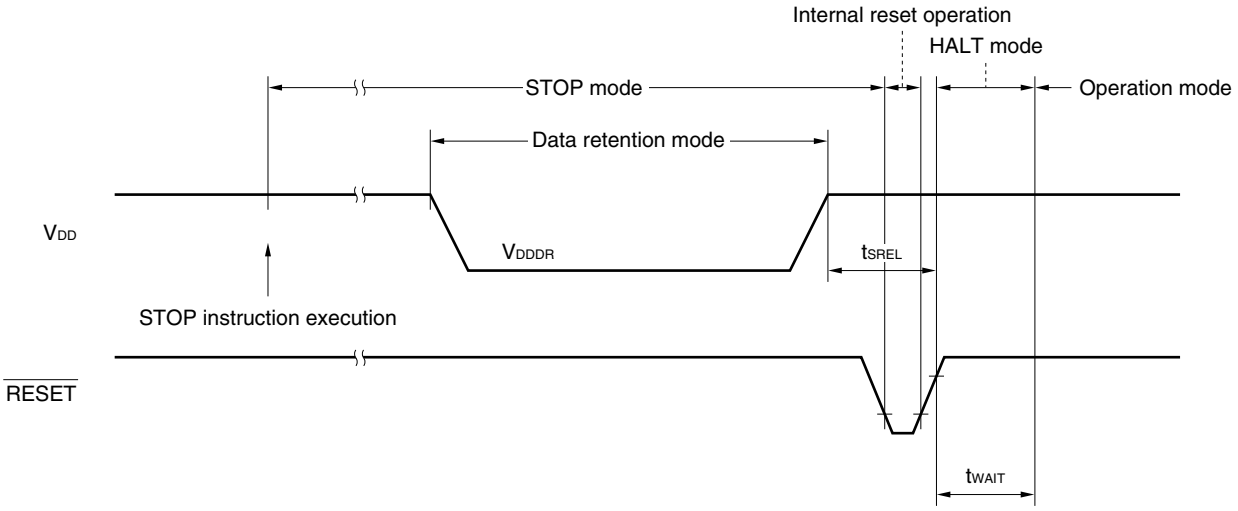
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{15}/f_X$		ms
		Release by interrupt request		**Note 2**		ms

Notes 1. The oscillation stabilization wait time is the time after oscillation has started during which the CPU is stopped to prevent unstable operation.

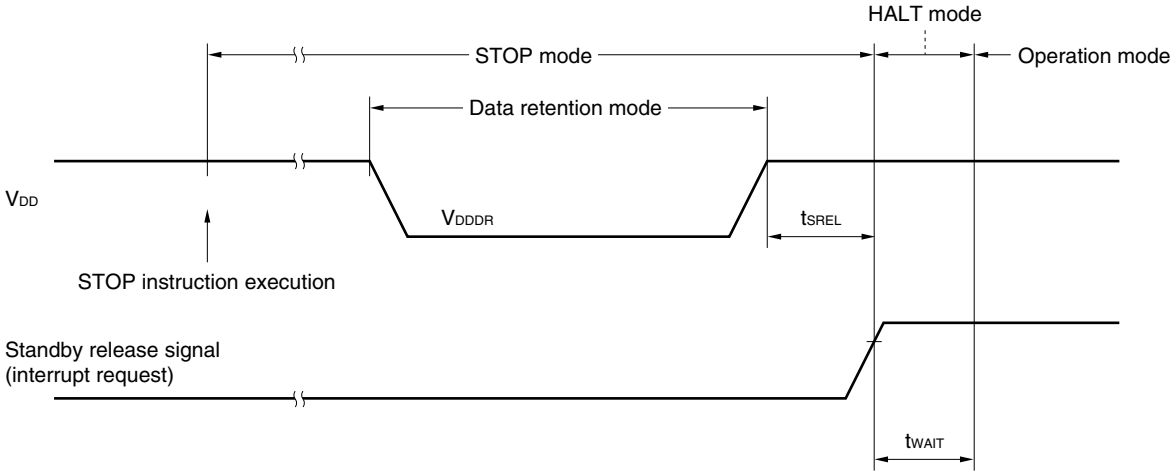
2. Selection of $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ is possible with bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register (OSTS).

Remark fx: Main system clock oscillation frequency

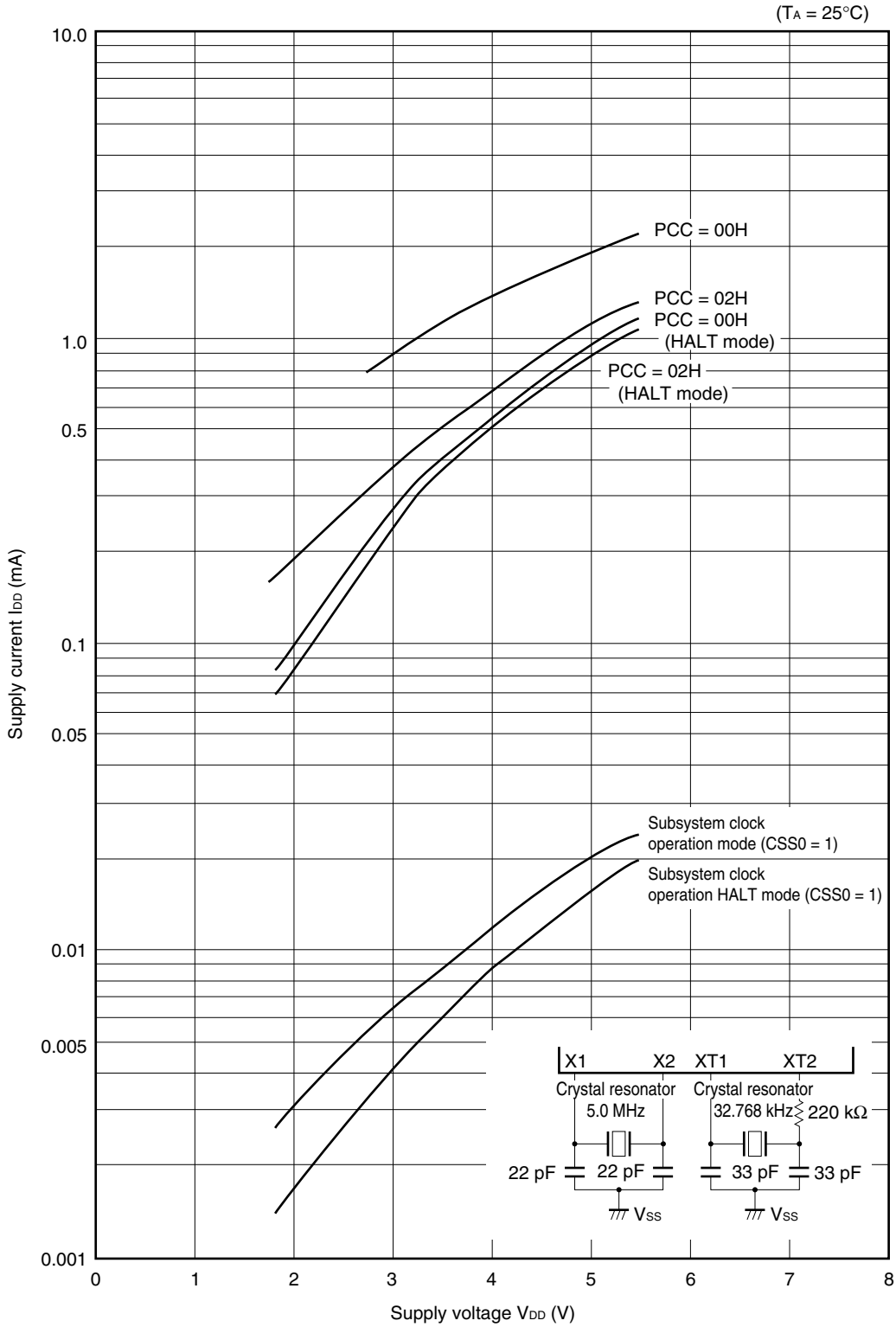
Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)



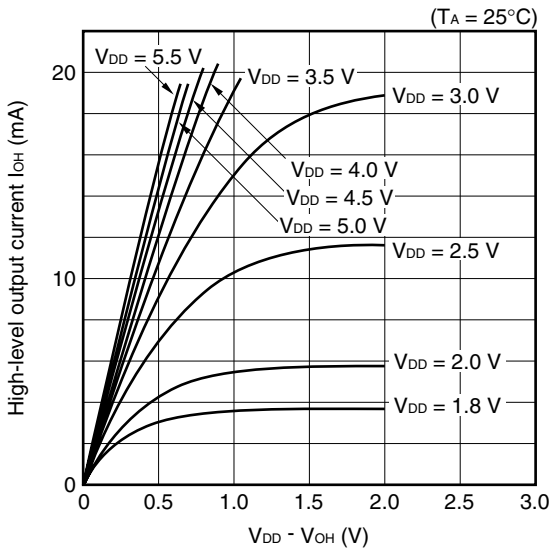
Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)



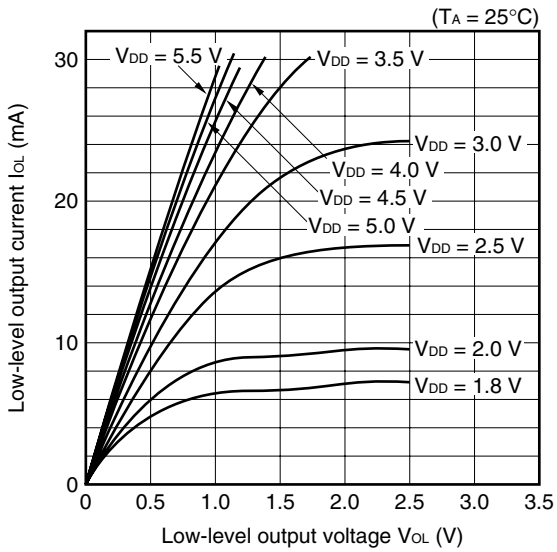
22.1 Characteristics Curves for Mask ROM Versions



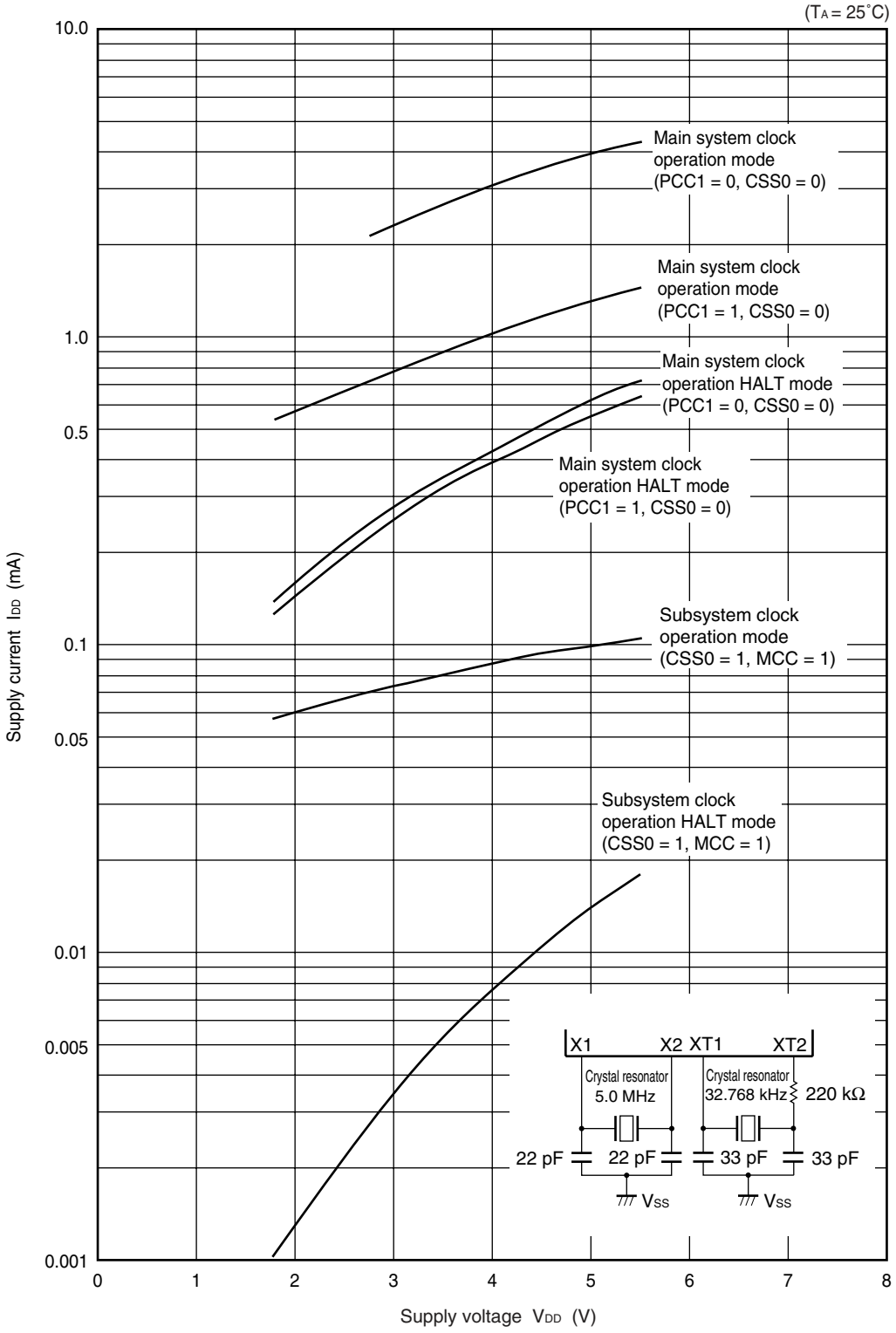
I_{OH} vs V_{DD} - V_{OH}



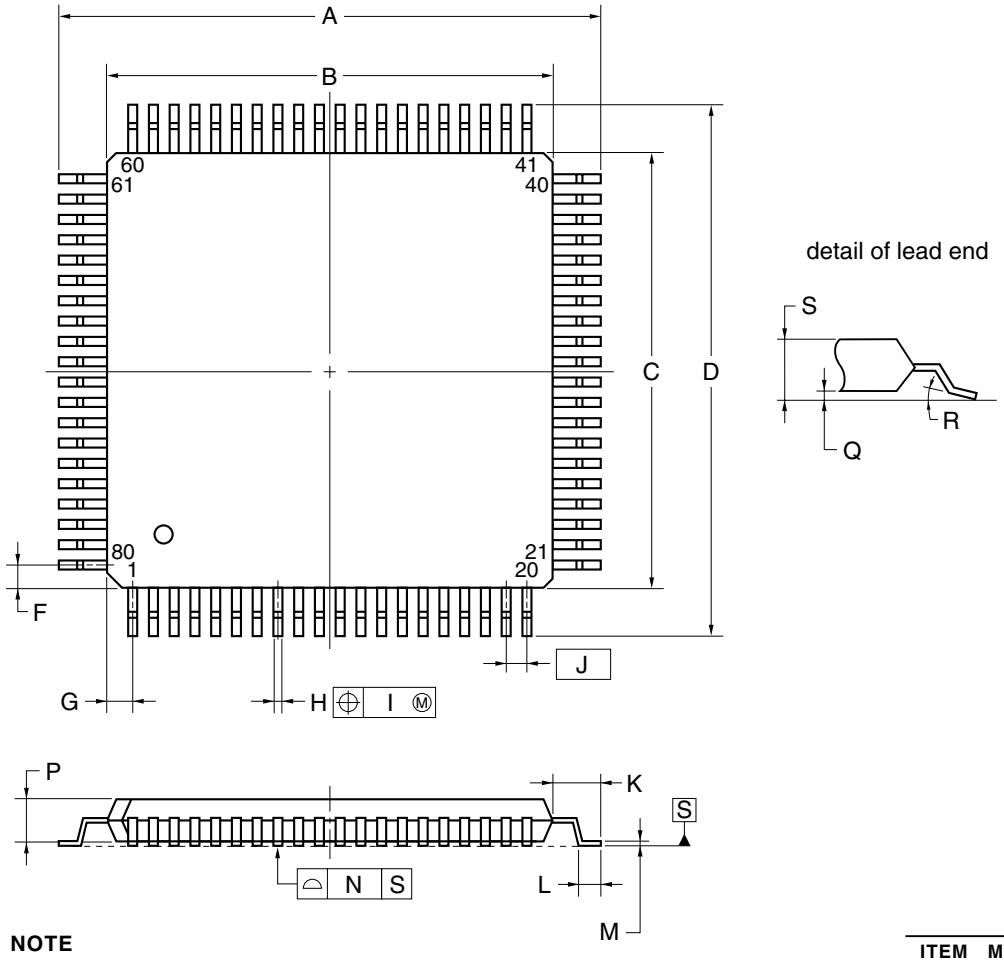
I_{OL} vs V_{OL}



22.2 Characteristics Curves for μPD78F9418A



80-PIN PLASTIC QFP (14x14)



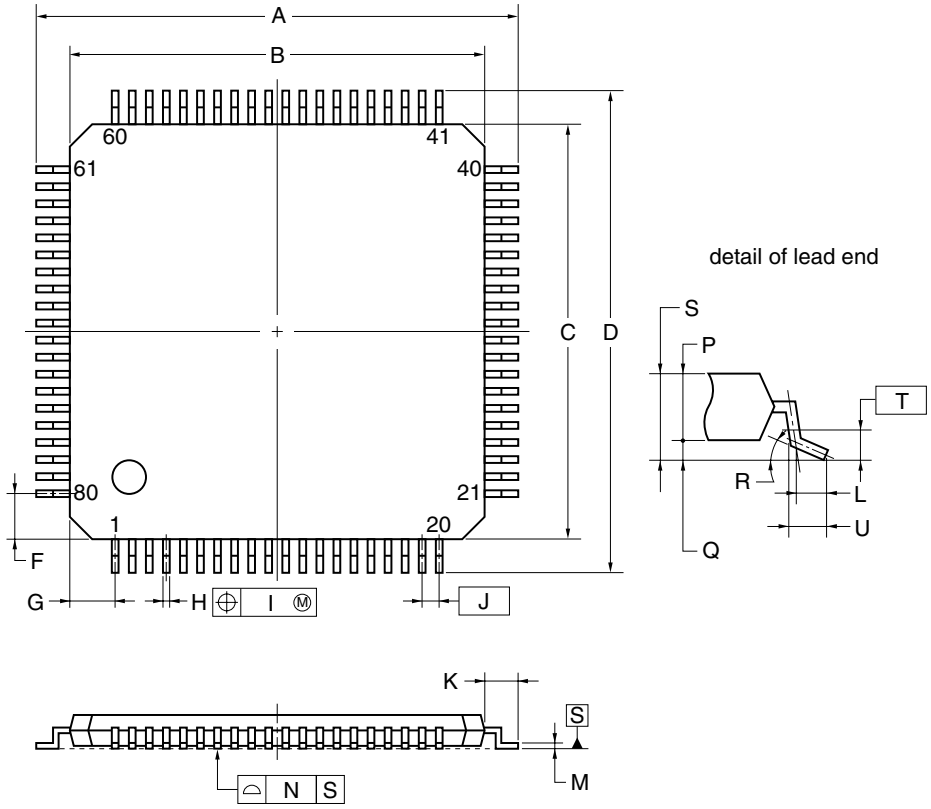
NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	17.20±0.20
B	14.00±0.20
C	14.00±0.20
D	17.20±0.20
F	0.825
G	0.825
H	0.32±0.06
I	0.13
J	0.65 (T.P.)
K	1.60±0.20
L	0.80±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	1.40±0.10
Q	0.125±0.075
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.70 MAX.

P80GC-65-8BT-1

80-PIN PLASTIC TQFP (FINE PITCH) (12x12)



NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	14.0±0.2
B	12.0±0.2
C	12.0±0.2
D	14.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5
M	0.145±0.05
N	0.08
P	1.0
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.1±0.1
T	0.25
U	0.6±0.15

P80GK-50-9EU-1

★ CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS

The μPD789407A and μPD789417A Subseries should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 24-1. Surface Mounting Type Soldering Conditions (1/3)

μPD789405AGC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD789406AGC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD789407AGC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD789415AGC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD789416AGC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD789417AGC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD78F9418AGC-8BT: 80-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: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less	VP15-00-2
Wave soldering	Soldering bath temperature: 260°C max., Time: 10 seconds max., Count: 1, 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 24-1. Surface Mounting Type Soldering Conditions (2/3)

μPD789405AGK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789406AGK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789407AGK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789415AGK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789416AGK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789417AGK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F9418GK-9EU: 80-pin plastic TQFP (fine pitch) (12 × 12)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-107-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Table 24-1. Surface Mounting Type Soldering Conditions (3/3)

μPD789405AGC-×××-8BT-A: 80-pin plastic QFP (14 × 14)
μPD789406AGC-×××-8BT-A: 80-pin plastic QFP (14 × 14)
μPD789407AGC-×××-8BT-A: 80-pin plastic QFP (14 × 14)
μPD789415AGC-×××-8BT-A: 80-pin plastic QFP (14 × 14)
μPD789416AGC-×××-8BT-A: 80-pin plastic QFP (14 × 14)
μPD789417AGC-×××-8BT-A: 80-pin plastic QFP (14 × 14)
μPD78F9418AGC-8BT-A: 80-pin plastic QFP (14 × 14)
μPD789405AGK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789406AGK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789407AGK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789415AGK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789416AGK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789417AGK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F9418GK-9EU-A: 80-pin plastic TQFP (fine pitch) (12 × 12)

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	When the pin pitch of the package is 0.65 mm or more, wave soldering can also be performed. 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.

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789407A and μPD789417A Subseries.

Figure A-1 shows development tools.

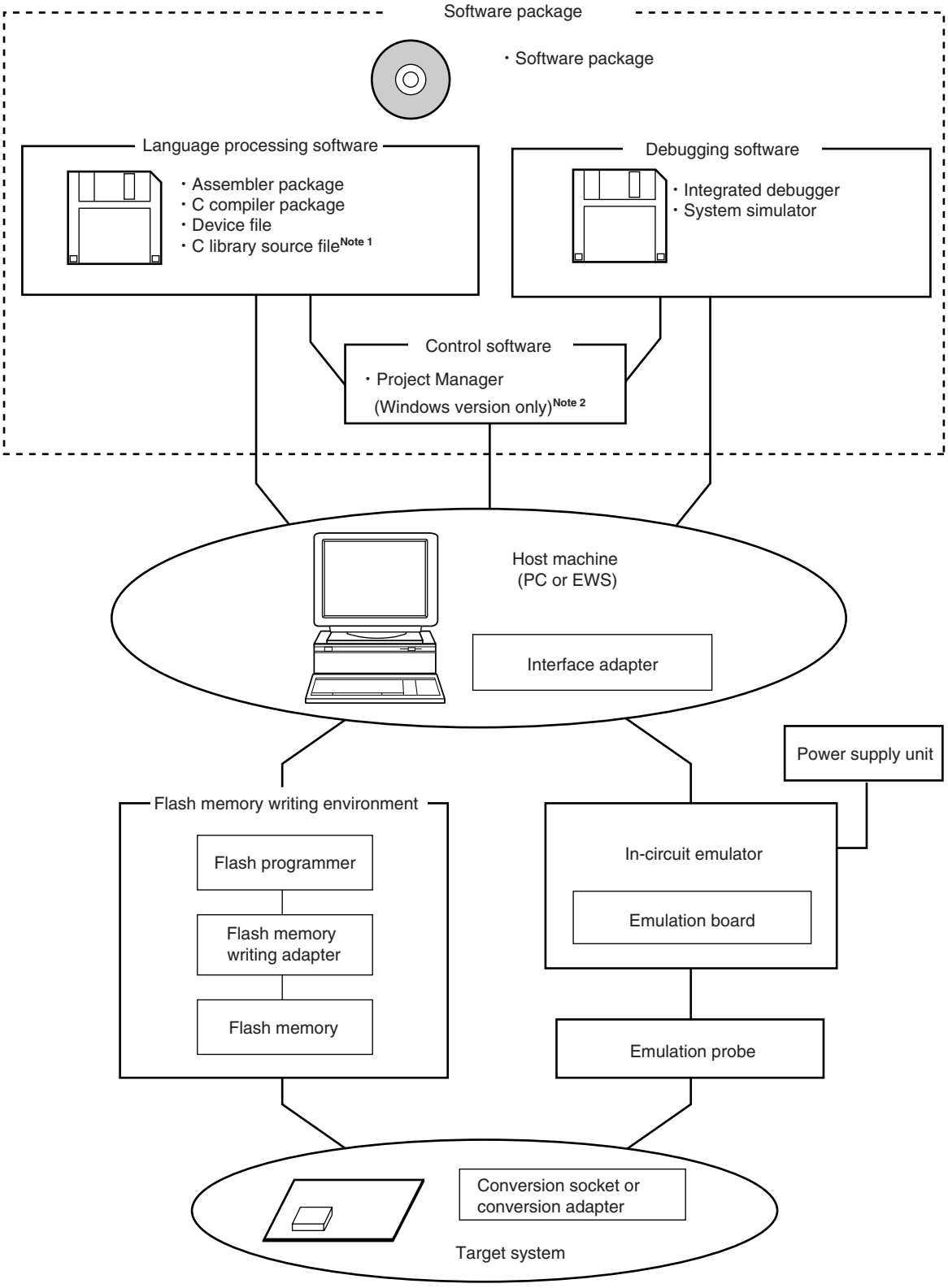
- Support of PC98-NX series

  Unless specified otherwise, the products supported by IBM PC/AT™ compatibles can be used in the PC98-NX series. When using the PC98-NX series, refer to the explanation of IBM PC/AT compatibles.

- Windows

  Unless specified otherwise, “Windows” indicates the following operating systems.
  - Windows 3.1
  - Windows 95, 98, 2000
  - Windows NT™ Ver.4.0

★ **Figure A-1. Development Tools**



Notes 1. C library source file is not included in the software package.

2. Project Manager is included in the assembler package.
Project Manager is used only in the Windows environment.

★ A.1 Software Package

SP78K0S Software package	Software tools for development of the 78K/0S Series are combined in this package. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S, and device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the operating system to be used.

μS××××SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object codes that can be executed by microcontroller. In addition, automatic functions to generate a symbol table and optimize branch instructions are also provided. Used in combination with a device file (DF789418) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C compiler package	Program that converts program written in C language into object codes that can be executed by microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789418) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789418[Note 1] Device file	File containing the information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789418
CC78K0S-L[Note 2] C library source file	Source file of functions for generating object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789418 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

Remark ××××in the part number differs depending on the host machine and operating system to be used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5-inch 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF789418
μS××××CC78K0S-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5-inch 2HD FD
BB13		English Windows	
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	Control software created for efficient development of the user program in the Windows environment. User program development operations such as editor startup, build, and debugger startup can be performed from the Project Manager. <Caution> The Project Manager is included in the assembler package (RA78K0S). The Project Manager is used only in the Windows environment.

A.4 Flash Memory Writing Tools

Flashpro III (FL-PR3, PG-FP3) Flashpro IV (FL-PR4, PG-FP4) Flash programmer	Dedicated flash programmer for microcontrollers incorporating flash memory
FA-80GC-8BT FA-80GK-9EU Flash memory writing adapter	Adapter for writing to flash memory and connected to Flashpro III or Flashpro IV. • FA-80GC-8BT: For 80-pin plastic QFP (GC-8BT type) • FA-80GK-9EU: For 80-pin plastic TQFP (GK-9EU type)

Remark The FL-PR3, FL-PR4, FA-80GC-8BT, and FA-80GK-9EU are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

A.5 Debugging Tools (Hardware)

IE-78K0S-NS In-circuit emulator		In-circuit emulator for debugging a hardware and software of application system using the 78K/0S Series. Supports an integrated debugger (ID78K0S-NS). Used in combination with an AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator		In-circuit emulator with functions expanded from the IE-78K0S-NS. The debug function has been further enhanced with the addition of a coverage function, and enhancement of the tracer function and timer function.
IE-70000-MC-PS-B AC adapter		Adapter for supplying power from AC 100 to 240 V outlet.
IE-70000-98-IF-C Interface adapter		Adapter necessary when using a PC-9800 series PC (except notebook type) as the host machine of the IE-78K0S-NS (C bus supported)
IE-70000-CD-IF-A PC card interface		PC card and interface cable necessary when using a notebook PC as the host machine of the IE-78K0S-NS (PCMCIA socket supported)
IE-70000-PC-IF-C Interface adapter		Adapter necessary when using an IBM PC/AT compatible as the host machine of the IE-78K0S-NS (ISA bus supported)
IE-70000-PCI-IF-A Interface adapter		Adapter necessary when using a personal computer incorporating the PCI bus as the host machine of the IE-78K0S-NS
IE-789418-NS-EM1 Emulation board		Board for emulating the peripheral hardware specific to the device. Used in combination with an in-circuit emulator.
NP-80GC Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the EV-9200GC-80.
	EV-9200GC-80 Conversion socket	Conversion socket to connect the NP-80GC to a target system board on which an 80-pin plastic QFP (GC-8BT type) can be mounted.
NP-80GC-TQ NP-H80GC-TQ Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the TGC-080SBP.
	TGC-080SBP Conversion adapter	Conversion adapter to connect the NP-80GC-TQ or NP-H80GC-TQ to a target system board on which an 80-pin plastic QFP (GC-8BT type) can be mounted.
NP-80GK NP-H80GK-TQ Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the TGK-080SDW.
	TGK-080SDW Conversion adapter	Conversion adapter to connect the NP-80GK or NP-H80GK-TQ to a target system board on which an 80-pin plastic TQFP (fine pitch) (GK-9EU type) can be mounted.

Remarks 1. The NP-80GC, NP-80GC-TQ, NP-H80GC-TQ, NP-80GK, and NP-H80GK-TQ are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

2. The TGC-080SBP and TGK-080SDW are products made by TOKYO ELETECH CORPORATION.
For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

3. The EV-9200GC-80 is sold in five units as a set.

4. The TGC-080SBP and TGK-080SDW are sold in one set units.

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	This debugger supports the in-circuit emulators IE-78K0S-NS and IE-78K0S-NS-A for the 78K/0S Series. The ID78K0S-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. Used in combination with a device file (DF789418) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	This is a system simulator for the 78K/0S Series. The SM78K0S is Windows-based software. It can be used to debug the target system at C source level or assembler level while simulating the operation of the target system on the host machine. Using SM78K0S, the logic and performance of the application can be verified independently of hardware development. Therefore, the development efficiency can be enhanced and the software quality can be improved. Used in combination with a device file (DF789418) (sold separately).
	Part number: μS××××SM78K0S
DF789418[Note] Device file	File containing the information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789418

Note DF789418 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

Remark ×××× in the part number differs depending on the operating system and supply medium to be used.

μS××××ID78K0S-NS
μS××××SM78K0S

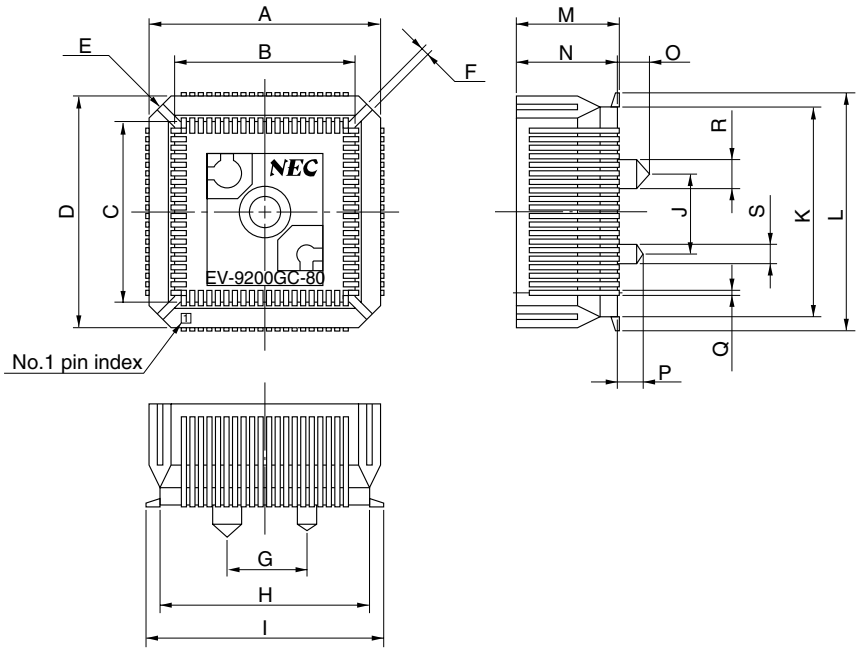
××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5-inch 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

A.7 Package Drawings of Conversion Socket and Conversion Adapter

A.7.1 Package drawing and recommended footprint of conversion socket (EV-9200GC-80)

Figure A-2. Package Drawing of EV-9200GC-80 (for Reference)

Based on EV-9200GC-80
(1) Package drawing (in mm)

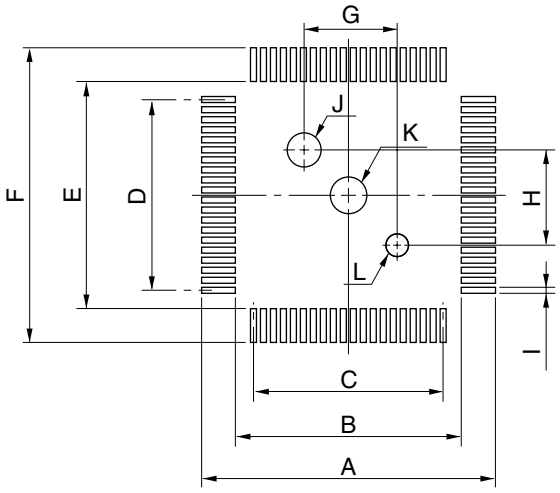


EV-9200GC-80-G1E

ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	14.4	0.567
C	14.4	0.567
D	18.0	0.709
E	4-C 2.0	4-C 0.079
F	0.8	0.031
G	6.0	0.236
H	16.0	0.63
I	18.7	0.736
J	6.0	0.236
K	16.0	0.63
L	18.7	0.736
M	8.2	0.323
N	8.0	0.315
O	2.5	0.098
P	2.0	0.079
Q	0.35	0.014
R	ϕ2.3	ϕ0.091
S	ϕ1.5	ϕ0.059

Figure A-3. Recommended Footprint of EV-9200GC-80 (for Reference)

Based on EV-9200GC-80
(2) Pad drawing (in mm)



EV-9200GC-80-P1E

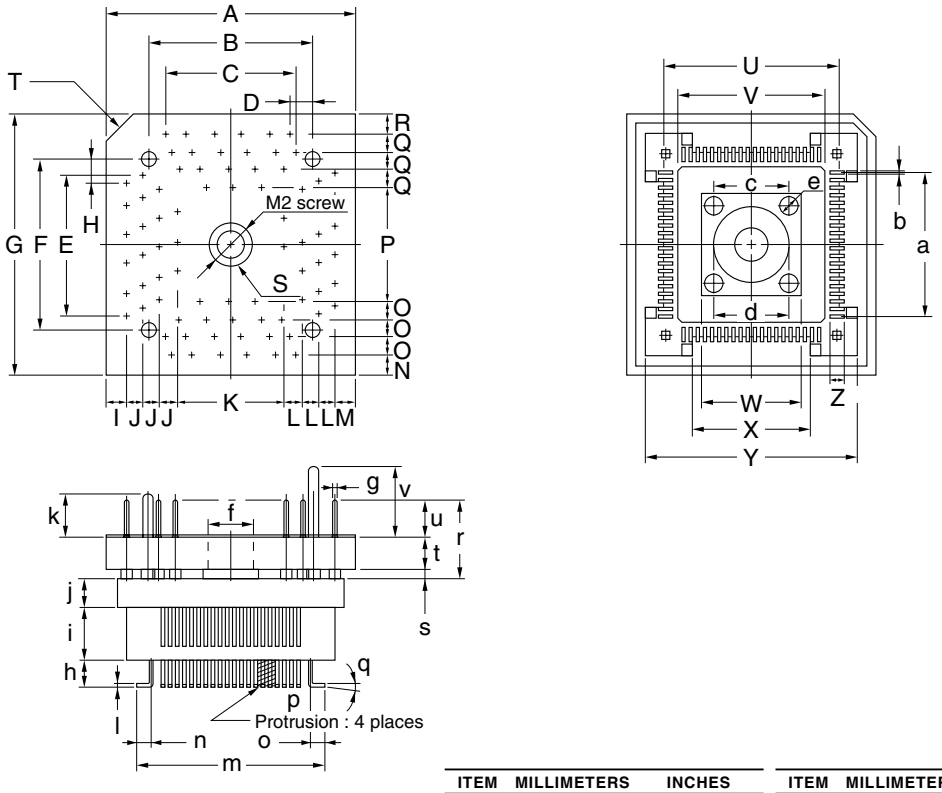
ITEM	MILLIMETERS	INCHES
A	19.7	0.776
B	15.0	0.591
C	$0.65\pm0.02\times19=12.35\pm0.05$	$0.026^{+0.001}_{-0.002}\times0.748=0.486^{+0.003}_{-0.002}$
D	$0.65\pm0.02\times19=12.35\pm0.05$	$0.026^{+0.001}_{-0.002}\times0.748=0.486^{+0.003}_{-0.002}$
E	15.0	0.591
F	19.7	0.776
G	6.0 ± 0.05	$0.236^{+0.003}_{-0.002}$
H	6.0 ± 0.05	$0.236^{+0.003}_{-0.002}$
I	0.35 ± 0.02	$0.014^{+0.001}_{-0.001}$
J	$\phi2.36\pm0.03$	$\phi0.093^{+0.001}_{-0.002}$
K	$\phi2.3$	$\phi0.091$
L	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$

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 Mount Manual" (http://www.necel.com/pkg/en/mount/index.html).

A.7.2 Package drawing of conversion adapter (TGK-080SDW)

Figure A-4. Package Drawing of TGK-080SDW (for Reference)

TGK-080SDW (TQPACK080SD + TQSOCKET080SDW) Package dimension (unit: mm)



ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	11.77	0.463
C	0.5x19=9.5	0.020x0.748=0.374
D	0.5	0.020
E	0.5x19=9.5	0.020x0.748=0.374
F	11.77	0.463
G	18.0	0.709
H	0.5	0.020
I	1.58	0.062
J	1.2	0.047
K	7.64	0.301
L	1.2	0.047
M	1.58	0.062
N	1.58	0.062
O	1.2	0.047
P	7.64	0.301
Q	1.2	0.047
R	1.58	0.062
S	ϕ3.55	ϕ0.140
T	C 2.0	C 0.079
U	12.31	0.485
V	10.17	0.400
W	6.8	0.268
X	8.24	0.324
Y	14.8	0.583
Z	1.4±0.2	0.055±0.008

ITEM	MILLIMETERS	INCHES
a	0.5x19=9.5±0.10	0.020x0.748=0.374±0.004
b	0.25	0.010
c	ϕ5.3	ϕ0.209
d	ϕ5.3	ϕ0.209
e	ϕ1.3	ϕ0.051
f	ϕ3.55	ϕ0.140
g	ϕ0.3	ϕ0.012
h	1.85±0.2	0.073±0.008
i	3.5	0.138
j	2.0	0.079
k	3.0	0.118
l	0.25	0.010
m	14.0	0.551
n	1.4±0.2	0.055±0.008
o	1.4±0.2	0.055±0.008
p	h=1.8 ϕ1.3	h=0.071 ϕ0.051
q	0~5°	0.000~0.197°
r	5.9	0.232
s	0.8	0.031
t	2.4	0.094
u	2.7	0.106
v	3.9	0.154

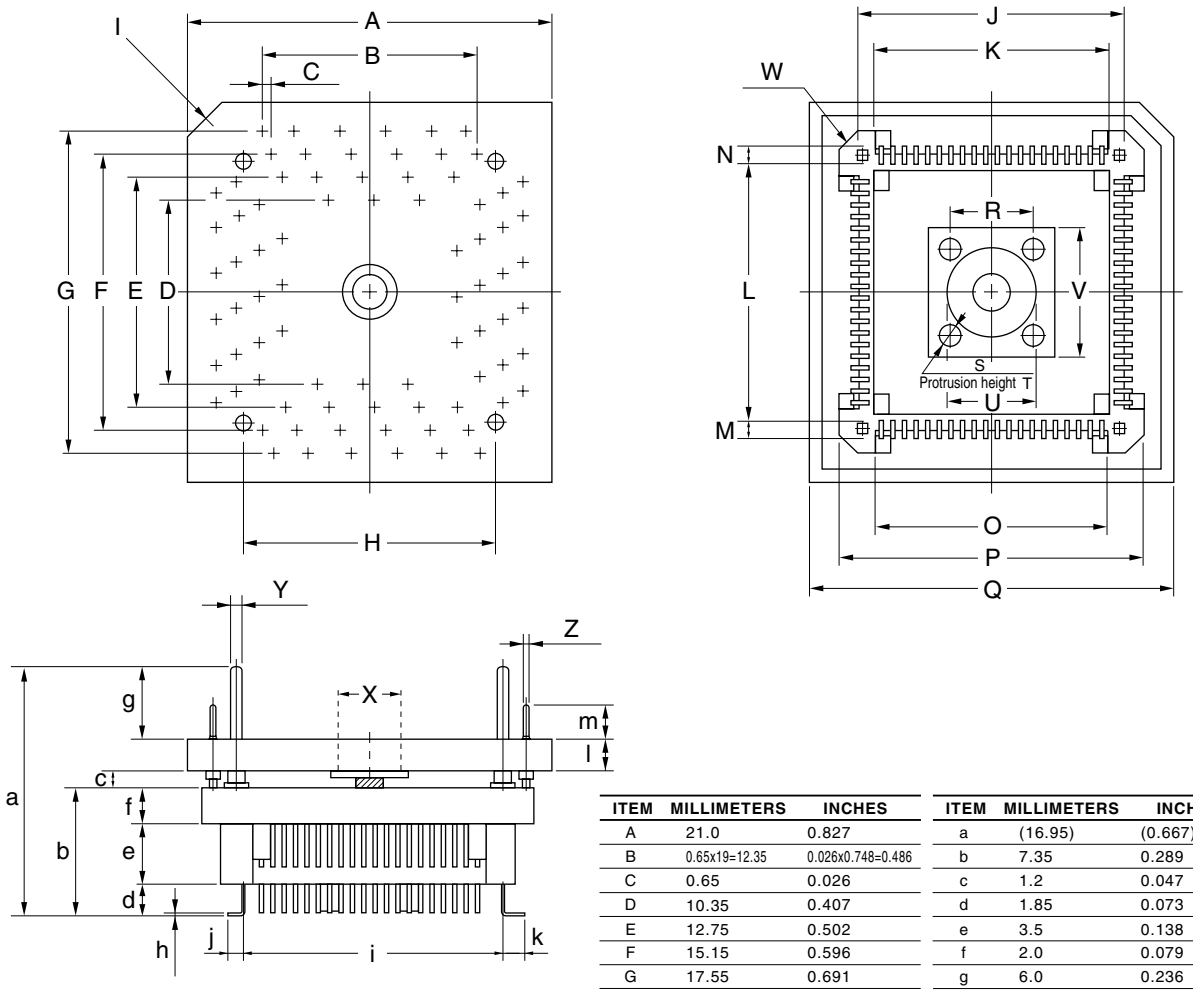
TGK-080SDW-G1E

note: Product by TOKYO ELETECH CORPORATION.

A.7.3 Package drawing of conversion adapter (TGC-080SBP)

Figure A-5. Package Drawing of TGC-080SBP (for Reference)

Reference diagram: TGC-080SBP (TQPACK080SB+TQSOCKET080SBP) Package dimension (unit: mm)



ITEM	MILLIMETERS	INCHES
A	21.0	0.827
B	0.65x19=12.35	0.026x0.748=0.486
C	0.65	0.026
D	10.35	0.407
E	12.75	0.502
F	15.15	0.596
G	17.55	0.691
H	14.47	0.570
I	C 2.0	C 0.079
J	14.95	0.589
K	13.95	0.549
L	13.7	0.539
M	1.15	0.045
N	1.15	0.045
O	12.62	0.497
P	17.52	0.690
Q	21.0	0.827
R	5.0	0.197
S	4-ϕ1.3	4-ϕ0.051
T	1.8	0.071
U	ϕ5.3	ϕ0.209
V	7.7	0.303
W	4-C 1.0	4-C 0.039
X	ϕ3.55	ϕ0.140
Y	ϕ0.9	ϕ0.035
Z	ϕ0.3	ϕ0.012

ITEM	MILLIMETERS	INCHES
a	(16.95)	(0.667)
b	7.35	0.289
c	1.2	0.047
d	1.85	0.073
e	3.5	0.138
f	2.0	0.079
g	6.0	0.236
h	0.25	0.010
i	13.95	0.549
j	1.025	0.040
k	1.025	0.040
l	2.4	0.094
m	2.7	0.106

TGC-080SBP-G0E

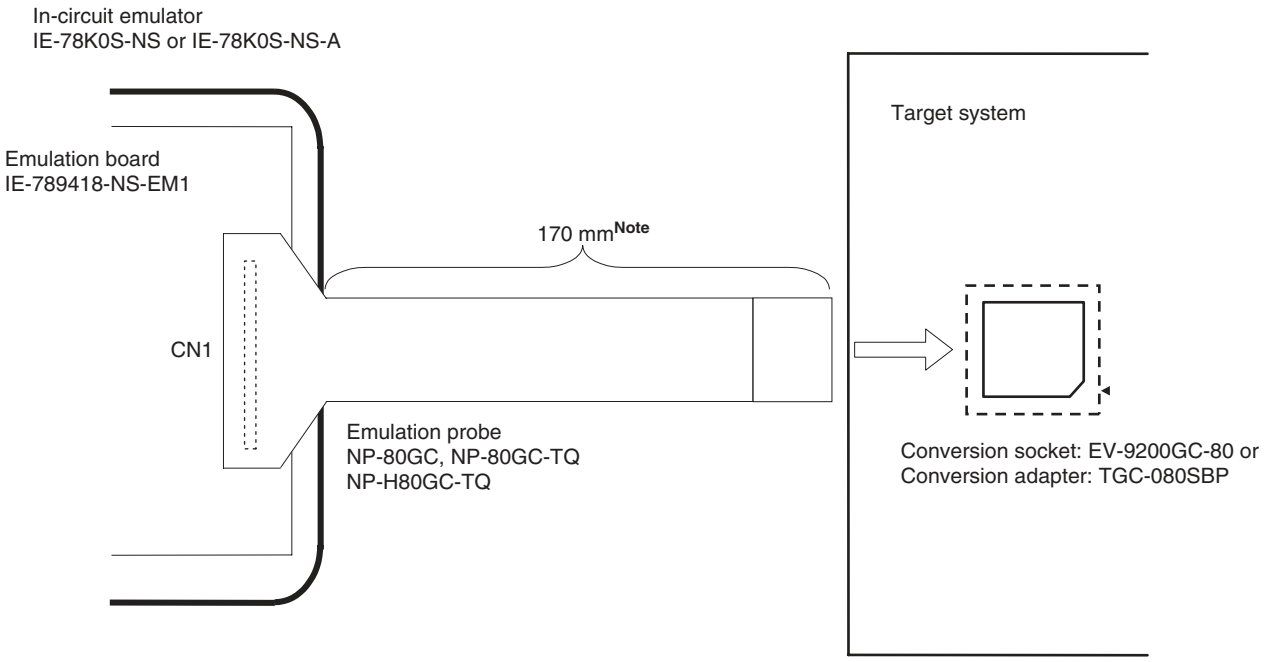
note: Product by TOKYO ELETECH CORPORATION.

★ APPENDIX B NOTES ON TARGET SYSTEM DESIGN

Figures B-1 to B-4 show the conditions when connecting the emulation probe to the conversion adapter or conversion socket. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

(1) NP-80GC, NP-80GC-TQ, NP-H80GC-TQ

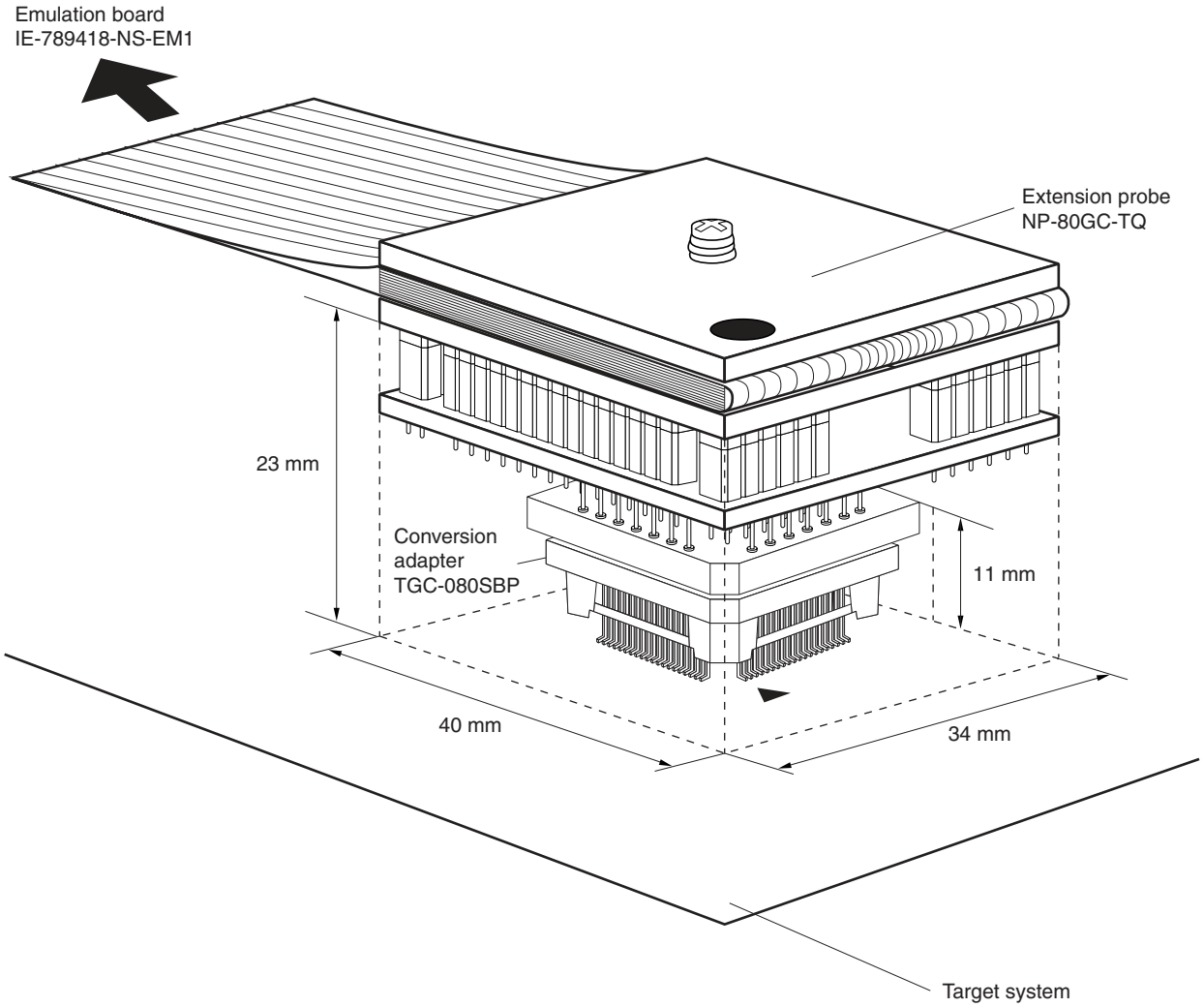
Figure B-1. Distance Between In-Circuit Emulator and Conversion Socket (80GC)



Note When NP-H80GC-TQ is used, the distance is 370 mm.

Remark NP-80GC, NP-80GC-TQ, and NP-H80GC-TQ are products of Naito Densei Machida Mfg. Co., Ltd.

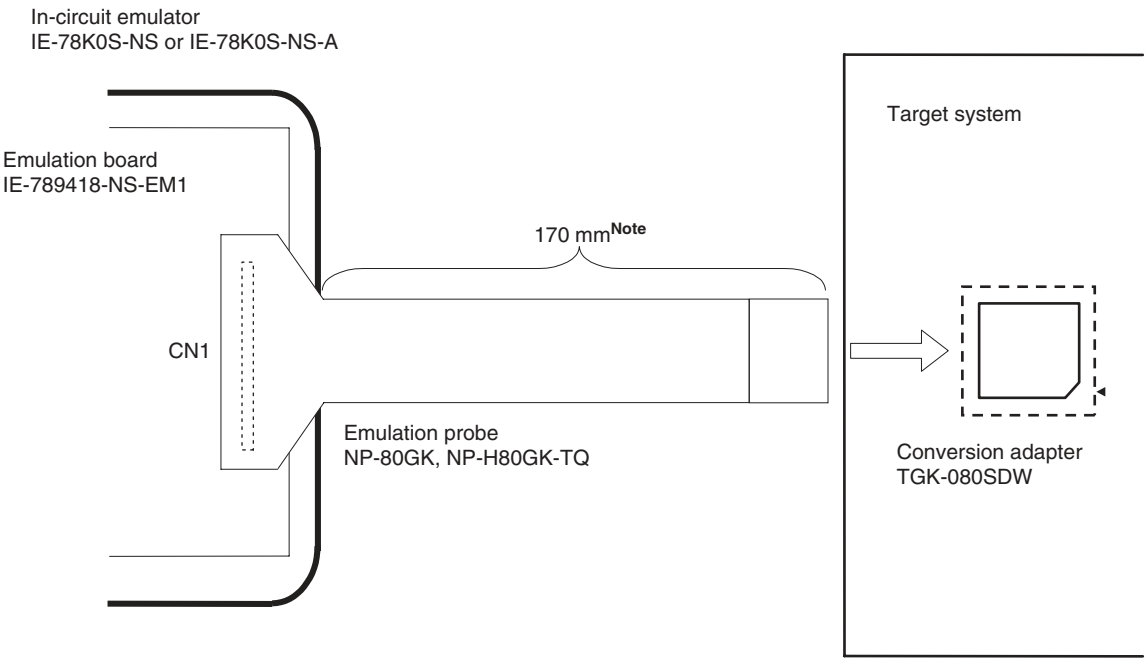
Figure B-2. Connection Condition of Target System (NP-80GC-TQ)



Remark NP-80GC-TQ is a product of Naito Densei Machida Mfg. Co., Ltd.
TGC-080SBP is a product of TOKYO ELETECH CORPORATION.

(2) NP-80GK, NP-H80GK-TQ

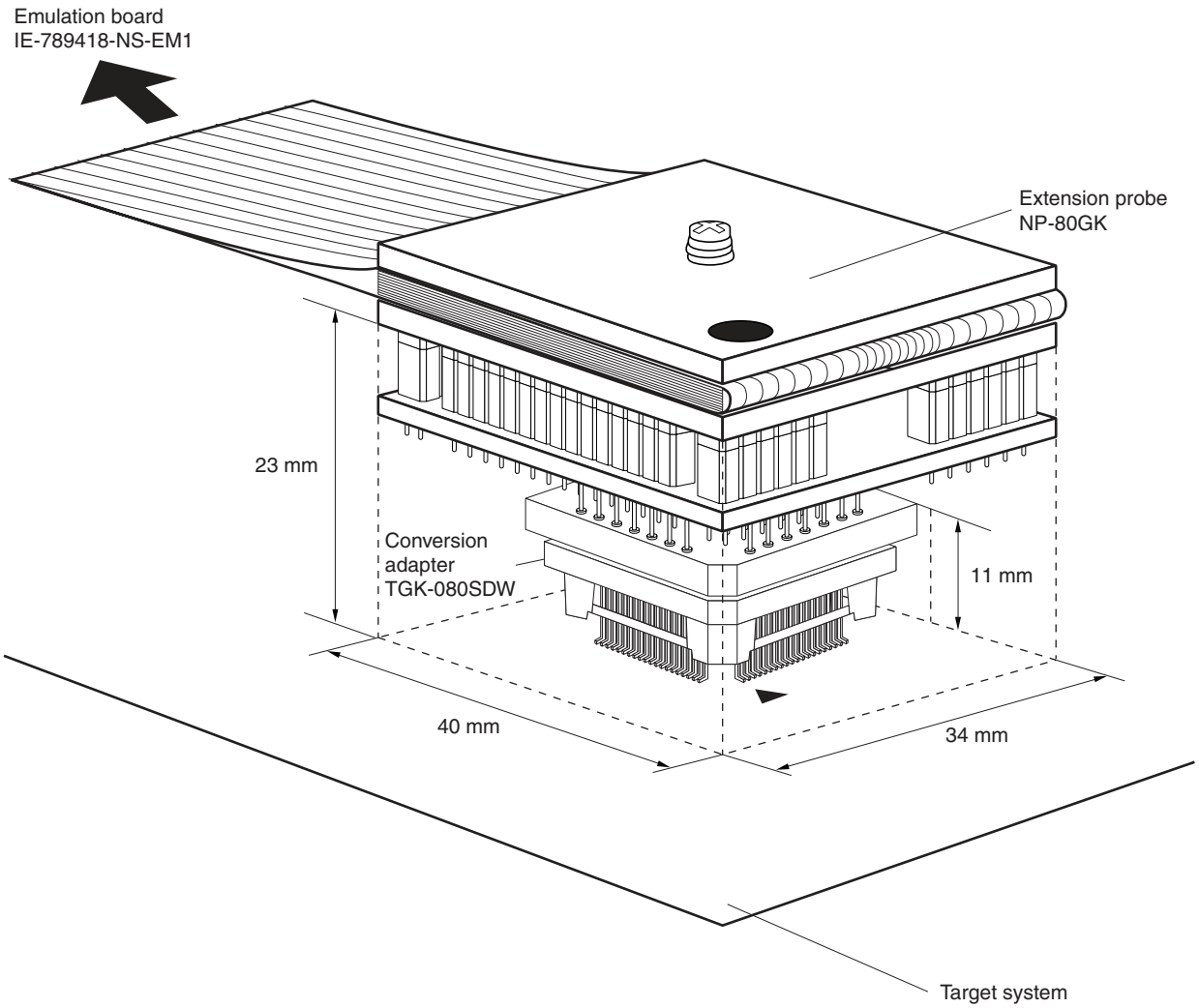
Figure B-3. Distance Between In-Circuit Emulator and Conversion Adapter (80GK)



Note When NP-H80GK-TQ is used, the distance is 370 mm.

Remark NP-80GK and NP-H80GK-TQ are products of Naito Densei Machida Mfg. Co., Ltd.
TGK-080SDW is a product of TOKYO ELETECH CORPORATION.

Figure B-4. Connection Condition of Target System (NP-80GK)



Remark NP-80GK is a product of Naito Densei Machida Mfg. Co., Ltd.
TGK-080SDW is a product of TOKYO ELETECH CORPORATION.

APPENDIX C REGISTER INDEX

C.1 Register Index (Alphabetic Order of Register Name)

C.2 Register Index (Alphabetic Order of Register Symbol)

APPENDIX D REVISION HISTORY

Here is the revision history of this manual. The “Applied to:” column indicates the chapters of each edition in which the revision was applied.

(1/2)

Edition	Revision from Previous Edition	Applied to:
2nd	Modification of packages • Deletion of 80-pin plastic TQFP (fine pitch) (GK-BE9 type) • Addition of 80-pin plastic TQFP (fine pitch) (GK-9EU type)	Throughout
	Modification of **Table 2-1 Types of Pin I/O Circuits**	CHAPTER 2 PIN FUNCTIONS
	Modification of **Table 4-3 Port Mode Register and Output Latch Settings When Using Alternate Functions**	CHAPTER 4 PORT FUNCTIONS
	Modification of Caution 2 in **6.2 Configuration of 16-Bit Timer (1) 16-bit compare register 50 (CR50)**	CHAPTER 6 16-BIT TIMER
	Modification of **Figure 6-2 Format of 16-Bit Timer Mode Control Register 50**	
	Addition of Caution in **6.4.1 Operation as timer interrupt**	
	Modification of **Figure 6-8 Settings of 16-Bit Timer Mode Control Register 50 for Capture Operation**	
	Addition of Caution in **7.4.3 Operation as square-wave output**	CHAPTER 7 8-BIT TIMER/ EVENT COUNTER
	Addition of Caution in **10.4.1 Basic operation of 8-bit A/D converter**	CHAPTER 10 8-BIT A/D CONVERTER (μPD789407A SUBSERIES)
	Addition of Caution in **11.4.1 Basic operation of 10-bit A/D converter**	CHAPTER 11 10-BIT A/D CONVERTER (μPD789417A SUBSERIES)
	Addition of Caution in **Table 18-1 Differences Between μPD78F9418A and Mask ROM Versions**	CHAPTER 18 μPD78F9418A
	Modification of **Table 18-2 Communication Mode** and addition of Note in it	
	Modification of **Figure 18-4 Flashpro III Connection Example in Pseudo 3-Wire Mode (When P0 Is Used)**	
	Modification of **Table 18-4 Example of Settings for PG-FP3**	
	Modification of product name of flash memory programming adapter in **A.2 Flash Memory Programming Tools**	APPENDIX A DEVELOPMENT TOOLS
	Addition of product name of conversion adapter corresponding to each emulation probe in **A.3.1 Hardware**	
3rd	Modification of pin handling of AV_{REF} pin and V_{PP} pin	CHAPTER 2 PIN FUNCTIONS
	Addition of Note related to feedback resistor	CHAPTER 5 CLOCK GENERATOR
	Addition of **6.5 Cautions on Using 16-Bit Timer 50**	CHAPTER 6 16-BIT TIMER 50
	Addition of **(8) Input impedance of ANI0 to ANI6 pins** in **10.5 Cautions on Using 8-Bit A/D Converter**	CHAPTER 10 8-BIT A/D CONVERTER (μPD789407A SUBSERIES)

Edition	Revision from Previous Edition	Applied to:
3rd	Modification of description of **(2) A/D conversion result register 0 (ADCR0)** in **11.2 Configuration of 10-Bit A/D Converter**	CHAPTER 11 10-BIT A/D CONVERTER (PD789417A SUBSERIES)
	Addition of **(8) Input impedance of ANI0 to ANI6 pins** in **11.5 Cautions on Using 10-Bit A/D Converter**	
	Addition of description on reading receive data of UART	CHAPTER 13 SERIAL INTERFACE 00
	Addition of Caution in **Figure 15-2 Format of Interrupt Request Flag Register**	CHAPTER 15 INTERRUPT FUNCTIONS
	Addition of Caution in **Figure 15-7 Format of Key Return Mode Register 00**	
	Addition of description on pull-up resistor and divider resistor for LCD driving in **Table 18-1 Differences Between PD78F9418A and Mask ROM Versions**	CHAPTER 18 PD78F9418A
	Overall revision of contents related to flash memory programming as **18.1 Flash Memory Characteristics**	
	Addition of electrical specifications	CHAPTER 21 ELECTRICAL SPECIFICATIONS
	Addition of characteristics curves (reference values)	CHAPTER 22 CHARACTERISTICS CURVES (REFERENCE VALUES)
	Addition of package drawings	CHAPTER 23 PACKAGE DRAWINGS
	Addition of recommended soldering conditions	CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS
	Overall revision of contents of development tools Deletion of embedded software	APPENDIX A DEVELOPMENT TOOLS
	Addition of notes on target system design	APPENDIX B NOTES ON TARGET SYSTEM DESIGN
3rd Edition (Modification Version)	Modification of **1.3 Ordering Information**	CHAPTER 1 GENERAL
	Addition of **Table 24-1. Surface Mounting Type Soldering Conditions (3/3)**	CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS