Preliminary User's Manual

μPD780024AS, 780034AS Subseries

8-Bit Single-Chip Microcontrollers

μPD780021AS
μPD780022AS
μPD780023AS
μPD780024AS
μPD780031AS
μPD780032AS
μPD780033AS
μPD780034AS
μPD78F0034BS

Document No. U16035EJ1V0UM00 (1st edition)
Date Published June 2002 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed: μPD78F0034BSGB-8ET

The customer must judge the need for a license for the following products:

μPD780021ASGB-xxx-8ET, 780022ASGB-xxx-8ET, 780023ASGB-xxx-8ET, 780024ASGB-xxx-8ET, 780031ASGB-xxx-8ET, 780032ASGB-xxx-8ET, 780033ASGB-xxx-8ET, 780034ASGB-xxx-8ET

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

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

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

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

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

M5D 98. 12

Regional Information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

J02.4

INTRODUCTION

Readers

This manual has been prepared for user engineers who understand the functions of the μPD780024AS, 780034AS Subseries and wish to design and develop application systems and programs for these devices.

μPD780024AS Subseries: μPD780021AS, 780022AS, 780023AS, 780024AS
μPD780034AS Subseries: μPD780031AS, 780032AS, 780033AS, 780034AS, 78F0034BS

Purpose

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

Organization

The μPD780024AS, 780034AS Subseries manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD780024AS, 780034AS Subseries User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupt • Other on-chip peripheral functions	• CPU functions • 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 gain a general understanding of functions:
  → Read this manual in the order of the **Contents**.
- How to interpret the register format:
  → For the bit number enclosed in square, the bit name is defined as a reserved word in RA78K0, and in CC78K0, already defined in the header file named sfrbit.h.
- To check the details of a register when you know the register name.
  → Refer to **APPENDIX D REGISTER INDEX.**

Differences Between μPD780024AS and 780034AS Subseries

The resolution of the A/D converter differ between the μPD780024AS and 780034AS Subseries products.

Item \ Subseries	μPD780024AS	μPD780034AS
A/D converter	8-bit resolution	10-bit resolution

Conventions

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

Related Documents

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

Documents Related to Devices

Document Name	Document No.
μPD780021A, 780022A, 780023A, 780024A, 780021AY, 780022AY, 780023AY, 780024AY Data Sheet	U14042E
μPD780021A(A), 780022A(A), 780023A(A), 780024A(A), 780021AY(A), 780022AY(A), 780023AY(A), 780024AY(A) Data Sheet	U15131E
μPD780031A, 780032A, 780033A, 780034A, 780031AY, 780032AY, 780033AY, 780034AY Data Sheet	U14044E
μPD780031A(A), 780032A(A), 780033A(A), 780034A(A), 780031AY(A), 780032AY(A), 780033AY(A), 780034AY(A) Data Sheet	U15132E
μPD78F0034A, 78F0034AY Data Sheet	U14040E
78K/0 Series Instructions User's Manual	U12326E
78K/0 Series Basic (I) Application Note	U12704E

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

Document Name		Document No.
RA78K0 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
SM78K0S, SM78K0 System Simulator Ver. 2.10 or Later Windows™ Based	Operation	U14611E
SM78K Series System Simulator Ver. 2.10 or Later	External Part User Open Interface Specifications	U15006E
ID78K0-NS Integrated Debugger Ver. 2.00 or Later Windows Based	Operation	U14379E
ID78K0 Integrated Debugger Windows Based	Reference	U11539E
	Guide	U11649E
RX78K0 Real-time OS	Fundamentals	U11537E
	Installation	U11536E
MX78K0 Embedded OS Windows Based	Fundamental	U12257E

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

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

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-780034-NS-EM1 Emulation Board	U14642E

Documents Related to Flash Memory Writing

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

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products & Packages -	X13769E
Semiconductor Device Mounting Technology Manual	C10535E
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

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

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 OUTLINE

1.1 Features

- Internal memory

Part Number \ Type	Program Memory (ROM/Flash Memory)	Data Memory (High-Speed RAM)
μPD780021AS, 780031AS	8 KB	512 bytes
μPD780022AS, 780032AS	16 KB	
μPD780023AS, 780033AS	24 KB	1024 bytes
μPD780024AS, 780034AS	32 KB	
μPD78F0034BS	32 KB[Note]	1024 bytes[Note]

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

- External memory expansion space: 64 KB
- Minimum instruction execution time changeable from high speed (0.24 μs: @ 8.38 MHz operation with main system clock) to ultra-low speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- I/O ports: 39
- 8-bit resolution A/D converter: 8 channels (μPD780024AS Subseries only)
- 10-bit resolution A/D converter: 8 channels (μPD780034AS Subseries only)
- Serial interface: 3 channels
  - 3-wire serial I/O mode: 2 channels
  - UART mode: 1 channel
- Timer: Five channels
  - 16-bit timer/event counter: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Vectored interrupt sources: 20
- Two types of on-chip clock oscillators (main system clock and subsystem clock)
- Power supply voltage: V_{DD} = 1.8 to 5.5 V

1.2 Applications

Home electric appliances, pagers, AV equipment, car audios, car electric equipment, office automation equipment, etc.

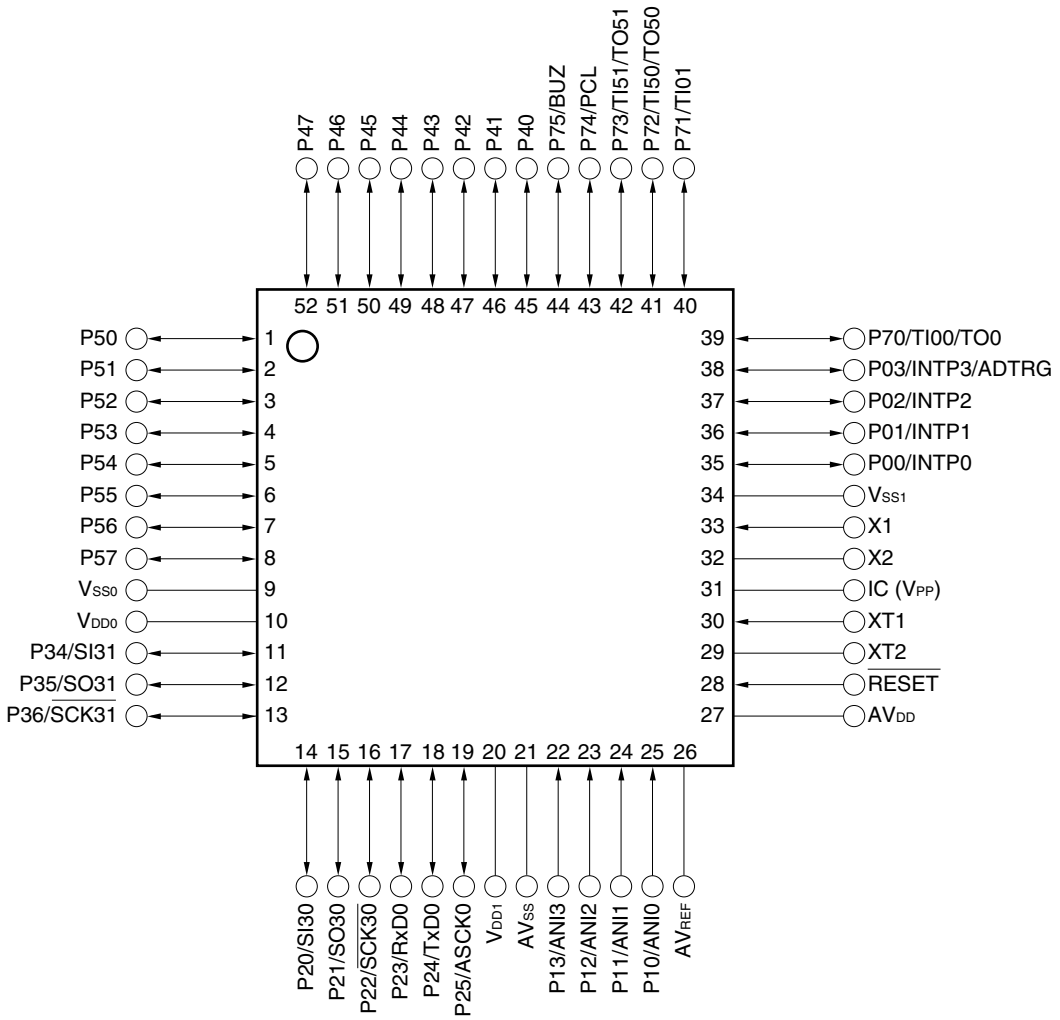
1.3 Ordering Information

Part Number	Package	Internal ROM
μPD780021ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD780022ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD780023ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD780024ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD780031ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD780032ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD780033ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD780034ASGB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F0034BSGB-8ET	52-pin plastic LQFP (10 × 10)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

- **52-pin plastic LQFP (10 × 10)**
  μPD780021ASGB-×××-8ET, 780022ASGB-×××-8ET,
  μPD780023ASGB-×××-8ET, 780024ASGB-×××-8ET,
  μPD780031ASGB-×××-8ET, 780032ASGB-×××-8ET,
  μPD780033ASGB-×××-8ET, 780034ASGB-×××-8ET,
  μPD780034BSGB-8ET



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

2. Connect AV_{SS} pin to V_{SS0}.

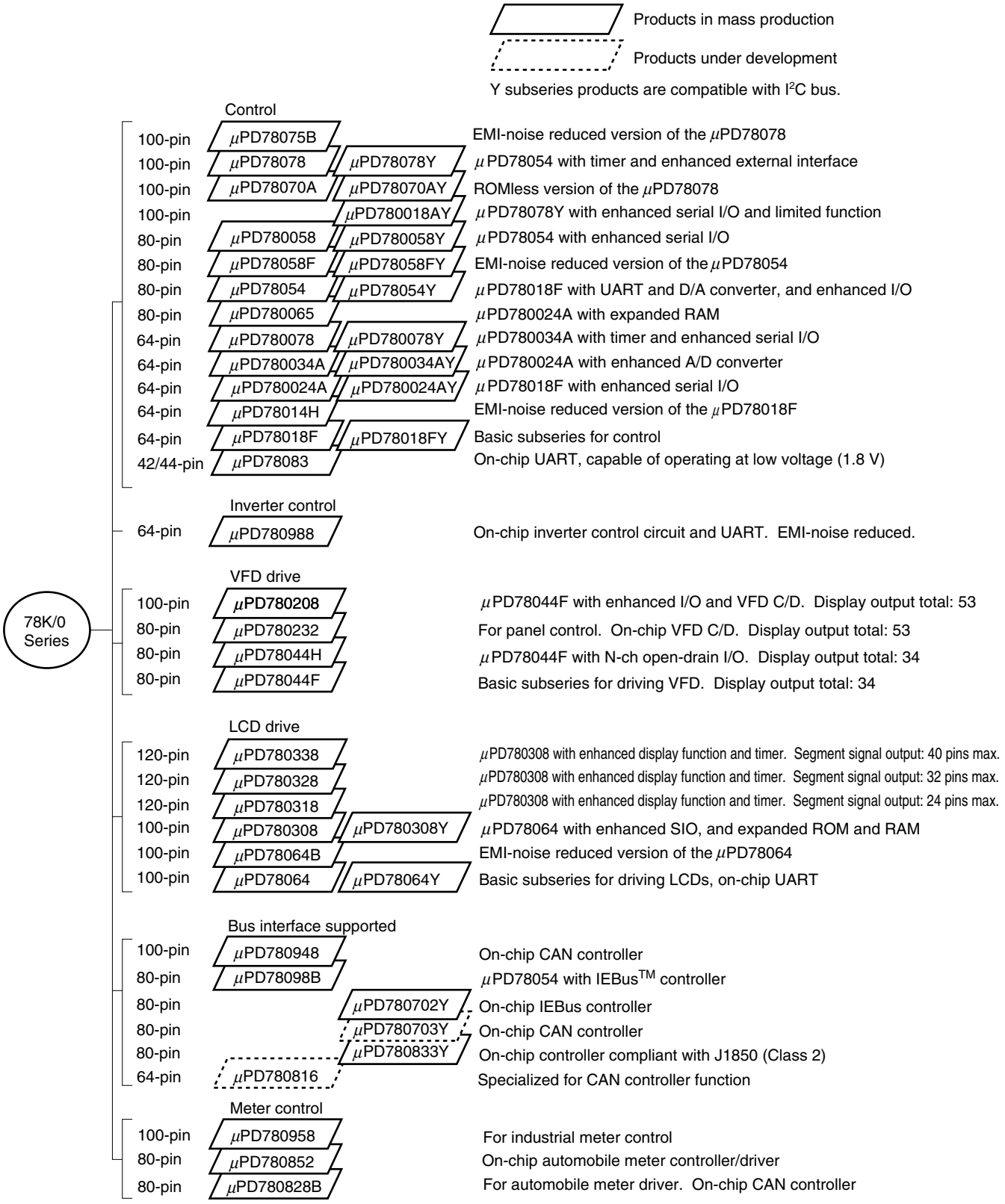
Remarks 1. When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of noise measures is recommended, such as supplying V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

2. Pin connection in parentheses is intended for the μPD78F0034BS.

ADTRG:	AD trigger input
ANI0 to ANI3:	Analog input
ASCK0:	Asynchronous serial clock
AV_{DD}:	Analog power supply
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog ground
BUZ:	Buzzer clock
IC:	Internally connected
INTP0 to INTP3:	External interrupt input
P00 to P03:	Port 0
P10 to P13:	Port 1
P20 to P25:	Port 2
P34 to P36:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P70 to P75:	Port 7
PCL:	Programmable clock
$\overline{\text{RESET}}$:	Reset
RxD0:	Receive data
$\overline{\text{SCK30}}$, $\overline{\text{SCK31}}$:	Serial clock
SI30, SI31:	Serial input
SO30, SO31:	Serial output
TI00, TI01, TI50, TI51:	Timer input
TO0, TO50, TO51:	Timer output
TxD0:	Transmit data
V_{DD0}, V_{DD1}:	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/0 Series Lineup

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



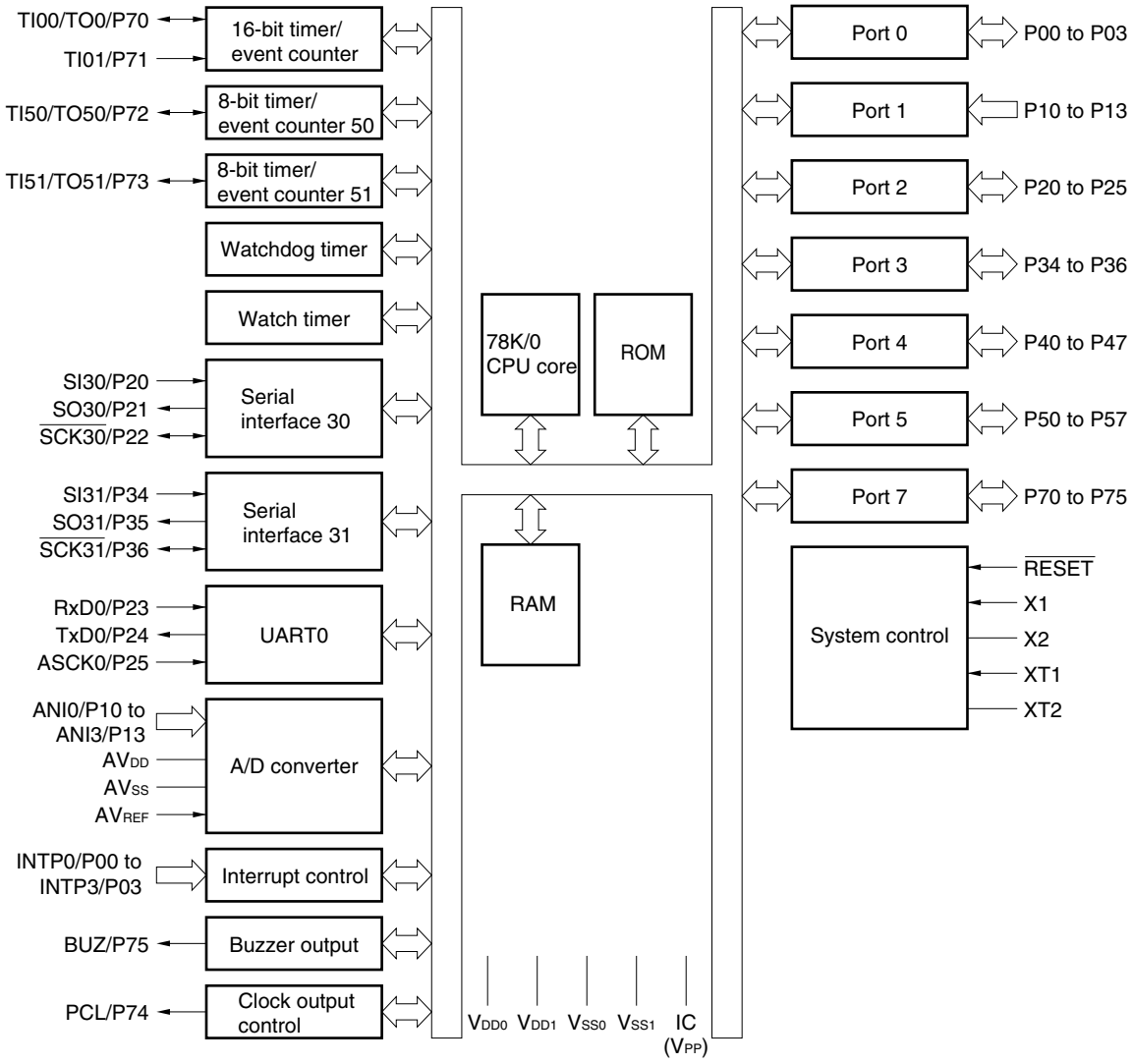
Remark VFD (Vacuum Fluorescent Display) is referred to as FIP™ (Fluorescent Indicator Panel) in some documents, but the functions of the two are the same.

The major functional differences among the subseries are listed below.

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

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

1.6 Block Diagram



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

2. Pin connection in parentheses is intended for the μPD78F0034BS.

1.7 Outline of Function

Item \ Part Number		μPD780021AS μPD780031AS	μPD780022AS μPD780032AS	μPD780023AS μPD780033AS	μPD780024AS μPD780034AS	μPD78F0034BS
Internal memory	ROM	8 KB (Mask ROM)	16 KB (Mask ROM)	24 KB (Mask ROM)	32 KB (Mask ROM)	32 KB[Note] (Flash memory)
	High-speed RAM	512 bytes		1024 bytes		1024 bytes[Note]
Memory space		64 KB				
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		Minimum instruction execution time changeable function				
	When main system clock selected	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (@ 8.38 MHz operation)				
	When subsystem clock selected	122 μs (@ 32.768 kHz operation)				
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.				
I/O port		Total: 39 • CMOS input: 4 • CMOS I/O: 35				
A/D converter		• 8-bit resolution × 4 channels (μPD780021AS, 780022AS, 780023AS, 780024AS) • 10-bit resolution × 4 channels (μPD780031AS, 780032AS, 780033AS, 780034AS, 78F0034BS) • Low-voltage operation: AV_{DD} = 1.8 to 5.5 V				
Serial interface		• 3-wire serial I/O mode: 2 channels • UART mode: 1 channel				
Timer		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel				
Timer output		Three outputs (8-bit PWM output enable: 2)				
Clock output		• 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (8.38 MHz with main system clock) • 32.768 kHz (32.768 kHz with subsystem clock)				
Buzzer output		1.02 kHz, 2.05 kHz, 4.10 kHz, 8.19 kHz (8.38 MHz with main system clock)				
Vectored interrupt source	Maskable	Internal: 13, External: 5				
	Non-maskable	Internal: 1				
	Software	1				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = –40 to +85°C				
Package		• 52-pin plastic LQFP (10 × 10)				

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

The outline of the timer/event counter is as follows (for details, refer to **CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0**, **CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50 AND 51**, **CHAPTER 8 WATCH TIMER**, and **CHAPTER 9 WATCHDOG TIMER**).

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

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

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

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0 4-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	Input	INTP0
P01				INTP1
P02				INTP2
P03				INTP3/ADTRG
P10 to P13	Input	Port 1 4-bit input-only port.	Input	ANI0 to ANI3
P20	I/O	Port 2 6-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	Input	SI30
P21				SO30
P22				$\overline{\text{SCK30}}$
P23				RxD0
P24				TxD0
P25				ASCK0
P34	I/O	Port 3 3-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	Input	SI31
P35				SO31
P36				$\overline{\text{SCK31}}$
P40 to P47	I/O	Port 4 8-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings. Interrupt request flag (KRIF) is set to 1 by falling edge detection.	Input	—
P50 to P57	I/O	Port 5 8-bit I/O port LEDs can be driven directly. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	Input	—
P70	I/O	Port 7 6-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	Input	TI00/TO0
P71				TI01
P72				TI50/TO50
P73				TI51/TO51
P74				PCL
P75				BUZ

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

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input with specifiable valid edges (rising edge, falling edge, both rising and falling edges)	Input	P00
INTP1				P01
INTP2				P02
INTP3				P03/ADTRG
SI30	Input	Serial interface serial data input	Input	P20
SI31				P34
SO30	Output	Serial interface serial data output	Input	P21
SO31				P35
$\overline{\text{SCK30}}$	I/O	Serial interface serial clock input/output	Input	P22
$\overline{\text{SCK31}}$				P36
RxD0	Input	Asynchronous serial interface serial data input	Input	P23
TxD0	Output	Asynchronous serial interface serial data output	Input	P24
ASCK0	Input	Asynchronous serial interface serial clock input	Input	P25
TI00	Input	External count clock input to 16-bit timer/event counter 0 Capture trigger input to 16-bit timer/event counter 0 capture register (CR00, CR01)	Input	P70/TO0
TI01		Capture trigger input to 16-bit timer/event counter 0 capture register (CR00)		P71
TI50		External count clock input to 8-bit timer/event counter 50		P72/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P73/TO51
TO0	Output	16-bit timer/event counter 0 output	Input	P70/TI00
TO50		8-bit timer/event counter 50 output (also used for 8-bit PWM output)	Input	P72/TI50
TO51		8-bit timer/event counter 51 output (also used for 8-bit PWM output)		P73/TI51
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P74
BUZ	Output	Buzzer output	Input	P75
ANI0 to ANI3	Input	A/D converter analog input	Input	P10 to P13
ADTRG	Input	A/D converter trigger signal input	Input	P03/INTP3
AV_{REF}	Input	A/D converter reference voltage input	—	—
AV_{DD}	—	A/D converter analog power supply. Connect to V_{DD0} or V_{DD1}.	—	—
AV_{SS}	—	A/D converter ground potential. Connect to V_{SS0} or V_{SS1}.	—	—
$\overline{\text{RESET}}$	Input	System reset input	Input	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	—	—
XT2	—		—	—
V_{DD0}	—	Positive power supply for ports	—	—
V_{DD1}	—	Positive power supply other than ports	—	—

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

Pin Name	I/O	Function	After Reset	Alternate Function
V_{SS0}	—	Ground potential for ports	—	—
V_{SS1}	—	Ground potential other than ports	—	—
IC	—	Internally connected. Connect directly to V_{SS0} or V_{SS1}.	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS0} or V_{SS1} in normal operating mode.	—	—

2.2 Description of Pin Functions

2.2.1 P00 to P03 (Port 0)

These are 4-bit I/O ports. Besides serving as I/O ports, they function as an external interrupt input, and A/D converter external trigger input.

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

(1) Port mode

These ports function as 4-bit I/O ports.

P00 to P03 can be specified as input or output ports in 1-bit units with port mode register 0 (PM0). On-chip pull-up resistors can be used by setting pull-up resistor option register 0 (PU0).

(2) Control mode

These ports function as an external interrupt request input, and A/D converter external trigger input.

(a) INTP0 to INTP3

INTP0 to INTP3 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges).

(b) ADTRG

A/D converter external trigger input pin.

Caution When P03 is used as an A/D converter external trigger input, specify the valid edge by bits 1, 2 (EGA00, EGA01) of A/D converter mode register (ADM0) and set interrupt mask flag (PMK3) to 1.

2.2.2 P10 to P13 (Port 1)

These are 4-bit input-only ports. Besides serving as input ports, they function as an A/D converter analog input.

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

(1) Port mode

These ports function as 4-bit input-only ports.

(2) Control mode

These ports function as A/D converter analog input pins (ANI0 to ANI3).

2.2.3 P20 to P25 (Port 2)

These are 6-bit I/O ports. Besides serving as I/O ports, they function as serial interface data I/O and clock I/O. The following operating modes can be specified in 1-bit units.

(1) Port mode

These ports function as 6-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 2 (PM2). On-chip pull-up resistors can be used by setting pull-up resistor option register 2 (PU2).

(2) Control mode

These ports function as serial interface data I/O and clock I/O.

(a) SI30 and SO30

Serial interface serial data I/O pins.

(b) $\overline{\text{SCK30}}$

Serial interface serial clock I/O pin.

(c) RxD0 and TxD0

Asynchronous serial interface serial data I/O pins.

(d) ASCK0

Asynchronous serial interface serial clock input pin.

2.2.4 P34 to P36 (Port 3)

These are 3-bit I/O ports. Besides serving as I/O ports, they function as serial interface data I/O and clock I/O. The following operating modes can be specified in 1-bit units.

(1) Port mode

These ports function as 3-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 3 (PM3). On-chip pull-up resistors can be used by setting pull-up resistor option register 3 (PU3).

(2) Control mode

These ports function as serial interface data I/O and clock I/O.

(a) SI31 and SO31

Serial interface serial data I/O pins.

(b) $\overline{\text{SCK31}}$

Serial interface serial clock I/O pin.

2.2.5 P40 to P47 (Port 4)

These are 8-bit I/O ports.

The interrupt request flag (KRIF) can be set to 1 by detecting a falling edge.

The following operating mode can be specified in 1-bit units.

Caution When using the falling edge detection interrupt (INTKR), be sure to set the memory expansion mode register (MEM) to 01H.

(1) Port mode

These ports function as 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 4 (PM4). On-chip pull-up resistors can be used by setting pull-up resistor option register 4 (PU4).

2.2.6 P50 to P57 (Port 5)

These are 8-bit I/O ports.

Port 5 can drive LEDs directly.

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

(1) Port mode

These ports function as 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 5 (PM5). On-chip pull-up resistors can be used by setting pull-up resistor option register 5 (PU5).

2.2.7 P70 to P75 (Port 7)

These are 6-bit I/O ports. Besides serving as I/O ports, they function as a timer I/O, clock output, and buzzer output. The following operating modes can be specified in 1-bit units.

(1) Port mode

Port 7 functions as a 6-bit I/O port. They can be specified as an input port or output port in 1-bit units with port mode register 7 (PM7). On-chip pull-up resistors can be used by setting pull-up resistor option register 7 (PU7). P70 and P71 are also 16-bit timer/event counter capture trigger signal input pins with a valid edge input.

(2) Control mode

Port 7 functions as timer I/O, clock output, and buzzer output.

(a) TI00

External count clock input pin to 16-bit timer/event counter and capture trigger signal input pin to 16-bit timer/event counter capture register (CR01).

(b) TI01

Capture trigger signal input pin to 16-bit timer/event counter capture register (CR00).

(c) TI50 and TI51

External count clock input pins to 8-bit timer/event counter.

(d) TO0, TO50, and TO51

Timer output pins.

(e) PCL

Clock output pin.

(f) BUZ

Buzzer output pin.

2.2.8 AV_{REF}

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

When no A/D converter is used, connect this pin to V_{SS0}.

2.2.9 AV_{DD}

This is an analog power supply pin of A/D converter. Always use the same potential as that of the V_{DD0} pin or V_{DD1} pin even when no A/D converter is used.

2.2.10 AV_{SS}

This is a ground potential pin of A/D converter. Always use the same potential as that of the V_{SS0} pin or V_{SS1} pin even when no A/D converter is used.

2.2.11 $\overline{\text{RESET}}$

This is a low-level active system reset input pin.

2.2.12 X1 and X2

Crystal resonator connect pins for main system clock oscillation.

For external clock supply, input clock signal to X1 and its inverted signal to X2.

2.2.13 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

For external clock supply, input the clock signal to XT1 and its inverted signal to XT2.

2.2.14 V_{DD0} and V_{DD1}

V_{DD0} is a positive power supply port pin.

V_{DD1} is a positive power supply pin other than port pin.

2.2.15 V_{SS0} and V_{SS1}

V_{SS0} is a ground potential port pin.

V_{SS1} is a ground potential pin other than port pin.

2.2.16 V_{PP} (flash memory versions only)

High-voltage apply pin for flash memory programming mode setting and program write/verify.

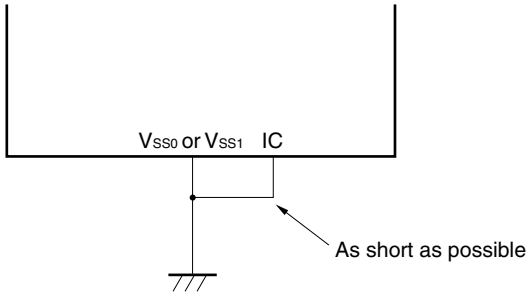
Connect directly to V_{SS0} or V_{SS1} in the normal operating mode.

2.2.17 IC (mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780024AS, 780034AS Subseries at delivery. Connect it directly to the V_{SS0} or V_{SS1} pin with the shortest possible wire in the normal operating mode.

When a potential difference is produced between the IC pin and V_{SS0} pin or V_{SS1} pin, because the wiring between those two pins is too long or an external noise is input to the IC pin, the user's program may not operate normally.

- **Connect IC pins to V_{SS0} pins or V_{SS1} pins directly.**



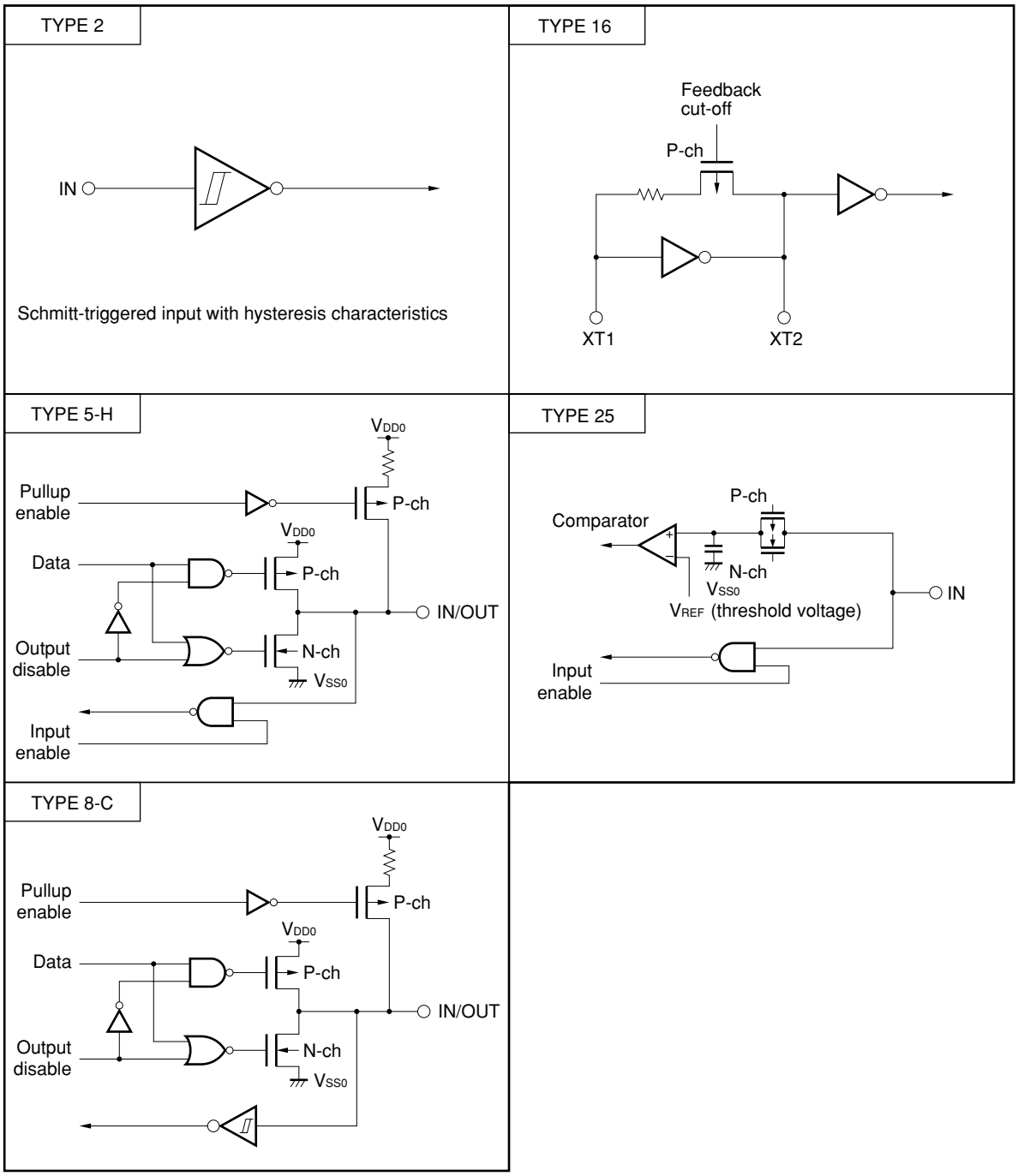
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

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

Table 2-1. Pin I/O Circuit Types

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP0 to P02/INTP2	8-C	I/O	Input: Independently connect to V_{SS0} via a resistor. Output: Leave open
P03/INTP3/ADTRG			
P10/ANI0 to P13/ANI3	25	Input	Connect to V_{DD0} or V_{SS0}.
P20/SI30	8-C	I/O	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P21/SO30	5-H		
P22/$\overline{\text{SCK30}}$	8-C		
P23/RxD0			
P24/TxD0	5-H		
P25/ASCK0	8-C		
P34/SI31	8-C	I/O	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P35/SO31	5-H		
P36/$\overline{\text{SCK31}}$	8-C		
P40 to P47	5-H	I/O	Input: Independently connect to V_{DD0} via a resistor. Output: Leave open.
P50 to P57	5-H	I/O	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P70/TI00/TO0	8-C	I/O	
P71/TI01			
P72/TI50/TO50			
P73/TI51/TO51			
P74/PCL	5-H		
P75/BUZ			
$\overline{\text{RESET}}$	2	Input	—
XT1	16		Connect to V_{DD0}.
XT2		—	Leave open.
AV_{DD}	—		Connect to V_{DD0} or V_{DD1}.
AV_{REF}			Connect to V_{SS0} or V_{SS1}.
AV_{SS}			
IC (for mask ROM version)			Connect directly to V_{SS0} or V_{SS1}.
V_{PP} (for flash memory version)			

Figure 2-1. Pin I/O Circuit List



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Spaces

μPD780024AS, 780034AS Subseries can access 64 KB memory space respectively.
Figures 3-1 to 3-5 show memory maps.

Caution In case of the internal memory capacity, the initial value of memory size switching register (IMS) of all products (μPD780024AS, 780034AS Subseries) is fixed (CFH). Therefore, set the value corresponding to each products indicated below.

μPD780021AS, 780031AS: 42H
μPD780022AS, 780032AS: 44H
μPD780023AS, 780033AS: C6H
μPD780024AS, 780034AS: C8H
μPD78F0034BS: Value for mask ROM version

Figure 3-1. Memory Map (μPD780021AS, 780031AS)

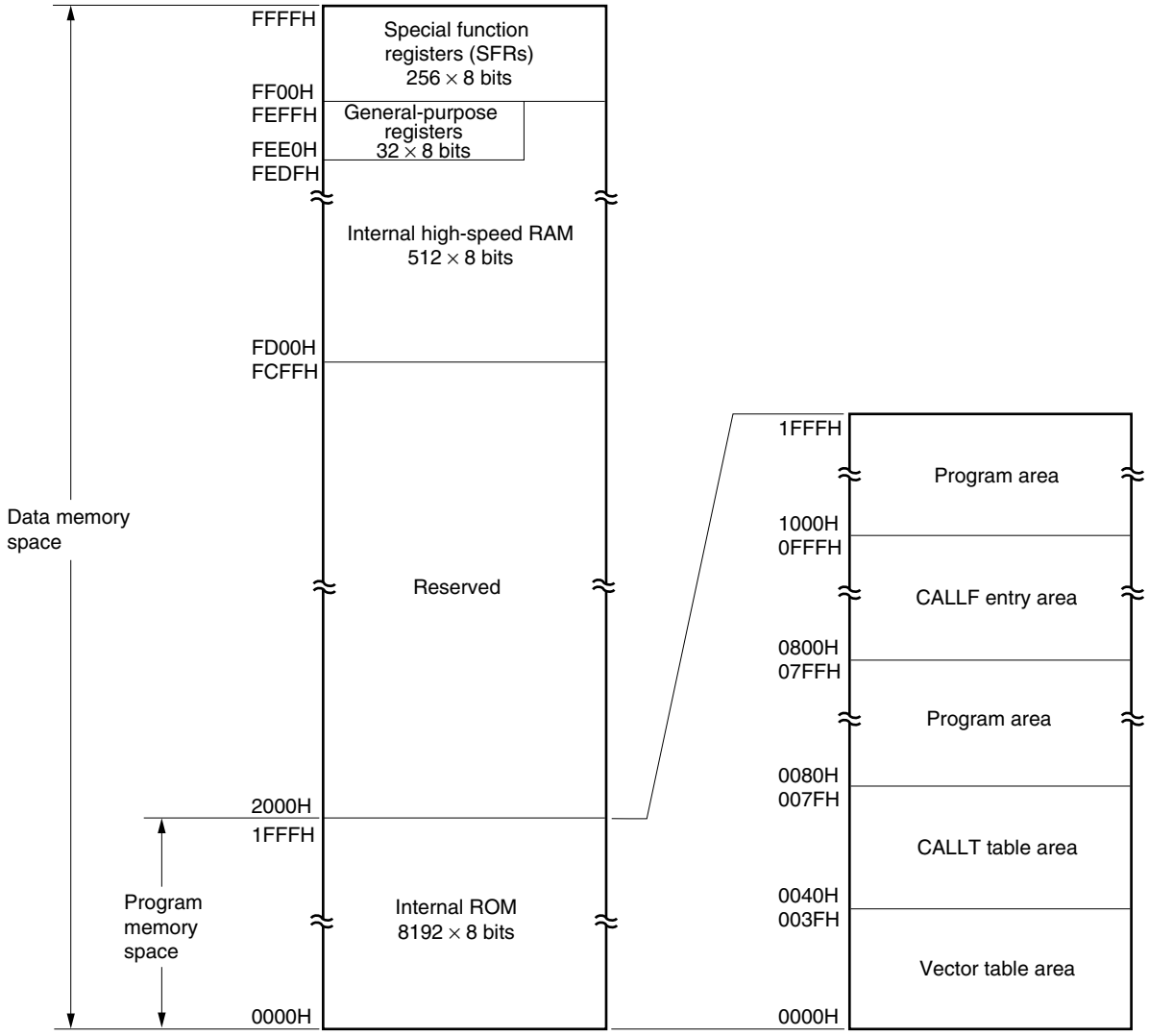


Figure 3-2. Memory Map (μPD780022AS, 780032AS)

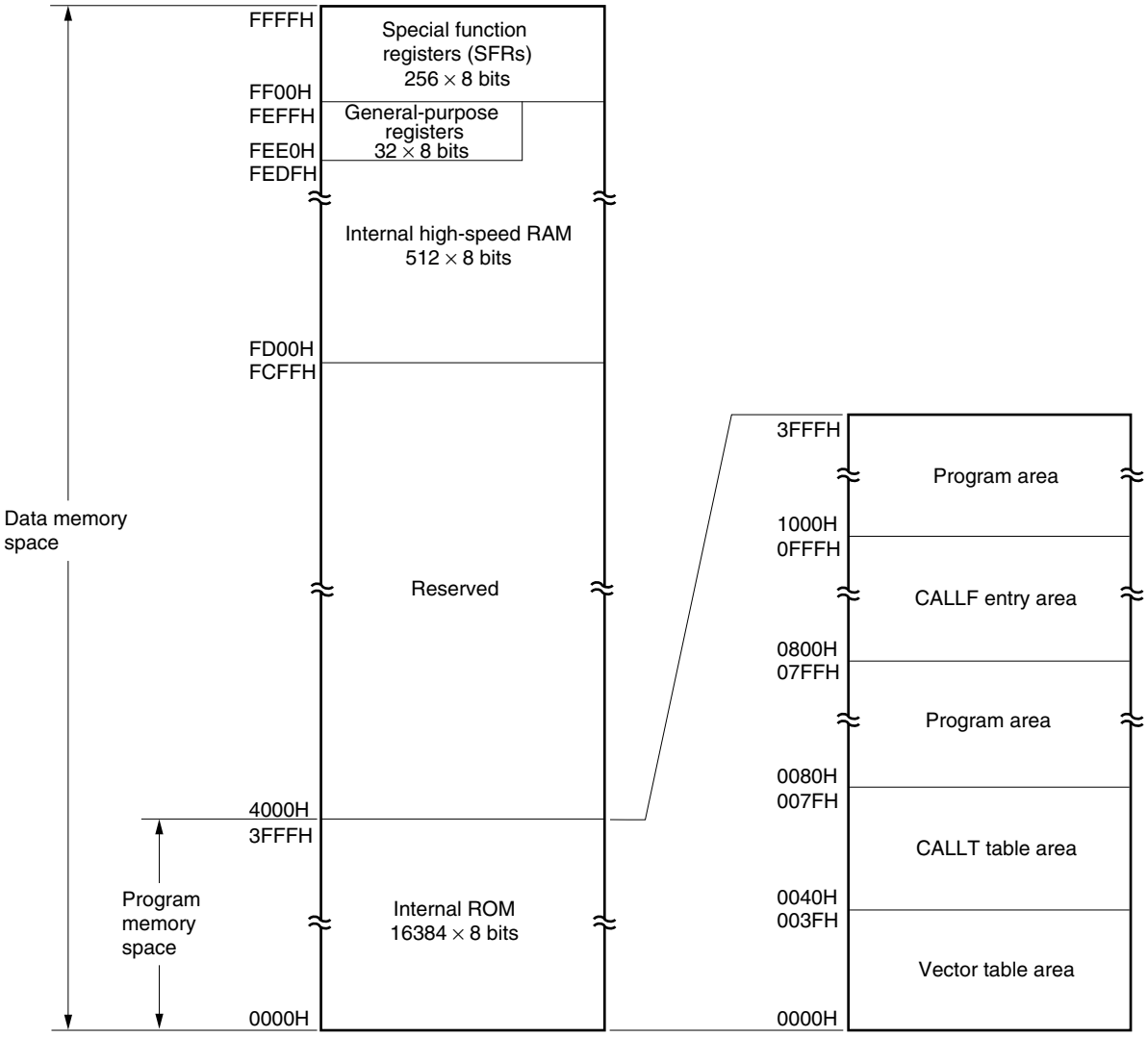


Figure 3-3. Memory Map (μPD780023AS, 780033AS)

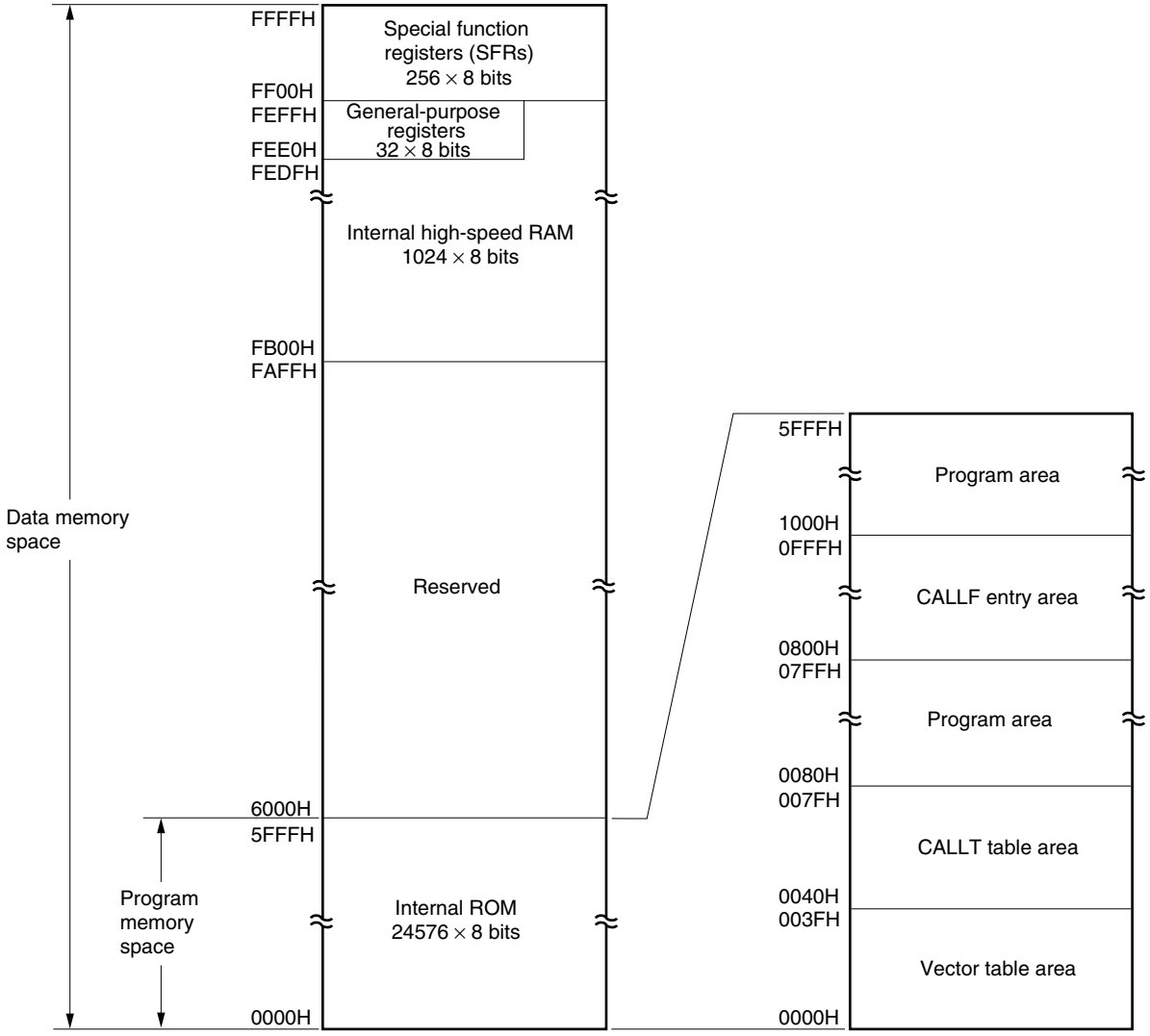


Figure 3-4. Memory Map (μPD780024AS, 780034AS)

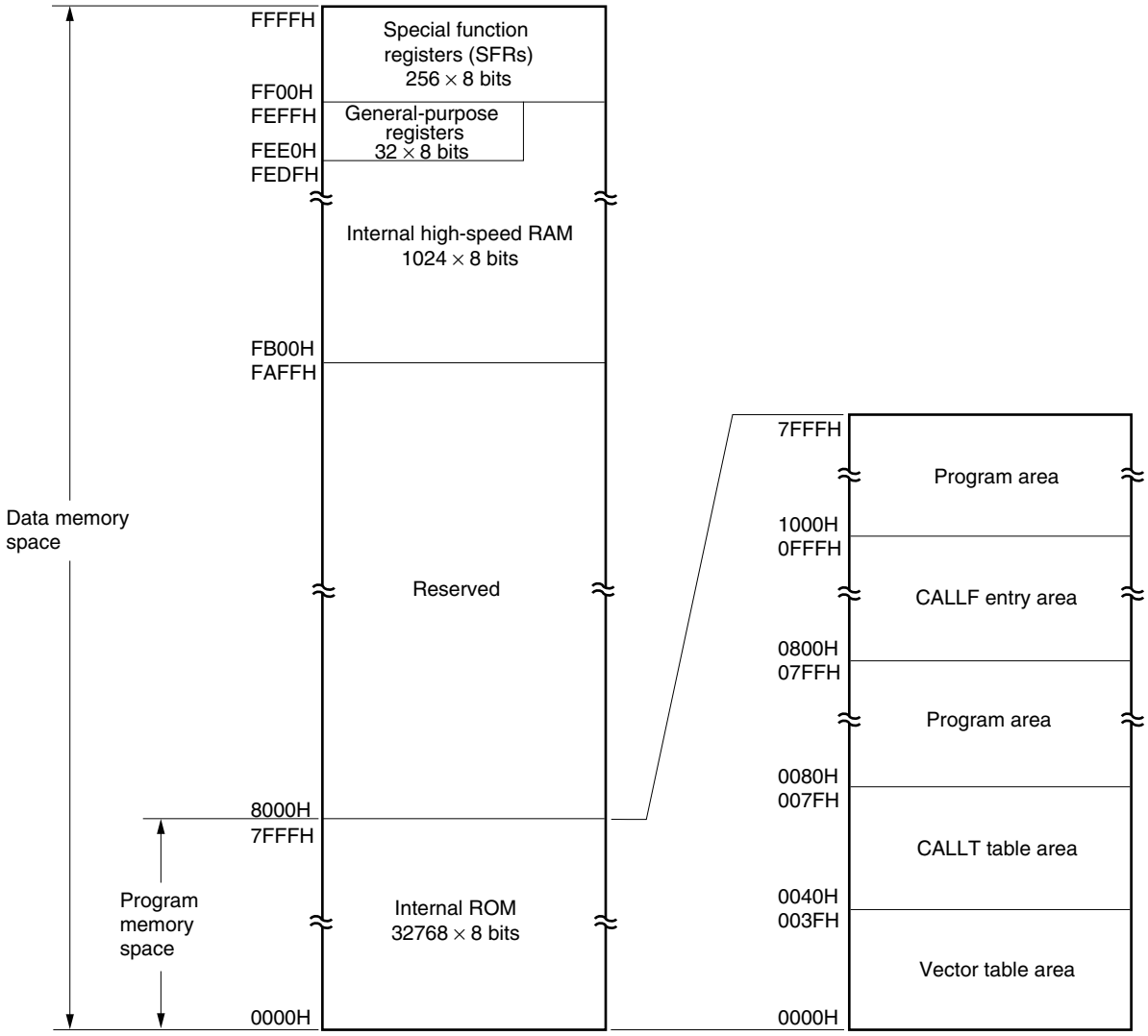
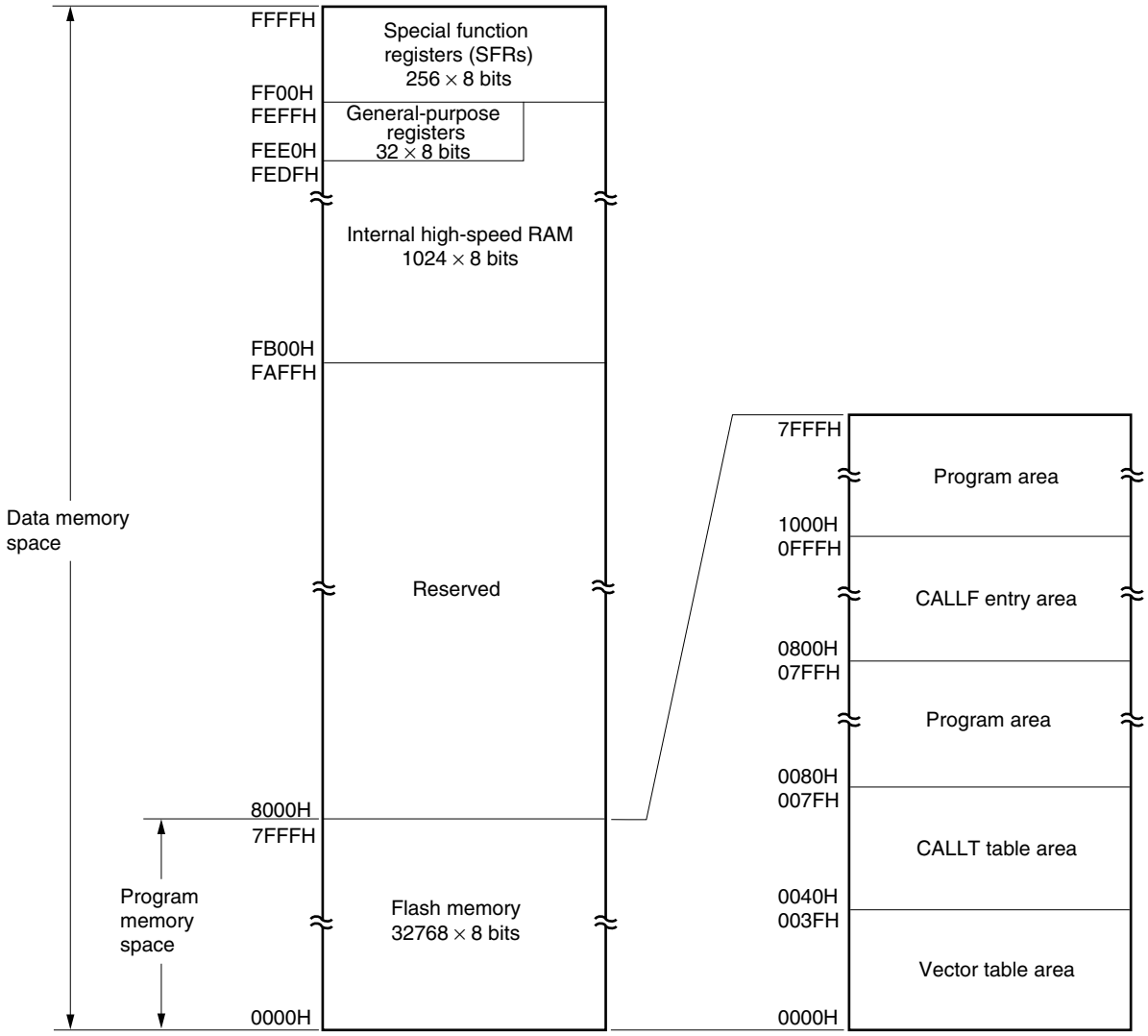


Figure 3-5. Memory Map (μPD78F0034BS)



3.1.1 Internal program memory space

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

The μPD780024AS, 780034AS Subseries products incorporate an on-chip ROM (or flash memory), as listed below.

Table 3-1. Internal ROM Capacity

Part Number	Type	Capacity
μPD780021AS, 780031AS	Mask ROM	8192 × 8 bits (0000H to 1FFFH)
μPD780022AS, 780032AS		16384 × 8 bits (0000H to 3FFFH)
μPD780023AS, 780033AS		24576 × 8 bits (0000H to 5FFFH)
μPD780024AS, 780034AS		32768 × 8 bits (0000H to 7FFFH)
μPD78F0034BS	Flash memory	32768 × 8 bits (0000H to 7FFFH)

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The $\overline{\text{RESET}}$ input and program start addresses for branch upon generation of each interrupt request are stored in the vector table area. Of the 16-bit address, lower 8 bits are stored at even addresses and higher 8 bits are stored at odd addresses.

Table 3-2. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input	0016H	INTCSI31
0004H	INTWDT	001AH	INTWTI
0006H	INTP0	001CH	INTTM00
0008H	INTP1	001EH	INTTM01
000AH	INTP2	0020H	INTTM50
000CH	INTP3	0022H	INTTM51
000EH	INTSER0	0024H	INTAD0
0010H	INTSR0	0026H	INTWT
0012H	INTST0	0028H	INTKR
0014H	INTCSI30	003EH	BRK

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The μPD780024AS, 780034AS Subseries products incorporate an internal high-speed RAM, as listed below.

Table 3-3. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD780021AS, 780031AS	512 × 8 bits (FD00H to FEFFH)
μPD780022AS, 780032AS	
μPD780023AS, 780033AS	1024 × 8 bits (FB00H to FEFFH)
μPD780024AS, 780034AS	
μPD78F0034BS	

The 32-byte area FEE0H to FEFFH is allocated four general-purpose register banks composed of eight 8-bit registers.

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

3.1.3 Special function register (SFR) area

An on-chip peripheral hardware special function register (SFR) is allocated in the area FF00H to FFFFH (refer to **3.2.3 Special function register (SFR) Table 3-5 Special Function Register List**).

Caution Do not access addresses where the SFR is not assigned.

3.1.4 External memory space

The external memory space is accessible with memory expansion mode register (MEM). External memory space can store program, table data, etc., and allocate peripheral devices.

3.1.5 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780024AS, 780034AS Subseries, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Data memory addressing is illustrated in Figures 3-6 to 3-10. For the details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-6. Data Memory Addressing (μPD780021AS, 780031AS)

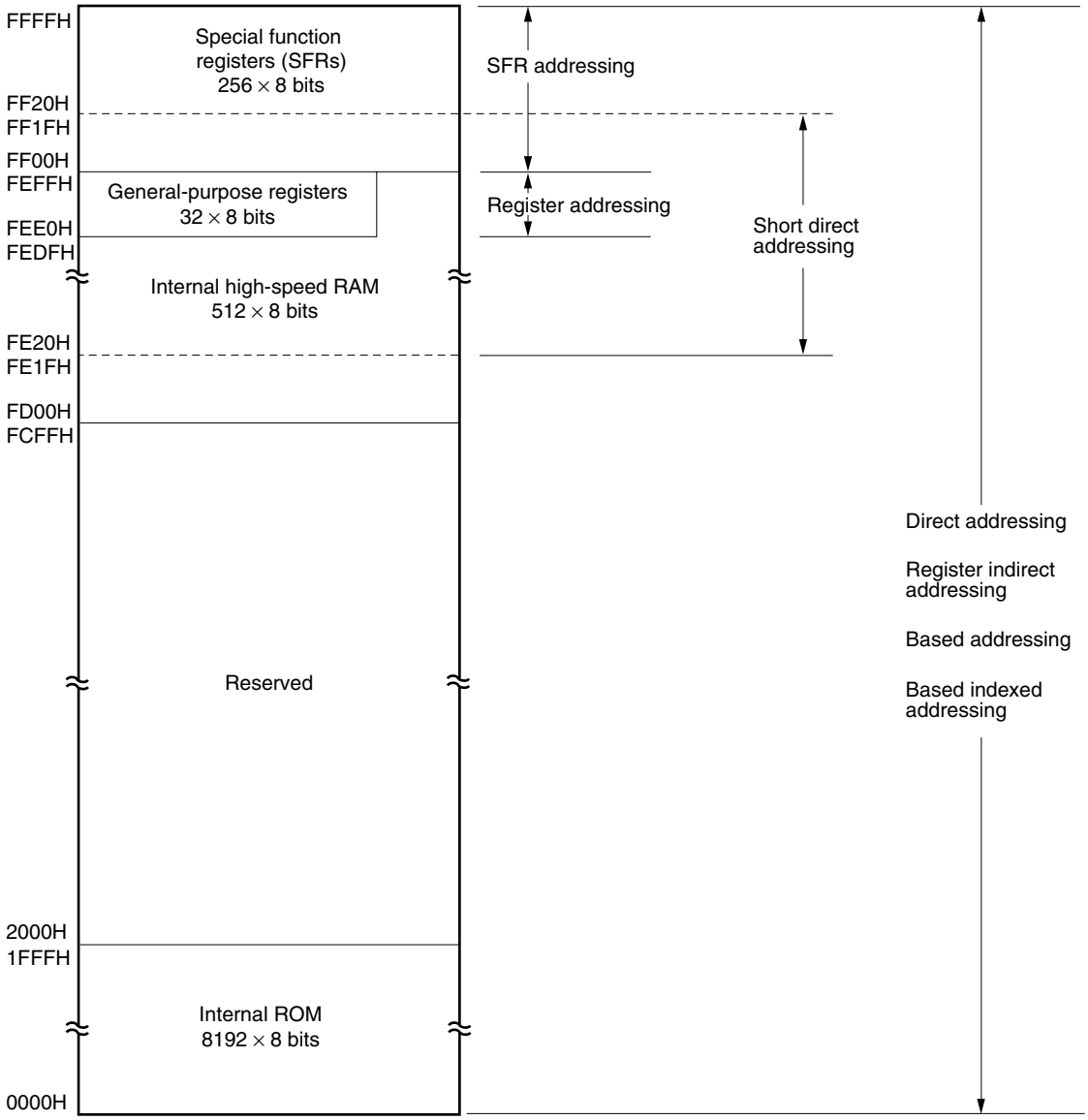


Figure 3-7. Data Memory Addressing (μPD780022AS, 780032AS)

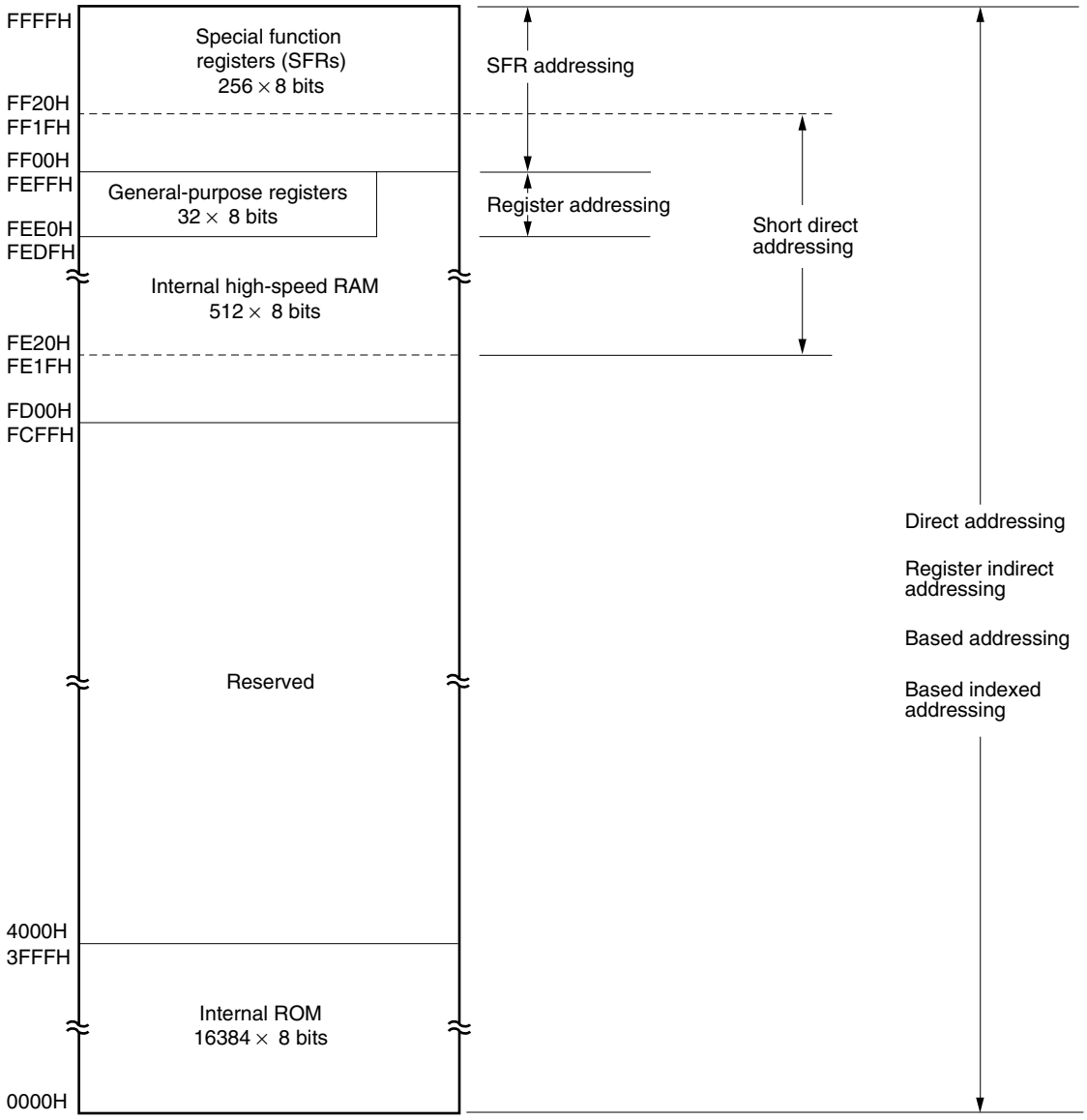


Figure 3-8. Data Memory Addressing (μPD780023AS, 780033AS)

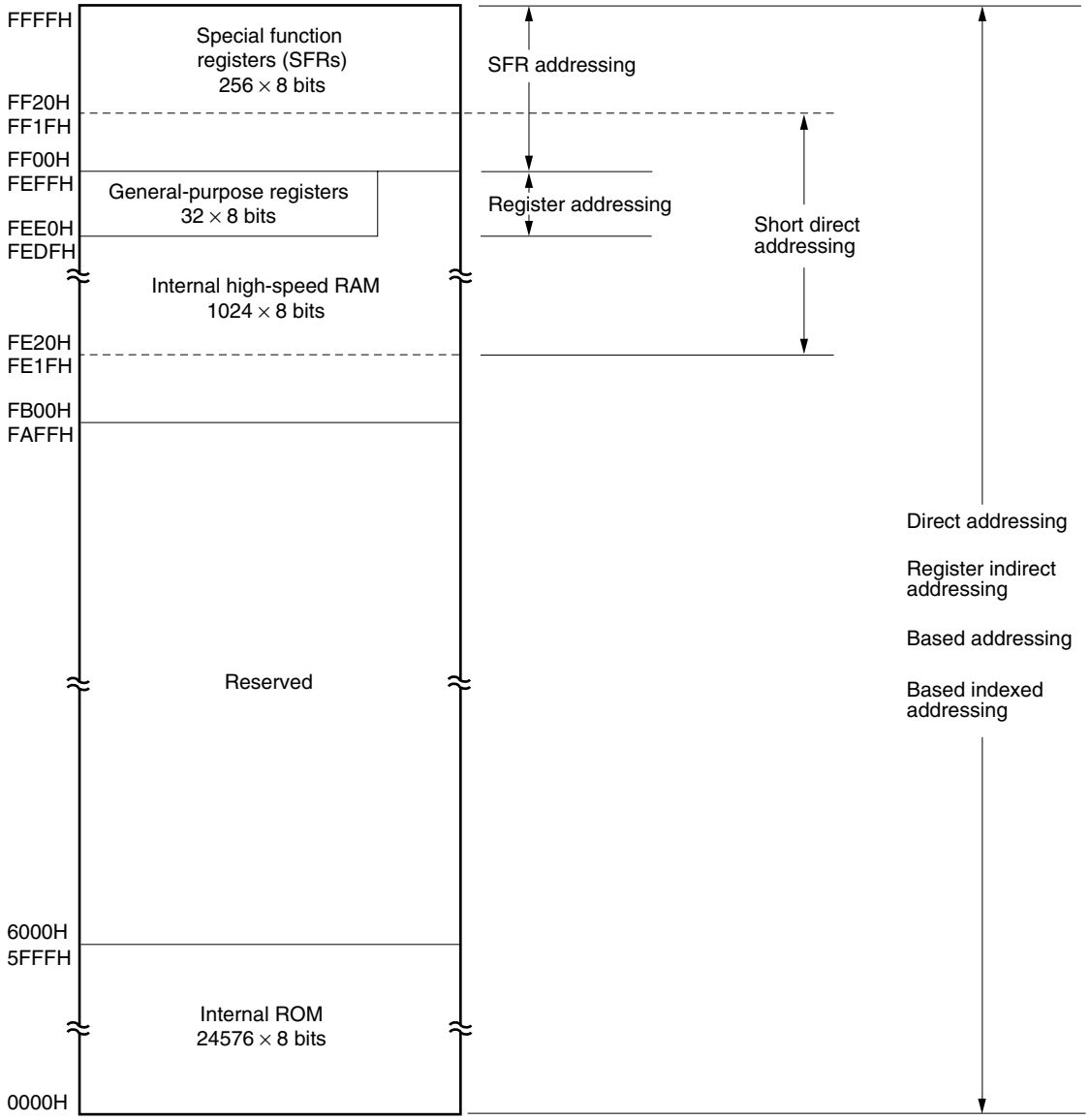


Figure 3-9. Data Memory Addressing (μPD780024AS, 780034AS)

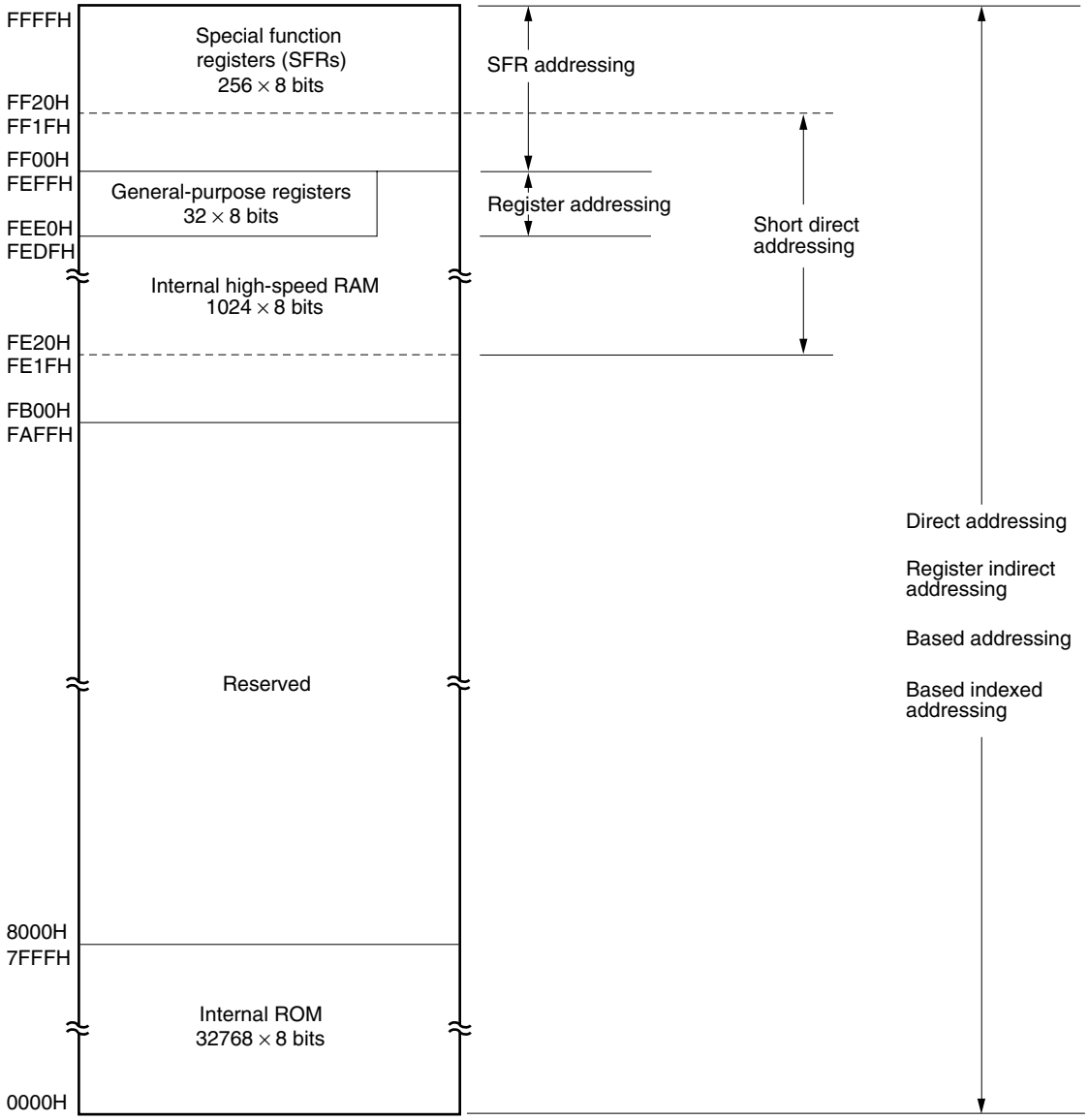
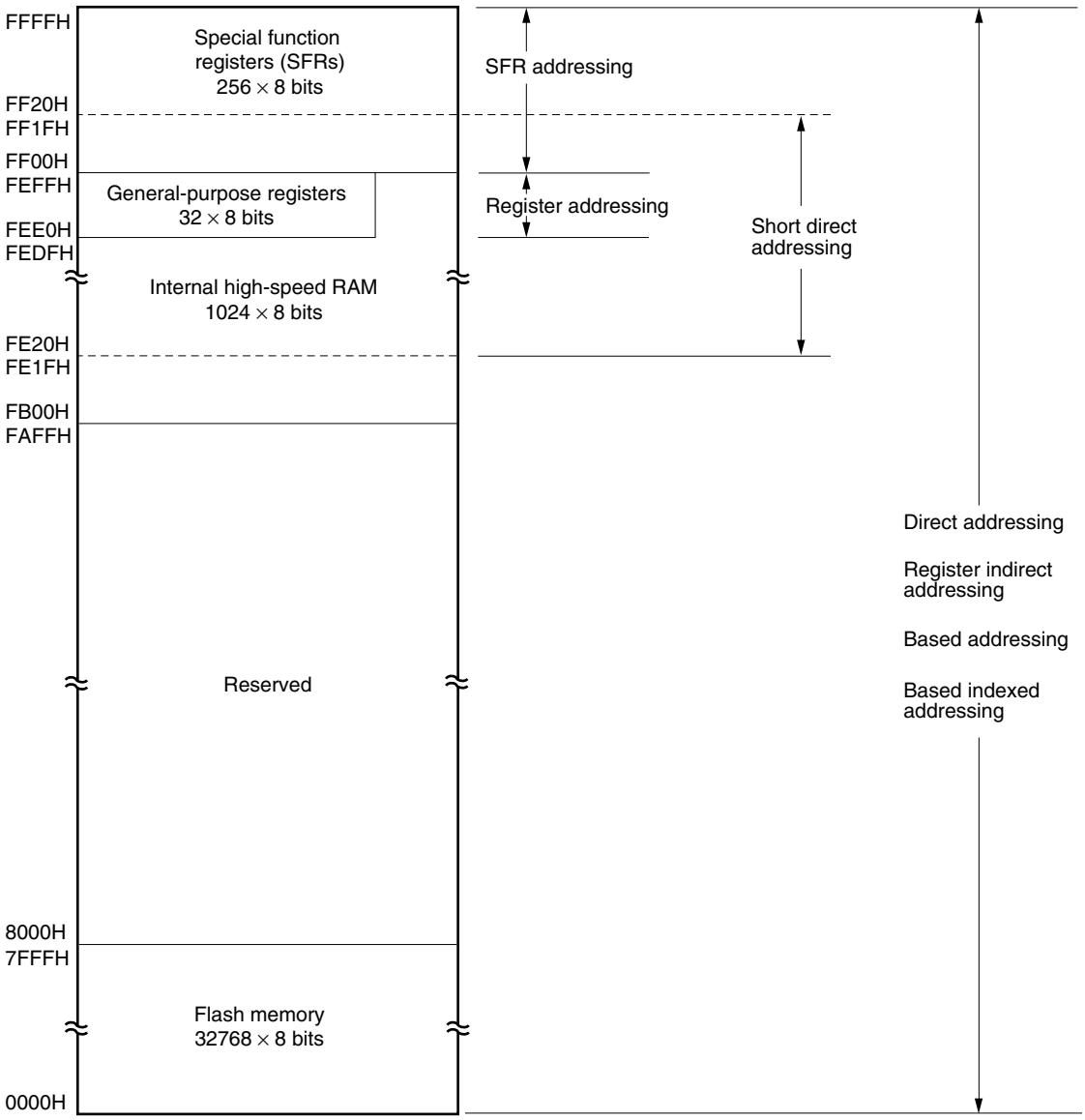


Figure 3-10. Data Memory Addressing (μPD78F0034BS)



3.2 Processor Registers

The μPD780024AS, 780034AS Subseries products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

(1) Program counter (PC)

The program counter is a 16-bit register which holds the address information of the next program to be executed. In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

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

Figure 3-11. Format of Program Counter

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

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags 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 reset upon execution of the RETB, RETI and POP PSW instructions.

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

Figure 3-12. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified with a priority specification flag register (PR0L, PR0H, PR1L) (refer to **15.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. When it is 1, all interrupts are acknowledgeable. Actual request acknowledgement is controlled with the interrupt enable flag (IE).

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Table 3-4. Internal High-Speed RAM Area

Part Number	Internal High-Speed RAM Area
μPD780021AS, 780022AS, 780031AS, 780032AS	FD00H to FEFFH
μPD780023AS, 780024AS, 780033AS, 780034AS, 78F0034BS	FB00H to FEFFH

Figure 3-13. Format of Stack Pointer

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

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

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

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

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

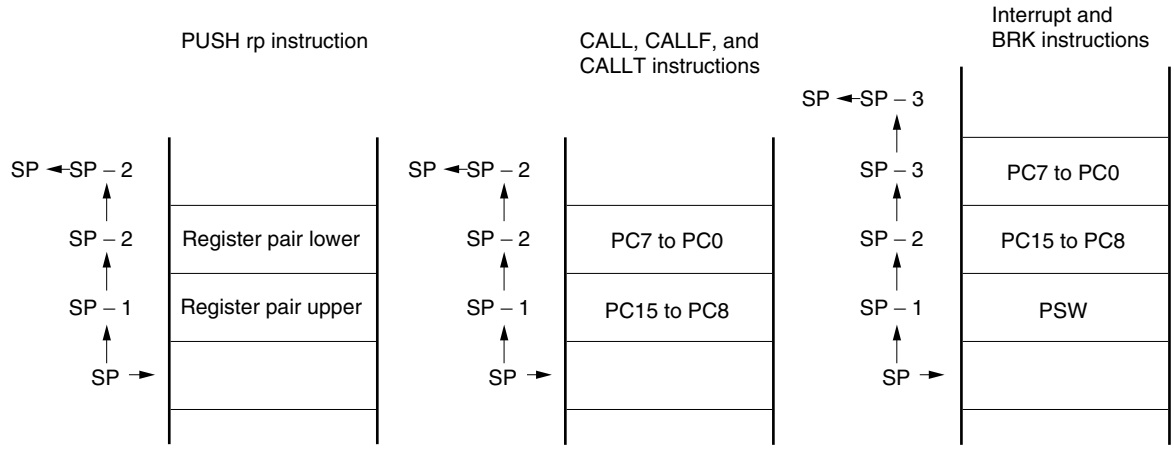
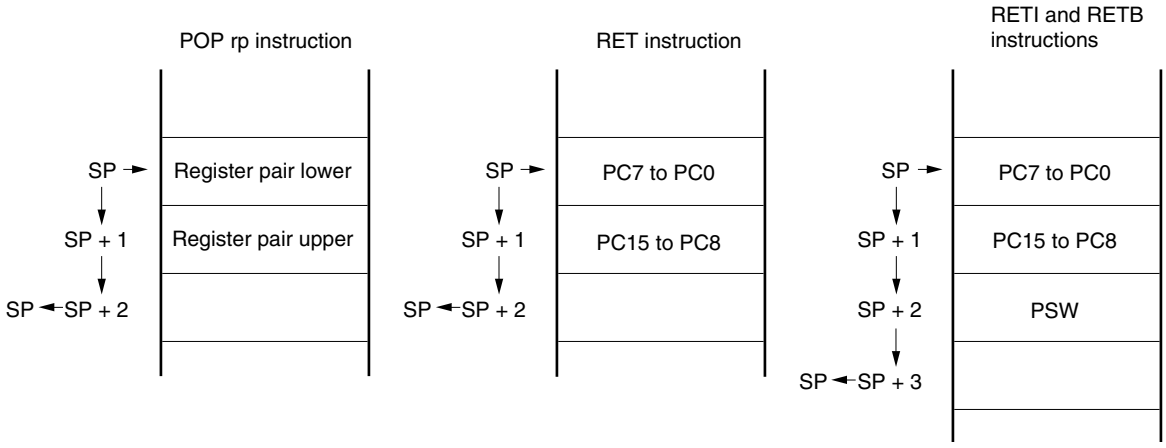


Figure 3-15. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

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

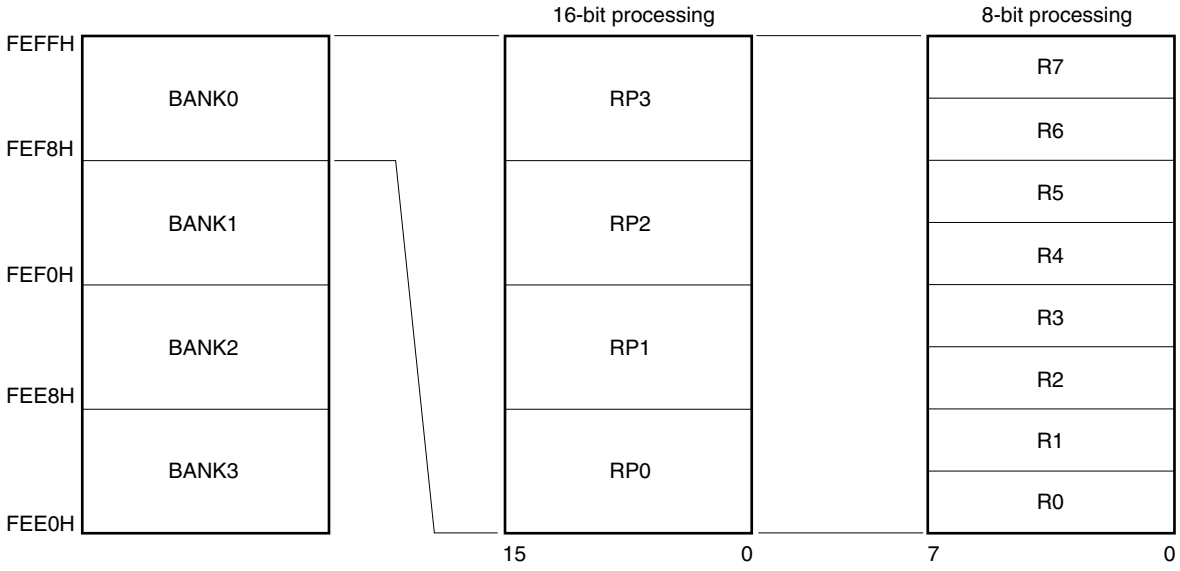
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE, and HL).

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

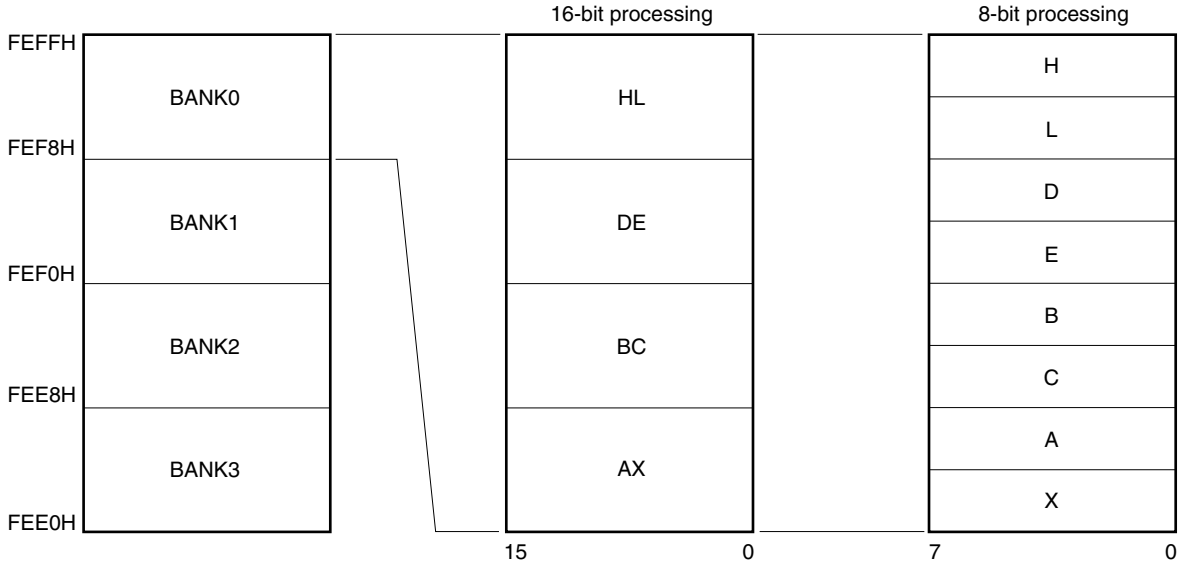
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-16. Configuration of General-Purpose Register

(a) Absolute name



(b) Function name



3.2.3 Special function register (SFR)

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

It is allocated in the FF00H to FFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

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

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

Table 3-5. Special Function Register List (1/2)

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
FF01H	Port 1	P1	R	√	√	—	
FF02H	Port 2	P2	R/W	√	√	—	
FF03H	Port 3	P3		√	√	—	
FF04H	Port 4	P4		√	√	—	
FF05H	Port 5	P5		√	√	—	
FF07H	Port 7	P7		√	√	—	
FF0AH	16-bit timer capture/compare register 00	CR00		—	—	√	Undefined
FF0BH							
FF0CH	16-bit timer capture/compare register 01	CR01		—	—	√	
FF0DH							
FF0EH	16-bit timer counter 0	TM0	R	—	—	√	0000H
FF0FH							
FF10H	8-bit timer compare register 50	CR50	R/W	—	√	—	Undefined
FF11H	8-bit timer compare register 51	CR51		—	√	—	
FF12H	8-bit timer counter 50	TM5 TM50	R	—	√	√	00H
FF13H	8-bit timer counter 51	TM51		—	√		
FF16H	A/D conversion result register 0	ADCR0		—	—	√	
FF17H				—	—		
FF18H	Transmit shift register 0	TXS0	W	—	√	—	FFH
	Receive buffer register 0	RXB0	R	—	√	—	
FF1AH	Serial I/O shift register 30	SIO30	R/W	—	√	—	Undefined
FF1BH	Serial I/O shift register 31	SIO31		—	√	—	
FF20H	Port mode register 0	PM0		√	√	—	FFH
FF22H	Port mode register 2	PM2		√	√	—	
FF23H	Port mode register 3	PM3		√	√	—	
FF24H	Port mode register 4	PM4		√	√	—	
FF25H	Port mode register 5	PM5		√	√	—	
FF27H	Port mode register 7	PM7		√	√	—	
FF30H	Pull-up resistor option register 0	PU0		√	√	—	00H
FF32H	Pull-up resistor option register 2	PU2		√	√	—	
FF33H	Pull-up resistor option register 3	PU3		√	√	—	
FF34H	Pull-up resistor option register 4	PU4		√	√	—	
FF35H	Pull-up resistor option register 5	PU5		√	√	—	
FF37H	Pull-up resistor option register 7	PU7		√	√	—	
FF40H	Clock output select register	CKS		√	√	—	
FF41H	Watch timer operation mode register	WTM		√	√	—	
FF42H	Watchdog timer clock select register	WDCS		—	√	—	

Table 3-5. 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	
FF47H	Memory expansion mode register	MEM		R/W	√	√	—	00H
FF48H	External interrupt rising edge enable register	EGP			√	√	—	
FF49H	External interrupt falling edge enable register	EGN			√	√	—	
FF60H	16-bit timer mode control register 0	TMC0			√	√	—	
FF61H	Prescaler mode register 0	PRM0			—	√	—	
FF62H	Capture/compare control register 0	CRC0			√	√	—	
FF63H	16-bit timer output control register 0	TOC0			√	√	—	
FF70H	8-bit timer mode control register 50	TMC50			√	√	—	
FF71H	Timer clock select register 50	TCL50			—	√	—	
FF78H	8-bit timer mode control register 51	TMC51			√	√	—	
FF79H	Timer clock select register 51	TCL51			—	√	—	
FF80H	A/D converter mode register 0	ADM0			√	√	—	
FF81H	Analog input channel specification register 0	ADS0			—	√	—	
FFA0H	Asynchronous serial interface mode register 0	ASIM0			√	√	—	
FFA1H	Asynchronous serial interface status register 0	ASIS0		R	—	√	—	
FFA2H	Baud rate generator control register 0	BRGC0		R/W	—	√	—	
FFB0H	Serial operation mode register 30	CSIM30			√	√	—	
FFB8H	Serial operation mode register 31	CSIM31			√	√	—	
FFD0H to FFDFH	External access area[Note 1]				√	√	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H		√	√		
FFE2H	Interrupt request flag register 1L	IF1L			√	√	—	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		√	√		
FFE6H	Interrupt mask flag register 1L	MK1L			√	√	—	
FFE8H	Priority level specification flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority level specification flag register 0H		PR0H		√	√		
FFEAH	Priority level specification flag register 1L	PR1L			√	√	—	
FFF0H	Memory size switching register	IMS			—	√	—	CFH[Note 2]
FFF9H	Watchdog timer mode register	WDTM			√	√	—	00H
FFFAH	Oscillation stabilization time select register	OSTS			—	√	—	04H
FFFBH	Processor clock control register	PCC			√	√	—	

Notes 1. The external access area cannot be accessed by SFR addressing. Access it with the direct addressing method.

2. The default is CFH, but set the value corresponding to each respective product as indicated below.
μPD780021AS, 780031AS: 42H
μPD780022AS, 780032AS: 44H
μPD780023AS, 780033AS: C6H
μPD780024AS, 780034AS: C8H
μPD78F0034BS: Value for mask ROM version

3.3 Instruction Address Addressing

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

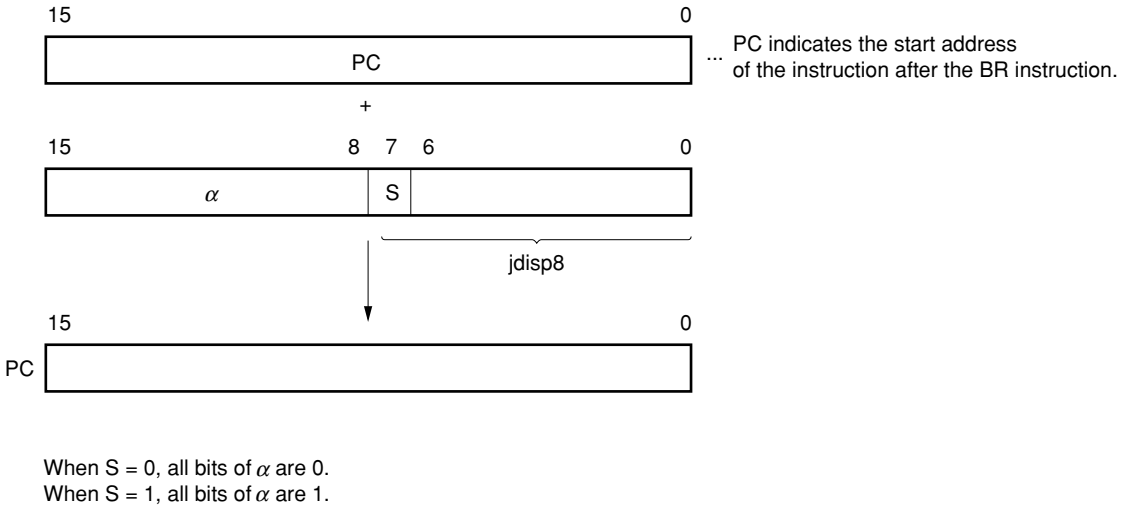
3.3.1 Relative addressing

[Function]

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

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

[Illustration]



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

3.3.2 Immediate addressing

[Function]

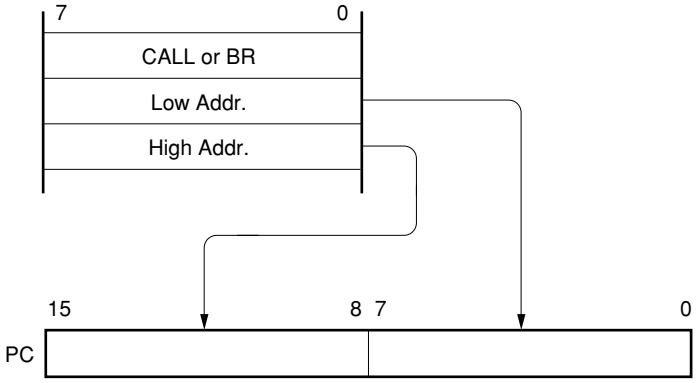
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

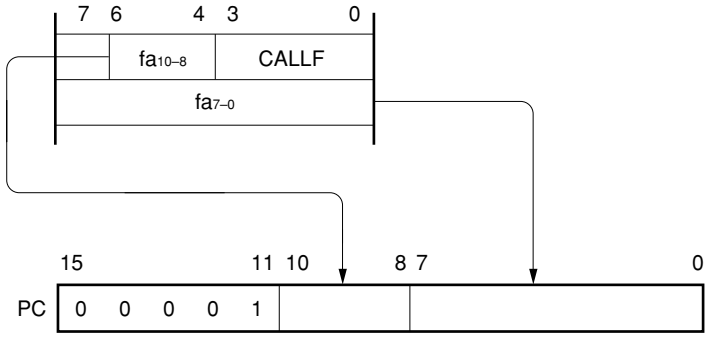
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

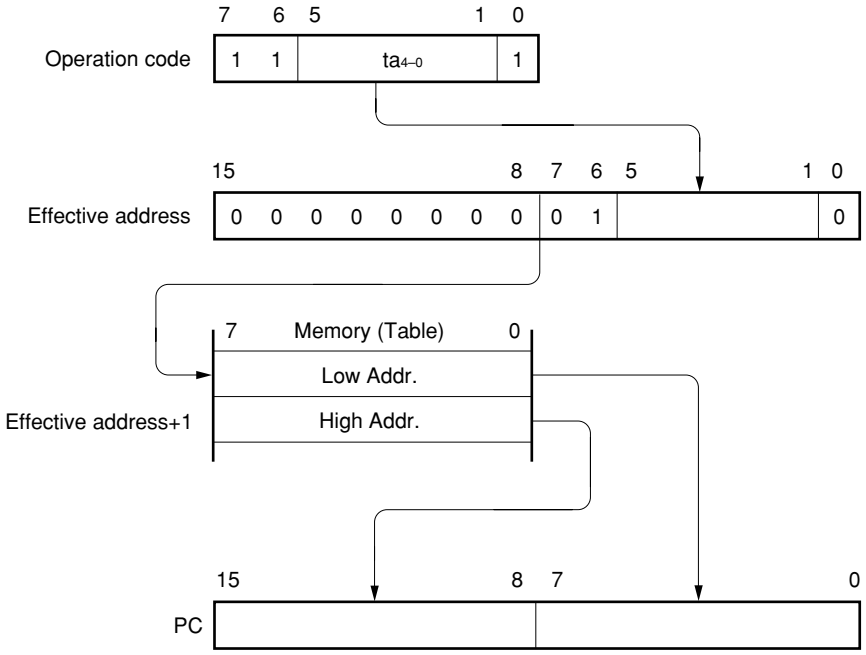
[Function]

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

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

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

[Illustration]



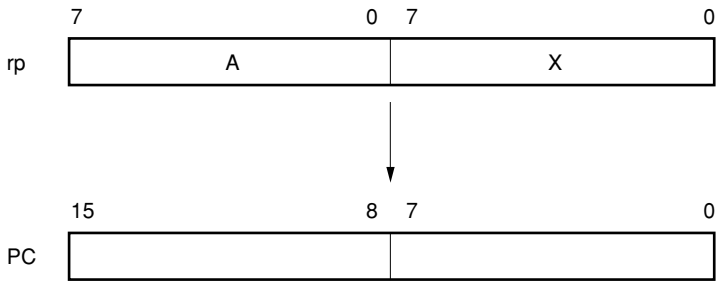
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the μPD780024AS, 780034AS Subseries instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register specify code (Rn and RPn) of an instruction word in the registered bank specified with the register bank select flag (RBS0 and RBS1). Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

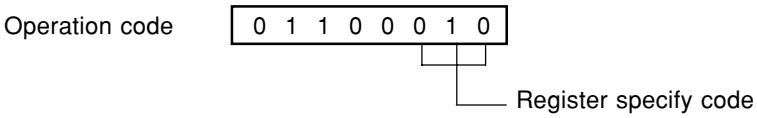
[Operand format]

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

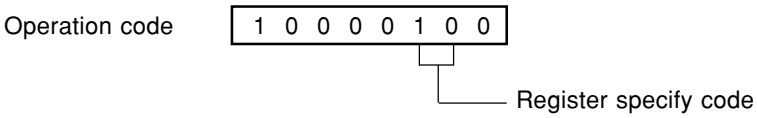
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

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

[Operand format]

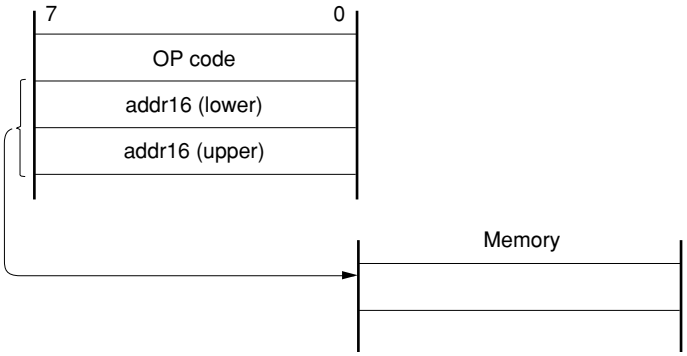
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word. This addressing is applied to the 256-byte space FE20H to FF1FH. An internal RAM and a special function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

If the SFR area (FF00H to FF1FH) where short direct addressing is applied, ports which are frequently accessed in a program and a compare register of the timer/event counter and a capture register of the timer/event counter are mapped 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. Refer to the **[Illustration]** on the next page.

[Operand format]

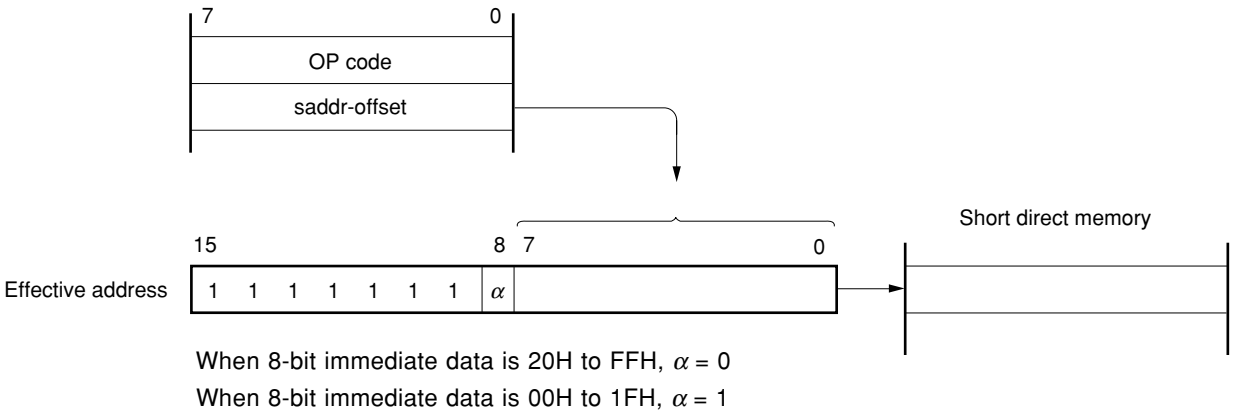
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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

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

[Illustration]



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

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

3.4.5 Special function register (SFR) addressing

[Function]

The 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 SFR mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

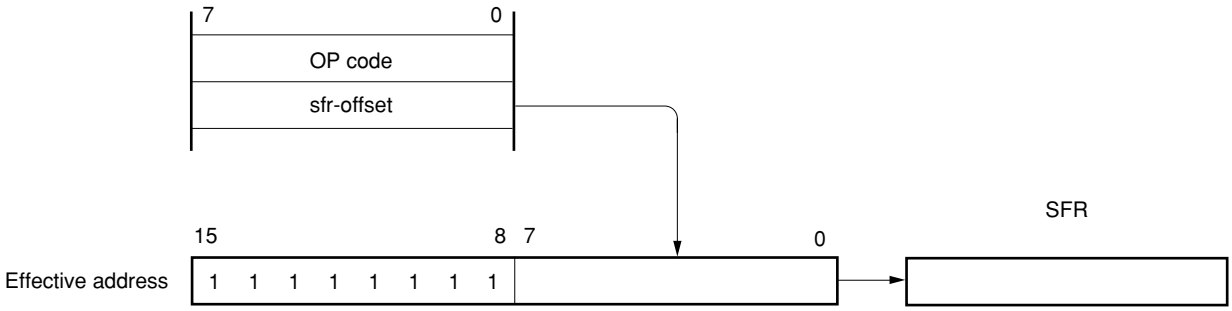
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

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

[Illustration]



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

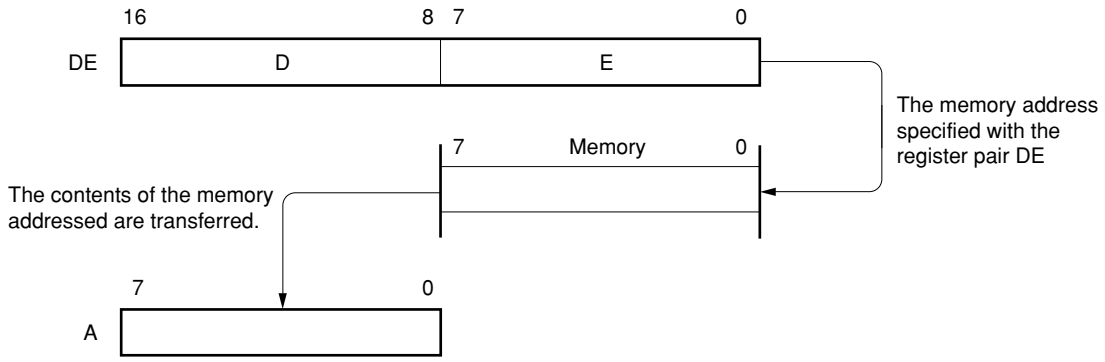
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

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

[Operand format]

Identifier	Description
—	[HL + byte]

[Description example]

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

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

3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

[Description example]

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

Operation code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

Stack addressing enables to address the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD780024AS, 780034AS Subseries products incorporate four input ports and 35 I/O ports. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware I/O pins.

Figure 4-1. Port Types

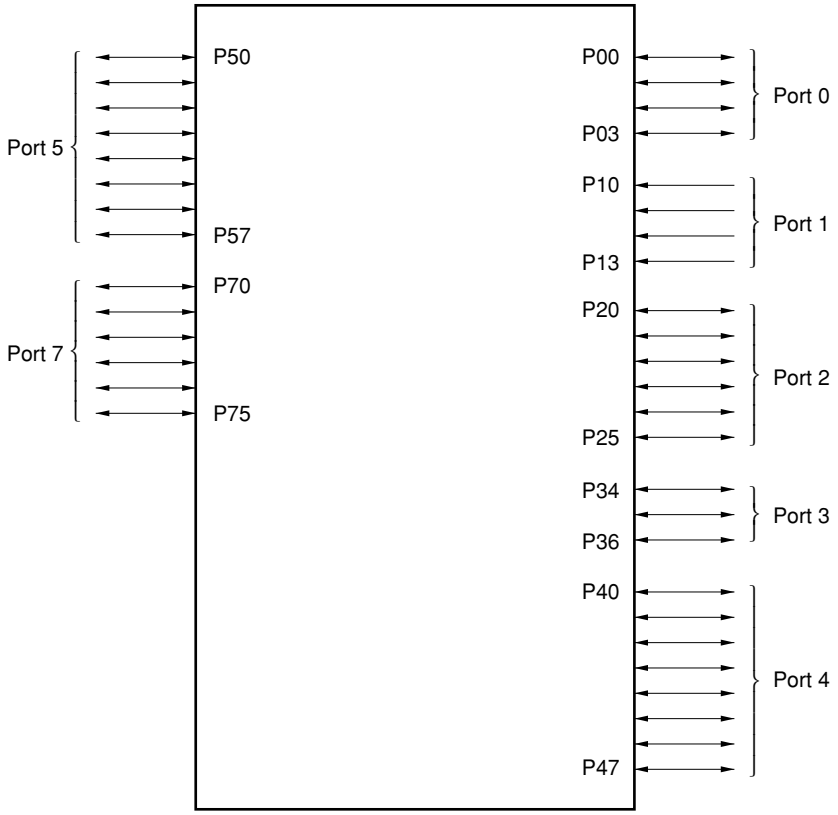


Table 4-1. Port Functions

Pin Name	Function	Alternate Function
P00	Port 0 4-bit I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	INTP0
P01		INTP1
P02		INTP2
P03		INTP3/ADTRG
P10 to P13	Port 1 4-bit input-only port.	ANI0 to ANI3
P20	Port 2 6-bit I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	SI30
P21		SO30
P22		$\overline{\text{SCK30}}$
P23		RxD0
P24		TxD0
P25		ASCK0
P34	Port 3 3-bit I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by software settings.	SI31
P35		SO31
P36		$\overline{\text{SCK31}}$
P40 to P47	Port 4 8-bit I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by software settings. Interrupt request flag (KRIF) is set to 1 by falling edge detection.	—
P50 to P57	Port 5 8-bit I/O port. LEDs can be driven directly. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	—
P70	Port 7 6-bit I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by software settings.	TI00/TO0
P71		TI01
P72		TI50/TO50
P73		TI51/TO51
P74		PCL
P75		BUZ

4.2 Configuration of Ports

A port consists of the following hardware.

Table 4-2. Configuration of Ports

Item	Configuration
Control register	Port mode register (PMm: m = 0, 2 to 5, 7) Pull-up resistor option register (PUm,: m = 0, 2 to 5, 7)
Port	Total: 39 ports (4 inputs, 35 inputs/outputs)
Pull-up resistor	Total: 39 (software control)

4.2.1 Port 0

Port 0 is a 4-bit I/O port with output latch. P00 to P03 pins can specify the input mode/output mode in 1-bit units with the port mode register 0 (PM0). An on-chip pull-up resistor of P00 to P03 pins can be used for them in 1-bit units with a pull-up resistor option register 0 (PU0).

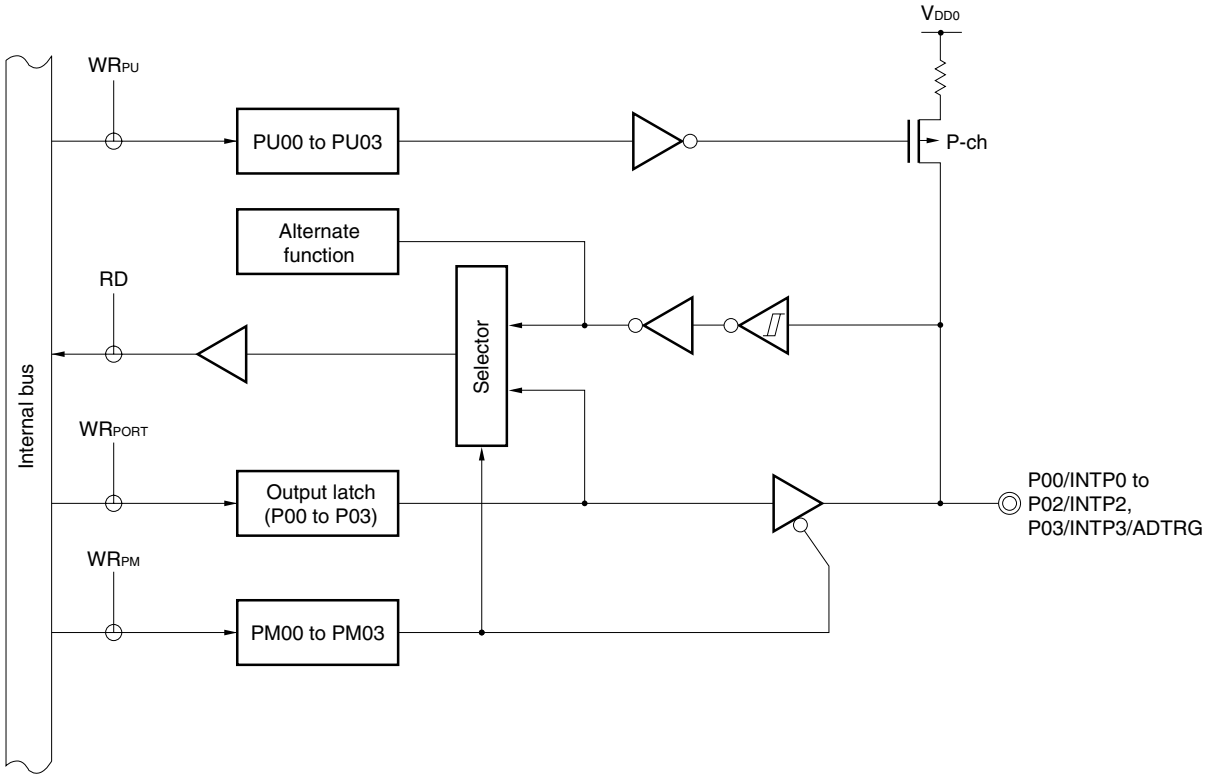
This port can also be used as an external interrupt request input, and A/D converter external trigger input.

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

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

Caution Because port 0 also serves for external interrupt request input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. Thus, when the output mode is used, set the interrupt mask flag to 1.

Figure 4-2. Block Diagram of P00 to P03



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

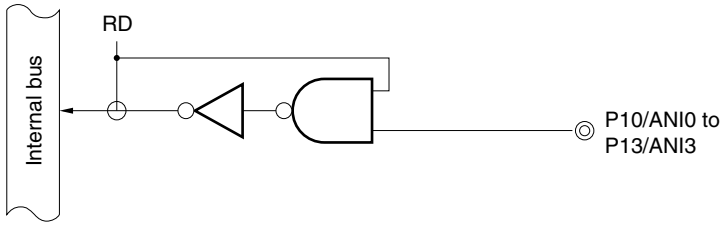
4.2.2 Port 1

Port 1 is an 4-bit input-only port.

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

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

Figure 4-3. Block Diagram of P10 to P13



RD: Port 1 read signal

Caution When port 1 is used as a input port, the input value can be read using an 8-bit memory manipulation instruction. However, in this case, do not use the higher 4 bits (P17 to P14) because they are undefined. Also, do not read the higher 4 bits using a 1-bit memory manipulation instruction.

4.2.3 Port 2

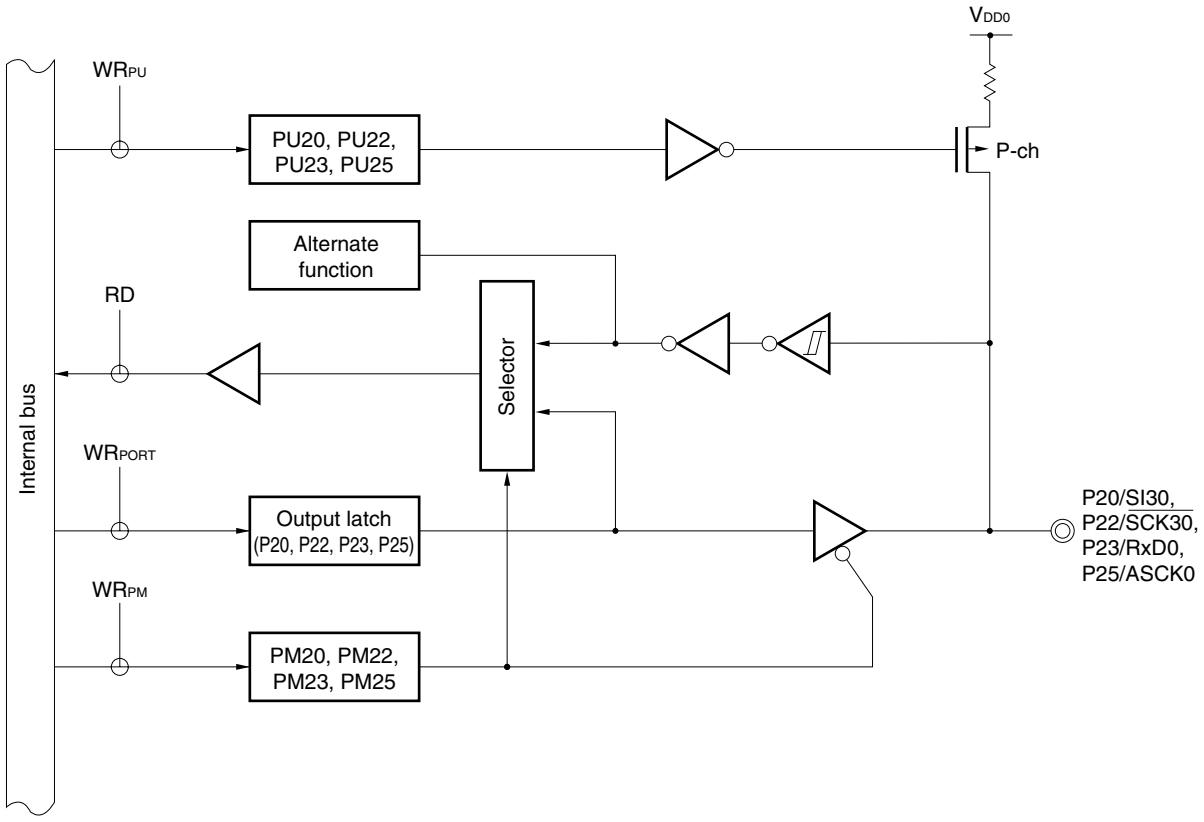
Port 2 is a 6-bit I/O port with output latch. P20 to P25 pins can specify the input mode/output mode in 1-bit units with the port mode register 2 (PM2). An on-chip pull-up resistor of P20 to P25 pins can be used for them in 1-bit units with a pull-up resistor option register 2 (PU2).

This port has also alternate functions as serial interface data I/O and clock I/O.

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

Figures 4-4 and 4-5 show block diagrams of port 2.

Figure 4-4. Block Diagram of P20, P22, P23, and P25



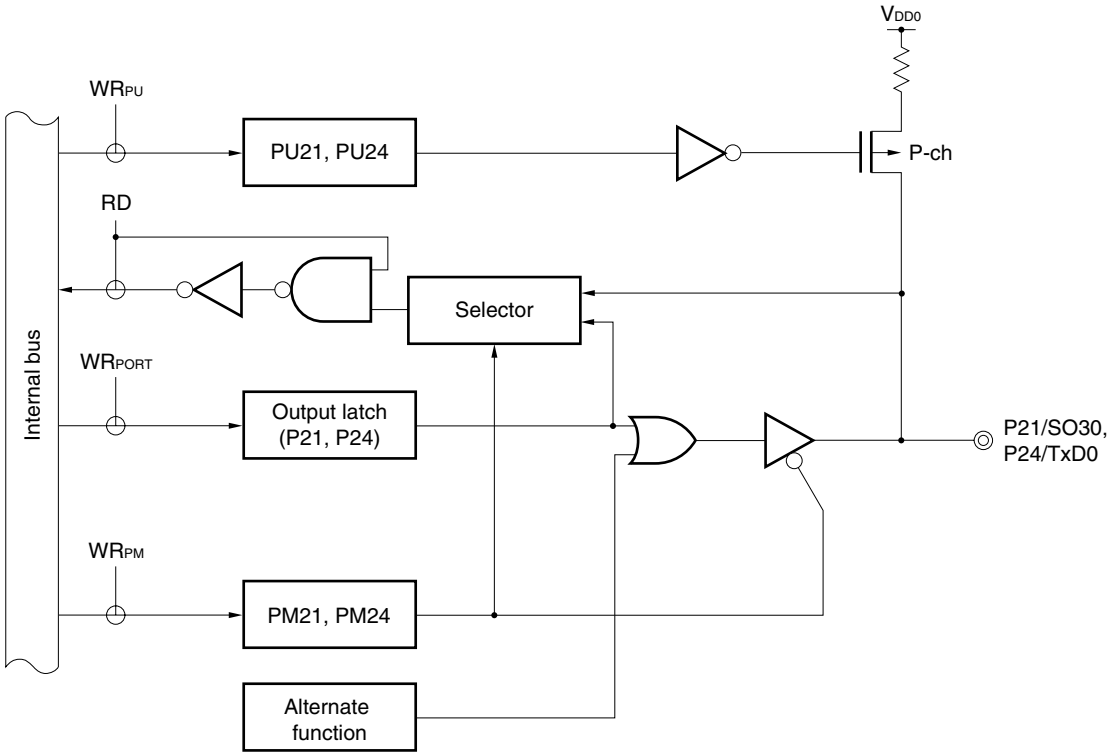
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Figure 4-5. Block Diagram of P21 and P24



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

4.2.4 Port 3

Port 3 is a 3-bit I/O port with output latch. P34 to P36 pins can specify the input mode/output mode in 1-bit units with port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified for the P34 to P36 pins in 1-bit units by pull-up resistor option register 3 (PU3).

Port 3 can also be used for serial interface data I/O and clock I/O.

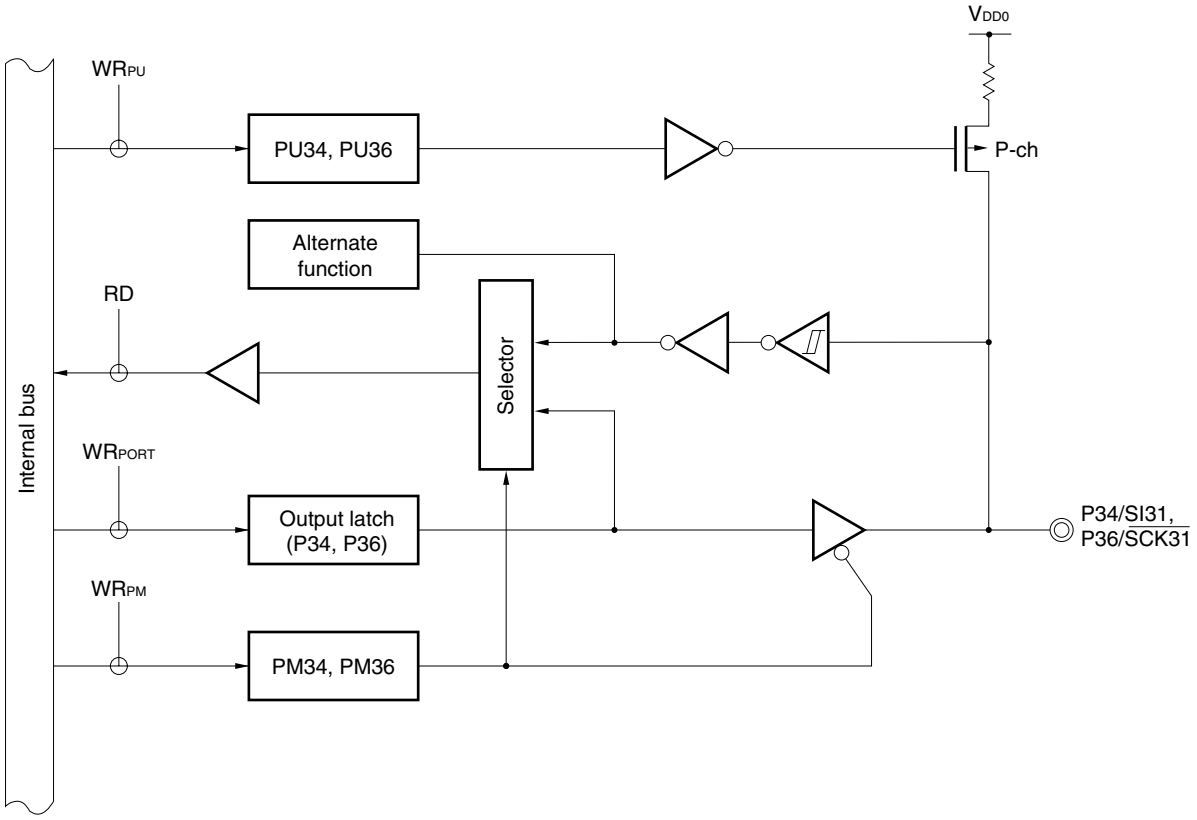
$\overline{\text{RESET}}$ input sets port 3 to input mode.

Figures 4-6 and 4-7 show block diagrams of port 3.

Cautions 1. When reading port 3 using an 8-bit memory manipulation instruction, do not use the lower 4 bits (P33 to P30) because they are undefined. When writing port 3 using an 8-bit memory manipulation instruction, any values can be written to the lower 4 bits. Execution of a 1-bit memory manipulation instruction for the lower 4 bits is prohibited.

2. Be sure to set the lower 4 bits (PM33 to PM30) of port mode register 3 (PM3) to 1.

Figure 4-6. Block Diagram of P34 and P36



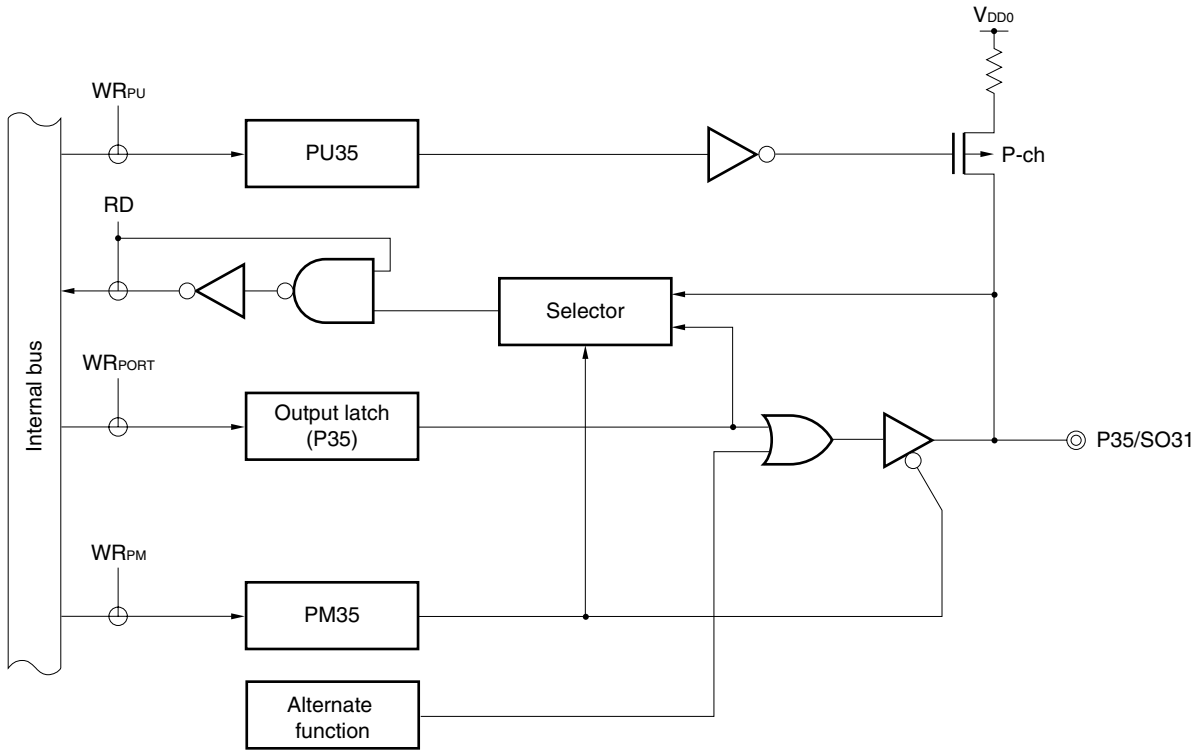
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 3 read signal

WR: Port 3 write signal

Figure 4-7. Block Diagram of P35



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

4.2.5 Port 4

Port 4 is an 8-bit I/O port with output latch. The P40 to P47 pins can specify the input mode/output mode in 1-bit units with port mode register 4 (PM4). An on-chip pull-up resistor of P40 to P47 pins can be used for them in 1-bit units with pull-up resistor option register 4 (PU4).

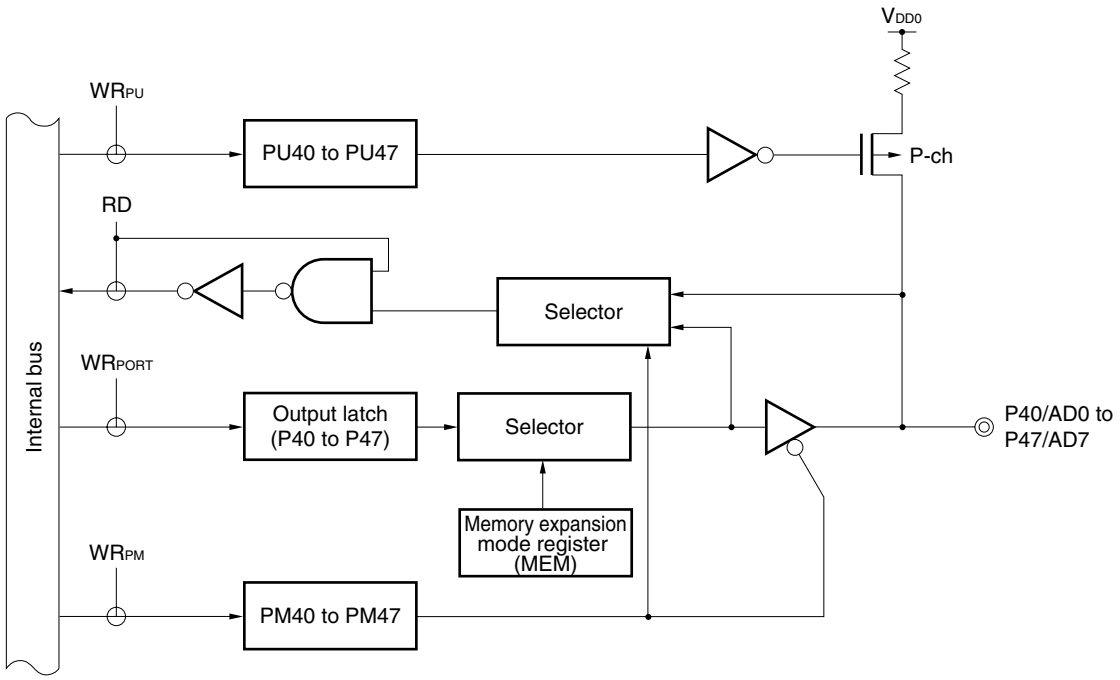
The interrupt request flag (KRIF) can be set to 1 by detecting falling edges.

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

Figures 4-8 and 4-9 show a block diagram of port 4 and block diagram of the falling edge detector, respectively.

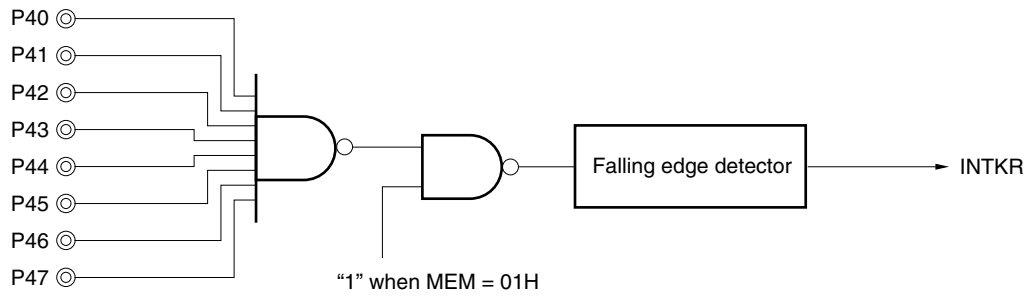
Caution When using the falling edge detection interrupt (INTKR), be sure to set the memory expansion mode register (MEM) to 01H.

Figure 4-8. Block Diagram of P40 to P47



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

Figure 4-9. Block Diagram of Falling Edge Detector



4.2.6 Port 5

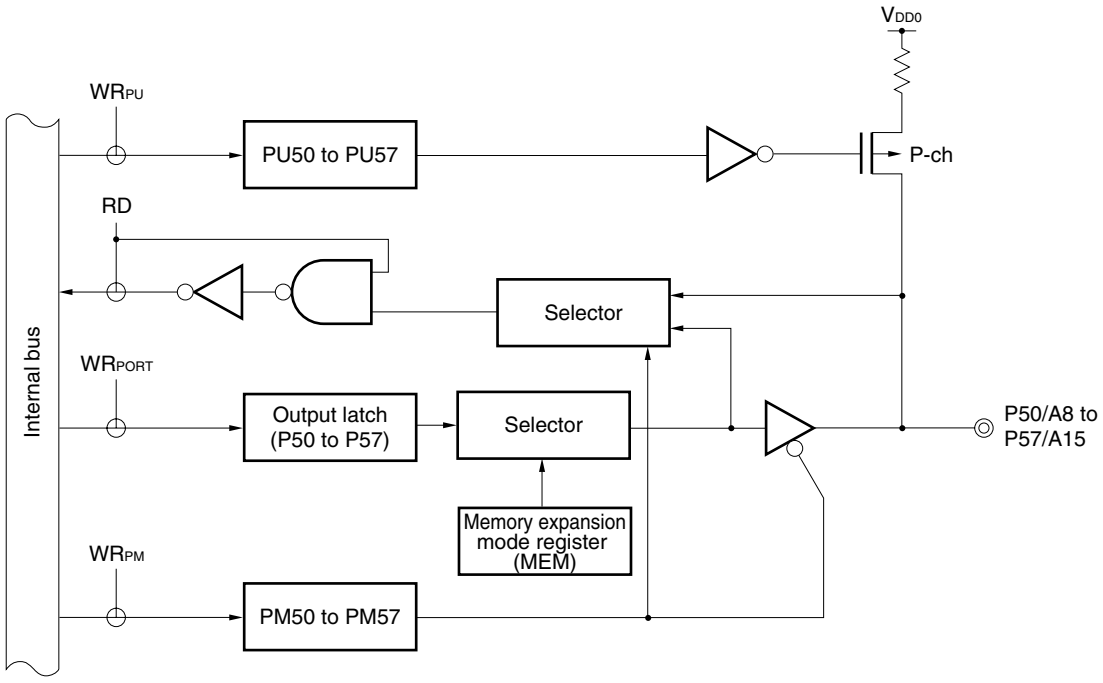
Port 5 is an 8-bit I/O port with output latch. The P50 to P57 pins can specify the input mode/output mode in 1-bit units with port mode register 5 (PM5). An on-chip pull-up resistor of P50 to P57 pins can be used for them in 1-bit units with pull-up resistor option register 5 (PU5).

Port 5 can drive LEDs directly.

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

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

Figure 4-10. Block Diagram of P50 to P57



PU: Pull-up resistor option register

PM: Port mode register

RD: Port 5 read signal

WR: Port 5 write signal

4.2.7 Port 7

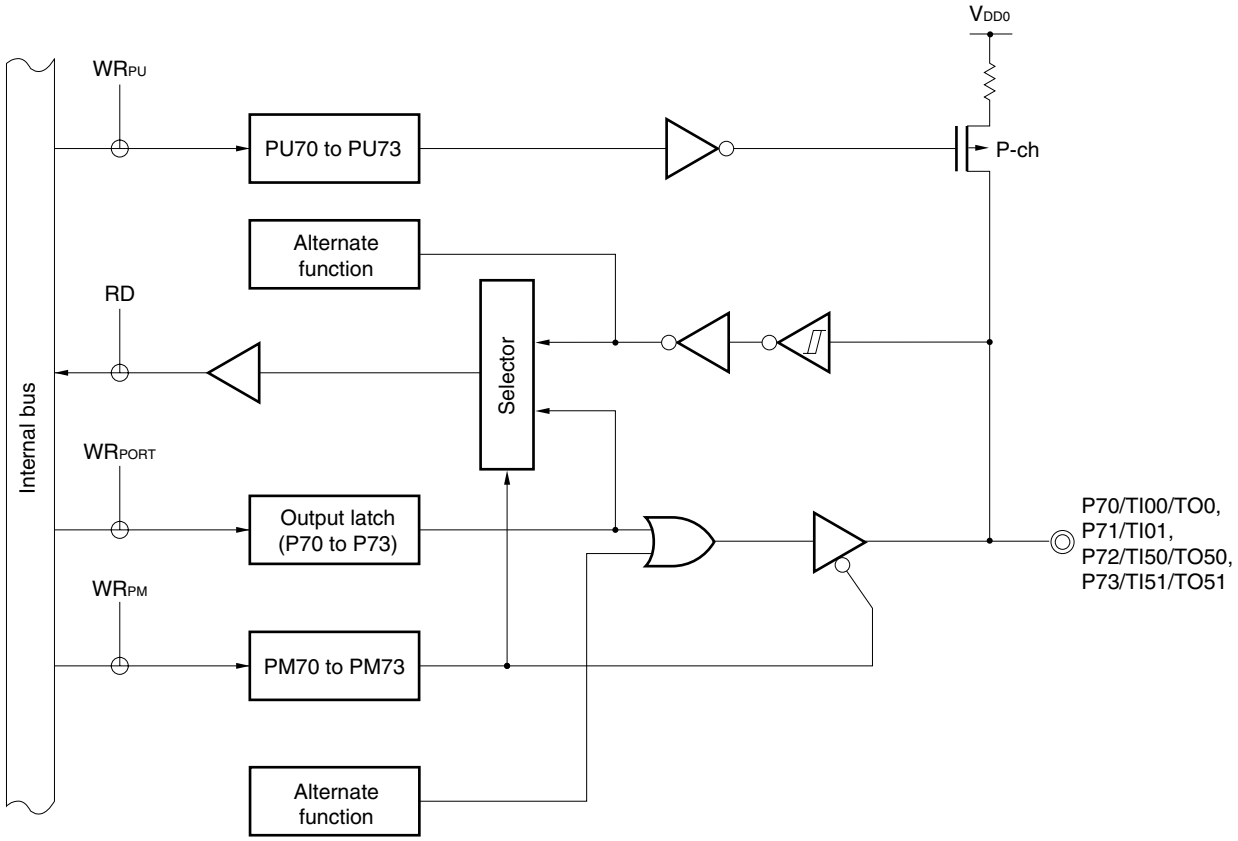
Port 7 is a 6-bit I/O port with output latch. The P70 to P75 pins can specify the input mode/output mode in 1-bit units with port mode register 7 (PM7). An on-chip pull-up resistor of P70 to P75 pins can be used for them in 1-bit units with pull-up resistor option register 7 (PU7).

This port can also be used as a timer I/O, clock output, and buzzer output.

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

Figures 4-11 and 4-12 show block diagrams of port 7.

Figure 4-11. Block Diagram of P70 to P73



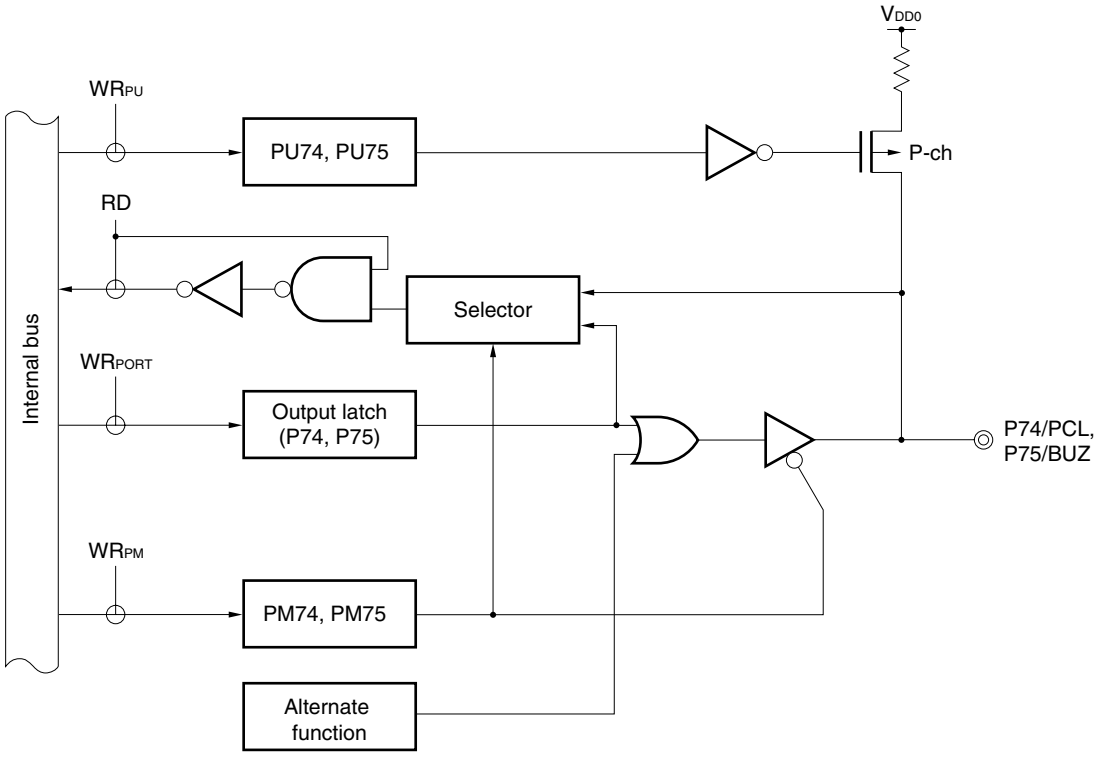
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 7 read signal

WR: Port 7 write signal

Figure 4-12. Block Diagram of P74 and P75



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

4.3 Registers to Control Port Function

The following two types of registers control the ports.

- Port mode registers (PM0, PM2 to PM5, PM7)
- Pull-up resistor option registers (PU0, PU2 to PU5, PU7)

(1) Port mode registers (PM0, PM2 to PM5, PM7)

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

PM0, PM2 to PM5, and PM7 are independently set by a 1-bit or 8-bit memory manipulation instruction.

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

Cautions 1. Pins P10 to P17 are input-only pins.

2. If a port has an alternate function pin and it is used as an alternate output function, set the output latches (P0 and P2 to P7) to 0.

Figure 4-13. Format of Port Mode Register (PM0, PM2 to PM5, PM7)

Address: FF20H After reset: FFH R/W

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

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	PM36	PM35	PM34	1 Note	1 Note	1 Note	1 Note

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

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

Note Since the lower 4 bits (PM33 to PM30) of PM3 are not fixed to 1, be sure to set the lower 4 bits to 1 when setting by an 8-bit memory manipulation instruction.

(2) Pull-up resistor option registers (PU0, PU2 to PU5, PU7)

These registers are used to set whether to use an on-chip pull-up resistor at each port or not. By setting PU0, PU2 to PU5, and PU7, the on-chip pull-up resistors of the port pins corresponding to the bits in PU0, PU2 to PU5, and PU7 can be used.

PU0, PU2 to PU5, and PU7 are independently set by a 1-bit or 8-bit memory manipulation instruction.

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

Cautions 1. The P10 to P13 pins do not incorporate a pull-up resistor.

2. When PUm is set to 1, the on-chip pull-up resistor is connected irrespective of the input/output mode. When using in output mode, therefore, set the bit of PUm to 0 (m = 0, 2 to 5, 7).

Figure 4-14. Format of Pull-Up Resistor Option Register (PU0, PU2 to PU5, PU7)

Address: FF30H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU0	0	0	0	0	PU03	PU02	PU01	PU00

Address: FF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU2	0	0	PU25	PU24	PU23	PU22	PU21	PU20

Address: FF33H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU3	0	PU36	PU35	PU34	0	0	0	0

Address: FF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40

Address: FF35H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50

Address: FF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 2 to 5, 7; n = 0 to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

4.4 Operations of Port Function

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

4.4.1 Writing to I/O port

(1) Output mode

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

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

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is OFF, the pin status does not change.

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

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

4.4.2 Reading from I/O port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.

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

(2) Input mode

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

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

CHAPTER 5 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 available.

(1) Main system clock oscillator

This circuit oscillates at frequencies of 1 to 8.38 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).

(2) Subsystem clock oscillator

The circuit oscillates at a frequency of 32.768 kHz. Oscillation cannot be stopped. If the subsystem clock oscillator is not used, the internal feedback resistor can be disabled by the processor clock control register (PCC). This enables to reduce the power consumption in the STOP mode.

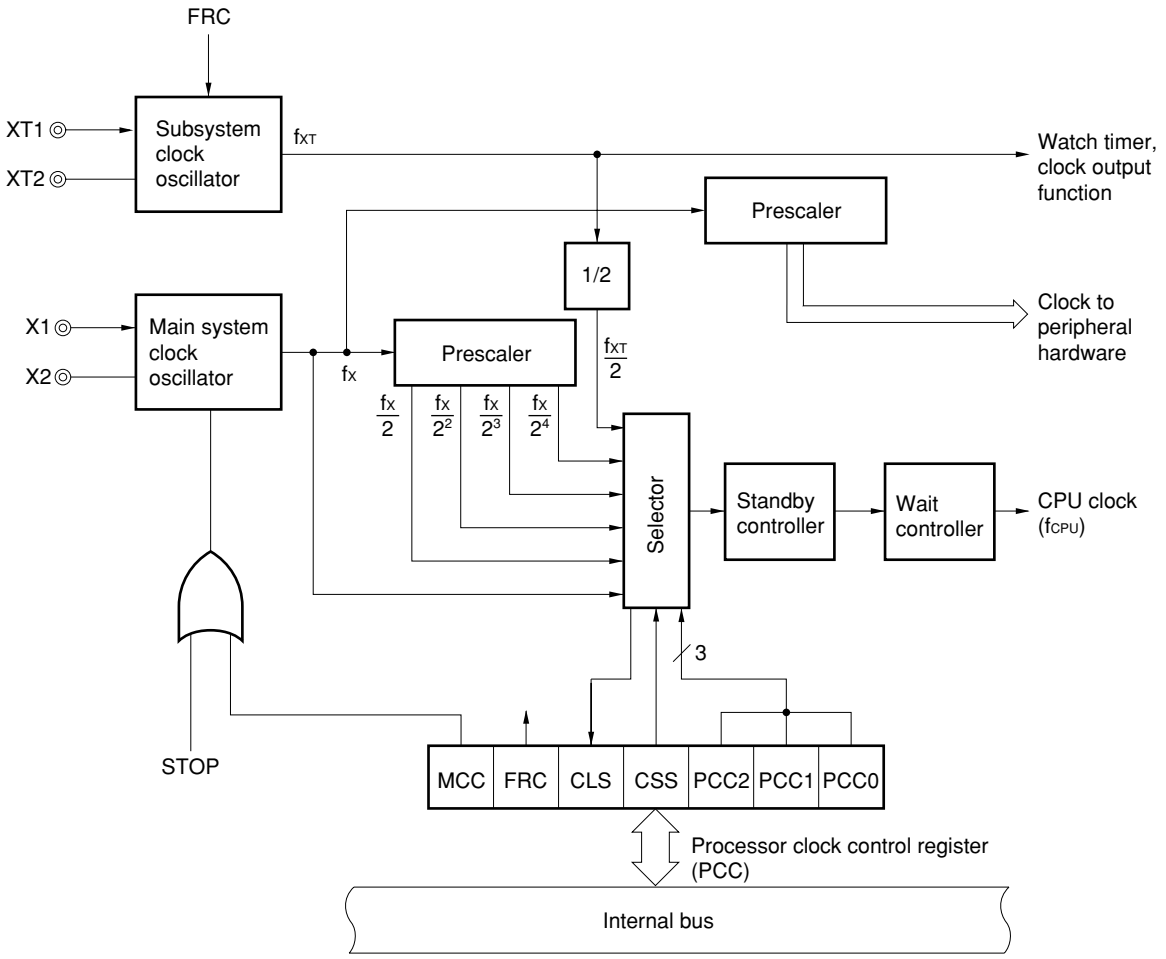
5.2 Configuration of Clock Generator

The clock generator consists of the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillators	Main system clock oscillator Subsystem clock oscillator

Figure 5-1. Block Diagram of Clock Generator



5.3 Registers to Control Clock Generator

The clock generator is controlled by the processor clock control register (PCC).

The PCC sets whether to use CPU clock selection, the ratio of division, main system clock oscillator operation/stop and subsystem clock oscillator internal feedback resistor.

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

$\overline{\text{RESET}}$ input sets the PCC to 04H.

Figure 5-2. Subsystem Clock Feedback Resistor

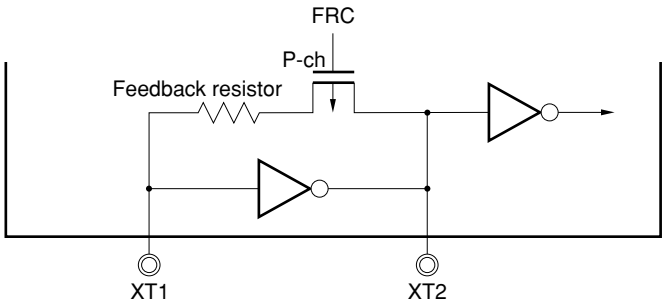


Figure 5-3. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 04H R/W[Note 1]

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

MCC	Main system clock oscillation control[Note 2]
0	Oscillation possible
1	Oscillation stopped

FRC	Subsystem clock feedback resistor selection[Note 3]
0	Internal feedback resistor used
1	Internal feedback resistor not used

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

CSS	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
0	0	0	0	f_X
	0	0	1	$f_X/2$
	0	1	0	$f_X/2^2$
	0	1	1	$f_X/2^3$
	1	0	0	$f_X/2^4$
1	0	0	0	$f_{XT}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

Notes 1. Bit 5 is Read Only.

2. When the CPU is operating on the subsystem clock, MCC should be used to stop the main system clock oscillation. A STOP instruction should not be used.

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

Cautions 1. Be sure to set bit 3 to 0.

2. When the external clock is input, MCC should not be set. This is because the X2 pin is connected to V_{DD1} via a pull-up resistor.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

The fastest instructions of μPD780024AS, 780034AS Subseries are carried out in two CPU clocks. The relationship of CPU clock (f_{CPU}) and minimum instruction execution time is shown in Table 5-2.

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

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$
f_X	0.24 μs
$f_X/2$	0.48 μs
$f_X/2^2$	0.95 μs
$f_X/2^3$	1.91 μs
$f_X/2^4$	3.81 μs
$f_{XT}/2$	122 μs

f_X = 8.38 MHz, f_{XT} = 32.768 kHz

f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

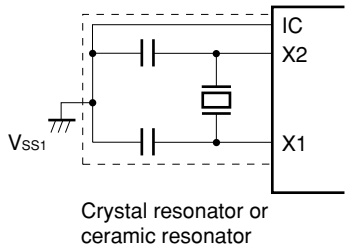
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (8.38 MHz TYP.) connected to the X1 and X2 pins.

External clocks can be input to the main system clock oscillator. In this case, input a clock signal to the X1 pin and an inverted-phase clock signal to the X2 pin.

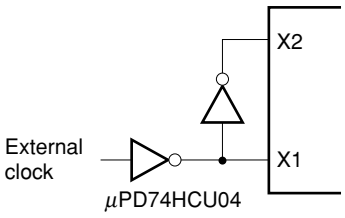
Figure 5-4 shows an external circuit of the main system clock oscillator.

Figure 5-4. External Circuit of Main System Clock Oscillator

(a) Crystal and ceramic oscillation



(b) External clock



Caution Do not execute the STOP instruction and do not set MCC (bit 7 of processor clock control register (PCC)) to 1 if an external clock is input. This is because when the STOP instruction or MCC is set to 1, the main system clock operation stops and the X2 pin is connected to V_{DD1} via a pull-up resistor.

5.4.2 Subsystem clock oscillator

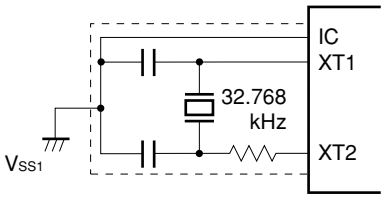
The subsystem clock oscillator oscillates with a crystal resonator (32.768 kHz TYP.) connected to the XT1 and XT2 pins.

External clocks can be input to the subsystem clock oscillator. In this case, input a clock signal to the XT1 pin and an inverted-phase clock signal to the XT2 pin.

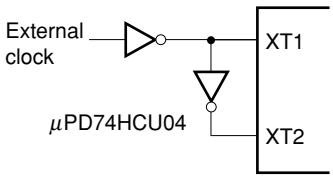
Figure 5-5 shows an external circuit of the subsystem clock oscillator.

Figure 5-5. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



Cautions are listed on the next page.

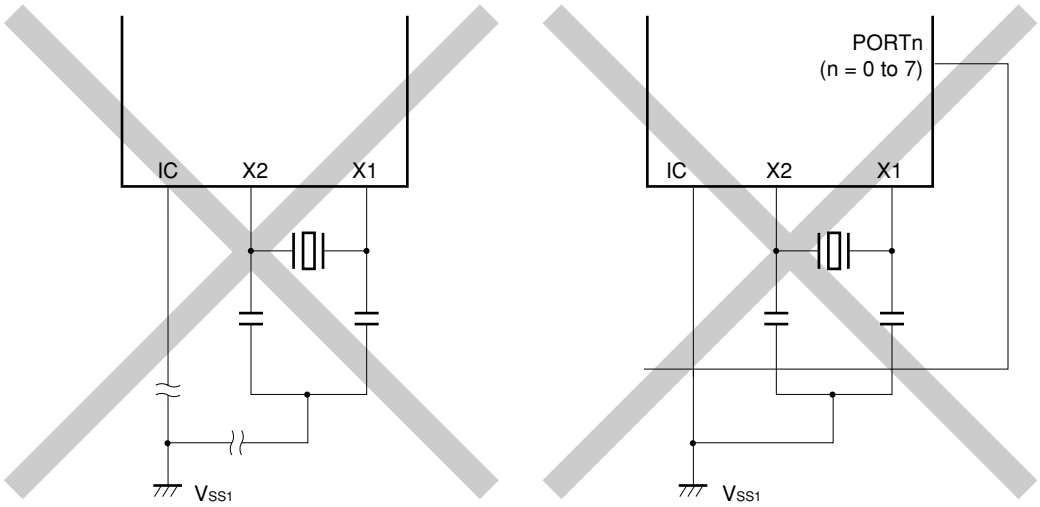
Cautions 1. When using the main system clock oscillator and a subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-4 and 5-5 to avoid an adverse effect from wiring capacitance.

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

Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.
Figure 5-6 shows examples of incorrect resonator connection.

Figure 5-6. Examples of Incorrect Resonator Connection (1/2)

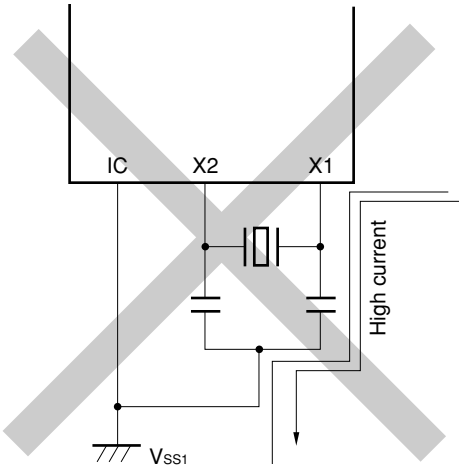
(a) Too long wiring **(b) Crossed signal line**



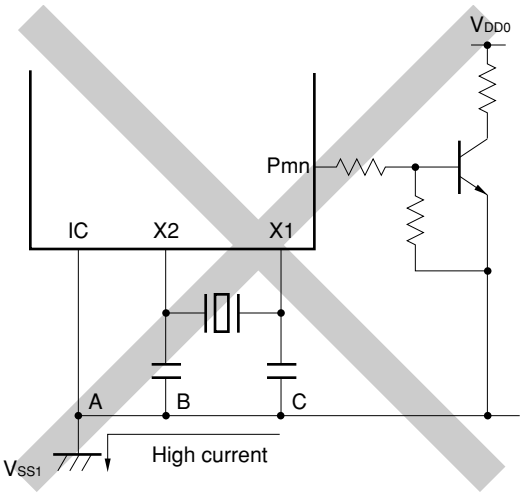
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Figure 5-6. Examples of Incorrect Resonator Connection (2/2)

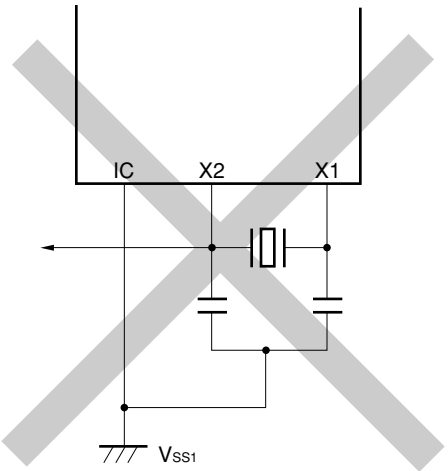
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

Cautions 2. When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.
To prevent that from occurring, it is recommended to wire X2 and XT1 so that they are not in parallel, and to connect the IC pin between X2 and XT1 directly to V_{SS1}.

5.4.3 Divider

The divider divides the main system clock oscillator output (fx) and generates various clocks.

5.4.4 When no subsystem clocks are used

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

XT1: Connect to V_{DD0}
XT2: Open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To minimize leakage current, the above internal feedback resistor can be removed with bit 6 (FRC) of the processor clock control register (PCC). In this case also, connect the XT1 and XT2 pins as described above.

5.5 Clock Generator Operations

The clock generator generates the following various types of clocks and controls the CPU operating mode including the standby mode.

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

The following clock generator functions and operations are determined with the processor clock control register (PCC).

(a) Upon generation of $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (3.81 μs @ 8.38 MHz operation) is selected (PCC = 04H). Main system clock oscillation stops while low level is applied to $\overline{\text{RESET}}$ pin.

(b) With the main system clock selected, one of the five CPU clock types (0.24 μs, 0.48 μs, 0.95 μs, 1.91 μs, 3.81 μs, @ 8.38 MHz operation) can be selected by setting the PCC.

(c) With the main system clock selected, two standby modes, the STOP and HALT modes, are available. To reduce current consumption in the STOP mode, the subsystem clock feedback resistor can be disconnected to stop the subsystem clock.

(d) The PCC can be used to select the subsystem clock and to operate the system with low-current consumption (122 μs @ 32.768 kHz operation).

(e) With the subsystem clock selected, main system clock oscillation can be stopped with the PCC. The HALT mode can be used. However, the STOP mode cannot be used (subsystem clock oscillation cannot be stopped).

(f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to the watch timer and clock output functions only. Thus the watch function and the clock output function can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped (except external input clock operation).

5.5.1 Main system clock operations

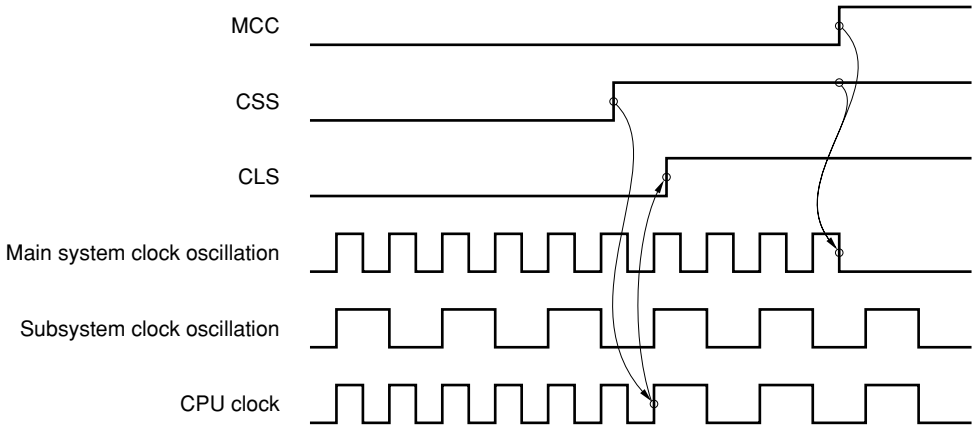
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

(a) Because the operation guarantee instruction execution speed depends on the power supply voltage, the minimum instruction execution time can be changed by bits 0 to 2 (PCC0 to PCC2) of the PCC.

(b) If bit 7 (MCC) of the PCC is set to 1 when operated with the main system clock, the main system clock oscillation does not stop. When bit 4 (CSS) of the PCC is set to 1 and the operation is switched to subsystem clock operation (CLS = 1) after that, the main system clock oscillation stops (see **Figure 5-7**).

Figure 5-7. Main System Clock Stop Function (1/2)

(a) Operation when MCC is set after setting CSS with main system clock operation



(b) Operation when MCC is set in case of main system clock operation

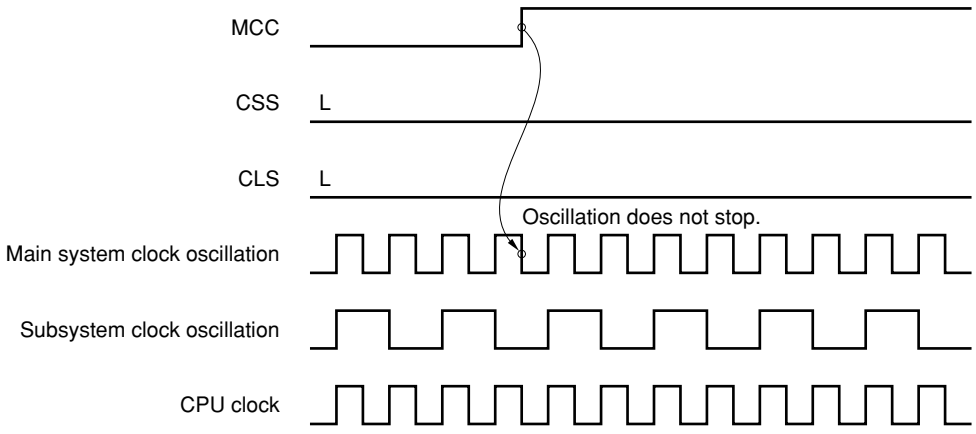
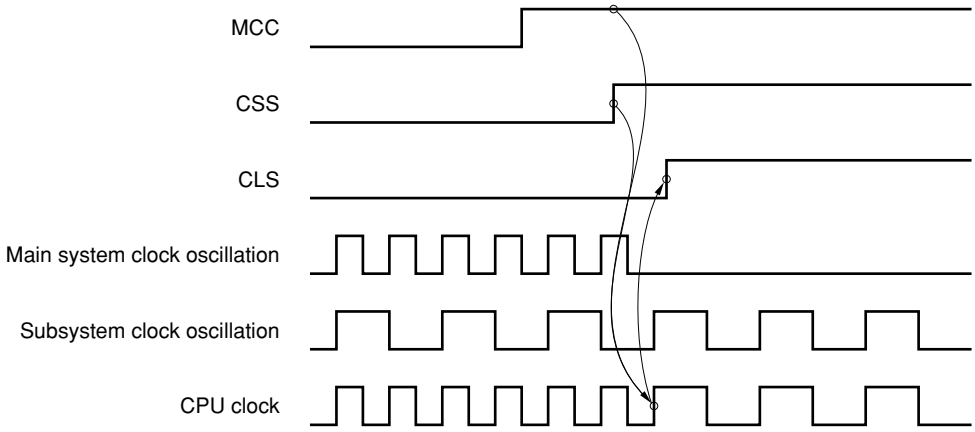


Figure 5-7. Main System Clock Stop Function (2/2)

(c) Operation when CSS is set after setting MCC with main system clock operation



5.5.2 Subsystem clock operations

When operated with the subsystem clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 1), the following operations are carried out.

(a) The minimum instruction execution time remains constant (122 μs @ 32.768 kHz operation) irrespective of bits 0 to 2 (PCC0 to PCC2) of the PCC.

(b) Watchdog timer counting stops.

Caution Do not execute the STOP instruction while the subsystem clock is in operation.

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

The actual switchover operation is not performed directly after writing to the PCC, but operation continues on the pre-switchover clock for several instructions (see **Table 5-3**).

Determination as to whether the system is operating on the main system clock or the subsystem clock is performed by bit 5 (CLS) of the PCC register.

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

Set Value Before Switchover				Set Value After Switchover																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
0	0	0	0					16 instructions				16 instructions				16 instructions				16 instructions				$f_X/2f_{XT}$ instruction (128 instructions)			
	0	0	1	8 instructions								8 instructions				8 instructions				8 instructions				$f_X/4f_{XT}$ instruction (64 instructions)			
	0	1	0	4 instructions				4 instructions								4 instructions				4 instructions				$f_X/8f_{XT}$ instruction (32 instructions)			
	0	1	1	2 instructions				2 instructions				2 instructions								2 instructions				$f_X/16f_{XT}$ instruction (16 instructions)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								$f_X/32f_{XT}$ instruction (8 instructions)			
1	×	×	×	1 instruction				1 instruction				1 instruction				1 instruction				1 instruction							

Remarks 1. One instruction is the minimum instruction execution time with the pre-switchover CPU clock.

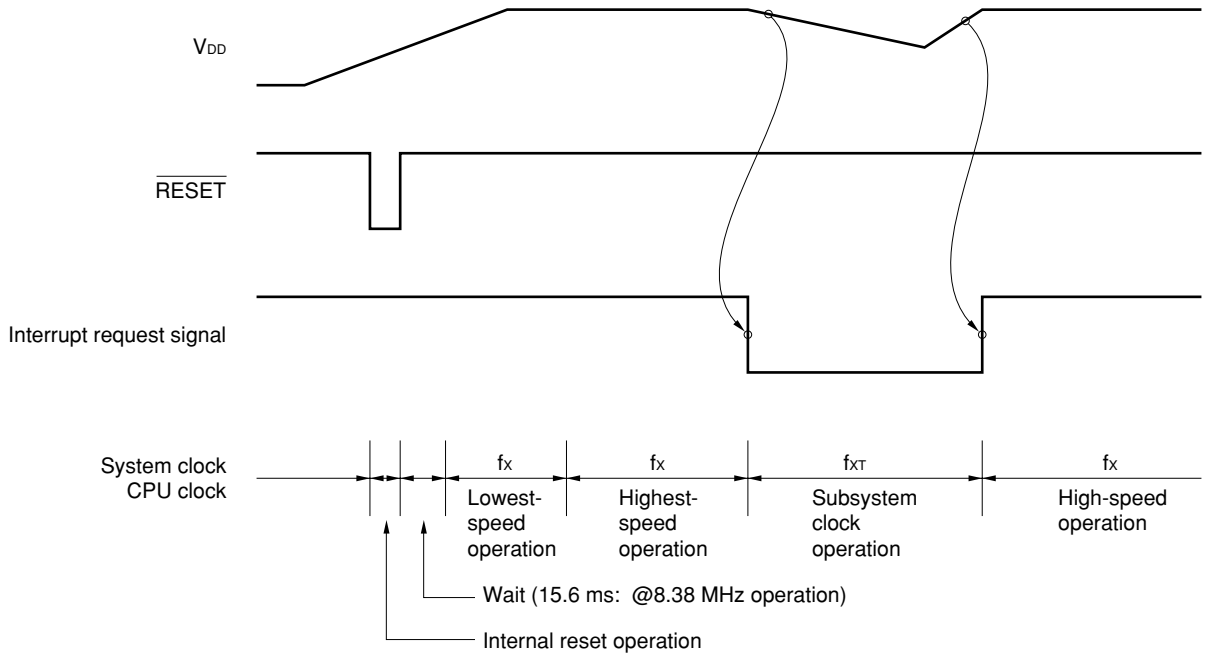
2. Figures in parentheses are for operation with f_X = 8.38 MHz and f_{XT} = 32.768 kHz.

Caution Selection of the CPU clock cycle dividing factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously. Simultaneous setting is possible, however, for selection of the CPU clock cycle dividing factor (PCC0 to PCC2) and switch over from the subsystem clock to the main system clock (changing CSS from 1 to 0).

5.6.2 System clock and CPU clock switching procedure

This section describes switching procedure between the system clock and CPU clock.

Figure 5-8. System Clock and CPU Clock Switching



<1> The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, main system clock starts oscillation. At this time, oscillation stabilization time (2^{17}/fx) is secured automatically.
After that, the CPU starts executing the instruction at the minimum speed of the main system clock (3.81 μs @ 8.38 MHz operation).

<2> After the lapse of a sufficient time for the V_{DD} voltage to increase to enable operation at maximum speeds, the PCC is rewritten and maximum-speed operation is carried out.

<3> Upon detection of a decrease of the V_{DD} voltage due to an interrupt request signal, the main system clock is switched to the subsystem clock (which must be in an oscillation stable state).

<4> Upon detection of V_{DD} voltage reset due to an interrupt, 0 is set to bit 7 (MCC) of the PCC and oscillation of the main system clock is started. After the lapse of time required for stabilization of oscillation, the PCC is rewritten and the maximum-speed operation is resumed.

Caution When subsystem clock is being operated while the main system clock is stopped, if switching to the main system clock is done again, be sure to switch after securing oscillation stabilization time by program.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0

6.1 Functions of 16-Bit Timer/Event Counter 0

The 16-bit timer/event counter 0 has the following functions.

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

(1) Interval timer

TM0 generates interrupt request at the preset time interval.

(2) PPG output

TM0 can output a square wave whose frequency and output pulse can be set freely.

(3) Pulse width measurement

TM0 can measure the pulse width of an externally input signal.

(4) External event counter

TM0 can measure the number of pulses of an externally input signal.

(5) Square-wave output

TM0 can output a square wave with any selected frequency.

6.2 Configuration of 16-Bit Timer/Event Counter 0

16-bit timer/event counter 0 consists of the following hardware.

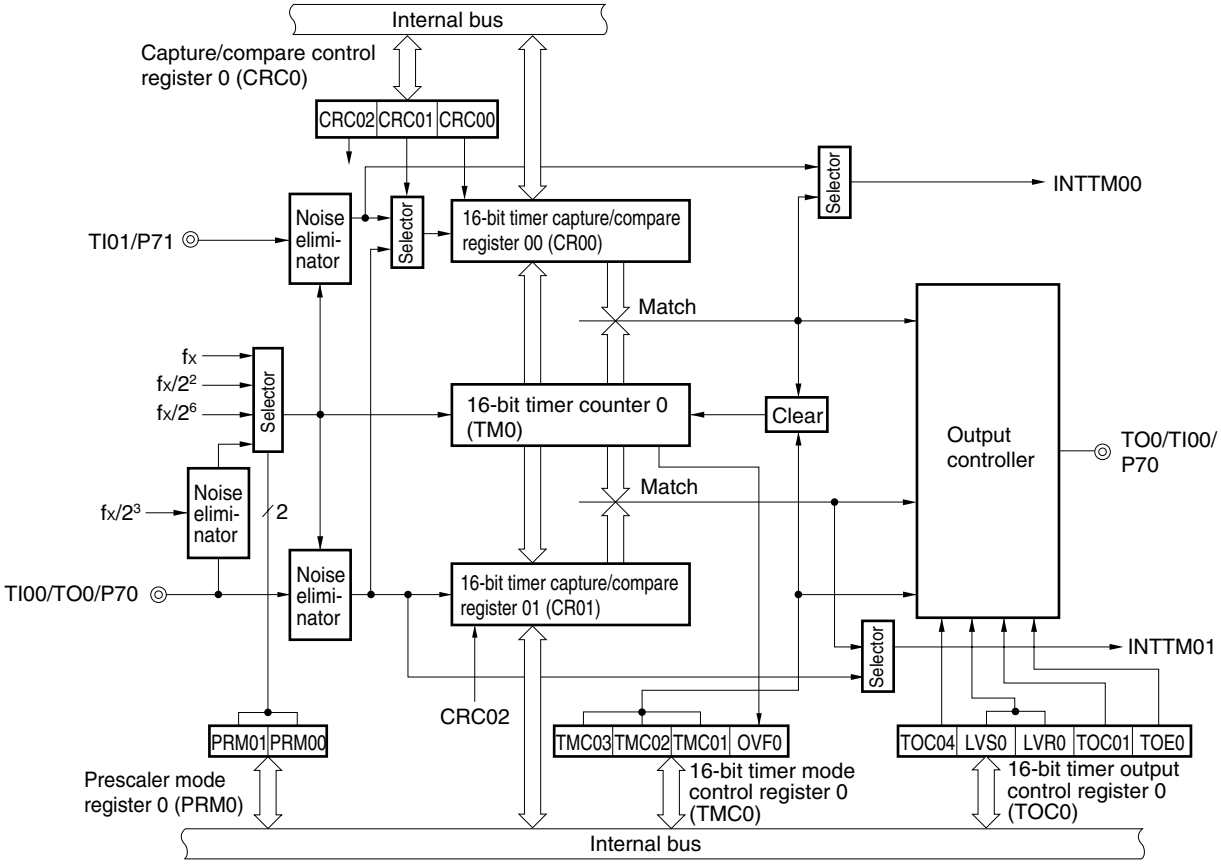
Table 6-1. Configuration of 16-Bit Timer/Event Counter 0

Item	Configuration
Timer/counter	16 bits × 1 (TM0)
Register	16-bit timer capture/compare register: 16 bits × 2 (CR00, CR01)
Timer output	1 (TO0)
Control registers	16-bit timer mode control register 0 (TMC0) Capture/compare control register 0 (CRC0) 16-bit timer output control register 0 (TOC0) Prescaler mode register 0 (PRM0) Port mode register 7 (PM7)**Note**

Note See **Figure 4-11 Block Diagram of P70 to P73** and **Figure 4-12 Block Diagram of P74 and P75.**

Figure 6-1 shows a block diagram.

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 0



(1) 16-bit timer counter 0 (TM0)

TM0 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of an input clock. If the count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read. The count value is reset to 0000H in the following cases:

<1> At $\overline{\text{RESET}}$ input
<2> If TMC03 and TMC02 are cleared
<3> If valid edge of TI00 is input in the clear & start mode by inputting valid edge of TI00
<4> If TM0 and CR00 match in the clear & start mode on match between TM0 and CR00

(2) 16-bit timer capture/compare register 00 (CR00)

CR00 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC00) of capture/compare control register 0 (CRC0).

- **When CR00 is used as a compare register**

  The value set in the CR00 is constantly compared with the 16-bit timer counter 0 (TM0) count value, and an interrupt request (INTTM00) is generated if they match. It can also be used as the register which holds the interval time when TM0 is set to interval timer operation.

- **When CR00 is used as a capture register**

  It is possible to select the valid edge of the TI00/TO0/P70 pin or the TI01/P71 pin as the capture trigger. Setting of the TI00 or TI01 valid edge is performed by means of prescaler mode register 0 (PRM0).

  If CR00 is specified as a capture register and capture trigger is specified to be the valid edge of the TI00/TO0/P70 pin, the situation is as shown in Table 6-2. On the other hand, when capture trigger is specified to be the valid edge of the TI01/P71 pin, the situation is as shown in Table 6-3.

Table 6-2. TI00/TO0/P70 Pin Valid Edge and CR00, CR01 Capture Trigger

ES01	ES00	TI00/TO0/P70 Pin Valid Edge	CR00 Capture Trigger	CR01 Capture Trigger
0	0	Falling edge	Rising edge	Falling edge
0	1	Rising edge	Falling edge	Rising edge
1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation	Both rising and falling edges

Table 6-3. TI01/P71 Pin Valid Edge and CR00 Capture Trigger

ES11	ES10	TI01/P71 Pin Valid Edge	CR00 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

CR00 is set by a 16-bit memory manipulation instruction.
After $\overline{\text{RESET}}$ input, the value of CR00 is undefined.

Cautions 1. Set a value other than 0000H in CR00 in the clear & start mode on match between TM0 and CR00. However, in the free-running mode and in the clear mode using the valid edge of TI00, if 0000H is set to CR00, an interrupt request (INTTM00) is generated following overflow (FFFFH).

2. If the value after CR00 is changed is smaller than that of 16-bit timer counter 0 (TM0), TM0 continues counting, overflows and then restarts counting from 0. Thus, if the value after CR00 is changed is smaller than the value before it was changed, it is necessary to restart the timer after changing CR00.

3. When P70 is used as the valid edge of TI00, it cannot be used as timer output (TO0). Moreover, when P70 is used as TO0, it cannot be used as the valid edge of TI00.

(3) 16-bit timer capture/compare register 01 (CR01)

CR01 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC02) of capture/compare control register 0 (CRC0).

- **When CR01 is used as a compare register**
  The value set in the CR01 is constantly compared with the 16-bit timer counter 0 (TM0) count value, and an interrupt request (INTTM01) is generated if they match.

- **When CR01 is used as a capture register**
  It is possible to select the valid edge of the TI00/TO0/P70 pin as the capture trigger. The TI00/TO0/P70 valid edge is set by means of prescaler mode register 0 (PRM0).

CR01 is set by a 16-bit memory manipulation instruction.
After $\overline{\text{RESET}}$ input, the value of CR01 is undefined.

Caution Set other than 0000H to CR01. This means 1-pulse count operation cannot be performed when CR01 is used as the event counter. However, in the free-running mode and in the clear mode using the valid edge of TI00, if 0000H is set to CR01, an interrupt request (INTTM01) is generated following overflow (FFFFH).

6.3 Registers to Control 16-Bit Timer/Event Counter 0

The following five types of registers are used to control the 16-bit timer/event counter 0.

- 16-bit timer mode control register 0 (TMC0)
- Capture/compare control register 0 (CRC0)
- 16-bit timer output control register 0 (TOC0)
- Prescaler mode register 0 (PRM0)
- Port mode register 7 (PM7)

(1) 16-bit timer mode control register 0 (TMC0)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 0 (TM0) clear mode, and output timing, and detects an overflow.

TMC0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC0 value to 00H.

Caution The 16-bit timer counter 0 (TM0) starts operation at the moment a value other than 0, 0 (operation stop mode) is set in TMC02 and TMC03, respectively. Set 0, 0 in TMC02 and TMC03 to stop the operation.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 0 (TMC0)

Address FF60H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0

TMC03	TMC02	TMC01	Operating mode and clear mode selection	TO0 output timing selection	Interrupt request generation
0	0	0	Operation stop (TM0 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM0 and CR00 or match between TM0 and CR01	Generated on match between TM0 and CR00, or match between TM0 and CR01
0	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	
1	0	0	Clear & start on TI00 valid edge	—	
1	0	1			
1	1	0	Clear & start on match between TM0 and CR00	Match between TM0 and CR00 or match between TM0 and CR01	
1	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	

OVF0	16-bit timer counter 0 (TM0) overflow detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Timer operation must be stopped before writing to bits other than the OVF0 flag.

2. Set the valid edge of the TI00/TO0/P70 pin with prescaler mode register 0 (PRM0).

3. If clear & start mode on match between TM0 and CR00 is selected, when the set value of CR00 is FFFFH and the TM0 value changes from FFFFH to 0000H, OVF0 flag is set to 1.

Remark TO0: 16-bit timer/event counter output pin
TI00: 16-bit timer/event counter input pin
TM0: 16-bit timer counter 0
CR00: 16-bit timer capture/compare register 00
CR01: 16-bit timer capture/compare register 01

(2) Capture/compare control register 0 (CRC0)

This register controls the operation of the 16-bit timer capture/compare registers (CR00, CR01).

CRC0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CRC0 value to 00H.

Figure 6-3. Format of Capture/Compare Control Register 0 (CRC0)

Address: FF62H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC0	0	0	0	0	0	CRC02	CRC01	CRC00

CRC02	CR01 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC01	CR00 capture trigger selection
0	Captures on valid edge of TI01
1	Captures on valid edge of TI00 by reverse phase

CRC00	CR00 operating mode selection
0	Operates as compare register
1	Operates as capture register

Cautions 1. Timer operation must be stopped before setting CRC0.

2. When clear & start mode on a match between TM0 and CR00 is selected with the 16-bit timer mode control register 0 (TMC0), CR00 should not be specified as a capture register.

3. If both the rising and falling edges have been selected as the valid edges of TI00, capture is not performed.

4. To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 0 (PRM0).

(3) 16-bit timer output control register 0 (TOC0)

This register controls the operation of the 16-bit timer/event counter 0 output controller. It sets R-S type flip-flop (LV0) setting/resetting, output inversion enabling/disabling, and 16-bit timer/event counter 0 timer output enabling/disabling.

TOC0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TOC0 value to 00H.

Figure 6-4 shows the TOC0 format.

Figure 6-4. Format of 16-Bit Timer Output Control Register 0 (TOC0)

Address: FF63H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOC0	0	0	0	TOC04	LVS0	LVR0	TOC01	TOE0

TOC04	Timer output F/F control by match of CR01 and TM0
0	Inversion operation disabled
1	Inversion operation enabled

LVS0	LVR0	16-bit timer/event counter timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOC01	Timer output F/F control by match of CR00 and TM0
0	Inversion operation disabled
1	Inversion operation enabled

TOE0	16-bit timer/event counter output control
0	Output disabled (output set to level 0)
1	Output enabled

Cautions 1. Timer operation must be stopped before setting TOC0.

2. If LVS0 and LVR0 are read after data is set, they will be 0.

(4) Prescaler mode register 0 (PRM0)

This register is used to set 16-bit timer counter 0 (TM0) count clock and TI00, TI01 input valid edges.

PRM0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PRM0 value to 00H.

Figure 6-5. Format of Prescaler Mode Register 0 (PRM0)

Address: FF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM0	ES11	ES10	ES01	ES00	0	0	PRM01	PRM00

ES11	ES10	TI01 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES01	ES00	TI00 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM01	PRM00	Count clock selection
0	0	f_X (8.38 MHz)
0	1	$f_X/2^2$ (2.09 MHz)
1	0	$f_X/2^6$ (131 kHz)
1	1	TI00 valid edge**Note**

Note The external clock requires a pulse two times longer than internal clock ($f_X/2^3$).

Cautions 1. If the valid edge of TI00 is to be set to the count clock, do not set the clear & start mode and the capture trigger at the valid edge of TI00.
Moreover, do not use the P70/TI00/TO0 pins as timer outputs (TO0).
2. Always set data to PRM0 after stopping the timer operation.
3. If the TI00 or TI01 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI00 pin or TI01 pin to enable the operation of the 16-bit timer counter 0 (TM0). Please be careful when pulling up the TI00 pin or the TI01 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

Remarks 1. f_X: Main system clock oscillation frequency
2. TI00, TI01: 16-bit timer/event counter input pin
3. Figures in parentheses are for operation with f_X = 8.38 MHz.

(5) Port mode register 7 (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P70/TO0/TI00 pin for timer output, set PM70 and the output latch of P70 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM7 value to FFH.

Figure 6-6. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.4 Operations of 16-Bit Timer/Event Counter 0

6.4.1 Interval timer operations

Setting the 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 6-7 allows operation as an interval timer. Interrupt request is generated repeatedly using the count value set in 16-bit timer capture/compare register 00 (CR00) beforehand as the interval.

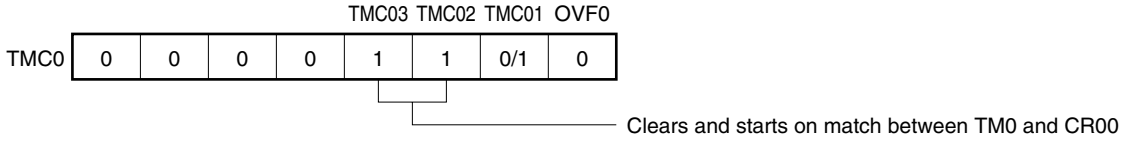
When the count value of the 16-bit timer counter 0 (TM0) matches the value set to CR00, counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM00) is generated.

Count clock of the 16-bit timer/event counter can be selected with bits 0 and 1 (PRM00, PRM01) of the prescaler mode register 0 (PRM0).

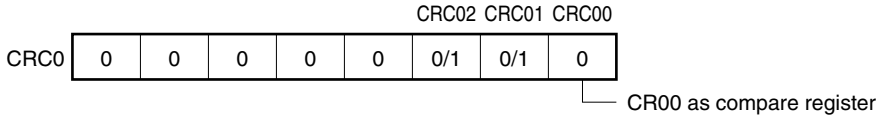
See **6.5 Cautions for 16-Bit Timer/Event Counter 0 (2)** about the operation when the compare register value is changed during timer count operation.

Figure 6-7. Control Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See **Figures 6-2** and **6-3**.

Figure 6-8. Interval Timer Configuration Diagram

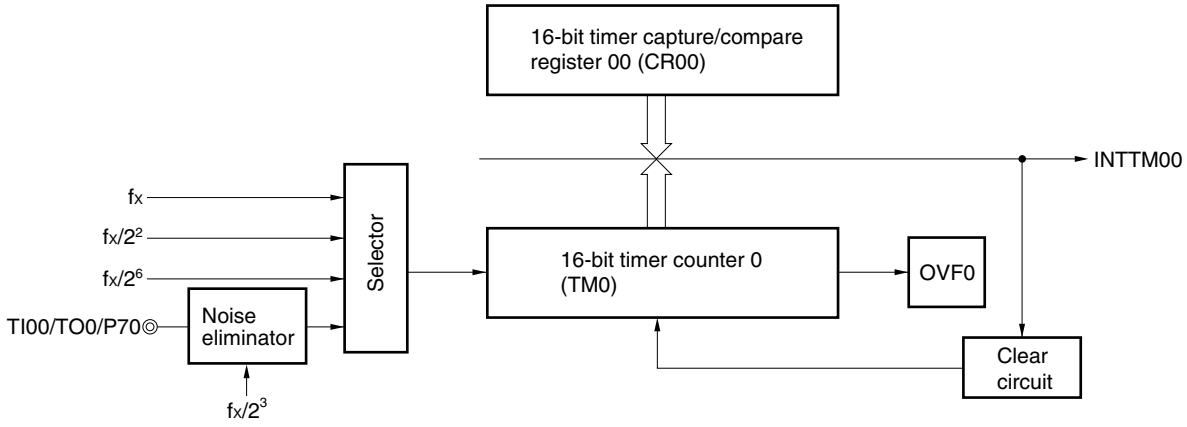
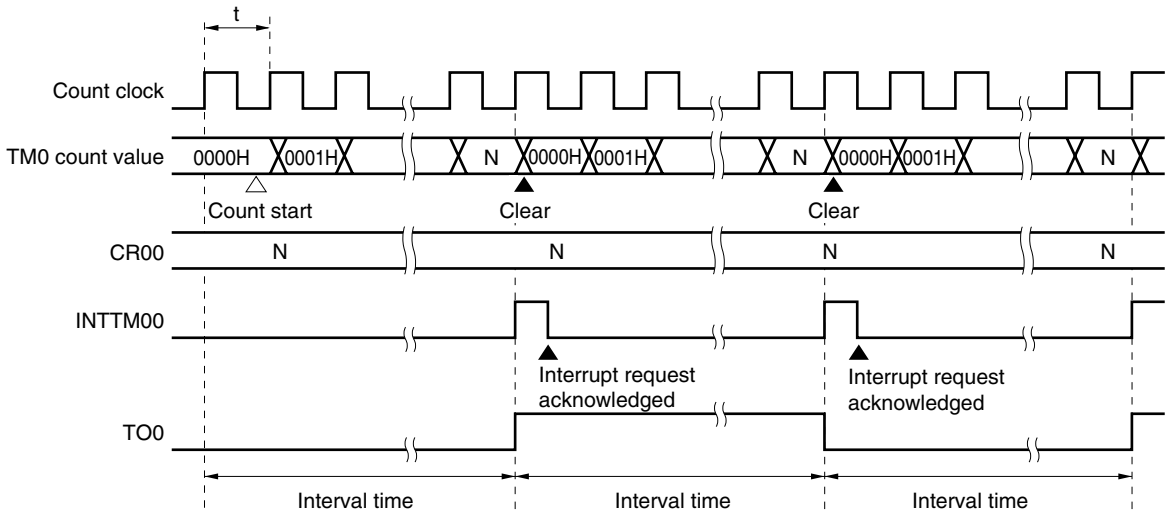


Figure 6-9. Timing of Interval Timer Operation



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

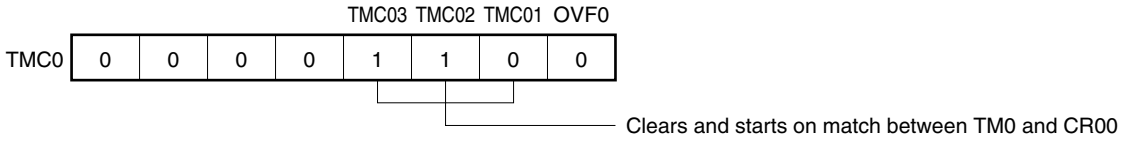
6.4.2 PPG output operations

Setting the 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 6-10 allows operation as PPG (Programmable Pulse Generator) output.

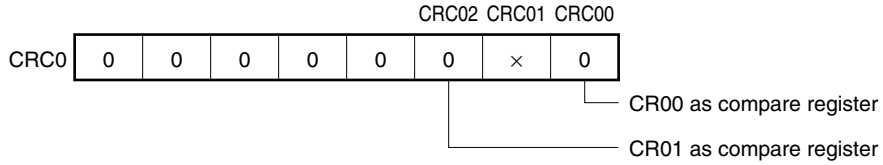
In the PPG output operation, square waves are output from the TO0/TI00/P70 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit timer capture/compare register 01 (CR01) and in 16-bit timer capture/compare register 00 (CR00), respectively.

Figure 6-10. Control Register Settings for PPG Output Operation

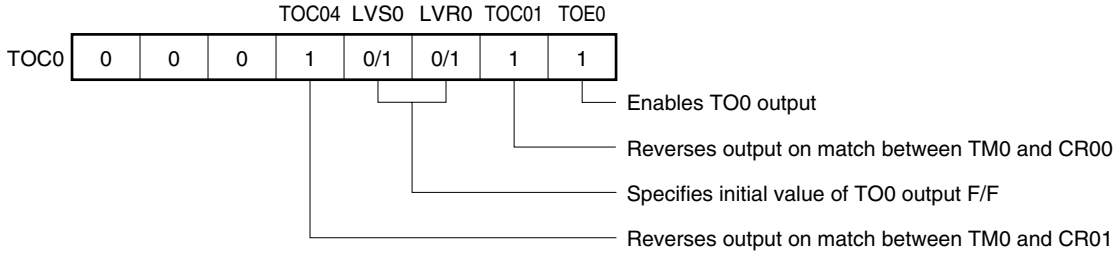
(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)



(c) 16-bit timer output control register 0 (TOC0)



Cautions 1. Values in the following range should be set in CR00 and CR01:
0000H < CR01 < CR00 ≤ FFFFH

2. The cycle of the pulse generated through PPG output (CR00 setting value + 1) has a duty of (CR01 setting value + 1)/(CR00 setting value + 1).

Remark ×: don't care

6.4.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/TO0/P70 pin and TI01/P71 pin using the 16-bit timer counter 0 (TM0).

There are two measurement methods: measuring with TM0 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00/TO0/P70 pin.

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

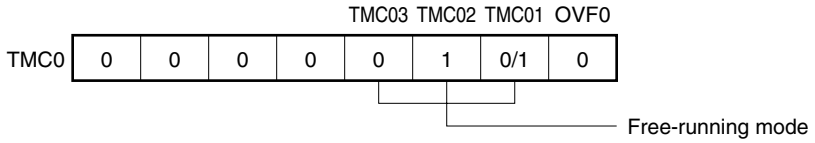
When the 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-11**), and the edge specified by prescaler mode register 0 (PRM0) is input to the TI00/TO0/P70 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

Any of three edge can be selected—rising, falling, or both edges—specified by means of bits 4 and 5 (ES00 and ES01) of PRM0.

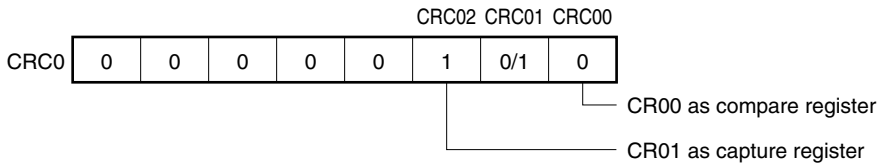
For valid edge detection, sampling is performed at the count clock selected by PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 6-11. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 6-2** and **6-3**.

Figure 6-12. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

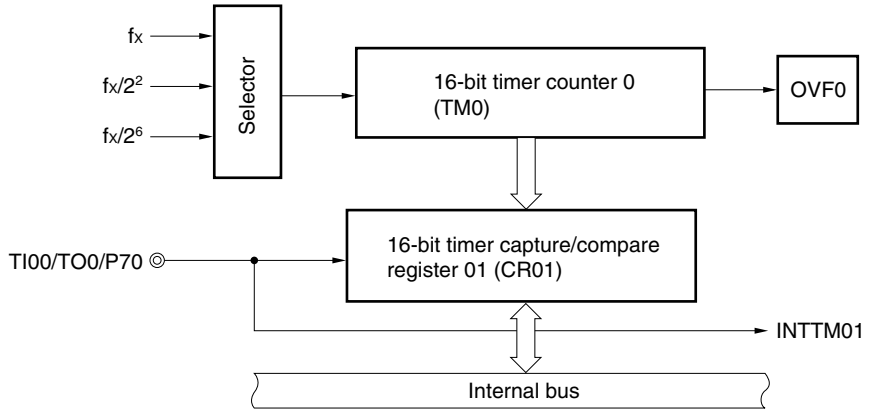
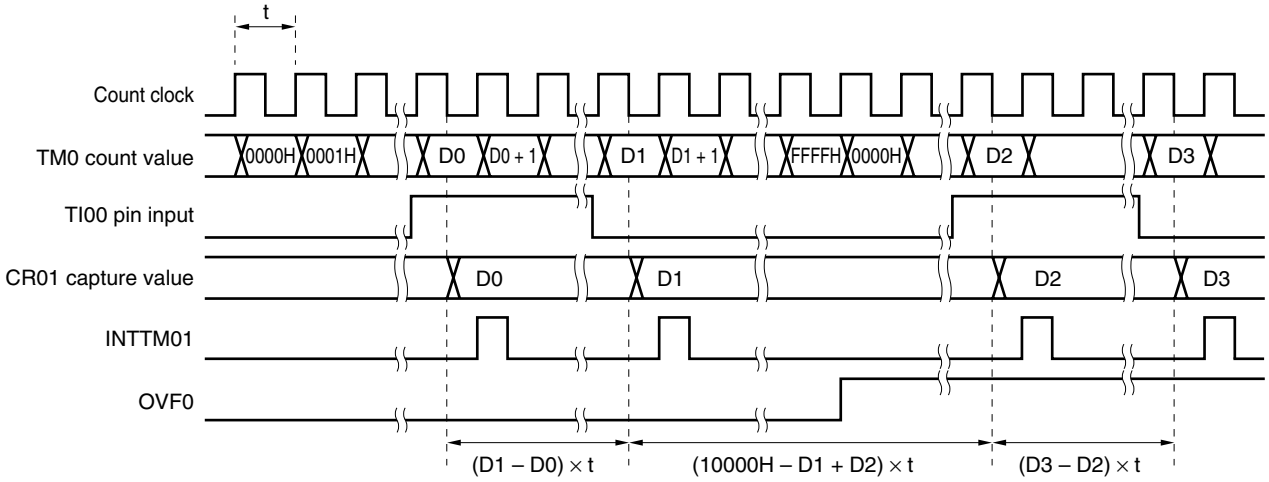


Figure 6-13. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



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

When the 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-14**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/TO0/P70 pin and the TI01/P71 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/TO0/P70 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

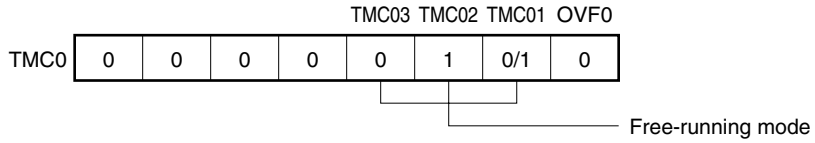
Also, when the edge specified by bits 6 and 7 (ES10 and ES11) of PRM0 is input to the TI01/P71 pin, the value of TM0 is taken into 16-bit timer capture/compare register 00 (CR00) and an interrupt request signal (INTTM00) is set.

Any of three edge can be selected—rising, falling, or both edges—as the valid edges for the TI00/TO0/P70 pin and the TI01/P71 pin specified by means of bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of PRM0, respectively.

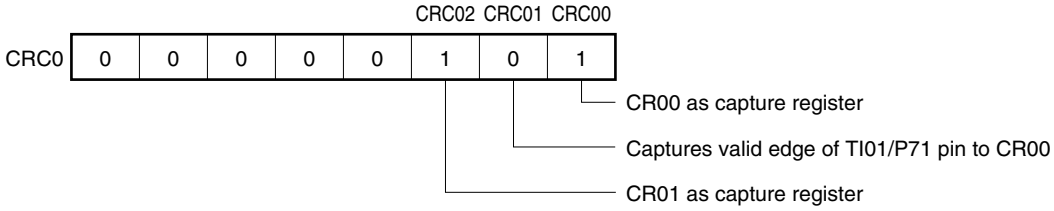
For valid edge detection of TI00/TO0/P70 and TI01/P71 pins, sampling is performed at the interval selected by means of the prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 6-14. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. For details, see **Figure 6-2**.

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

  Capture register operation in capture trigger input is shown.

Figure 6-15. CR01 Capture Operation with Rising Edge Specified

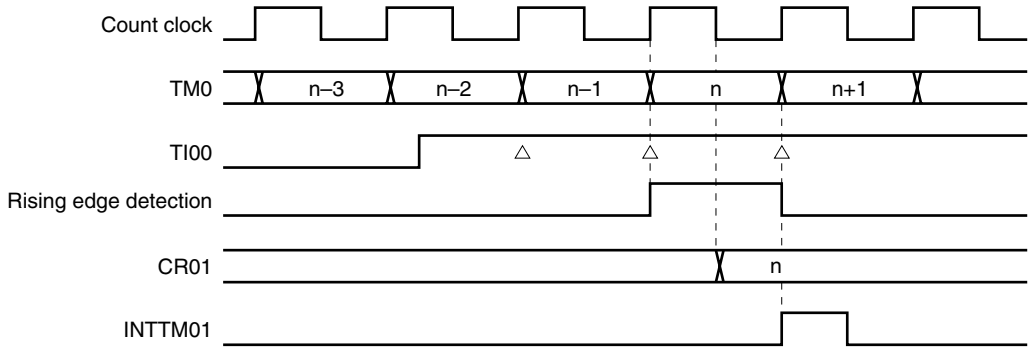
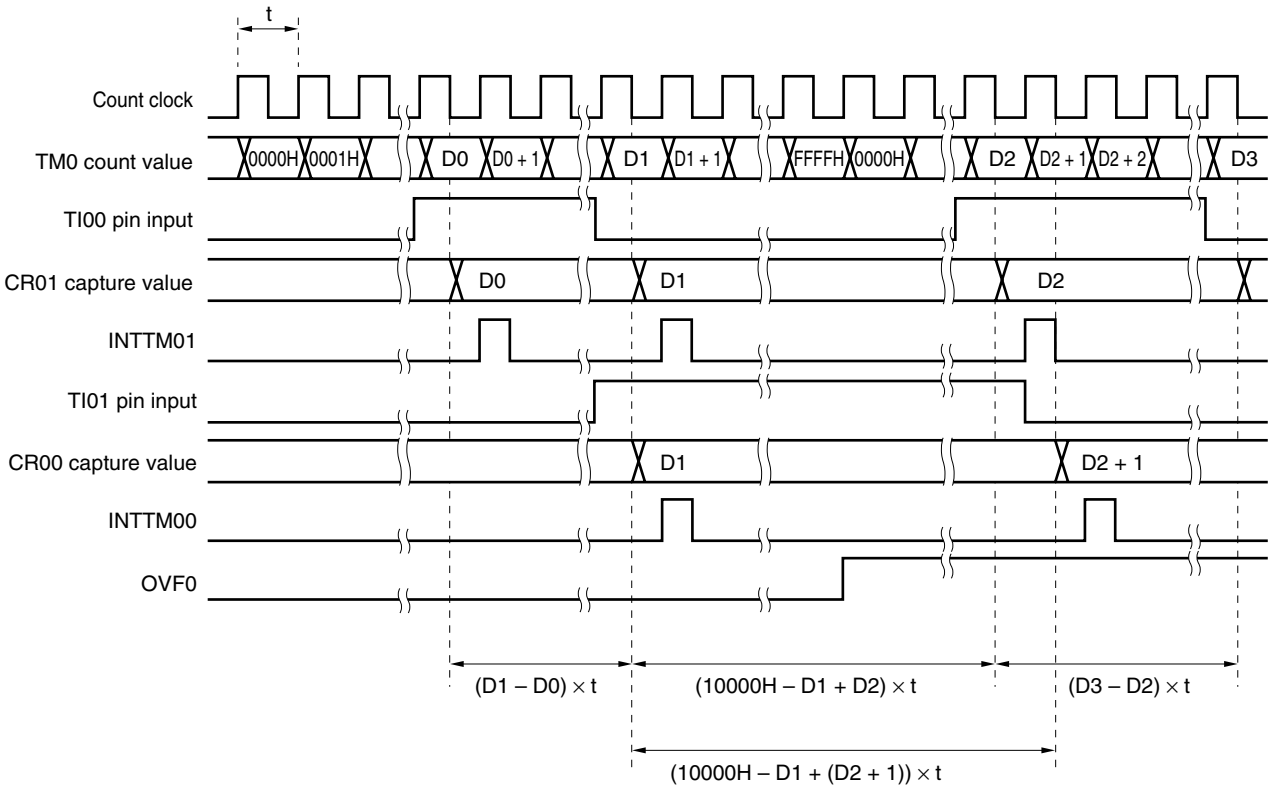


Figure 6-16. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



(3) Pulse width measurement with free-running counter and two capture registers

When the 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-17**), it is possible to measure the pulse width of the signal input to the TI00//TO0/P70 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/TO0/P70 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

Also, on the inverse edge input of that of the capture operation into CR01, the value of TM0 is taken into 16-bit timer capture/compare register 00 (CR00).

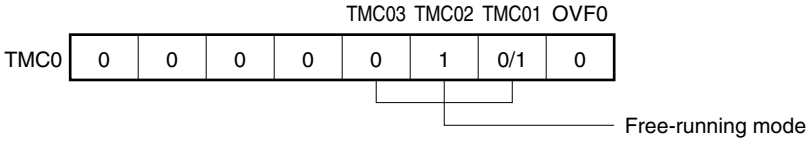
Either of two edge can be selected—rising or falling—as the valid edges for the TI00/TO0/P70 pin specified by means of bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

For TI00/TO0/P70 pin valid edge detection, sampling is performed at the interval selected by means of the prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

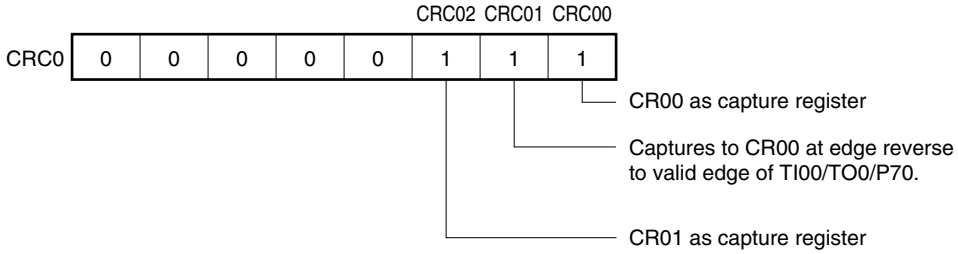
Caution If the valid edge of TI00/TO0/P70 pin is specified to be both rising and falling edges, the 16-bit timer capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

(a) 16-bit timer mode control register 0 (TMC0)

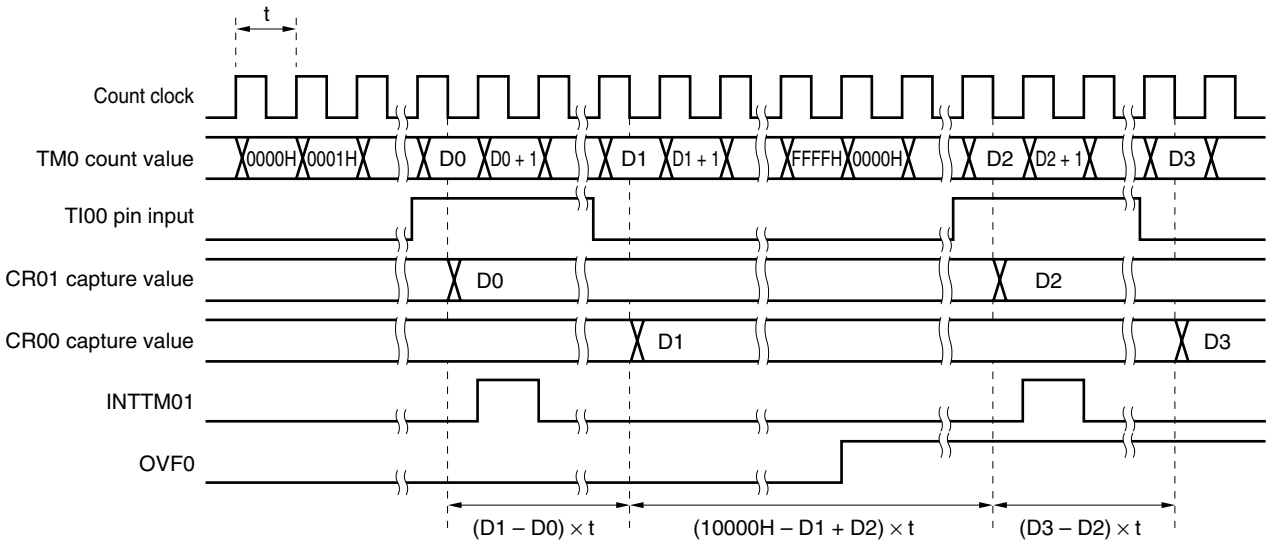


(b) Capture/compare control register 0 (CRC0)



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

Figure 6-18. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



(4) Pulse width measurement by means of restart

When input of a valid edge to the TI00/TO0/P70 pin is detected, the count value of the 16-bit timer counter 0 (TM0) is taken into 16-bit timer capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/TO0/P70 pin is measured by clearing TM0 and restarting the count (see register settings in **Figure 6-19**).

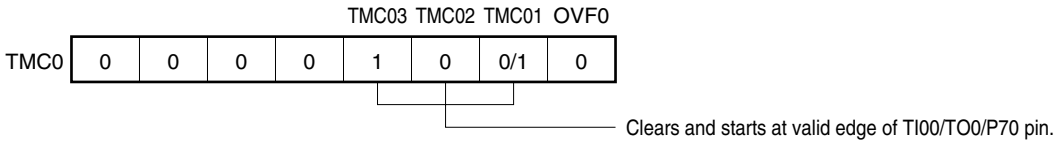
The edge specification can be selected from two types, rising and falling edges by bits 4 and 5 (ES00 and ES01) of the prescaler mode register 0 (PRM0).

In a valid edge detection, the sampling is performed by a cycle selected by the prescaler mode register 0 (PRM0) and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

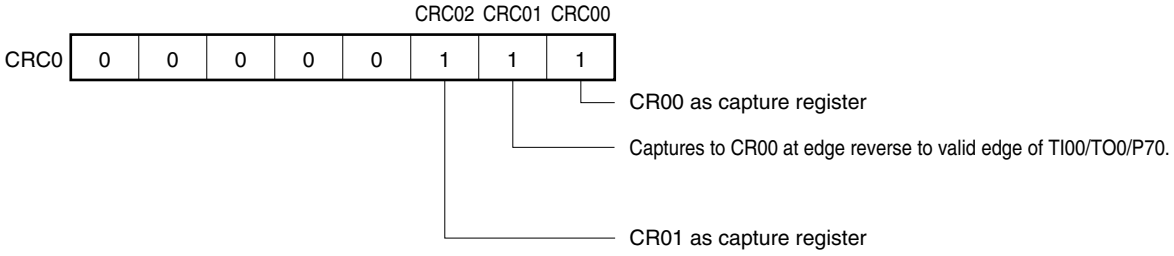
Caution If the valid edge of TI00/TO0/P70 pin is specified to be both rising and falling edges, the 16-bit timer capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-19. Control Register Settings for Pulse Width Measurement by Means of Restart

(a) 16-bit timer mode control register 0 (TMC0)

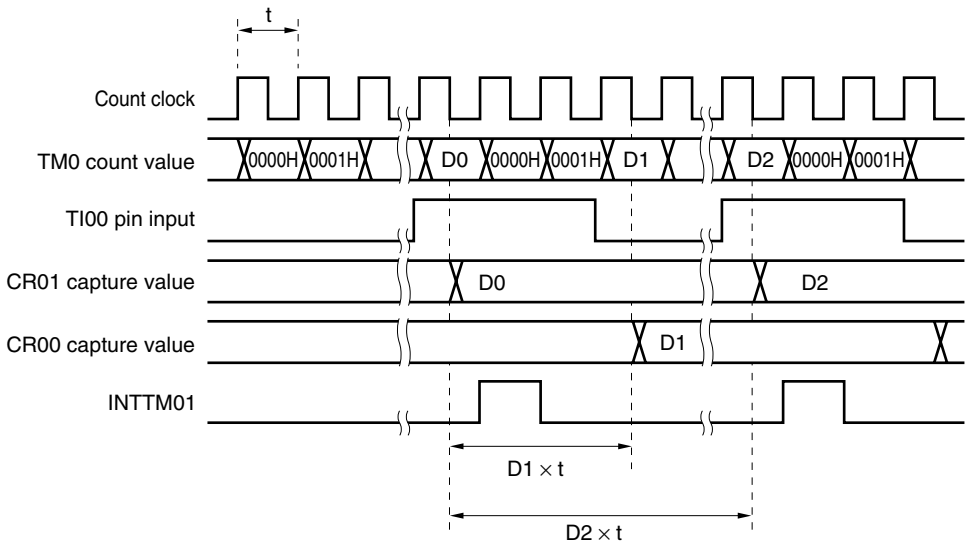


(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figure 6-2**.

Figure 6-20. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.4.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/TO0/P70 pin with the 16-bit timer counter 0 (TM0).

TM0 is incremented each time the valid edge specified with the prescaler mode register 0 (PRM0) is input.

When the TM0 counted value matches the 16-bit timer capture/compare register 00 (CR00) value, TM0 is cleared to 0 and the interrupt request signal (INTTM00) is generated.

Input the value except 0000H to CR00 (count operation with a pulse cannot be carried out).

The rising edge, the falling edge, or both edges can be selected with bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

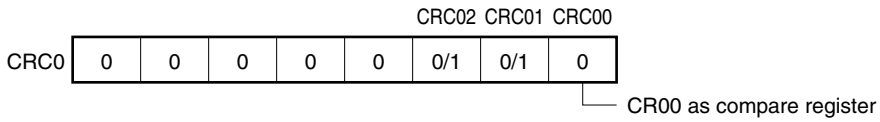
Because operation is carried out only after the valid edge is detected twice by sampling with the internal clock ($f_X/2^3$), noise with short pulse widths can be eliminated.

Caution When used as an external event counter, the P70/TI00/TO0 pin cannot be used as timer output (TO0).

Figure 6-21. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control register 0 (TMC0)

(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See **Figures 6-2** and **6-3**.

Figure 6-22. External Event Counter Configuration Diagram

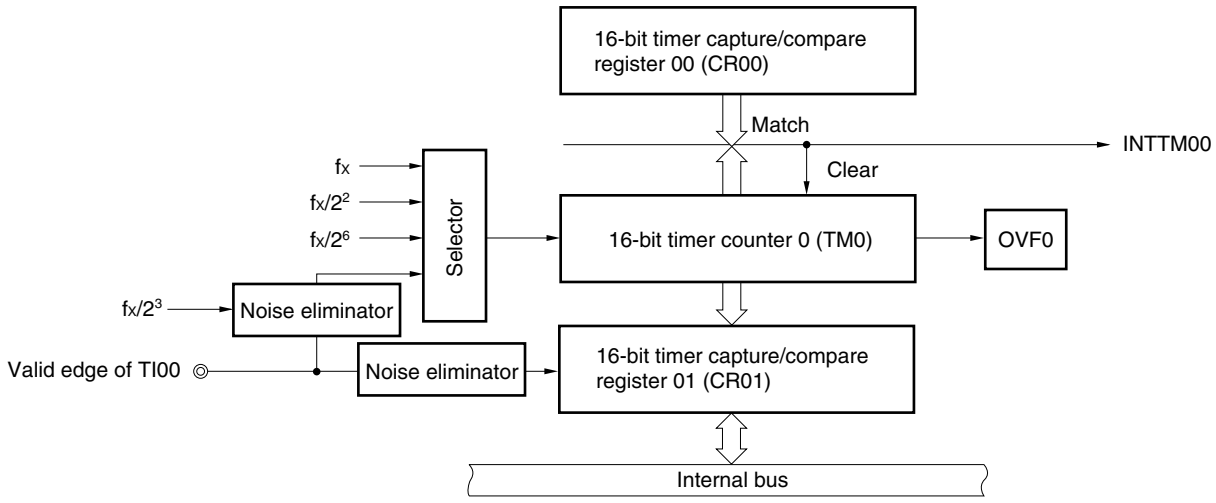
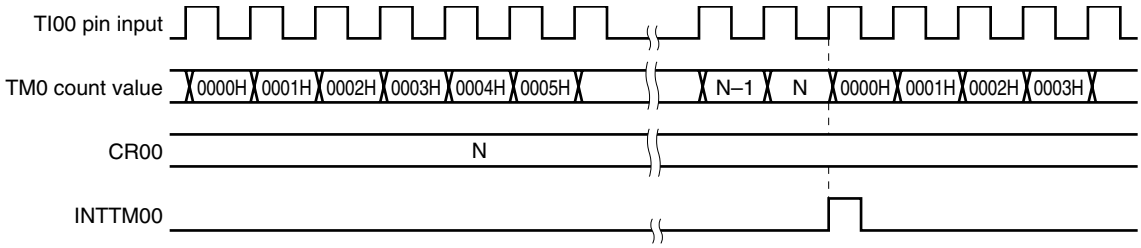


Figure 6-23. External Event Counter Operation Timings (with Rising Edge Specified)



Caution When reading the external event counter count value, TM0 should be read.

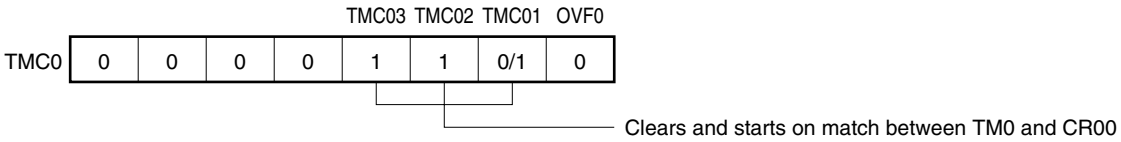
6.4.5 Square-wave output operation

A square wave with any selected frequency to be output at intervals of the count value preset to the 16-bit timer capture/compare register 00 (CR00) operates.

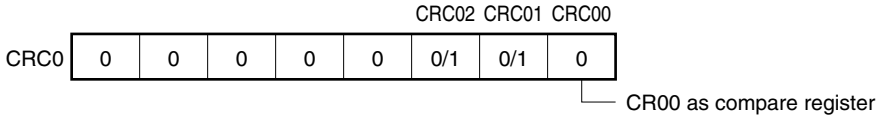
The TO0 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of the 16-bit timer output control register 0 (TOC0) to 1. This enables a square wave with any selected frequency to be output.

Figure 6-24. Control Register Settings in Square-Wave Output Mode

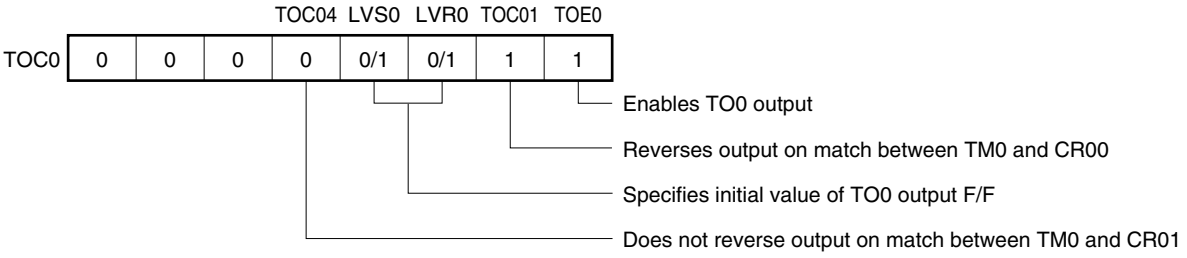
(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)

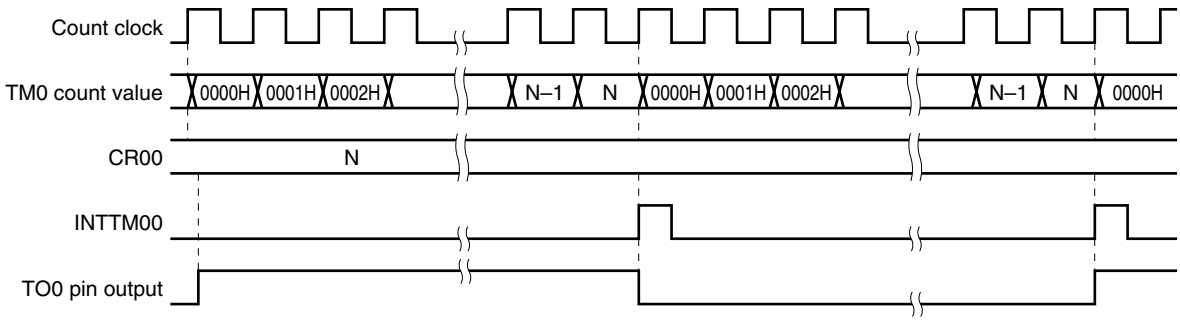


(c) 16-bit timer output control register 0 (TOC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See **Figures 6-2**, **6-3**, and **6-4**.

Figure 6-25. Square-Wave Output Operation Timing

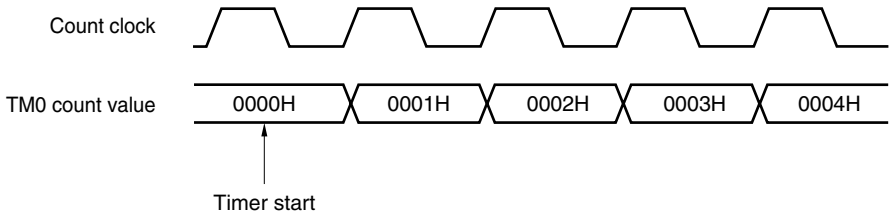


6.5 Cautions for 16-Bit Timer/Event Counter 0

(1) Timer start errors

An error with a maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because the 16-bit timer counter 0 (TM0) is started asynchronously with the count clock.

Figure 6-26. 16-Bit Timer Counter 0 (TM0) Start Timing



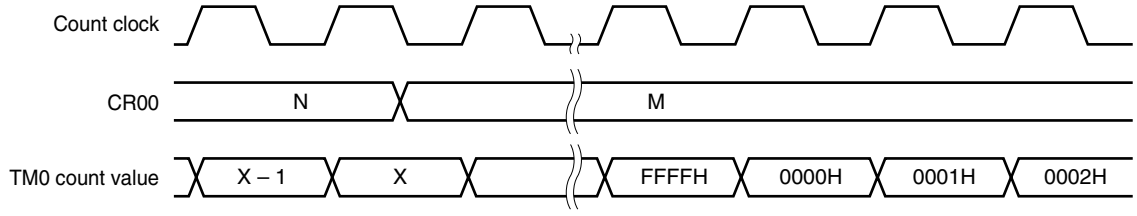
(2) 16-bit timer compare register setting (in clear & start mode on match between TM0 and CR00)

Set other than 0000H to 16-bit timer capture/compare registers 00, 01 (CR00, CR01). This means 1-pulse count operation cannot be performed when it is used as the event counter.

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

If the value after the 16-bit timer capture/compare register 00 (CR00) is changed is smaller than that of 16-bit timer counter 0 (TM0), TM0 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR00 is changed is smaller than the value (N) before it was changed, it is necessary to reset and restart the timer after changing CR00.

Figure 6-27. Timings After Change of Compare Register During Timer Count Operation

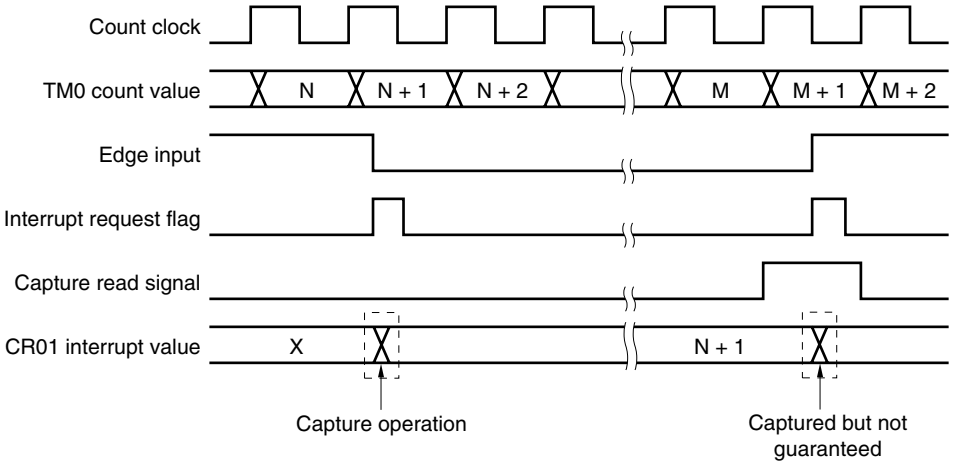


Remark N > X > M

(4) Capture register data retention timings

If the valid edge of the TI00/TO0/P70 pin is input during 16-bit timer capture/compare register 01 (CR01) read, CR01 carries out capture operation but the capture value at this time is not guaranteed. However, the interrupt request flag (TMIF01) is set upon detection of the valid edge.

Figure 6-28. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00/TO0/P70 pin after setting bits 2 and 3 (TMC02 and TMC03) of the 16-bit timer mode control register 0 (TMC0) to 0, 0, respectively, and then stopping timer operation. Valid edge is set with bits 4 and 5 (ES00 and ES01) of the prescaler mode register 0 (PRM0).

(6) Operation of OVF0 flag

<1> OFV0 flag is set to 1 in the following case.

Either the clear & start mode on match between TM0 and CR00 or the free-running mode that clears and starts at the valid edge of TIn is selected.

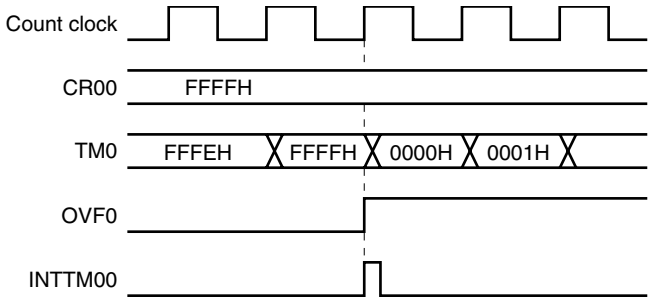
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 6-29. Operation Timing of OVF0 Flag



<2> Even if the OVF0 flag is cleared before the next count clock (before TM0 becomes 0001H) after the occurrence of TM0 overflow, the OVF0 flag is reset newly and clear is disabled.

(7) Contending operations

<1> The contending operation between the read time of 16-bit timer capture/compare register (CR00/CR01) and capture trigger input (CR00/CR01 used as capture register)
Capture trigger input is prior to the other. The data read from CR00/CR01 is not defined.

<2> The match timing of contending operation between the write period of 16-bit timer capture/compare register (CR00/CR01) and 16-bit timer counter 0 (TM0) (CR00/CR01 used as a compare register)
The match discrimination is not performed normally. Do not write any data to CR00/CR01 near the match timing.

(8) Timer operation

<1> Even if the 16-bit timer counter 0 (TM0) is read, the value is not captured by 16-bit timer capture/compare register 01 (CR01).

<2> Regardless of the CPU's operation mode, when the timer stops, the input signals to pins TI00/TI01 are not acknowledged.

(9) Capture operation

<1> If TI00 is specified as the valid edge of the count clock, capture operation by the capture register specified as the trigger for TI00 is not possible.
<2> If both the rising and falling edges are selected as the valid edges of TI00, capture is not performed.
<3> To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 0 (PRM0).
<4> The capture operation is performed at the fall of the count clock. An interrupt request input (INTTM0n), however, is generated at the rise of the next count clock.

(10) Compare operation

<1> When the 16-bit timer capture/compare register (CR00/CR01) is overwritten during timer operation, match interrupt may be generated or clear operation may not be performed normally if that value is close to the timer value and larger than the timer value.
<2> Capture operation may not be performed for CR00/CR01 set in compare mode even if a capture trigger has been input.

(11) Edge detection

<1> If the TI00 pin or the TI01 pin is high level immediately after system reset and rising edge or both the rising and falling edges are specified as the valid edge for the TI00 pin or TI01 pin to enable the 16-bit timer counter 0 (TM0) operation, a rising edge is detected immediately after. Be careful when pulling up the TI00 pin or the TI01 pin. However, the rising edge is not detected at restart after the operation has been stopped once.
<2> The sampling clock used to eliminate noise differs when a TI00 valid edge is used as count clock and when it is used as a capture trigger. In the former case, the count clock is $f_X/2^3$, and in the latter case the count clock is selected by prescaler mode register 0 (PRM0). When a valid edge is detected twice by sampling, the capture operation is started, therefore noise with short pulse widths can be eliminated.

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50 AND 51

7.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 (TM50, TM51) have the following two modes.

- Mode using 8-bit timer/event counters alone (single mode)
- Mode using the cascade connection (16-bit resolution: cascade connection mode)

These two modes are described next.

(1) Mode using 8-bit timer/event counters alone (single mode)

The timer operates as an 8-bit timer/event counter.
It has the following functions.

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

(2) Mode using the cascade connection (16-bit resolution: cascade connection mode)

The timer operates as a 16-bit timer/event counter by connecting in cascade.
It has the following functions.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

Figures 7-1 and 7-2 show 8-bit timer/event counter block diagrams.

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

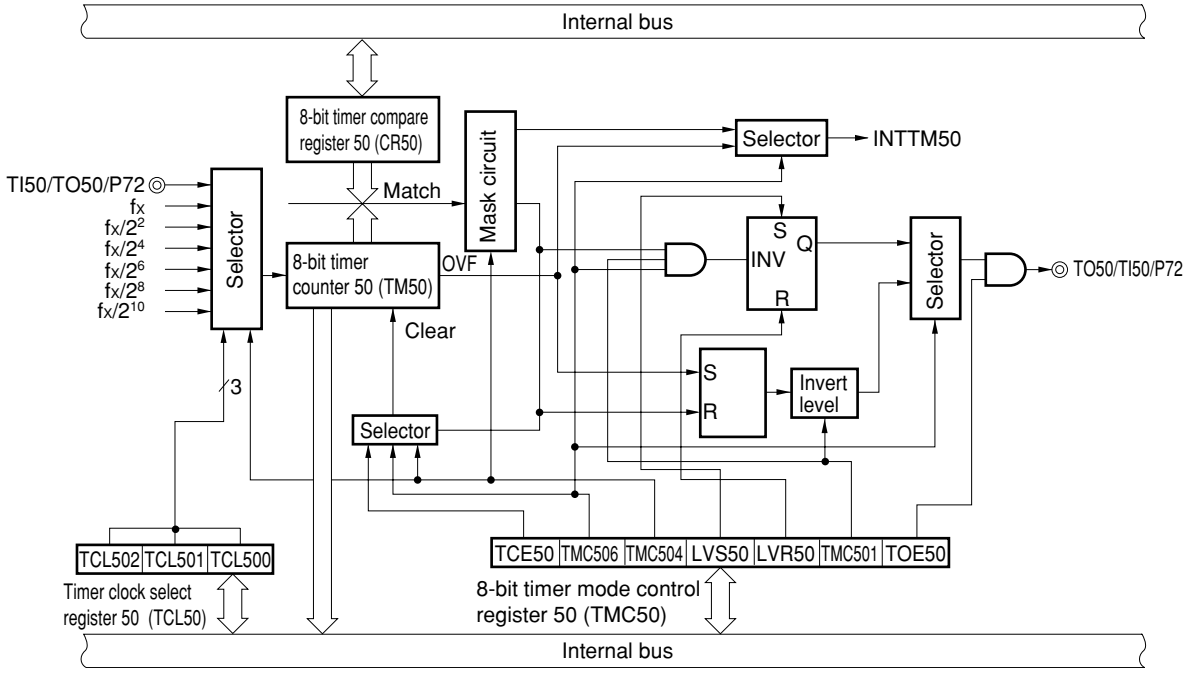
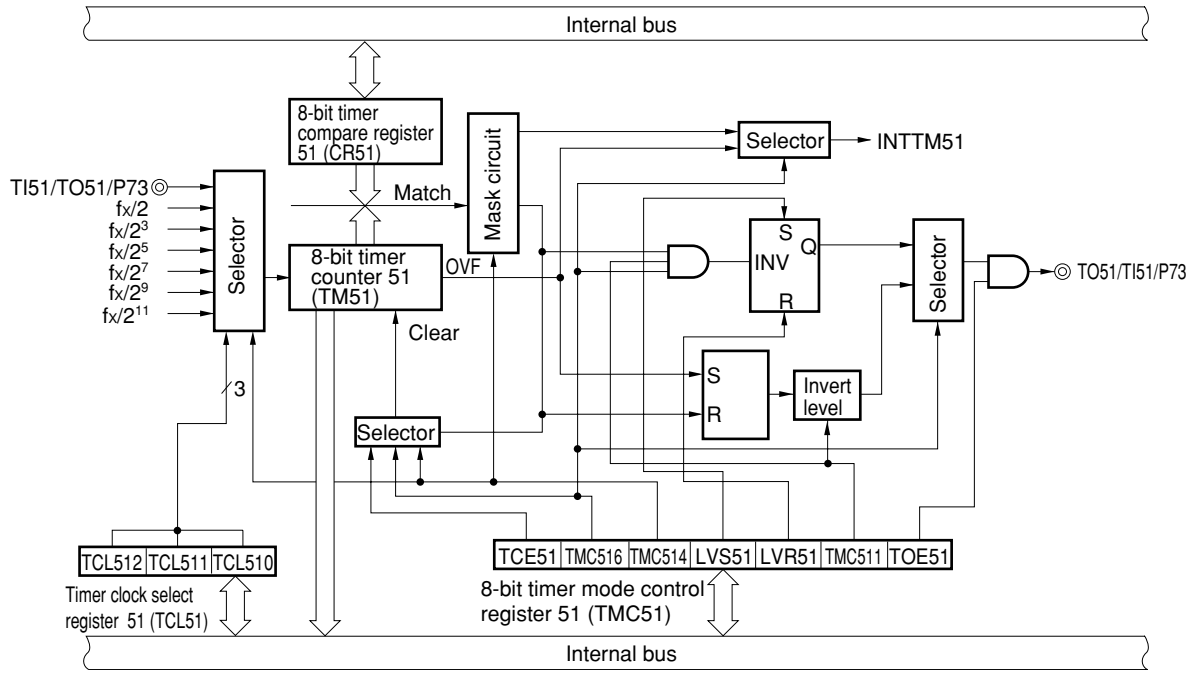


Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51



7.2 Configurations of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 consist of the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer output	2 (TO5n)
Control registers	Timer clock select register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 7 (PM7)**Note**

Note See **Figure 4-11 Block Diagram of P70 to P73.**

Remark n = 0, 1

(1) 8-bit timer counter 5n (TM5n: n = 0, 1)

TM5n is an 8-bit read-only register which counts the count pulses.

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

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

When TM50 and TM51 can be connected in cascade and used as a 16-bit timer, they can be read by a 16-bit memory operation instruction. However, since they are connected by an internal 8-bit bus, TM50 and TM51 are read separately in two times. Thus, take read during count change into consideration and compare them in two times reading. When count value is read during operation, count clock input is temporary stopped, and then the count value is read. In the following situations, count value is set to 00H.

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

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in clear & start mode on match between TM5n and CR5n.

Caution In cascade connection mode, the count value is reset to 00H when the lowest timer TCE5n is cleared.

Remark n = 0, 1

(2) 8-bit timer compare register 5n (CR5n: n = 0, 1)

The value set in CR5n is constantly compared with the 8-bit timer counter (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match (except PWM mode).

It is possible to rewrite the value of CR5n within 00H to FFH during count operation.

When TM50 and TM51 can be connected in cascade and used as a 16-bit timer, CR50 and CR51 operate as the 16-bit compare register. It compares count value with register value, and if the values are matched, an interrupt request (INTTM50) is generated. INTTM51 interrupt request is also generated at this time. Thus, when TM50 and TM51 are used as cascade connection, mask INTTM51 interrupt request.

Caution When changing the set value of 8-bit compare register 5n (CR5n) in cascade connection mode, change the value after stopping the timer operation of cascade-connected 8-bit timer counter 5n (TM5n).

Remark n = 0, 1

7.3 Registers to Control 8-Bit Timer/Event Counters 50 and 51

The following three types of registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock select register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 7 (PM7)

n = 0, 1

(1) Timer clock select register 5n (TCL5n: n = 0, 1)

This register sets count clocks of 8-bit timer/event counter 5n and the valid edge of TI50, TI51 input.

TCL5n is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL5n to 00H.

Figure 7-3. Format of Timer Clock Select Register 50 (TCL50)

Address: FF71H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection
0	0	0	TI50 falling edge
0	0	1	TI50 rising edge
0	1	0	f_X (8.38 MHz)
0	1	1	$f_X/2^2$ (2.09 MHz)
1	0	0	$f_X/2^4$ (523 kHz)
1	0	1	$f_X/2^6$ (131 kHz)
1	1	0	$f_X/2^8$ (32.7 kHz)
1	1	1	$f_X/2^{10}$ (8.18 kHz)

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

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

Remarks 1. When cascade connection is used, only the settings of TCL502 to TCL00 are valid.

2. fx: Main system clock oscillation frequency

3. Figures in parentheses are for operation with fx = 8.38 MHz

Figure 7-4. Format of Timer Clock Select Register 51 (TCL51)

Address: FF79H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection
0	0	0	TI51 falling edge
0	0	1	TI51 rising edge
0	1	0	$f_X/2$ (4.19 MHz)
0	1	1	$f_X/2^3$ (1.04 MHz)
1	0	0	$f_X/2^5$ (261 kHz)
1	0	1	$f_X/2^7$ (65.4 kHz)
1	1	0	$f_X/2^9$ (16.3 kHz)
1	1	1	$f_X/2^{11}$ (4.09 kHz)

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

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

Remarks 1. When cascade connection is used, the settings of TCL5n0 to TCL5n2 (n = 0, 1) are valid only for the lowermost timer.

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) 8-bit timer mode control register 5n (TMC5n: n = 0, 1)

TMC5n is a register which sets up the following six types.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Single mode/cascade connection mode selection
<4> Timer output F/F (flip flop) status setting
<5> Active level selection in timer F/F control or PWM (free-running) mode
<6> Timer output control

TMC5n is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC5n to 00H.

Figure 7-5 shows the TMC5n format.

Figure 7-5. Format of 8-Bit Timer Mode Control Register 5n (TMC5n)

Address: FF70H (TMC50) FF78H (TMC51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC5n	TCE5n	TMC5n6	0	TMC5n4	LVS5n	LVR5n	TMC5n1	TOE5n

TCE5n	TM5n count operation control
0	After clearing to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMC5n6	TM5n operating mode selection
0	Clear and start mode by matching between TM5n and CR5n
1	PWM (free-running) mode

TMC5n4	Single mode/cascade connection mode selection
0	Single mode (use the lowest timer)
1	Cascade connection mode (connect to lower timer)

LVS5n	LVR5n	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC5n1	In other modes (TMC5n6 = 0)	In PWM mode (TMC5n6 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOE5n	Timer output control
0	Output disabled (port mode)
1	Output enabled

Caution Before clearing TCE5n to 0, set the interrupt mask flag (TMMK5n) to 1. This is because an interrupt may occur after TCE5n has been cleared.
Clear TCE5n to 0 using the following procedure.

```
TMMK5n = 1      ; Mask set
TCE5n = 0       ; Timer clear
TMIF5n = 0      ; Interrupt request flag clear
TMMK5n = 0      ; Mask clear
    :
TCE5n = 1       ; Timer start
    :
```

Remarks 1. In PWM mode, PWM output will be inactive because of TCE5n = 0.

2. If LVS5n and LVR5n are read after data is set, 0 is read.

3. n = 0, 1

(3) Port mode register 7 (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P72/TO50/TI50 and P73/TI51/TO51 pins for timer output, set PM72, PM73, and output latches of P72 and P73 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM7 to FFH.

Figure 7-6. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

7.4 Operations of 8-Bit Timer/Event Counters 50 and 51

7.4.1 Interval timer (8-bit) operation

The 8-bit timer/event counters operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count values of the 8-bit timer counter 5n (TM5n) match the values set to CR5n, counting continues with the TM5n values cleared to 0 and the interrupt request signals (INTTM5n) are generated.

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

See **7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51 (2)** about the operation when the compare register value is changed during timer count operation.

[Setting]

<1> Set the registers.

- TCL5n: Select count clock
- CR5n: Compare value
- TMC5n: Clear and start mode by match of TM5n and CR5n
  (TMC5n = 0000×××0B × = don't care)

<2> After TCE5n = 1 is set, count operation starts.

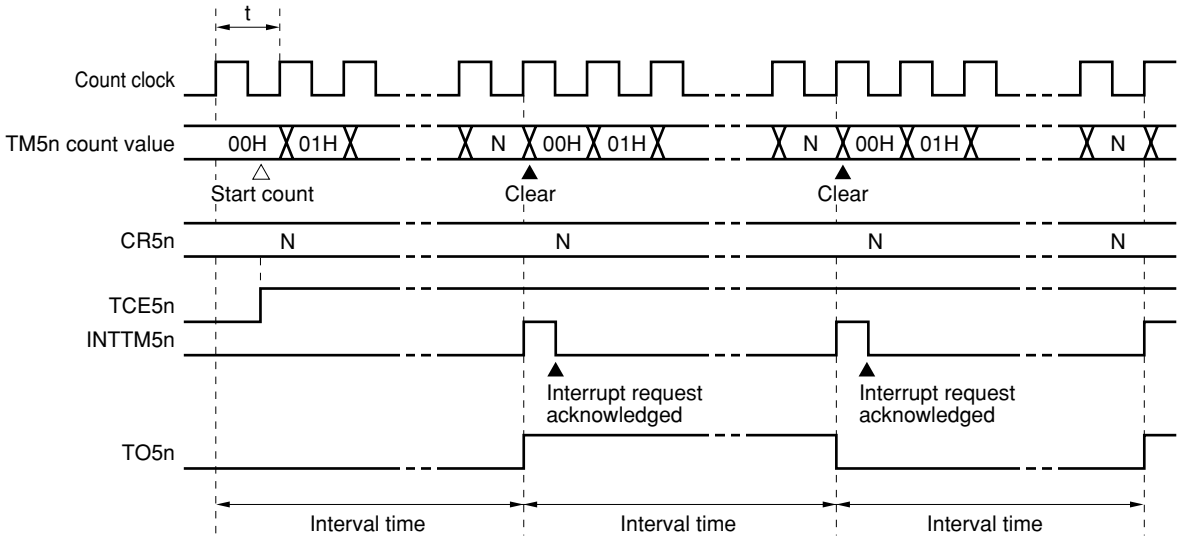
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n generates repeatedly at the same interval. Set TCE5n to 0 to stop count operation.

Remark n = 0, 1

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

(a) Basic operation

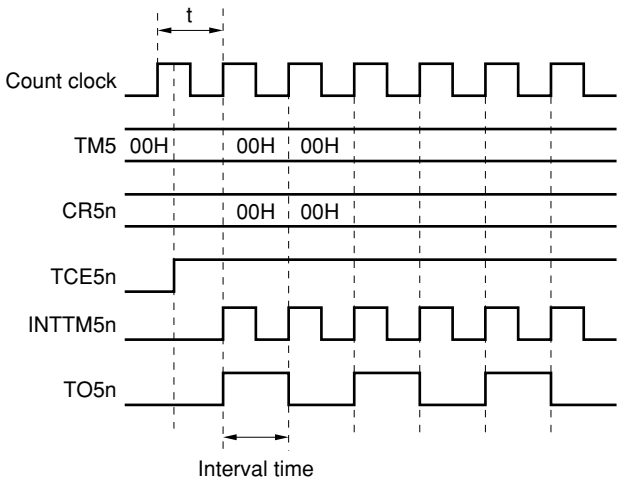


Remarks 1. Interval time = (N + 1) × t
N = 00H to FFH

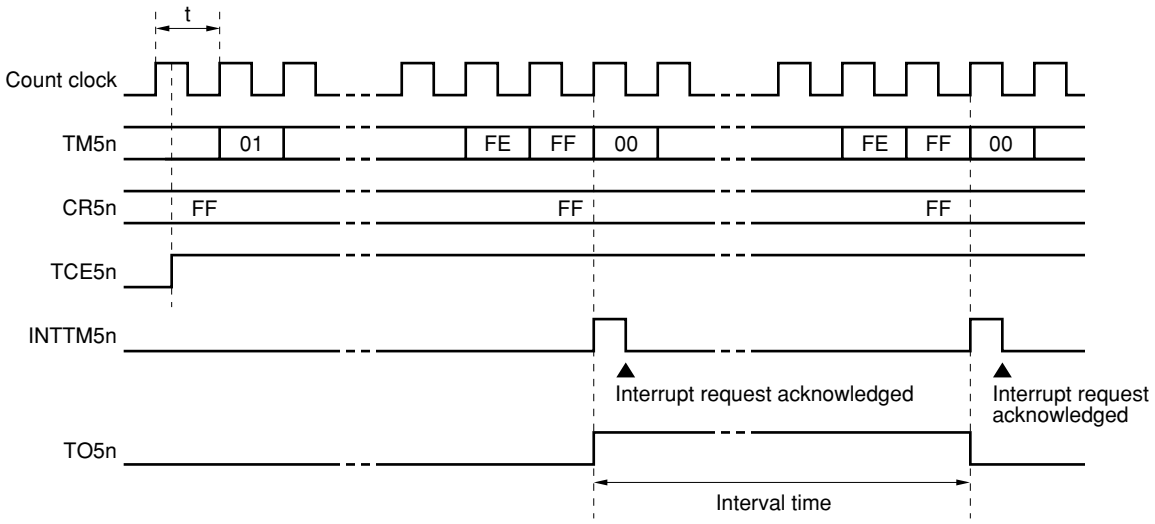
2. n = 0, 1

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

(b) When CR5n = 00H



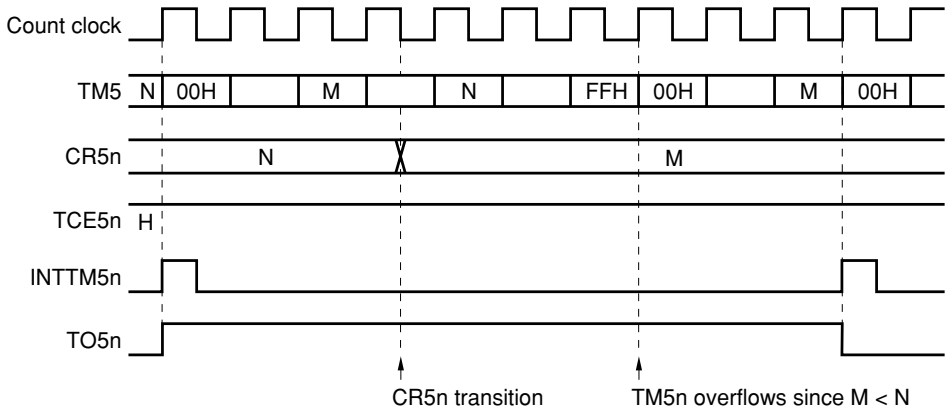
(c) When CR5n = FFH



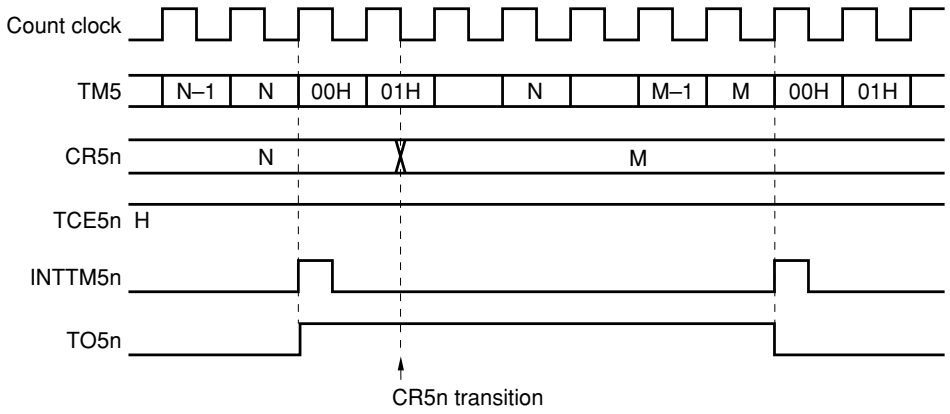
n = 0, 1

Figure 7-7. Interval Timer Operation Timings (3/3)

(d) Operated by CR5n transition (M < N)



(e) Operated by CR5n transition (M > N)



n = 0, 1

7.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI5n by the 8-bit timer counter 5n (TM5n).

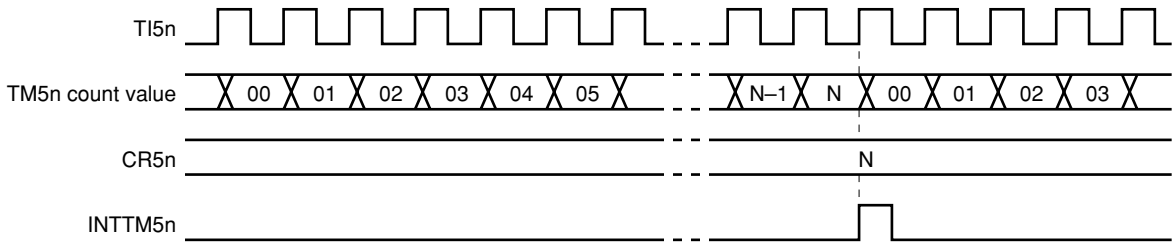
TM5n is incremented each time the valid edge specified with the timer clock select register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n counted values match the values of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and the interrupt request signal (INTTM5n) is generated.

Whenever the TM5n counted value matches the value of CR5n, INTTM5n is generated.

Remark n = 0, 1

Figure 7-8. External Event Counter Operation Timing (with Rising Edge Specified)



N = 00H to FFH

n = 0, 1

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

A square wave with any selected frequency is output at intervals of the value preset to the 8-bit timer compare register 5n (CR5n).

TO5n pin output status is reversed at intervals of the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register.

- Set port latch and port mode register to 0
- TCL5n: Select count clock
- CR5n: Compare value
- TMC5n: Clear and start mode by match of TM5n and CR5n

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F reverse enable
Timer output enable → TOE5n = 1

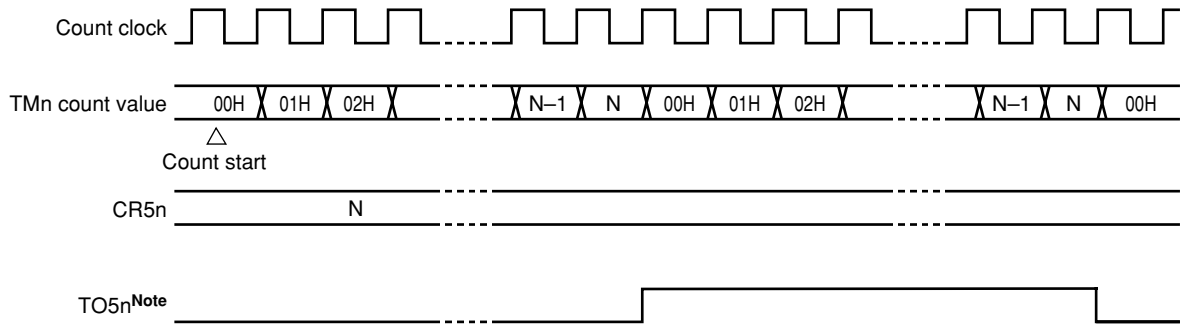
<2> After TCE5n = 1 is set, count operation starts.

<3> Timer output F/F is reversed by match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> Timer output F/F is reversed at the same interval and square wave is output from TO5n.

Remark n = 0, 1

Figure 7-9. Square-Wave Output Operation Timing



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

Remark n = 0, 1

7.4.4 8-bit PWM output operation

8-bit timer/event counter operates as PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty rate pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

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

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

Enable/disable for PWM output can be selected with bit 0 of TMC5n (TOE5n).

Caution Rewrite of CR5n in PWM mode is allowed only once in a cycle.

Remark n = 0, 1

(1) PWM output basic operation

[Setting]

<1> Set port latch (P72, P73) and port mode register 7 (PM72, PM73) to 0.

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

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

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

<5> Count operation starts when bit 7 of TMC5n (TCE5n) is set to 1.
Set TCE5n to 0 to stop count operation.

[PWM output operation]

<1> PWM output (output from TO5n) outputs inactive level after count operation starts until overflow is generated.

<2> When overflow is generated, the active level set in <1> of setting is output.
The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).

<3> After the CR5n matches the count value, PWM output outputs the inactive level again until overflow is generated.

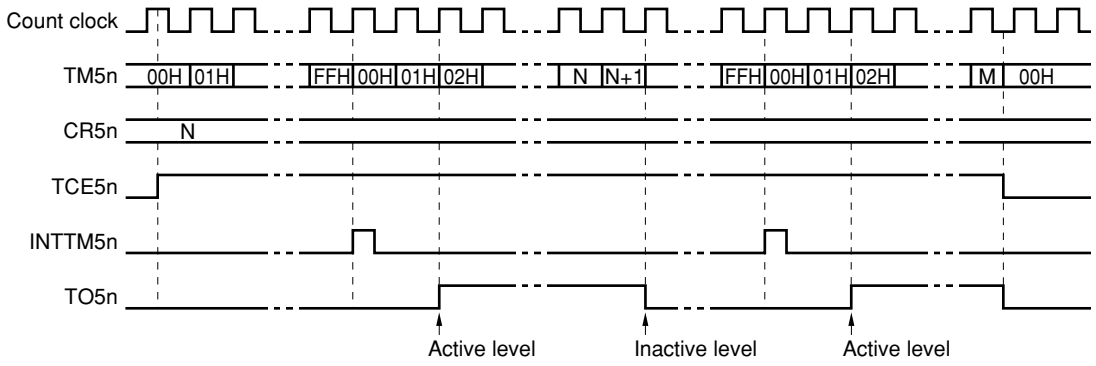
<4> Operations <2> and <3> are repeated until the count operation stops.

<5> When the count operation is stopped with TCE5n = 0, PWM output comes to inactive level.

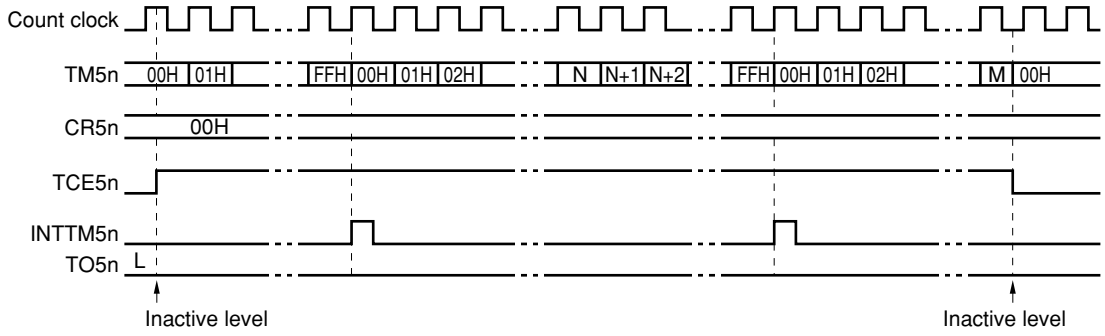
Remark n = 0, 1

Figure 7-10. PWM Output Operation Timing

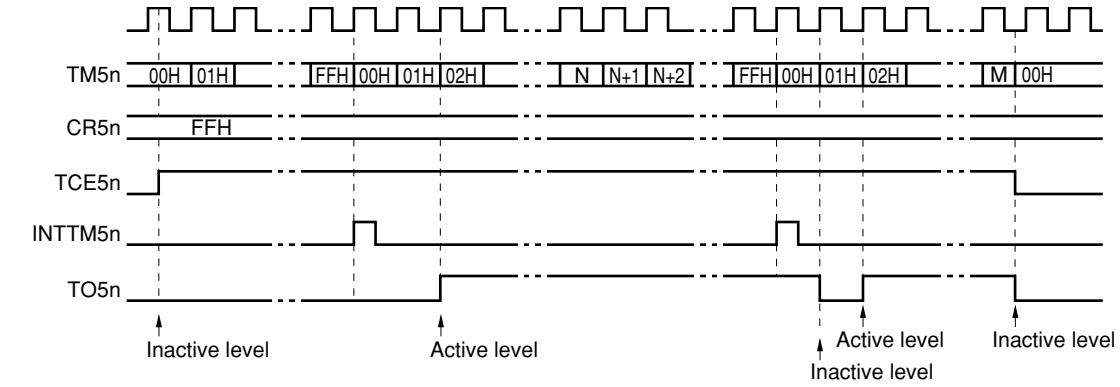
(a) Basic operation (active level = H)



(b) CR5n = 0



(c) CR5n = FFH

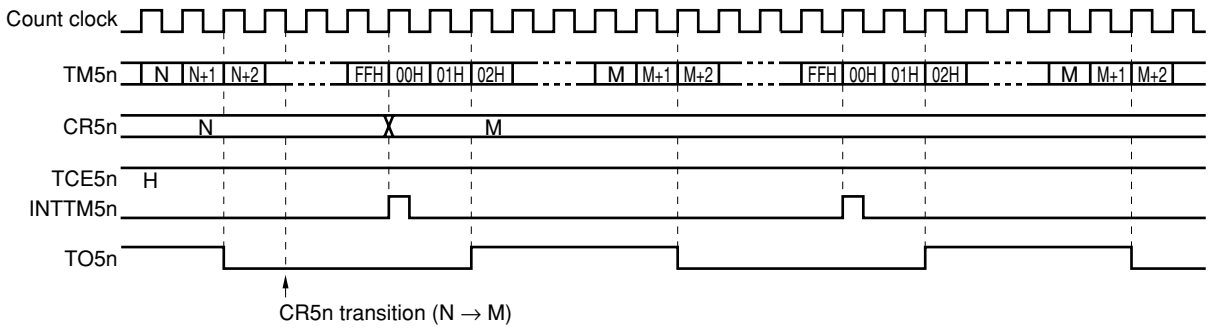


n = 0, 1

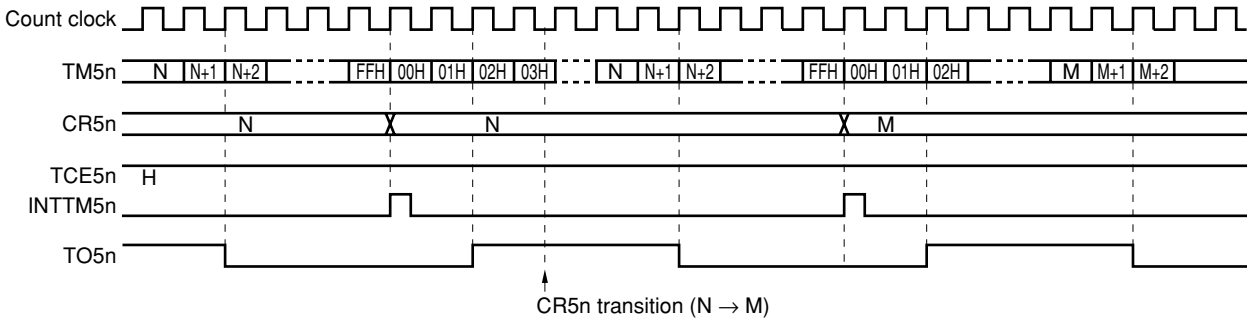
(2) Operated by CR5n transition

Figure 7-11. Timing of Operation by CR5n Transition

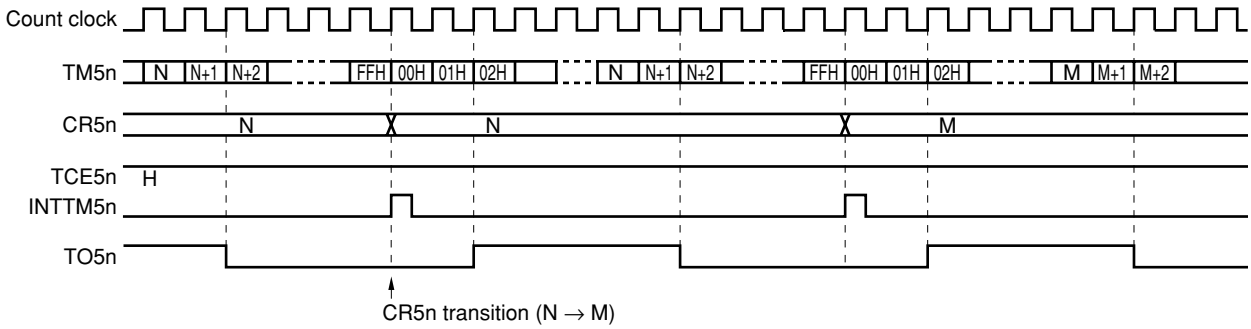
(a) CR5n value transits from N to M before overflow of TM5n



(b) CR5n value transits from N to M after overflow of TM5n



(c) CR5n value transits from N to M between two clocks (00H and 01H) after overflow of TM5n



n = 0, 1

7.4.5 Interval timer (16-bit) operation

When "1" is set in bit 4 (TMC514) of 8-bit timer mode control register 51 (TMC51), the 16-bit resolution timer/counter mode is entered.

The 8-bit timer/event counter operates as an interval timer which generates interrupt requests repeatedly at intervals of the count value preset to the 8-bit timer compare registers (CR50, CR51).

[Setting]

<1> Set each register.

TCL50: Select count clock in TM50.
Cascade-connected TM51 need not be selected.

CR50, CR51: Compare value (each value can be set at 00H to FFH)

TMC50, TMC51: Select the clear & start mode by match of TM50 and CR50 (TM51 and CR51).

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

<2> When TMC51 is set to TCE51 = 1 and then, TMC50 is set to TCE50 = 1, count operation starts.

<3> When the values of TM50 and CR50 of cascade-connected timer match, INTTM50 of TM50 is generated (TM50 and TM51 are cleared to 00H).

<4> INTTM50 generates repeatedly at the same interval.

Cautions 1. Stop timer operation without fail before setting compare register (CR50, CR51).

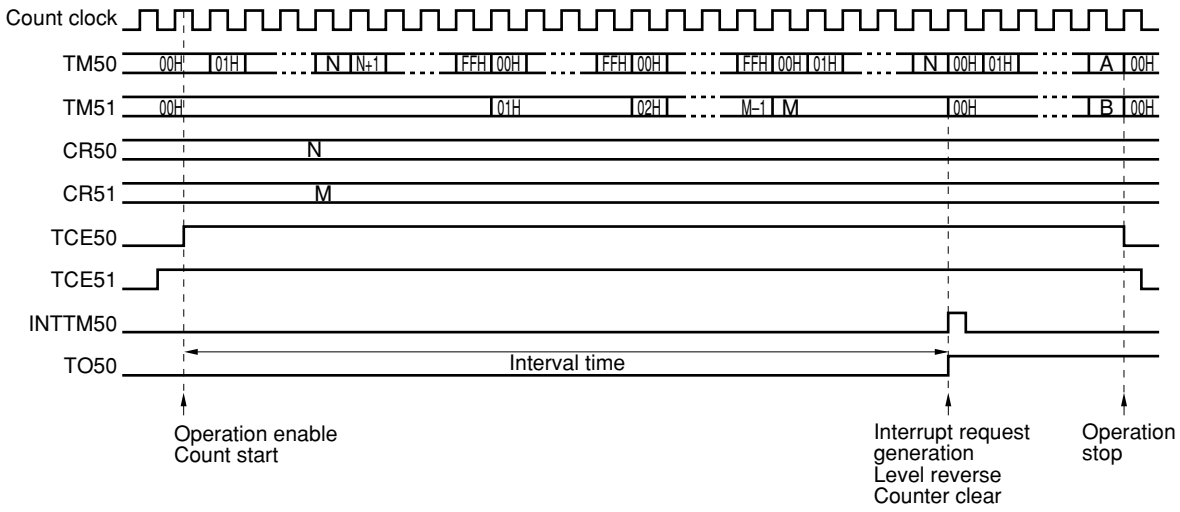
2. INTTM51 of TM51 is generated when TM51 count value matches CR51, even if cascade connection is used. Ensure to mask TM51 to disable interrupt.

3. Set TCE50 and TCE51 in a sequential order of TM51 and TM50.

4. Count restart/stop can only be controlled by setting TCE50 of TM50 to 1/0.

Figure 7-12 shows an example of 16-bit resolution cascade connection mode timing.

Figure 7-12. 16-Bit Resolution Cascade Connection Mode

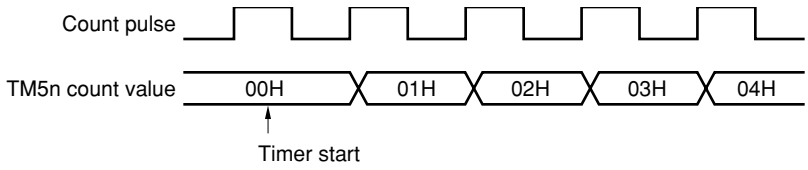


7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(1) Timer start errors

An error with the maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because the 8-bit timer counter 5n (TM5n) is started asynchronously with the count pulse.

Figure 7-13. 8-Bit Timer Counter Start Timing

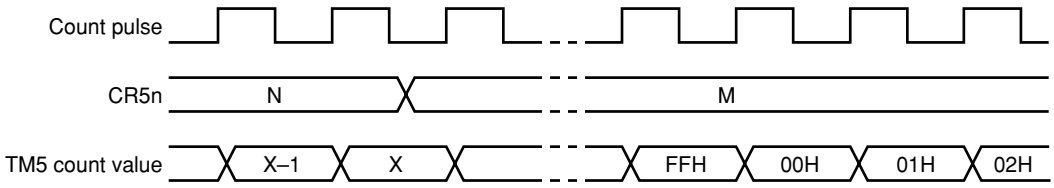


n = 0, 1

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

If the value after the 8-bit timer compare register 5n (CR5n) is changed is smaller than the value of 8-bit timer counter 5n (TM5n), TM5n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR5n is changed is smaller than the value (N) before it was changed, it is necessary to restart the timer after changing CR5n.

Figure 7-14. Timing After Change of Compare Register During Timer Count Operation



Caution Except when the TI5n input is selected, always set TCE5n = 0 before setting the stop state.

Remarks 1. N > X > M

2. n = 0, 1

(3) TM5n (n = 0, 1) reading during timer operation

When reading TM5n during operation, select count clock having high/low level waveform longer than two cycles of CPU clock because count clock stops temporary. For example, in the case where CPU clock (f_{CPU}) is f_X, when the selected count clock is $f_X/4$ or below, it can be read.

Remark n = 0, 1

CHAPTER 8 WATCH TIMER

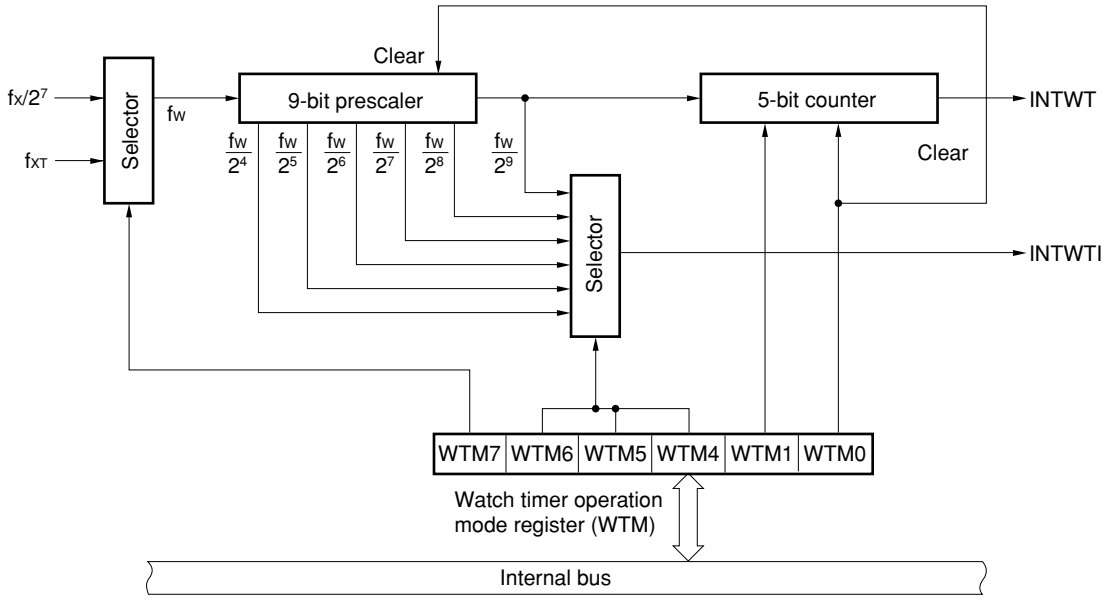
8.1 Functions of Watch Timer

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 8-1 shows the watch timer block diagram.

Figure 8-1. Block Diagram of Watch Timer



(1) Watch timer

When the main system clock or subsystem clock is used, interrupt requests (INTWT) are generated at 2^{14}/fw second intervals.

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

(2) Interval timer

Interrupt requests (INTWTI) are generated at the preset time interval.

Table 8-1. Interval Timer Interval Time

Interval Time		When Operated at f_X = 8.38 MHz	When Operated at f_X = 4.19 MHz	When Operated at f_{XT} = 32.768 kHz
$2^{11} \times 1/f_X$	$2^4 \times 1/f_{XT}$	244 μs	489 μs	488 μs
$2^{12} \times 1/f_X$	$2^5 \times 1/f_{XT}$	489 μs	978 μs	977 μs
$2^{13} \times 1/f_X$	$2^6 \times 1/f_{XT}$	978 μs	1.96 ms	1.95 ms
$2^{14} \times 1/f_X$	$2^7 \times 1/f_{XT}$	1.96 ms	3.91 ms	3.91 ms
$2^{15} \times 1/f_X$	$2^8 \times 1/f_{XT}$	3.91 ms	7.82 ms	7.81 ms
$2^{16} \times 1/f_X$	$2^9 \times 1/f_{XT}$	7.82 ms	15.6 ms	15.6 ms

Remark fx: 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 operation mode register (WTM)

8.3 Register to Control Watch Timer

Watch timer operation mode register (WTM) is a register to control watch timer.

- **Watch timer operation mode register (WTM)**
  This register sets the watch timer count clock, enables/disables operation, prescaler interval time, and 5-bit counter operation control.
  WTM is set by an 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input sets WTM to 00H.

Figure 8-2. Format of Watch Timer Operation Mode Register (WTM)

Address: FF41H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WTM	WTM7	WTM6	WTM5	WTM4	0	0	WTM1	WTM0

WTM7	Watch timer count clock selection
0	$f_X/2^7$ (65.4 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler interval time selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
Other than above			Setting prohibited

WTM1	5-bit counter operation control
0	Clear after operation stop
1	Start

WTM0	Watch timer operation enable
0	Operation stop (clear both prescaler and timer)
1	Operation enable

Caution Do not change the count clock and interval time (by setting bits 4 to 7 (WTM4 to WTM7) of WTM) during watch timer operation.

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$ or f_{XT})
2. f_X: Main system clock oscillation frequency
3. f_{XT}: Subsystem clock oscillation frequency
4. Figures in parentheses apply to operation with f_X = 8.38 MHz, f_{XT} = 32.768 kHz.

8.4 Operations of Watch Timer

8.4.1 Watch timer operation

The watch timer generates an interrupt request (INTWT) at a specific time interval (2^{14}/fw seconds) by using the main system clock or subsystem clock.

The interrupt request is generated at the following time interval.

- If main system clock (8.38 MHz) is selected: 0.25 seconds
- If subsystem clock (32.768 kHz) is selected: 0.5 seconds

The watch timer generates an interrupt request (INTWT) at a specific time interval.

When bit 0 (WTM0) and bit 1 (WTM1) of the watch timer operation mode register (WTM) are set to 1, the count operation starts. When these bits are set to 0, the 5-bit counter is cleared and the count operation stops.

When the interval timer is simultaneously operated, zero-second start can be achieved only for the watch timer by setting WTM1 to 0. In this case, however, the 9-bit prescaler is not cleared. Therefore, an error up to $2^9 \times 1$/fw seconds occurs in the first overflow (INTWT) after zero-second start.

Remark fx: Main system clock oscillation frequency
fxT: Subsystem clock oscillation frequency
fw: Watch timer clock frequency (fx/2^7 or fxT)

8.4.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt requests (INTWTI) repeatedly at an interval of the preset count value.

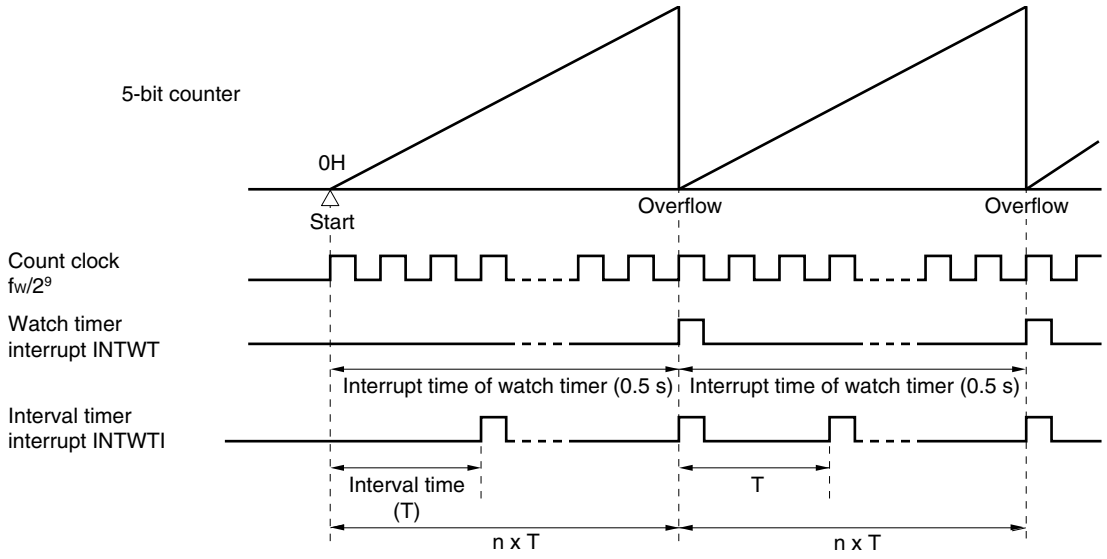
The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer operation mode register (WTM).

Table 8-3. Interval Timer Interval Time

WTM6	WTM5	WTM4	Interval Time	When Operated at fx = 8.38 MHz	When Operated at fx = 4.19 MHz	When Operated at fxT = 32.768 kHz
0	0	0	$2^4 \times 1$/fw	244 μs	489 μs	488 μs
0	0	1	$2^5 \times 1$/fw	489 μs	978 μs	977 μs
0	1	0	$2^6 \times 1$/fw	978 μs	1.96 ms	1.95 ms
0	1	1	$2^7 \times 1$/fw	1.96 ms	3.91 ms	3.91 ms
1	0	0	$2^8 \times 1$/fw	3.91 ms	7.82 ms	7.81 ms
1	0	1	$2^9 \times 1$/fw	7.82 ms	15.6 ms	15.6 ms
Other than above			Setting prohibited			

Remark fx: Main system clock oscillation frequency
fxT: Subsystem clock oscillation frequency
fw: Watch timer clock frequency (fx/2^7 or fxT)

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



Caution When operation of the watch timer and 5-bit counter is enabled by the watch timer mode control register (WTM) (by setting bits 0 (WTM0) and 1 (WTM1) of WTM to 1), the interval until the first interrupt request (INTWT) is generated after the register is set does not exactly match the specification made with bit 3 (WTM3) of WTM. This is because there is a delay of one 9-bit prescaler output cycle until the 5-bit counter starts counting. Subsequently, however, the INTWT signal is generated at the specified intervals.

Remark fw: Watch timer clock frequency

n: The number of times of interval timer operations

Figures in parentheses are for operation with fw = 32.768 kHz

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

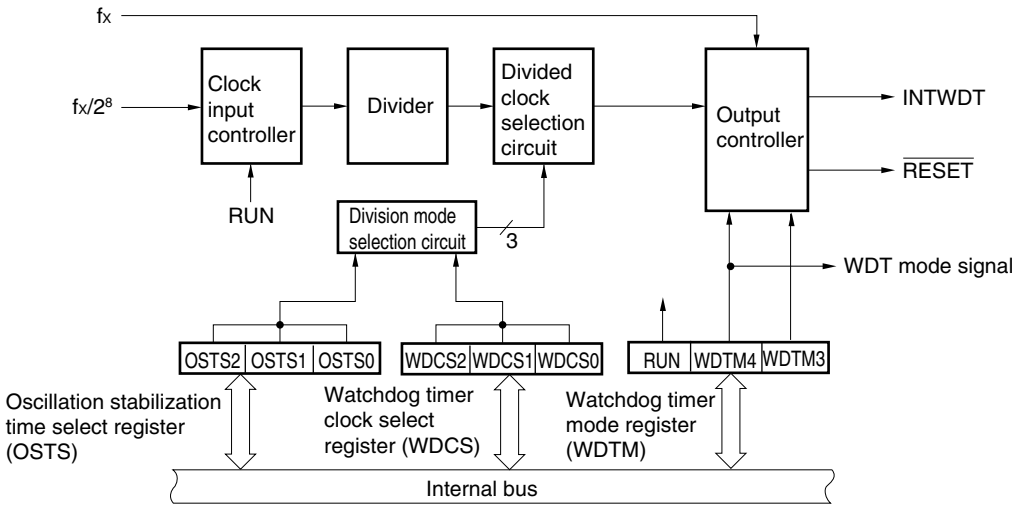
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Oscillation stabilization time selection

Caution Select the watchdog timer mode or the interval timer mode with the watchdog timer mode register (WDTM) (The watchdog timer and the interval timer cannot be used simultaneously).

Figure 9-1 shows a block diagram of the watchdog timer.

Figure 9-1. Block Diagram of Watchdog Timer



(1) Watchdog timer mode

A program loop is detected. Upon detection of the program loop, a non-maskable interrupt request or $\overline{\text{RESET}}$ can be generated.

Table 9-1. Watchdog Timer Program Loop Detection Time

Program Loop Detection Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 9-2. Interval Time

Interval Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz

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	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time select register (OSTS)

9.3 Registers to Control Watchdog Timer

The following three types of registers are used to control the watchdog timer.

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)
- Oscillation stabilization time select register (OSTS)

(1) Watchdog timer clock select register (WDCS)

This register sets overflow time of the watchdog timer and the interval timer.

WDCS is set by an 8-bit memory manipulation instruction.

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

Figure 9-2. Format of Watchdog Timer Clock Select Register (WDCS)

Address: FF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer
0	0	0	$2^{12}/f_X$ (489 μs)
0	0	1	$2^{13}/f_X$ (978 μs)
0	1	0	$2^{14}/f_X$ (1.96 ms)
0	1	1	$2^{15}/f_X$ (3.91 ms)
1	0	0	$2^{16}/f_X$ (7.82 ms)
1	0	1	$2^{17}/f_X$ (15.6 ms)
1	1	0	$2^{18}/f_X$ (31.3 ms)
1	1	1	$2^{20}/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) Watchdog timer mode register (WDTM)

This register sets the watchdog timer operating mode and enables/disables counting.

WDTM is set by 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 (WDTM)

Address: FFF9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Watchdog timer operation mode selection[Note 1]
0	Count stop
1	Counter is cleared and counting starts

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	×	Interval timer mode[Note 3] (Maskable interrupt request occurs upon generation of an overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow)

Notes 1. Once set to 1, RUN cannot be cleared to 0 by software.
Thus, once counting starts, it can only be stopped by $\overline{\text{RESET}}$ input.

2. Once set to 1, WDTM3 and WDTM4 cannot be cleared to 0 by software.

3. The watchdog timer starts operations as the interval timer when 1 is set to RUN.

Caution When 1 is set to RUN so that the watchdog timer is cleared, the actual overflow time is up to $2^8/f_X$ seconds shorter than the time set by watchdog timer clock select register (WDCS).

Remark ×: don't care

(3) Oscillation stabilization time select register (OSTS)

A register to select oscillation stabilization time from reset time or STOP mode released time to the time when oscillation is stabilized.

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H. Thus, when releasing the STOP mode by $\overline{\text{RESET}}$ input, the time required to release is 2^{17}/fx.

Figure 9-4. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFAH After reset: 04H R/W

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	2^{12}/fx (488 μs)
0	0	1	2^{14}/fx (1.95 ms)
0	1	0	2^{15}/fx (3.91 ms)
0	1	1	2^{16}/fx (7.81 ms)
1	0	0	2^{17}/fx (15.6 ms)
Other than the above			Setting prohibited

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz

9.4 Watchdog Timer Operations

9.4.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer is operated to detect any program loops.

The program loop detection time interval is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set program loop time interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the program loop detection time is exceeded, system reset or a non-maskable interrupt request is generated according to WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions 1. The actual program loop detection time may be shorter than the set time by a maximum of 2^8/fx seconds.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 9-4. Watchdog Timer Program Loop Detection Time

Program Loop Detection Time
$2^{12} \times 1/fx$ (489 μs)
$2^{13} \times 1/fx$ (978 μs)
$2^{14} \times 1/fx$ (1.96 ms)
$2^{15} \times 1/fx$ (3.91 ms)
$2^{16} \times 1/fx$ (7.82 ms)
$2^{17} \times 1/fx$ (15.6 ms)
$2^{18} \times 1/fx$ (31.3 ms)
$2^{20} \times 1/fx$ (125 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz.

9.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at an interval of the preset count value when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0.

The interval time of interval timer is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). When 1 is set to bit 7 (RUN) of WDTM, the watchdog timer operates as the interval timer.

When the watchdog timer operated as the interval timer, the interrupt mask flag (WDTMK) and priority specify flag (WDTPR) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, INTWDT has the highest priority at default.

The interval timer continues operating in the HALT mode but it stops in STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (this selects the watchdog timer mode), the interval timer mode is not set unless $\overline{\text{RESET}}$ input is applied.

2. The interval time just after setting by WDTM may be shorter than the set time by a maximum of 2^8/fx seconds.

3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 9-5. Interval Timer Interval Time

Interval Time
$2^{12} \times 1/fx$ (489 μs)
$2^{13} \times 1/fx$ (978 μs)
$2^{14} \times 1/fx$ (1.96 ms)
$2^{15} \times 1/fx$ (3.91 ms)
$2^{16} \times 1/fx$ (7.82 ms)
$2^{17} \times 1/fx$ (15.6 ms)
$2^{18} \times 1/fx$ (31.3 ms)
$2^{20} \times 1/fx$ (125 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz.

CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

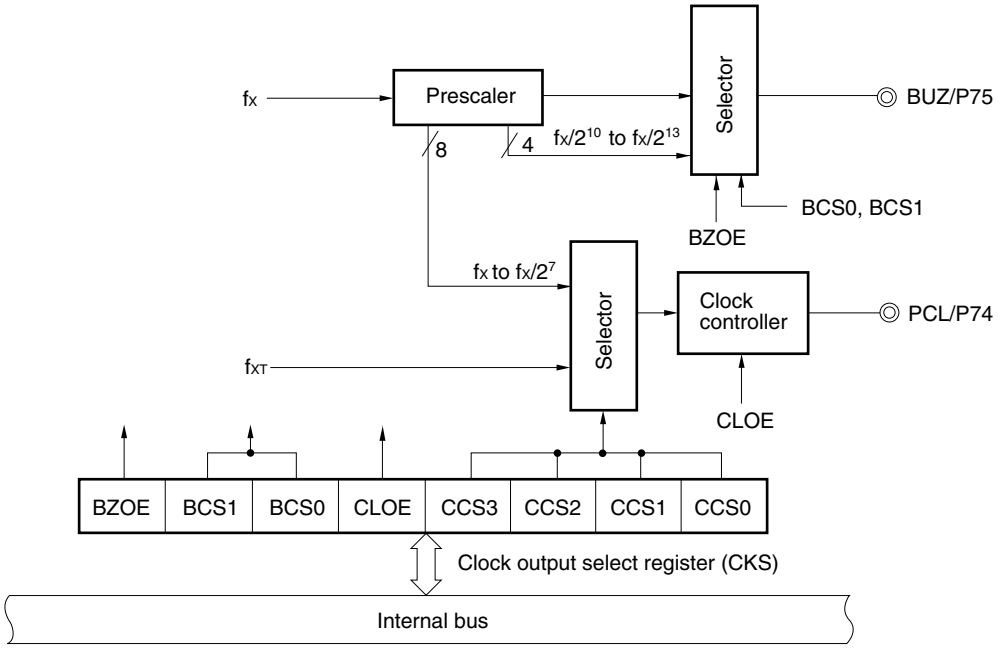
10.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output select register (CKS) is output.

In addition, the buzzer output is intended for square wave output of buzzer frequency selected with CKS.

Figure 10-1 shows the block diagram of clock output/buzzer output controller.

Figure 10-1. Block Diagram of Clock Output/Buzzer Output Controller



10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller consists of the following hardware.

Table 10-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select register (CKS) Port mode register (PM7)[Note]

Note See **Figure 4-12 Block Diagram of P74 and P75**.

10.3 Registers to Control Clock Output/Buzzer Output Controller

The following two types of registers are used to control the clock output/buzzer output controller.

- Clock output select register (CKS)
- Port mode register (PM7)

(1) Clock output select register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

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

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

Figure 10-2. Format of Clock Output Select Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Stop clock division circuit operation. BUZ fixed to low level.
1	Enable clock division circuit operation. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection
0	0	$f_X/2^{10}$ (8.18 kHz)
0	1	$f_X/2^{11}$ (4.09 kHz)
1	0	$f_X/2^{12}$ (2.04 kHz)
1	1	$f_X/2^{13}$ (1.02 kHz)

CLOE	PCL output enable/disable specification
0	Stop clock division circuit operation. PCL fixed to low level.
1	Enable clock division circuit operation. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	0	f_X (8.38 MHz)
0	0	0	1	$f_X/2$ (4.19 MHz)
0	0	1	0	$f_X/2^2$ (2.09 MHz)
0	0	1	1	$f_X/2^3$ (1.04 MHz)
0	1	0	0	$f_X/2^4$ (524 kHz)
0	1	0	1	$f_X/2^5$ (262 kHz)
0	1	1	0	$f_X/2^6$ (131 kHz)
0	1	1	1	$f_X/2^7$ (65.5 kHz)
1	0	0	0	f_{XT} (32.768 kHz)
Other than above				Setting prohibited

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P74/PCL pin for clock output and the P75/BUZ pin for buzzer output, set PM74, PM75 and the output latch of P74, P75 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM7 to FFH.

Figure 10-3. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

10.4 Operations of Clock Output/Buzzer Output Controller

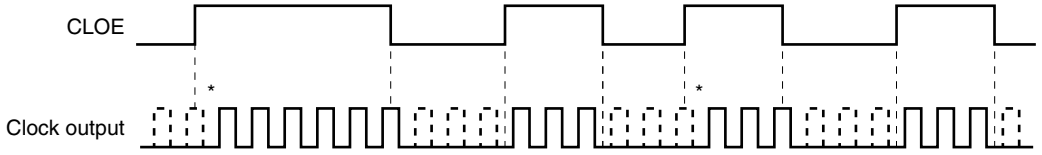
10.4.1 Operation as clock output

The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output select register (CKS) (clock pulse output in disabled status).
<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 10-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after securing high level of the clock.

Figure 10-4. Remote Control Output Application Example



10.4.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output select register (CKS) (buzzer output in disabled status).
<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

CHAPTER 11 8-BIT A/D CONVERTER (μPD780024AS SUBSERIES)

11.1 Functions of A/D Converter

A/D converter is an 8-bit resolution converter that converts analog inputs into digital values. It can control up to 4 analog input channels (ANI0 to ANI3).

(1) Hardware start

Conversion is started by trigger input (ADTRG: rising edge, falling edge, or both rising and falling edges can be specified).

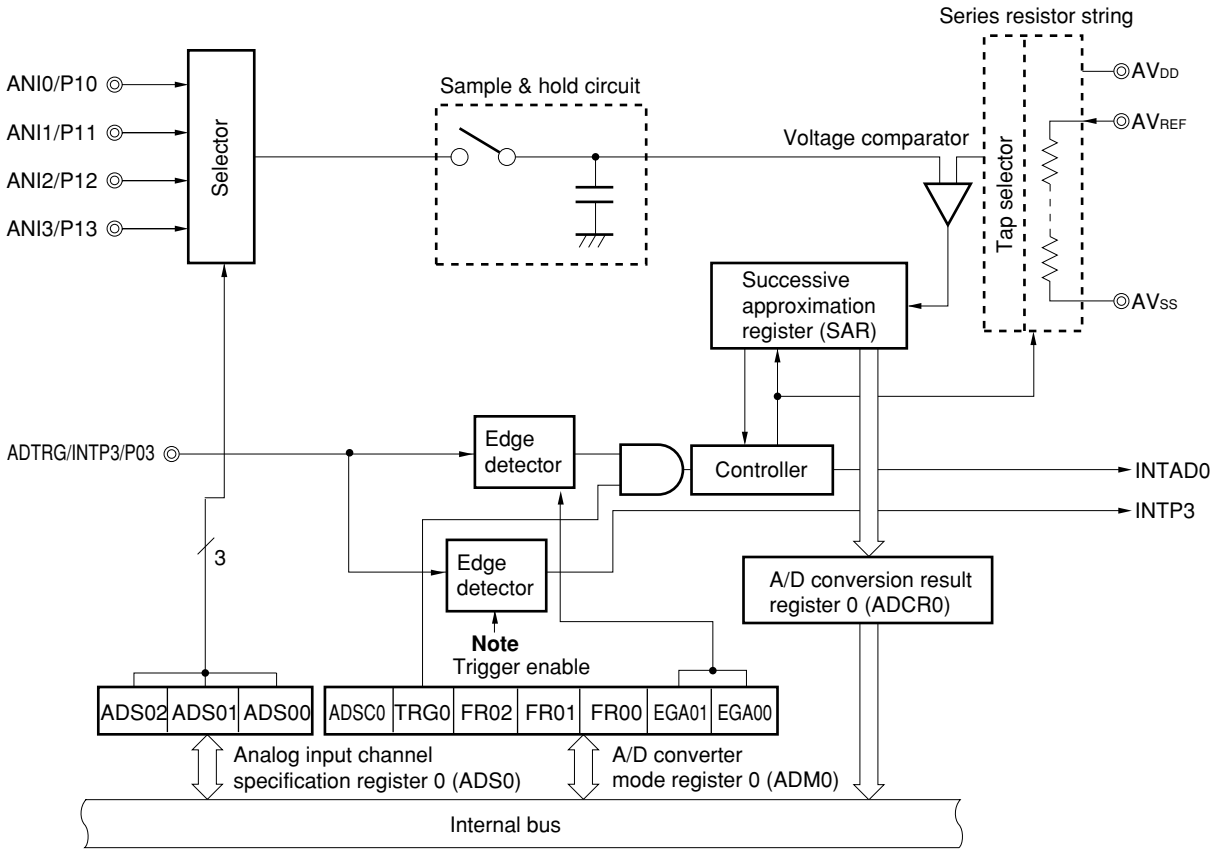
(2) Software start

Conversion is started by setting the A/D converter mode register 0 (ADM0).

Select one channel for analog input from ANI0 to ANI3 to perform A/D conversion. In the case of hardware start, A/D conversion stops when an A/D conversion operation ends and an interrupt request (INTAD0) is generated. In the case of software start, A/D conversion is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD0) is generated.

Caution Although the μPD78F0034BS incorporate a 10-bit A/D converter, this converter can be operated as an 8-bit A/D converter by using the device file DF780024.

Figure 11-1. Block Diagram of 8-Bit A/D Converter



Note The valid edge is specified by bit 3 of the EGP and EGN registers (see **Figure 11-4 Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)**).

11.2 Configuration of A/D Converter

The A/D converter consists of the following hardware.

Table 11-1. Configuration of A/D Converter

Item	Configuration
Analog input	4 channels (ANI0 to ANI3)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) Analog input channel specification register 0 (ADS0) External interrupt rising edge enable register (EGP) External interrupt falling edge enable register (EGN)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is held (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register 0 (ADCR0).

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

The ADCR0 is an 8-bit register that stores the A/D conversion result. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

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

$\overline{\text{RESET}}$ input sets ADCR0 to 00H.

Caution When writing is performed to the A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI3 pins

These are four analog input pins to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI3 are alternate-function pins that can also be used for digital input.

Cautions 1. Use ANI0 to ANI3 input voltages within the specification range. If a voltage higher than AV_{REF} or lower than AV_{SS} is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

2. Analog input (ANI0 to ANI3) pins are alternate function pins that can also be used as input port (P10 to P13) pins. When A/D conversion is performed by selecting any one of ANI0 to ANI3, do not execute any input instruction to port 1 during conversion. It may cause the lower conversion resolution.

When a digital pulse is applied to a pin adjacent to the pin in the process of A/D conversion, A/D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply any pulse to a pin adjacent to the pin in the process of A/D conversion.

(7) AV_{REF} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI3 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.

Caution A series resistor string of several 10 kΩ is connected between the AV_{REF} pin and AV_{SS} pin. Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

(8) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always keep it at the same potential as the V_{SS0} or V_{SS1} pin even when not using the A/D converter.

(9) AV_{DD} pin

This is the A/D converter analog power supply pin. Always keep it at the same potential as the V_{DD0} or V_{DD1} pin even when not using the A/D converter.

11.3 Registers to Control A/D Converter

The following 4 types of registers are used to control the A/D converter.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register 0 (ADS0)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)

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

This register sets the conversion time for analog input to be A/D converted, conversion start/stop, and external trigger.

ADM0 is set by 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 (ADM0)

Address: FF80H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	0

ADCS0	A/D conversion operation control
0	Stop conversion operation.
1	Enable conversion operation.

TRG0	Software start/hardware start selection
0	Software start
1	Hardware start

FR02	FR01	FR00	Conversion time selection[Note 1]
0	0	0	144/fx (17.1 μs)
0	0	1	120/fx (14.3 μs)
0	1	0	96/fx (Setting prohibited[Note 2])
1	0	0	72/fx (Setting prohibited[Note 2])
1	0	1	60/fx (Setting prohibited[Note 2])
1	1	0	48/fx (Setting prohibited[Note 2])
Other than above			Setting prohibited

EGA01	EGA00	External trigger signal, edge specification
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Both falling and rising edge detection

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because A/D conversion time is less than 14 μs.

Caution When rewriting FR00 to FR02 to other than the same data, stop A/D conversion operations once prior to performing rewrite.

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz.

(2) Analog input channel specification register 0 (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set by an 8-bit memory manipulation instruction.

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

Figure 11-3. Format of Analog Input Channel Specification Register 0 (ADS0)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	0[Note]	ADS01	ADS00

ADS01	ADS00	Analog input channel specification
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

Note Be sure to set bit 2 to 0.

(3) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP3.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets EGP and EGN to 00H.

Figure 11-4. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 3)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

11.4 Operations of A/D Converter

11.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion with the analog input channel specification register 0 (ADS0).

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

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Bit 7 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset.

<6> Next, bit 6 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 7, as described below.

- Bit 7 = 1: (3/4) AV_{REF}
- Bit 7 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated as follows.

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

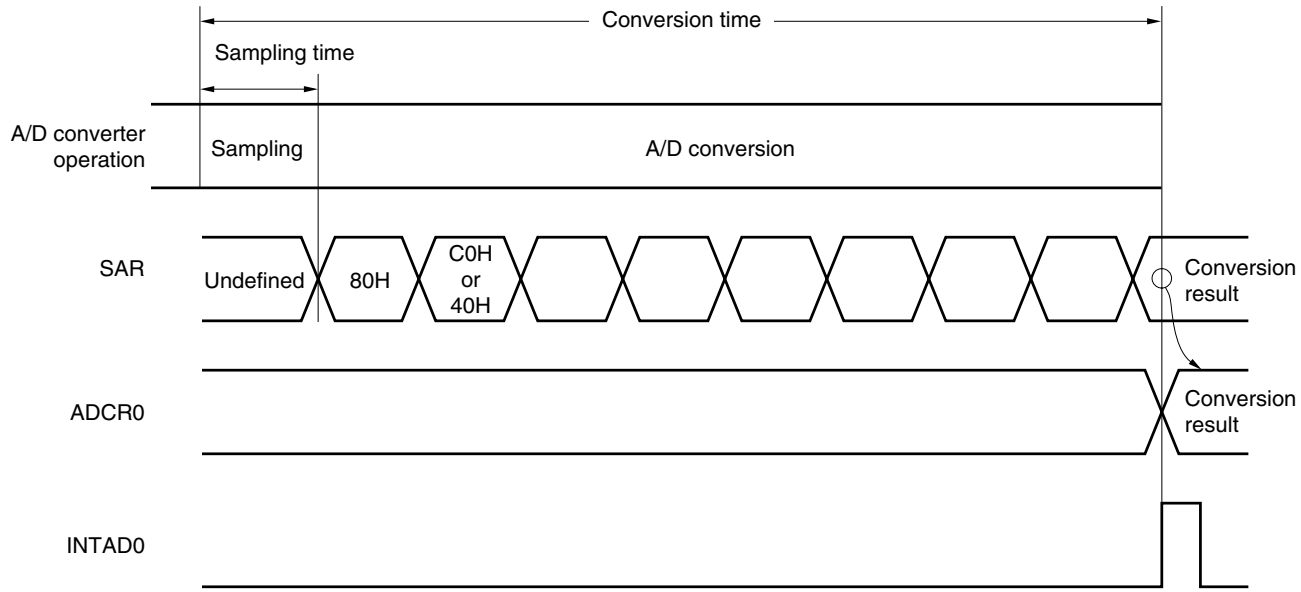
<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 8 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in the A/D conversion result register 0 (ADCR0).

At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.

Caution The first A/D conversion value just after A/D conversion operations start may not fall within the rating.

Figure 11-5. Basic Operation of 8-Bit A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed to the ADM0 or the analog input channel specification register 0 (ADS0) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS0 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register 0 (ADCR0) to 0000H.

11.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3) and the A/D conversion result (stored in the A/D conversion result register 0 (ADCR0)) is shown by the following expression.

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

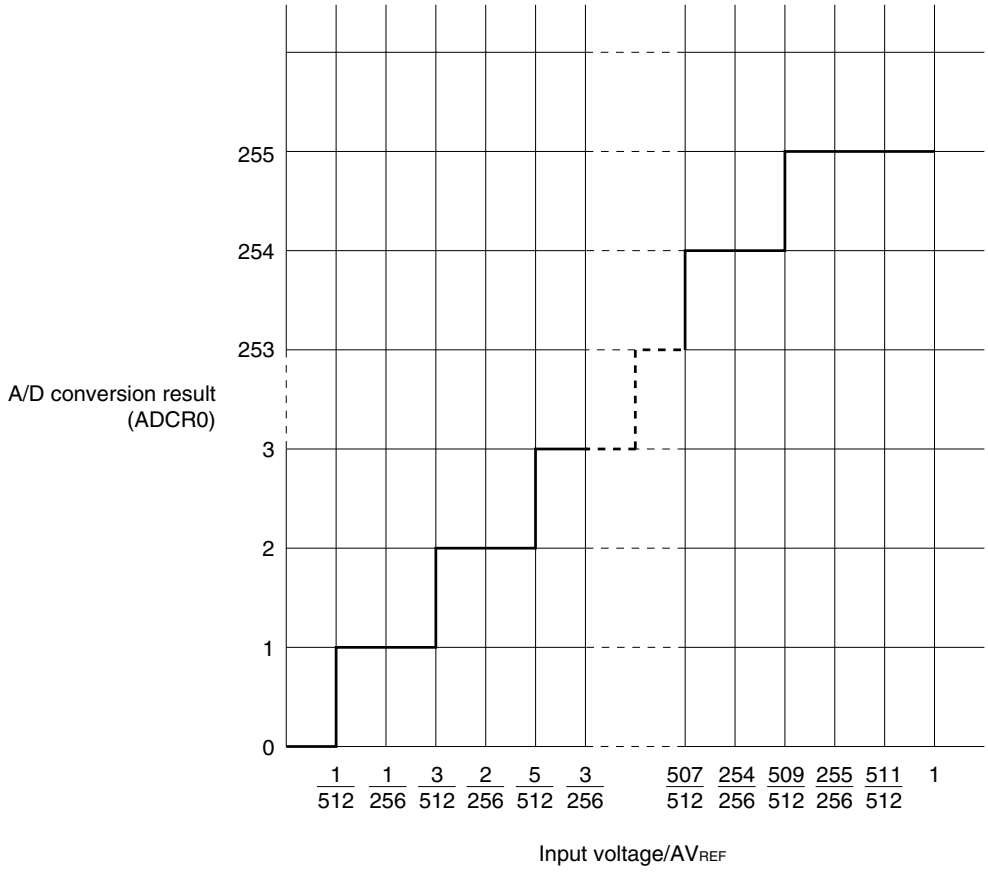
or

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

where, INT( ): Function which returns integer part of value in parentheses
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR0: A/D conversion result register 0 (ADCR0) value

Figure 11-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-6. Relationship Between Analog Input Voltage and A/D Conversion Result



11.4.3 A/D converter operation mode

Select one analog input channel from among ANI0 to ANI3 by the analog input channel specification register 0 (ADS0) to start A/D conversion.

A/D conversion can be started in either of the following two ways.

- Hardware start: Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges specified).
- Software start: Conversion is started by specifying the A/D converter mode register 0 (ADM0).

The A/D conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is simultaneously generated.

(1) A/D conversion by hardware start

When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) are set to 1, the A/D conversion standby state is set. When the external trigger signal (ADTRG) is input, A/D conversion of the voltage applied to the analog input pins specified by the analog input channel specification register 0 (ADS0) starts.

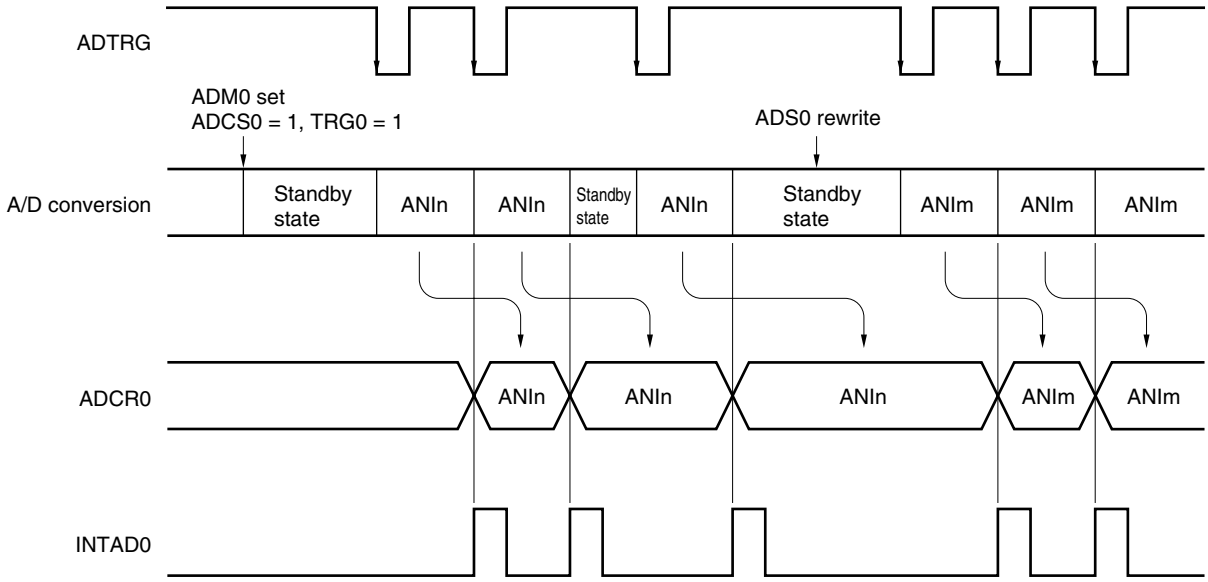
Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is not started until a new external trigger signal is input.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning. If ADS0 is rewritten during A/D conversion waiting, A/D conversion starts when the following external trigger input signal is input.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Caution When P03/INTP3/ADTRG is used as the external trigger input (ADTRG), specify the valid edge by bits 1 and 2 (EGA00, EGA01) of the A/D converter mode register 0 (ADM0) and set the interrupt mask flag (PMK3) to 1.

Figure 11-7. A/D Conversion by Hardware Start (When Falling Edge Is Specified)



Remarks 1. n = 0, 1, ..., 3

2. m = 0, 1, ..., 3

(2) A/D conversion by software start

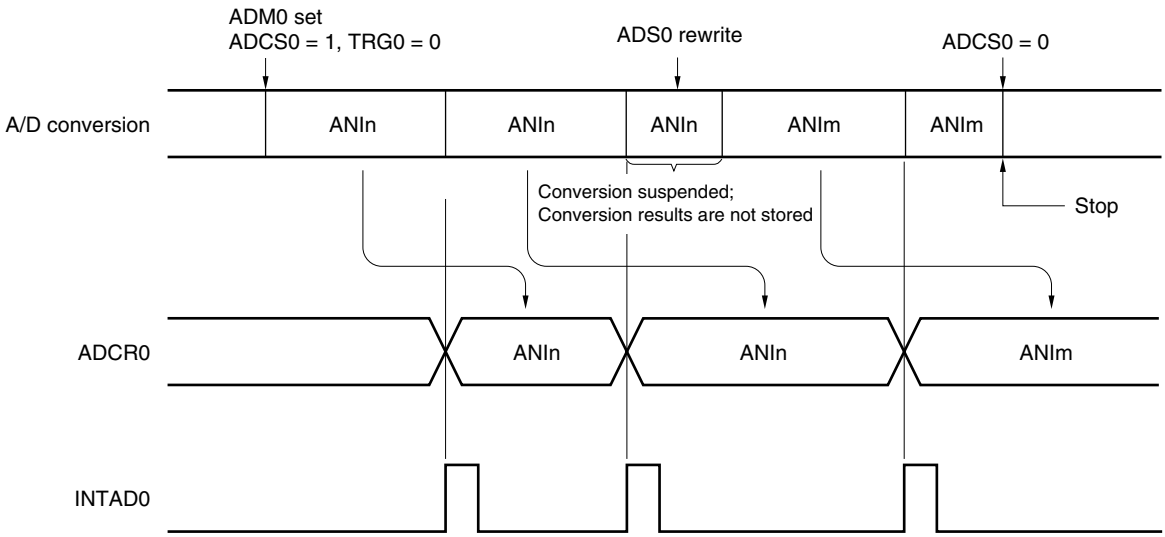
When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) are set to 0 and 1, respectively, A/D conversion of the voltage applied to the analog input pin specified by the analog input channel specification register 0 (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS0.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion and A/D conversion of the newly selected analog input channel is started.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 11-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 3

2. m = 0, 1, ..., 3

11.5 How to Read A/D Converter Characteristics Table

Here we will explain the special terms unique to A/D converters.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per 1 bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

When the resolution is 8 bits,

$1LSB = 1/2^8 = 1/256$
$= 0.4\%FSR$

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

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero scale offset, full scale offset, integral linearity error, differential linearity error and errors which are combinations of these express overall error.
Furthermore, quantization error is not included in overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, there naturally occurs a ±1/2LSB error. In an A/D converter, an analog input voltage in a range of ±1/2LSB are converted to the same digital code, so a quantization error cannot be avoided.
Furthermore, it is not included in the overall error, zero scale offset, full scale offset, integral linearity error, and differential linearity error in the characteristics table.

Figure 11-9. Overall Error

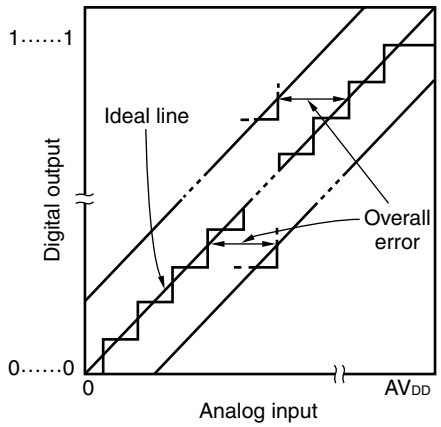
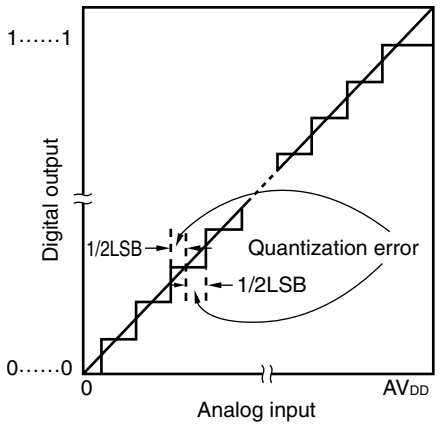


Figure 11-10. Quantization Error



(4) Zero scale offset

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001. If the actual measured value is greater than the theoretical value, it shows the difference between the actual measured value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0.....001 to 0.....010.

(5) Full scale offset

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (full scale –3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measured value and the ideal straight line when the zero scale offset and full scale offset are 0.

(7) Differential linearity error

Although the ideal output width for a given code is 1LSB, this value shows the difference between the actual measured value and the ideal value of the width when outputting a particular code.

Figure 11-11. Zero Scale Offset

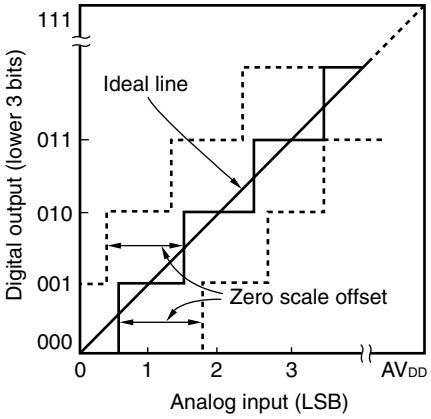


Figure 11-12. Full Scale Offset

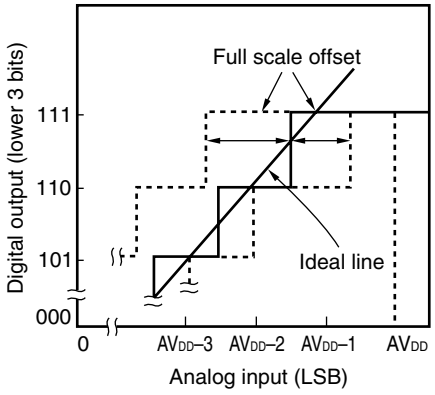


Figure 11-13. Integral Linearity Error

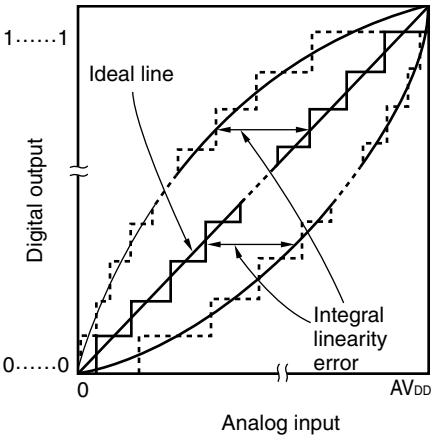
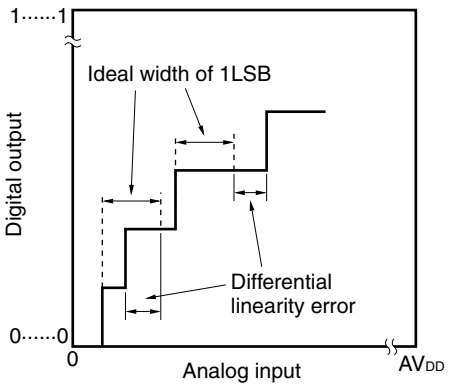


Figure 11-14. Differential Linearity Error



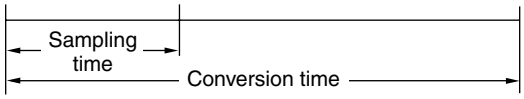
(8) Conversion time

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

Sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



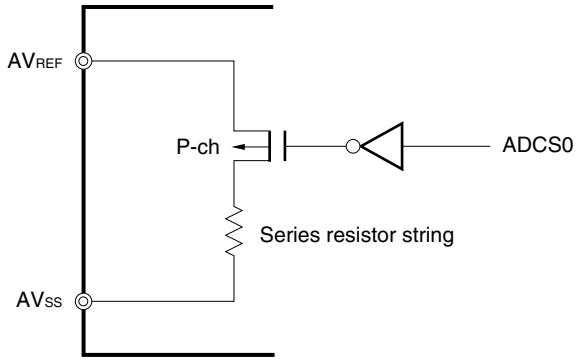
11.6 Cautions for A/D Converter

(1) Current consumption in standby mode

A/D converter stops operating in the standby mode. At this time, current consumption can be reduced by stopping the conversion operation (by setting bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) to 0).

Figure 11-15 shows how to reduce the current consumption in the standby mode.

Figure 11-15. Example of Method of Reducing Current Consumption in Standby Mode



(2) Input range of ANI0 to ANI3

The input voltages of ANI0 to ANI3 should be within the specification range. In particular, if a voltage higher than AV_{REF} or lower than AV_{SS} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Contending operations

<1> Contention between A/D conversion result register 0 (ADCR0) write and ADCR0 read by instruction upon the end of conversion

ADCR0 read is given priority. After the read operation, the new conversion result is written to ADCR0.

<2> Contention between ADCR0 write and external trigger signal input upon the end of conversion

The external trigger signal is not accepted during A/D conversion. Therefore, the external trigger signal is not accepted during ADCR0 write.

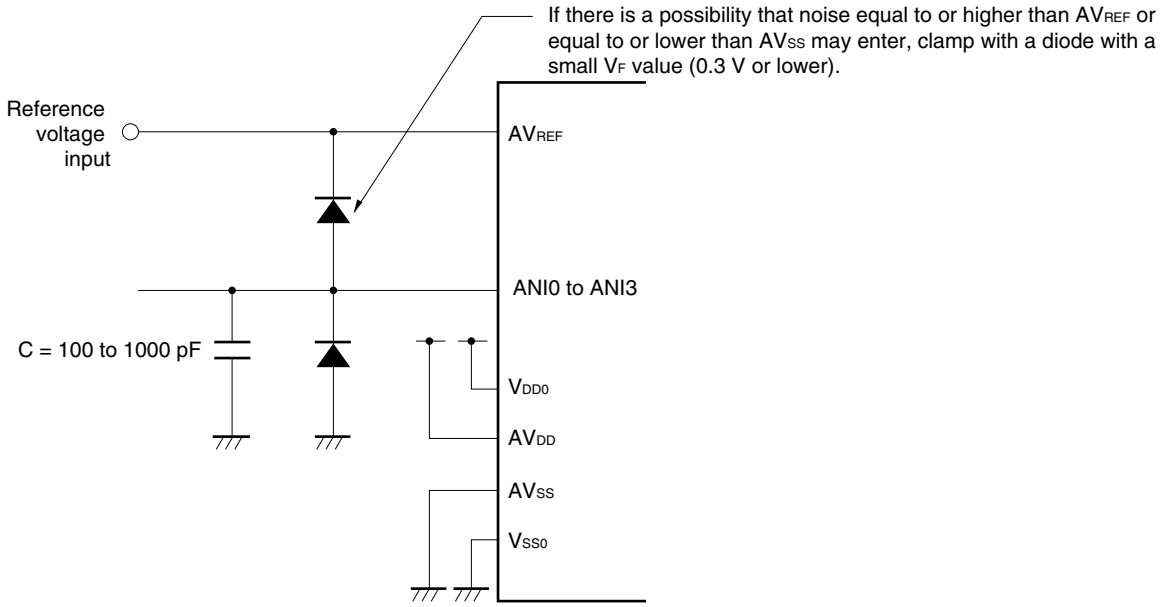
<3> Contention between ADCR0 write and A/D converter mode register 0 (ADM0) write or analog input channel specification register 0 (ADS0) write upon the end of conversion

ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD0) generated.

(4) Noise countermeasures

To maintain the 8-bit resolution, attention must be paid to noise input to pin AV_{REF} and pins ANI0 to ANI3. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 11-16 to reduce noise.

Figure 11-16. Analog Input Pin Connection



(5) ANI0 to ANI3

The analog input pins (ANI0 to ANI3) also function as input port pins (P10 to P13).

When A/D conversion is performed with any of pins ANI0 to ANI3 selected, do not execute an input instruction to port 1 while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to a pin adjacent to the pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

(6) AV_{REF} pin input impedance

A series resistor string of several 10 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

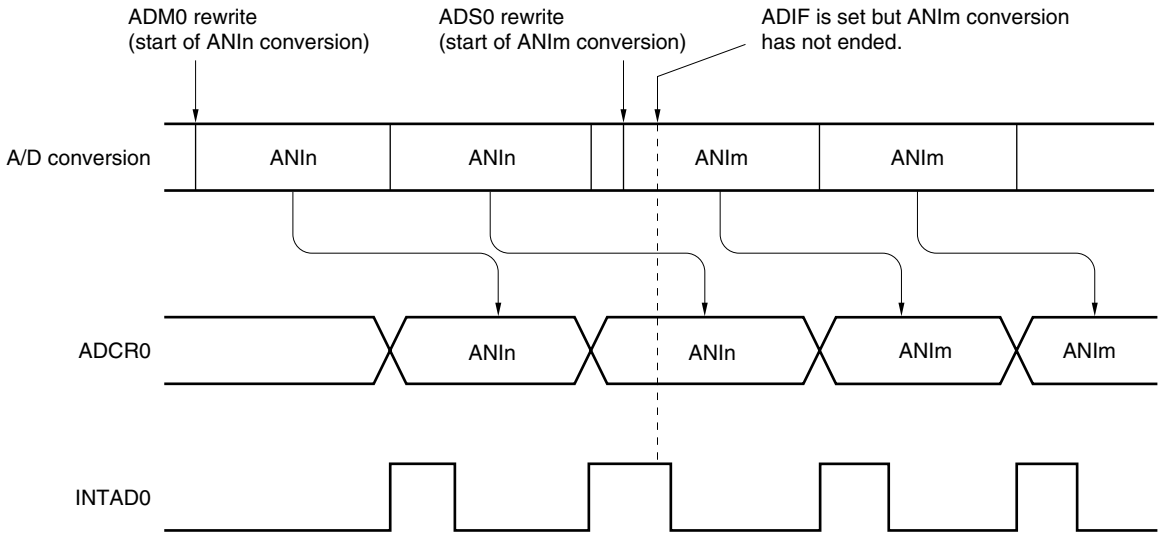
(7) Interrupt request flag (ADIF0)

The interrupt request flag (ADIF0) is not cleared even if the analog input channel specification register 0 (ADS0) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS0 rewrite. Caution is therefore required since, at this time, when ADIF0 is read immediately just after the ADS0 rewrite, ADIF0 is set despite the fact that the A/D conversion for the post-change analog input has not ended.

When the A/D conversion is restarted after it is stopped, clear ADIF0 before restart.

Figure 11-17. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 3

2. m = 0, 1, ..., 3

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

The first A/D conversion value just after A/D conversion operations start may not fall within the rating. Polling A/D conversion end interrupt request (INTAD0) and take measures such as removing the first conversion results.

(9) A/D conversion result register 0 (ADCR0) read operation

When writing is performed to the A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Timing at which A/D conversion result is undefined

The A/D conversion value may be undefined if the timing of completion of A/D conversion and the timing of stopping the A/D conversion conflict with each other. Therefore, read the A/D conversion result during the A/D conversion operation. To read the conversion result after stopping the A/D conversion operation, be sure to stop the A/D conversion before the next conversion ends.

Figures 11-18 and 11-19 show the timing of reading the conversion result.

Figure 11-18. Timing of Reading Conversion Result (When Conversion Result Is Undefined)

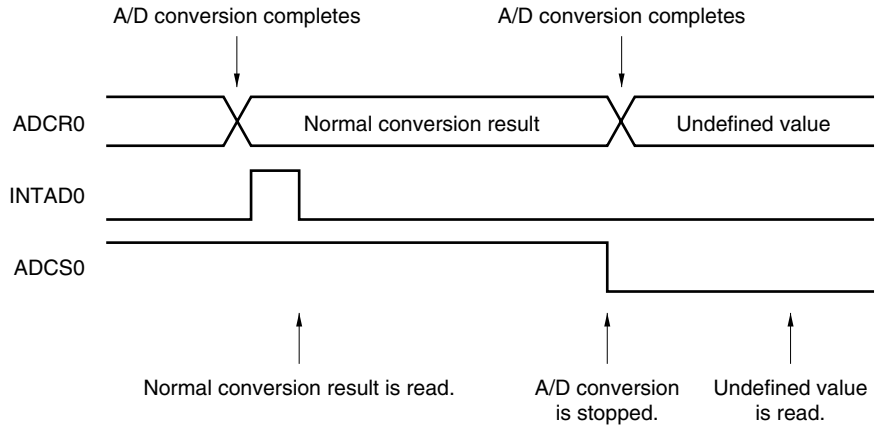
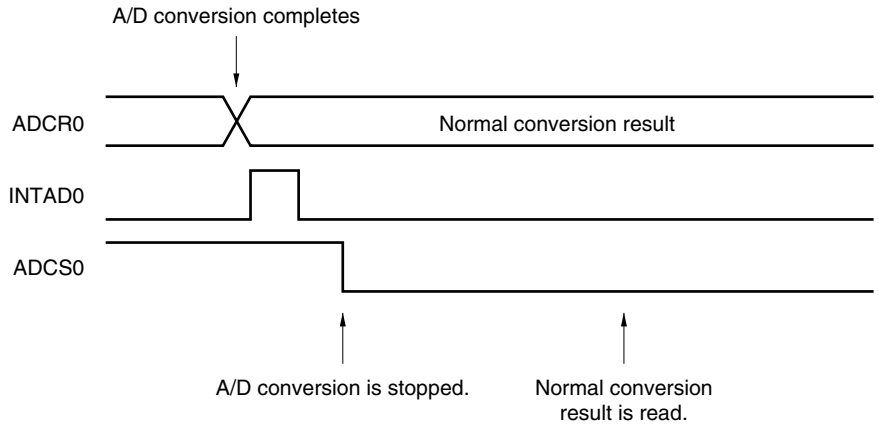


Figure 11-19. Timing of Reading Conversion Result (When Conversion Result Is Normal)



(11) Notes on board design

Locate analog circuits as far away from digital circuits as possible on the board because the analog circuits may be affected by the noise of the digital circuits. In particular, do not cross an analog signal line with a digital signal line, or wire an analog signal line in the vicinity of a digital signal line. Otherwise, the A/D conversion characteristics may be affected by the noise of the digital line.

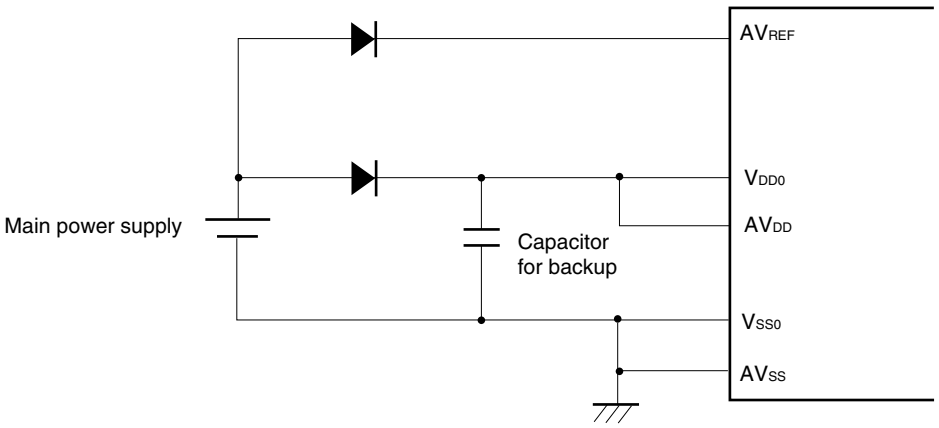
Connect AV_{SS0} and V_{SS0} at one location on the board where the voltages are stable.

(12) AV_{DD} pin

The AV_{DD} pin is the analog circuit power supply pin. It supplies power to the input circuits of the ANI0 to ANI3 pins.

Therefore, be sure to apply the same potential as V_{DD0} to this pin even for applications designed to switch to a backup battery for power supply.

Figure 11-20. AV_{DD} Pin Connection

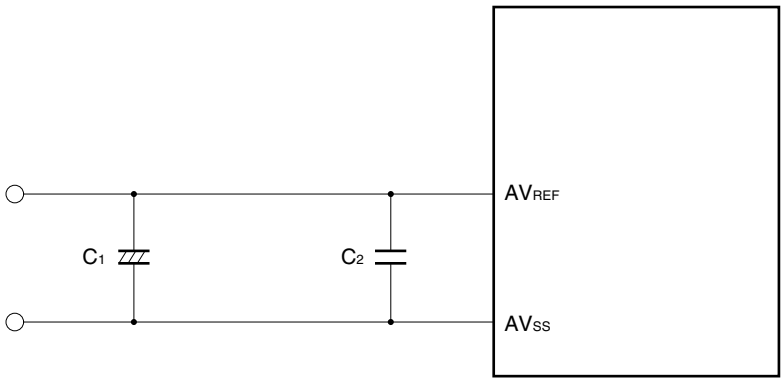


(13) AV_{REF} pin

Connect a capacitor to the AV_{REF} pin to minimize conversion errors due to noise. If an A/D conversion operation has been stopped and then is started, the voltage applied to the AV_{REF} pin becomes unstable, causing the accuracy of the A/D conversion to drop. To prevent this, also connect a capacitor to the AV_{REF} pin.

Figure 11-21 shows an example of connecting a capacitor.

Figure 11-21. Example of Connecting Capacitor to AV_{REF} Pin



Remark C1: 4.7 μF to 10 μF (reference value)
C2: 0.01 μF to 0.1 μF (reference value)
Connect C2 as close to the pin as possible.

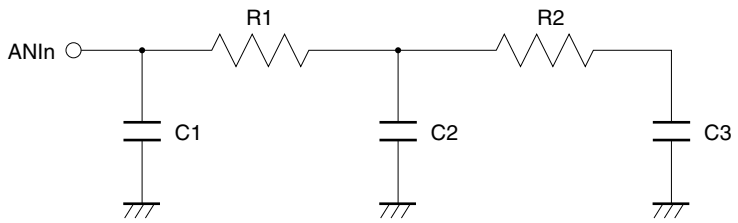
(14) Internal equivalent circuit of ANI0 to ANI3 pins and permissible signal source impedance

To complete sampling within the sampling time with sufficient A/D conversion accuracy, the impedance of the signal source such as a sensor must be sufficiently low. Figure 11-22 shows the internal equivalent circuit of the ANI0 to ANI3 pins.

If the impedance of the signal source is high, connect capacitors with a high capacitance to the pins ANI0 to ANI3. An example of this is shown in Figure 11-23. In this case, however, the microcontroller cannot follow an analog signal with a high differential coefficient because a lowpass filter is created.

To convert a high-speed analog signal or to convert an analog signal in the scan mode, insert a low-impedance buffer.

Figure 11-22. Internal Equivalent Circuit of Pins ANI0 to ANI3



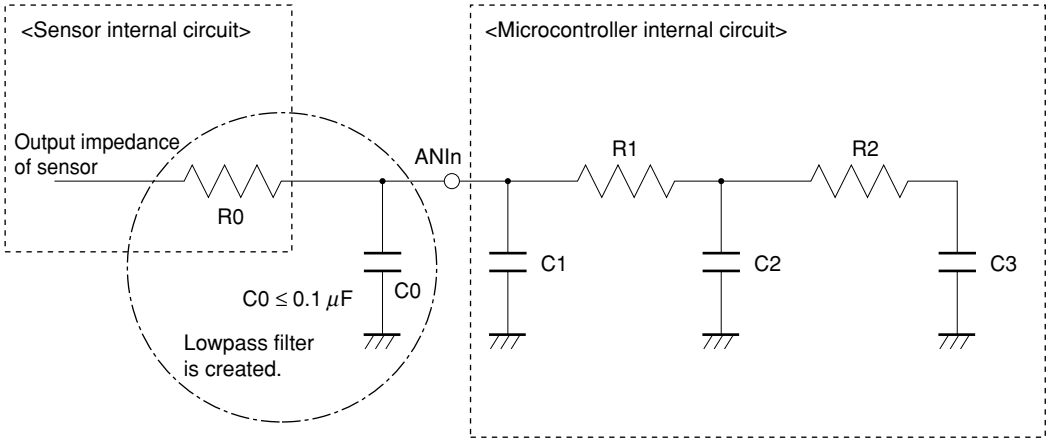
Remark n = 0 to 3

Table 11-2. Resistance and Capacitance of Equivalent Circuit (Reference Values)

AV_{REF}	R1	R2	C1	C2	C3
1.8 V	75 kΩ	30 kΩ	8 pF	4 pF	2 pF
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF

Caution The resistance and capacitance in Table 11-2 are not guaranteed values.

Figure 11-23. Example of Connection If Signal Source Impedance Is High



Remark n = 0 to 3

12.1 Functions of A/D Converter

A/D converter is a 10-bit resolution converter that converts analog inputs into digital signals. It can control up to 4 analog input channels (ANI0 to ANI3).

(1) Hardware start

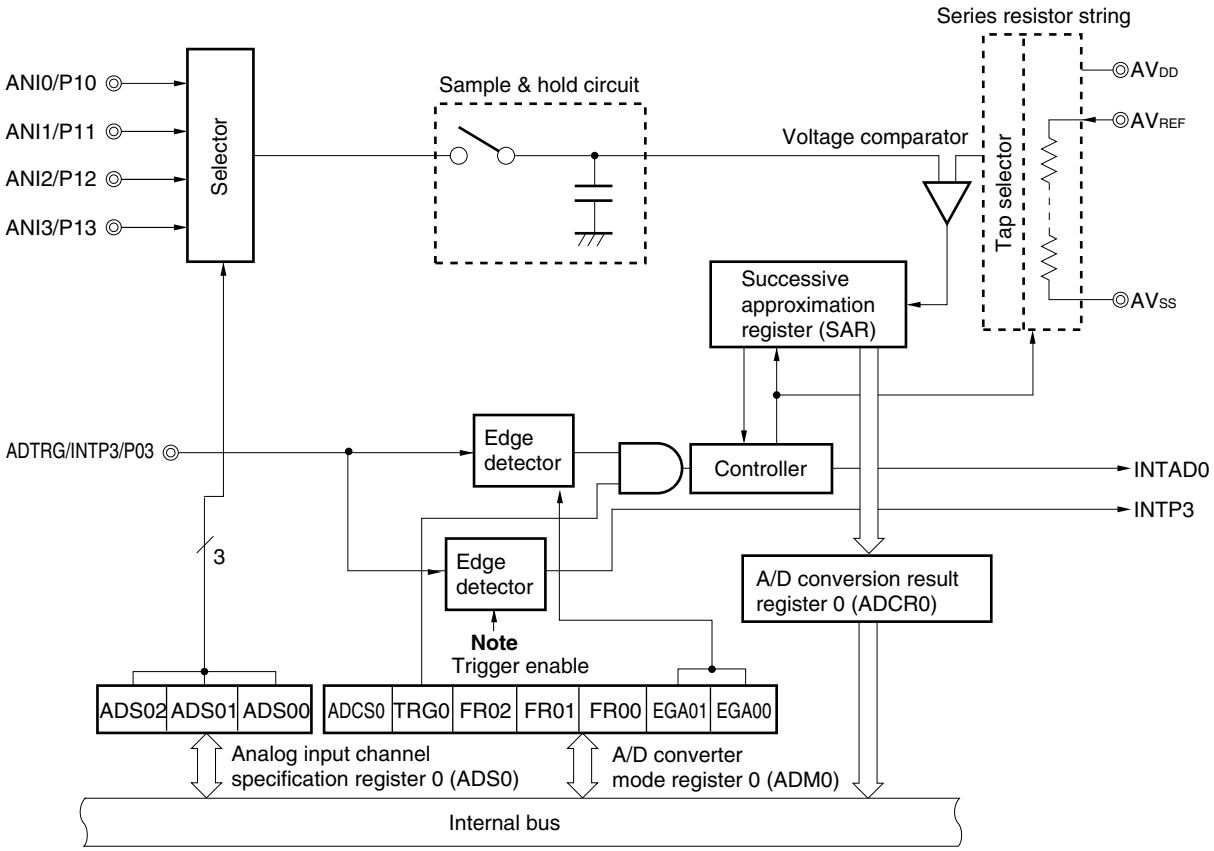
Conversion is started by trigger input (ADTRG: rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

Conversion is started by setting the A/D converter mode register 0 (ADM0).

Select one channel for analog input from ANI0 to ANI3 to start A/D conversion. In the case of hardware start, the A/D converter stops when A/D conversion is completed, and an interrupt request (INTAD0) is generated. In the case of software start, A/D conversion is repeated. Each time as A/D conversion operation ends, an interrupt request (INTAD0) is generated.

Figure 12-1. Block Diagram of 10-Bit A/D Converter



Note The valid edge is specified by bit 3 of the EGP and EGN registers (see **Figure 12-4 Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)**).

12.2 Configuration of A/D Converter

A/D converter consists of the following hardware.

Table 12-1. Configuration of A/D Converter

Item	Configuration
Analog input	4 channels (ANI0 to ANI3)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) Analog input channel specification register 0 (ADS0) External interrupt rising edge enable register (EGP) External interrupt falling edge enable register (EGN)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is hold (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register 0 (ADCR0).

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

The ADCR0 is a 16-bit register that stores the A/D conversion results. Lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

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

$\overline{\text{RESET}}$ input sets ADCR0 to 00H.

Caution When writing is performed to the A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI3 pins

These are four analog input pins to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI3 are alternate-function pins that can also be used for digital input.

Cautions 1. Use ANI0 to ANI3 input voltages within the specification range. If a voltage higher than AV_{REF} or lower than AV_{SS} is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

2. Analog input (ANI0 to ANI3) pins are alternate function pins that can also be used as input port (P10 to P13) pins. When A/D conversion is performed by selecting any one of ANI0 to ANI3, do not execute any input instruction to port 1 during conversion. It may cause the lower conversion resolution.

When a digital pulse is applied to a pin adjacent to the pin in the process of A/D conversion, A/D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply any pulse to a pin adjacent to the pin in the process of A/D conversion.

(7) AV_{REF} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI3 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.

Caution A series resistor string of several 10 kΩ is connected between the AV_{REF} and AV_{SS} pins. Therefore, when output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

(8) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always keep it at the same potential as the V_{SS0} or V_{SS1} pin when not using the A/D converter.

(9) AV_{DD} pin

This is the A/D converter analog power supply pin. Always keep it at the same potential as the V_{DD0} or V_{DD1} pin even when not using the A/D converter.

12.3 Registers to Control A/D Converter

The following 4 types of registers are used to control the A/D converter.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register 0 (ADS0)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)

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

This register sets the conversion time for analog input to be A/D converted, conversion start/stop, and external trigger.

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

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

Figure 12-2. Format of A/D Converter Mode Register 0 (ADM0)

Address: FF80H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	0

ADCS0	A/D conversion operation control
0	Stop conversion operation.
1	Enable conversion operation.

TRG0	Software start/hardware start selection
0	Software start
1	Hardware start

FR02	FR01	FR00	Conversion time selection[Note 1]
0	0	0	144/fx (17.1 μs)
0	0	1	120/fx (14.3 μs)
0	1	0	96/fx (Setting prohibited[Note 2])
1	0	0	72/fx (Setting prohibited[Note 2])
1	0	1	60/fx (Setting prohibited[Note 2])
1	1	0	48/fx (Setting prohibited[Note 2])
Other than above			Setting prohibited

EGA01	EGA00	External trigger signal, edge specification
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Both falling and rising edge detection

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because A/D conversion time is less than 14 μs.

Caution When rewriting FR00 to FR02 to other than the same data, stop A/D conversion operations once prior to performing rewrite.

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz.

(2) Analog input channel specification register 0 (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set by an 8-bit memory manipulation instruction.

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

Figure 12-3. Format of Analog Input Channel Specification Register 0 (ADS0)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	0[Note]	ADS01	ADS00

ADS01	ADS00	Analog input channel specification
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

Note Be sure to set bit 2 to 0.

(3) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP3.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets EGP and EGN to 00H.

Figure 12-4. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 3)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

12.4 Operations of A/D Converter

12.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion with the analog input channel specification register 0 (ADS0).

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

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset.

<6> Next, bit 8 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

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

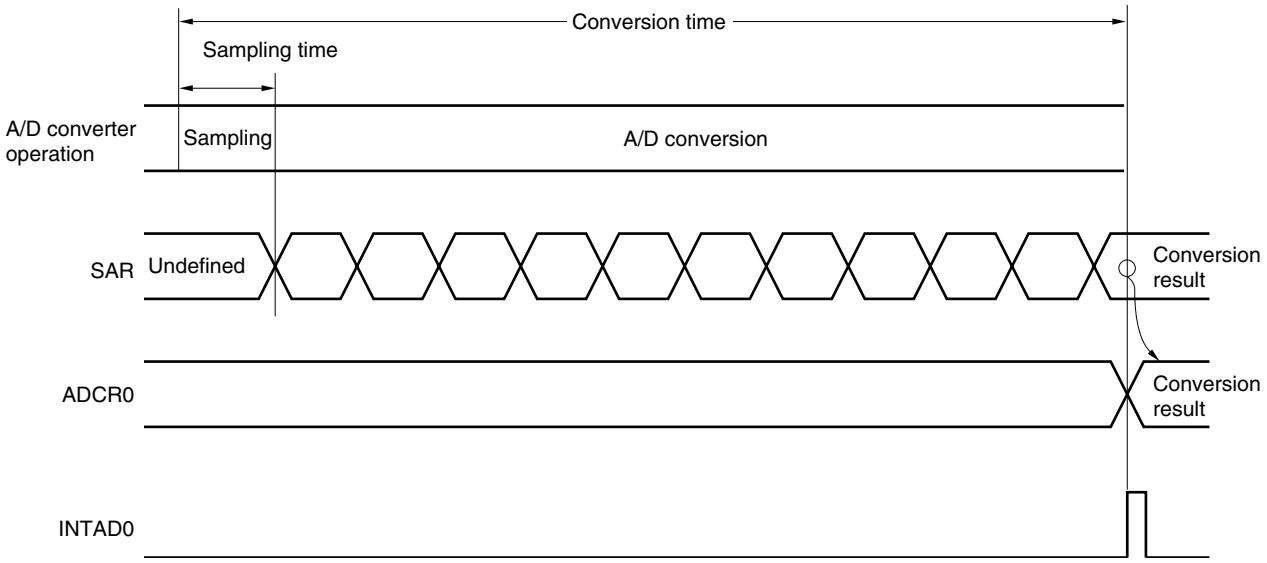
<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in the A/D conversion result register 0 (ADCR0).

At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.

Caution The first A/D conversion value just after A/D conversion operations start may not fall within the rating.

Figure 12-5. Basic Operation of 10-Bit A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed to the ADM0 or the analog input channel specification register 0 (ADS0) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS0 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register 0 (ADCR0) to 0000H.

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (stored in the A/D conversion result register 0 (ADCR0)) is shown by the following expression.

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

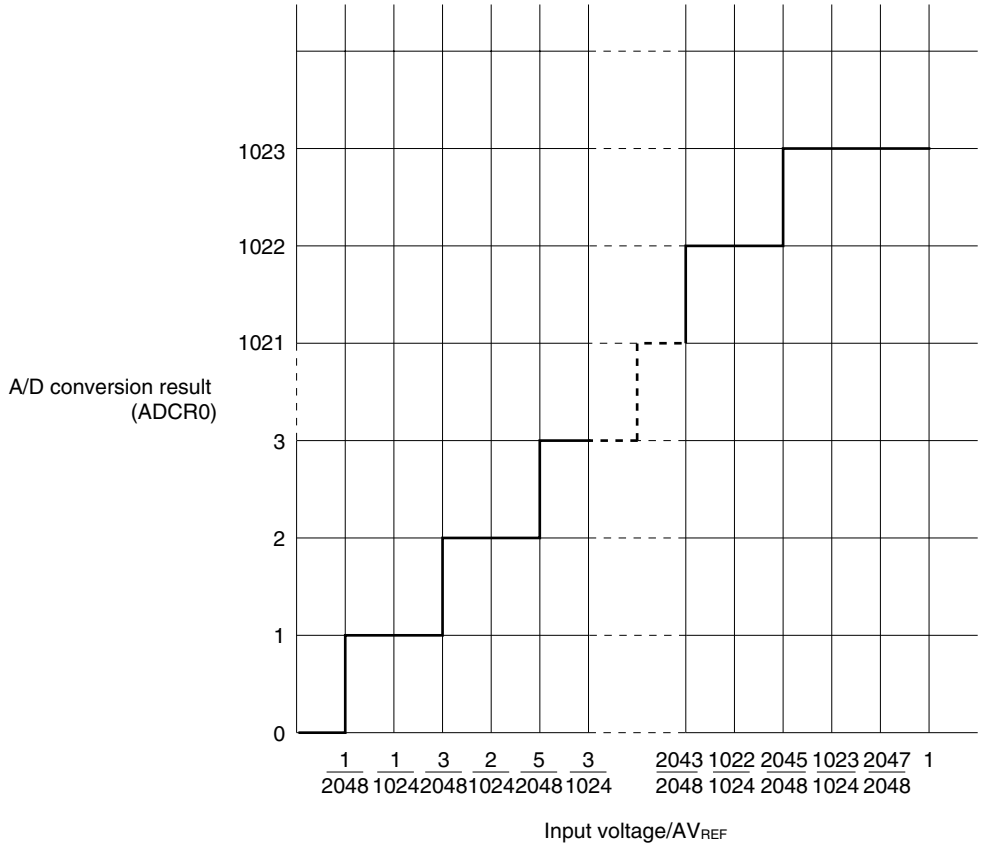
or

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

where, INT( ): Function which returns integer part of value in parentheses
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR0: A/D conversion result register 0 (ADCR0) value

Figure 12-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-6. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operation mode

Select one analog input channel from among ANI0 to ANI3 by the analog input channel specification register 0 (ADS0) to start A/D conversion.

A/D conversion can be started in either of the following two ways.

- Hardware start: Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges specified).
- Software start: Conversion is started by specifying the A/D converter mode register 0 (ADM0).

The A/D conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is simultaneously generated.

(1) A/D conversion by hardware start

When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) are set to 1, the A/D conversion standby state is set. When the external trigger signal (ADTRG) is input, A/D conversion of the voltage applied to the analog input pin specified by the analog input channel specification register 0 (ADS0) starts.

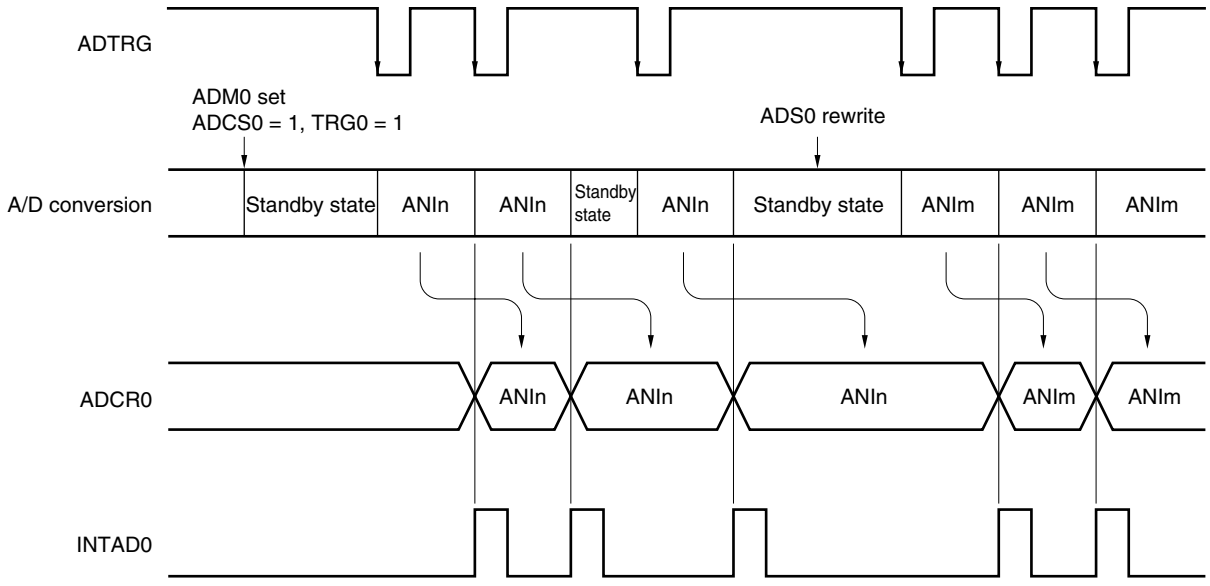
Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is not started until a new external trigger signal is input.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning. If ADS0 is rewritten during A/D conversion waiting, A/D conversion starts when the following external trigger input signal is input.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Caution When P03/INTP3/ADTRG is used as the external trigger input (ADTRG), specify the valid edge by bits 1 and 2 (EGA00, EGA01) of the A/D converter mode register 0 (ADM0) and set the interrupt mask flag (PMK3) to 1.

Figure 12-7. A/D Conversion by Hardware Start (When Falling Edge Is Specified)



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(2) A/D conversion by software start

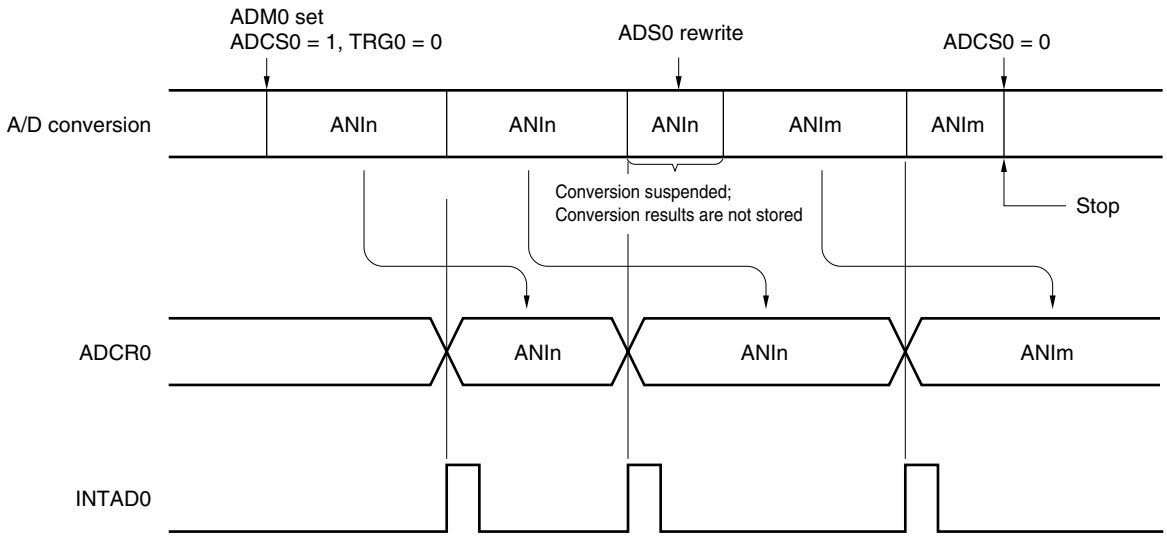
When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) are set to 0 and 1, respectively, A/D conversion of the voltage applied to the analog input pin specified by the analog input channel specification register 0 (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS0.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion and A/D conversion of the new selected analog input channel is started.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 12-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

12.5 How to Read A/D Converter Characteristics Table

Here we will explain the special terms unique to A/D converters.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per 1 bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

When the resolution is 10 bits,

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

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

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero scale offset, full scale offset, integral linearity error, differential linearity error and errors which are combinations of these express overall error.
Furthermore, quantization error is not included in overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, there naturally occurs a ±1/2LSB error. In an A/D converter, an analog input voltage in a range of ±1/2LSB are converted to the same digital code, so a quantization error cannot be avoided.
Furthermore, it is not included in the overall error, zero scale offset, full scale offset, integral linearity error, and differential linearity error in the characteristics table.

Figure 12-9. Overall Error

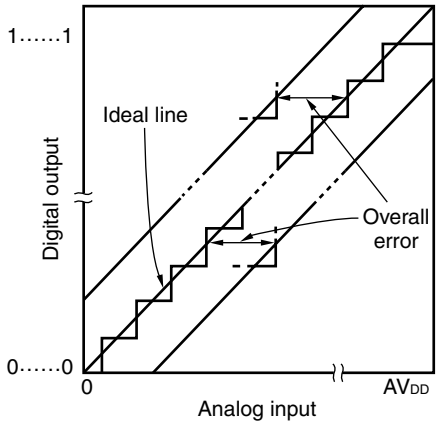
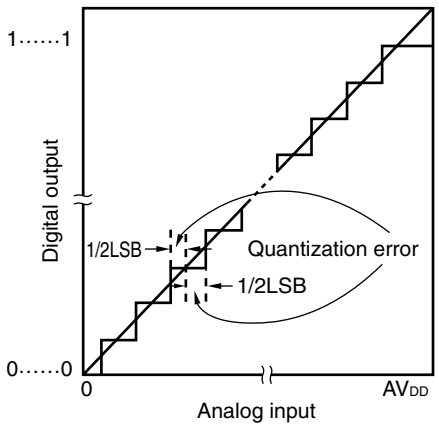


Figure 12-10. Quantization Error



(4) Zero scale offset

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0……000 to 0……001. If the actual measured value is greater than the theoretical value, it shows the difference between the actual measured value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full scale offset

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (full scale –3/2LSB) when the digital output changes from 1……110 to 1……111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measured value and the ideal straight line when the zero scale offset and full scale offset are 0.

(7) Differential linearity error

Although the ideal output width for a given code is 1LSB, this value shows the difference between the actual measured value and the ideal value of the width when outputting a particular code.

Figure 12-11. Zero Scale Offset

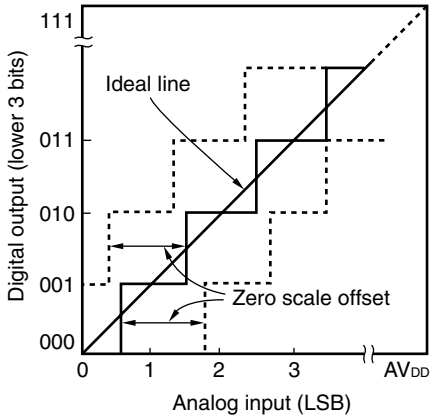


Figure 12-12. Full Scale Offset

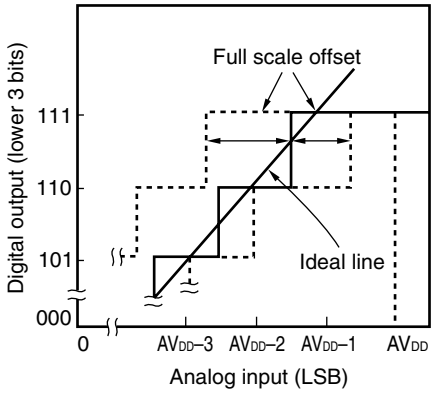


Figure 12-13. Integral Linearity Error

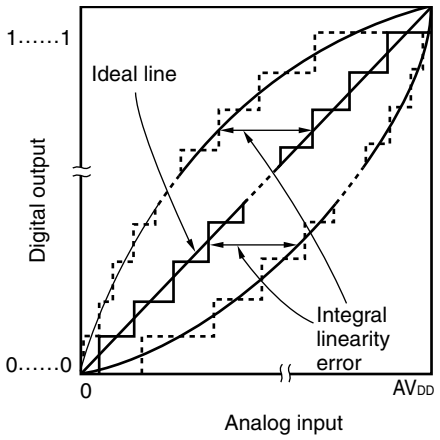
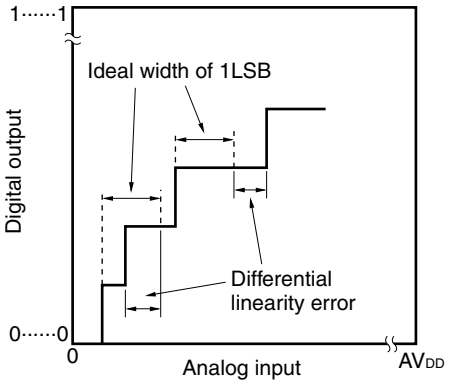


Figure 12-14. Differential Linearity Error



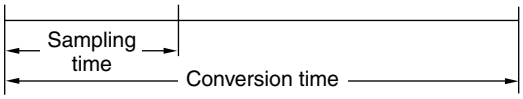
(8) Conversion time

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

Sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



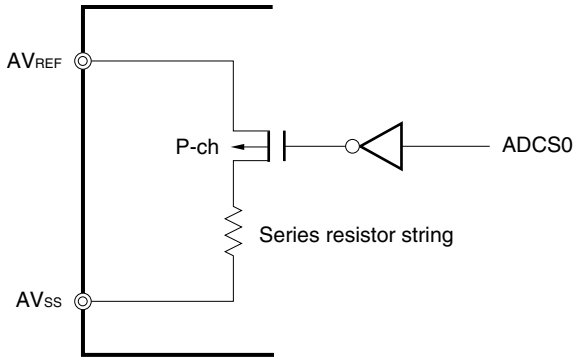
12.6 Cautions for A/D Converter

(1) Current consumption in standby mode

A/D converter stops operating in the standby mode. At this time, current consumption can be reduced by stopping the conversion operation (by setting bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) to 0).

Figure 12-15 shows how to reduce the current consumption in the standby mode.

Figure 12-15. Example of Method of Reducing Current Consumption in Standby Mode



(2) Input range of ANI0 to ANI3

The input voltages of ANI0 to ANI3 should be within the specification range. In particular, if a voltage higher than AV_{REF} or lower than AV_{SS} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Contending operations

<1> Contention between A/D conversion result register 0 (ADCR0) write and ADCR0 read by instruction upon the end of conversion

ADCR0 read is given priority. After the read operation, the new conversion result is written to ADCR0.

<2> Contention between ADCR0 write and external trigger signal input upon the end of conversion

The external trigger signal is not accepted during A/D conversion. Therefore, the external trigger signal is not accepted during ADCR0 write.

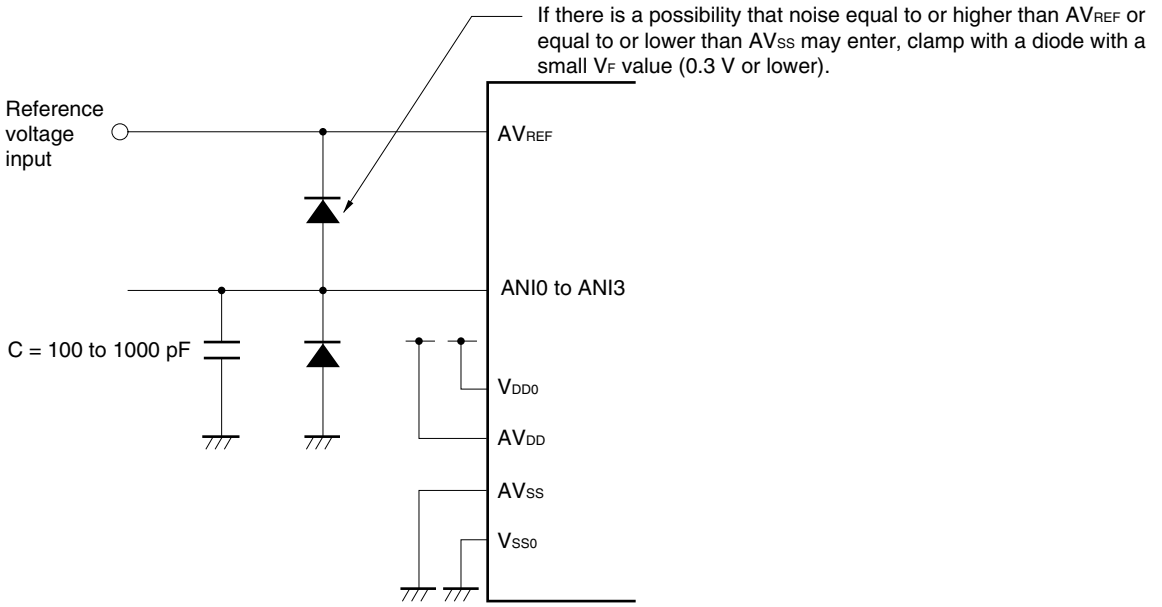
<3> Contention between ADCR0 write and A/D converter mode register 0 (ADM0) write or analog input channel specification register 0 (ADS0) write upon the end of conversion

ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD0) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to pin AV_{REF} and pins ANI0 to ANI3. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 12-16 to reduce noise.

Figure 12-16. Analog Input Pin Connection



(5) ANI0 to ANI3

The analog input pins (ANI0 to ANI3) also function as port pins.

When A/D conversion is performed with any of pins ANI0 to ANI3 selected, do not execute an input instruction to port 1 while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to a pin adjacent to the pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

(6) AV_{REF} pin input impedance

A series resistor string of several 10 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

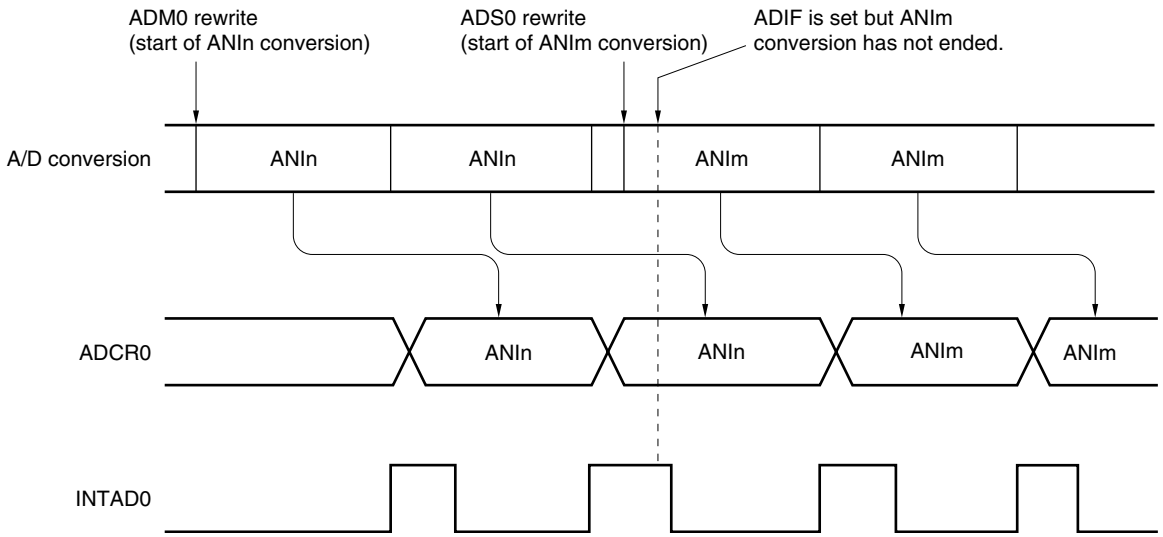
(7) Interrupt request flag (ADIF0)

The interrupt request flag (ADIF0) is not cleared even if the analog input channel specification register 0 (ADS0) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS0 rewrite. Caution is therefore required since, at this time, when ADIF0 is read immediately just after the ADS0 rewrite, ADIF0 is set despite the fact that the A/D conversion for the post-change analog input has not ended.

When A/D conversion is restarted after it is stopped, clear ADIF0 before restart.

Figure 12-17. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 3

2. m = 0, 1, ..., 3

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

The first A/D conversion value just after A/D conversion operations start may not fall within the rating. Polling A/D conversion end interrupt request (INTAD0) and take measures such as removing the first conversion results.

(9) A/D conversion result register 0 (ADCR0) read operation

When writing is performed to the A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Timing at which A/D conversion result is undefined

The A/D conversion value may be undefined if the timing of completion of A/D conversion and the timing of stopping the A/D conversion conflict with each other. Therefore, read the A/D conversion result during the A/D conversion operation. To read the conversion result after stopping the A/D conversion operation, be sure to stop the A/D conversion before the next conversion ends.

Figures 12-18 and 12-19 show the timing of reading the conversion result.

Figure 12-18. Timing of Reading Conversion Result (When Conversion Result Is Undefined)

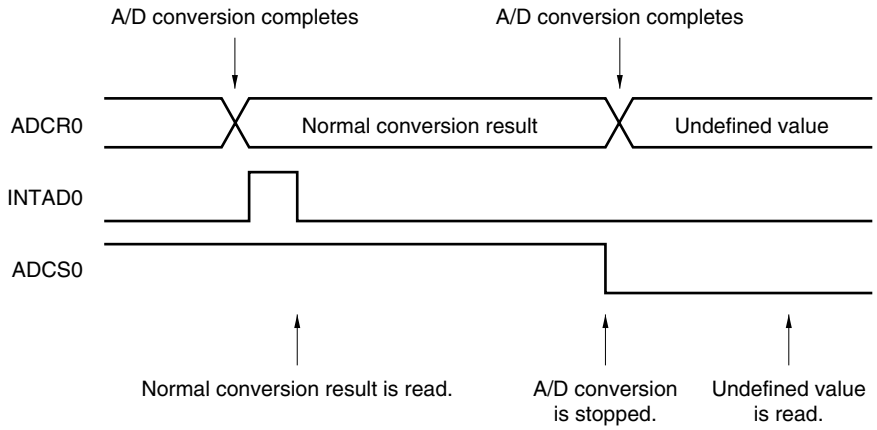
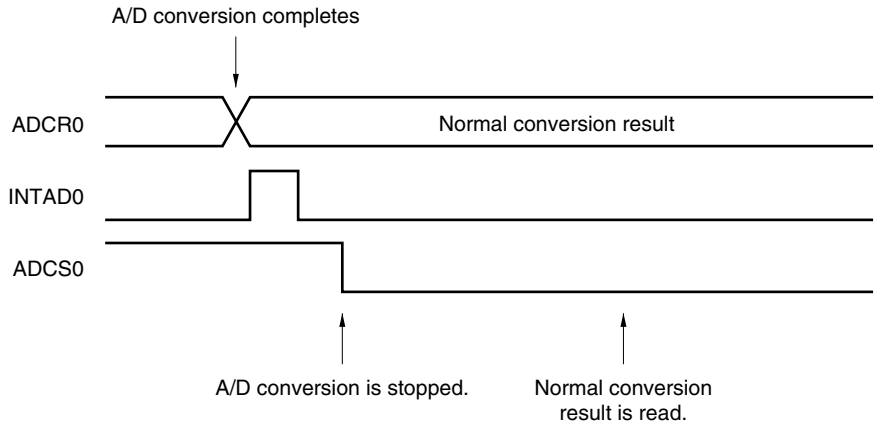


Figure 12-19. Timing of Reading Conversion Result (When Conversion Result Is Normal)



(11) Notes on board design

Locate analog circuits as far away from digital circuits as possible on the board because the analog circuits may be affected by the noise of the digital circuits. In particular, do not cross an analog signal line with a digital signal line, or wire an analog signal line in the vicinity of a digital signal line. Otherwise, the A/D conversion characteristics may be affected by the noise of the digital line.

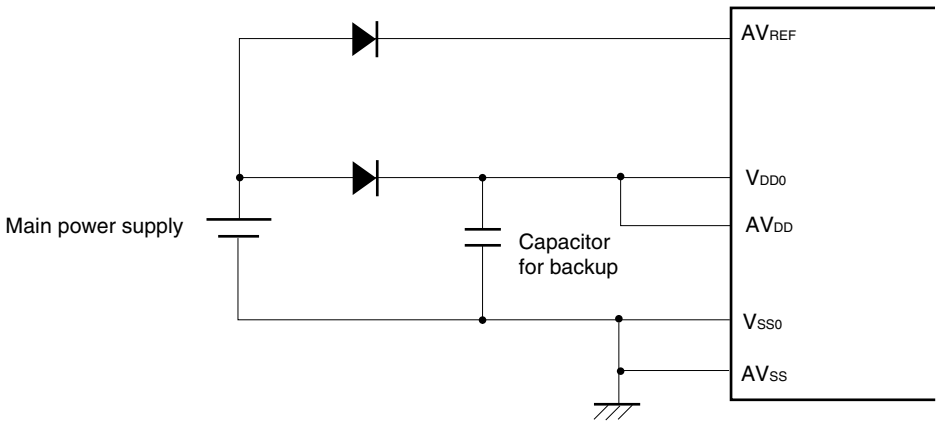
Connect AV_{SS0} and V_{SS0} at one location on the board where the voltages are stable.

(12) AV_{DD} pin

The AV_{DD} pin is the analog circuit power supply pin. It supplies power to the input circuits of the ANI0 to ANI3 pins.

Therefore, be sure to apply the same potential as V_{DD0} to this pin even for applications designed to switch to a backup battery for power supply.

Figure 12-20. AV_{DD} Pin Connection

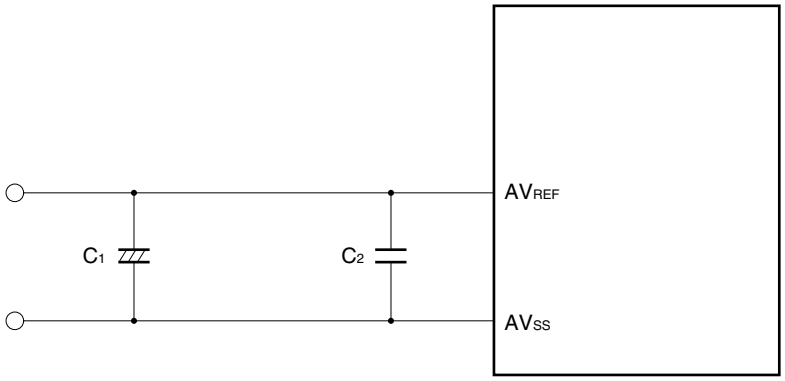


(13) AV_{REF} pin

Connect a capacitor to the AVREF pin to minimize conversion errors due to noise. If an A/D conversion operation has been stopped and then is started, the voltage applied to the AV_{REF} pin becomes unstable, causing the accuracy of the A/D conversion to drop. To prevent this, also connect a capacitor to the AV_{REF} pin.

Figure 12-21 shows an example of connecting a capacitor.

Figure 12-21. Example of Connecting Capacitor to AV_{REF} Pin



Remark C1: 4.7 μF to 10 μF (reference value)
C2: 0.01 μF to 0.1 μF (reference value)
Connect C2 as close to the pin as possible.

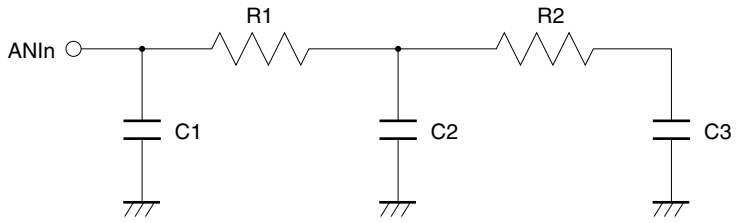
(14) Internal equivalent circuit of ANI0 to ANI3 pins and permissible signal source impedance

To complete sampling within the sampling time with sufficient A/D conversion accuracy, the impedance of the signal source such as a sensor must be sufficiently low. Figure 12-22 shows the internal equivalent circuit of the ANI0 to ANI3 pins.

If the impedance of the signal source is high, connect capacitors with a high capacitance to the pins ANI0 to ANI3. An example of this is shown in Figure 12-23. In this case, however, the microcontroller cannot follow an analog signal with a high differential coefficient because a lowpass filter is created.

To convert a high-speed analog signal or to convert an analog signal in the scan mode, insert a low-impedance buffer.

Figure 12-22. Internal Equivalent Circuit of Pins ANI0 to ANI3



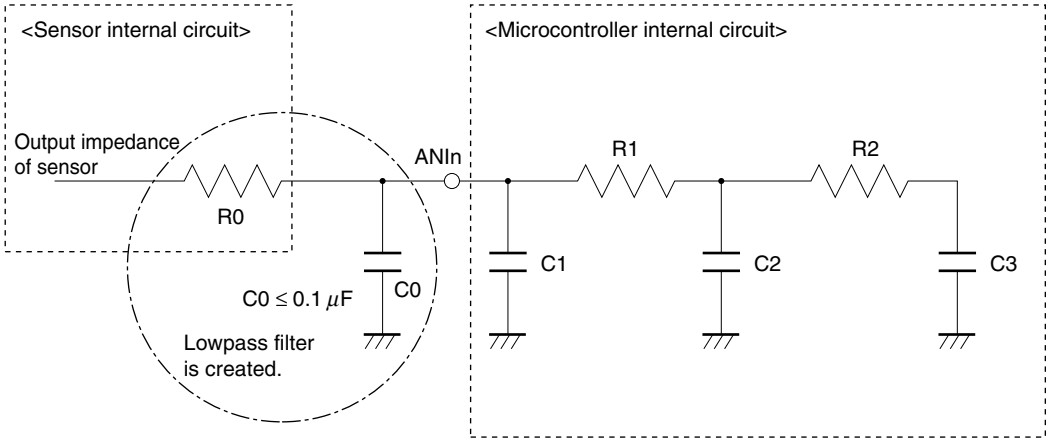
Remark n = 0 to 3

Table 12-2. Resistance and Capacitance of Equivalent Circuit (Reference Values)

AV_{REF}	R_1	R_2	C1	C2	C3
1.8 V	75 kΩ	30 kΩ	8 pF	4 pF	2 pF
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF

Caution The resistance and capacitance in Table 12-2 are not guaranteed values.

Figure 12-23. Example of Connection If Signal Source Impedance Is High



Remark **n =** 0 to 3

CHAPTER 13 SERIAL INTERFACE (UART0)

13.1 Functions of Serial Interface

The serial interface (UART0) has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.
For details, see **13.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

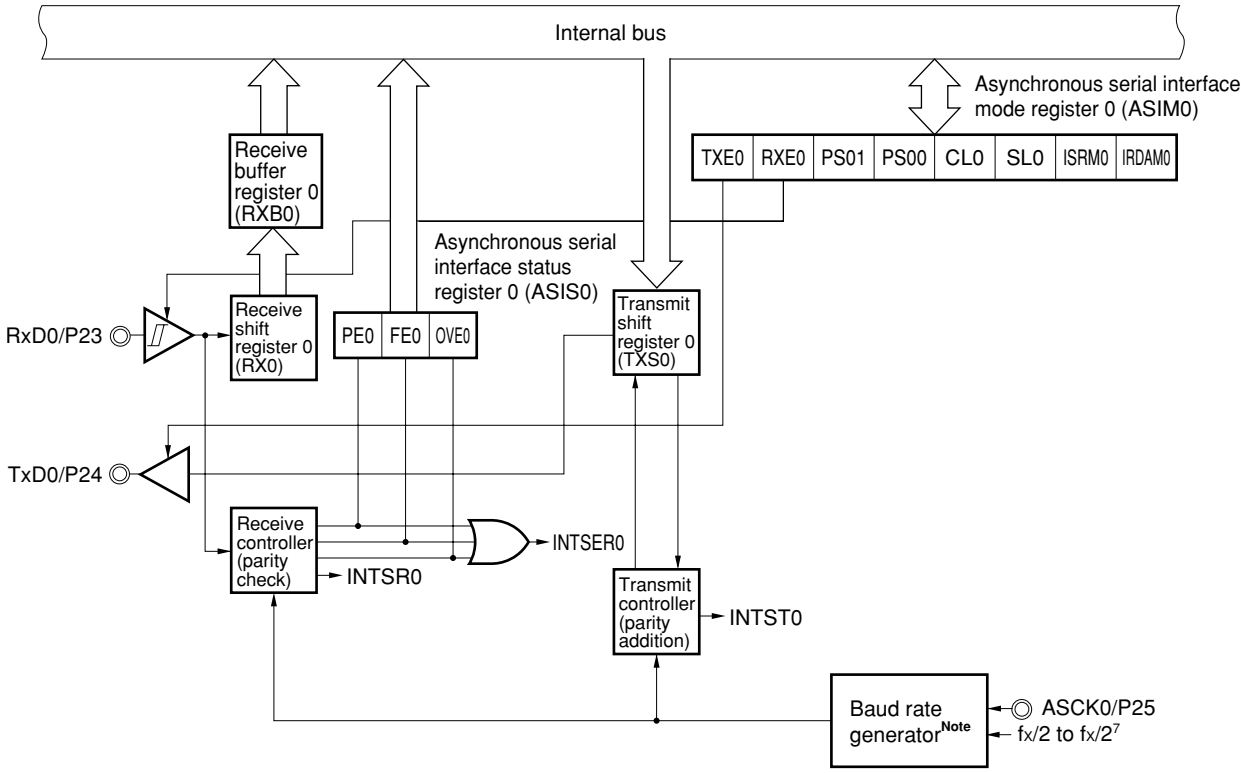
This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted and received.
The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates. In addition, a baud rate can also be defined by dividing clocks input to the ASCK0 pin.
The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).
For details, see **13.4.2 Asynchronous serial interface (UART) mode**.

(3) Infrared data transfer mode

For details, see **13.4.3 Infrared data transfer mode**.

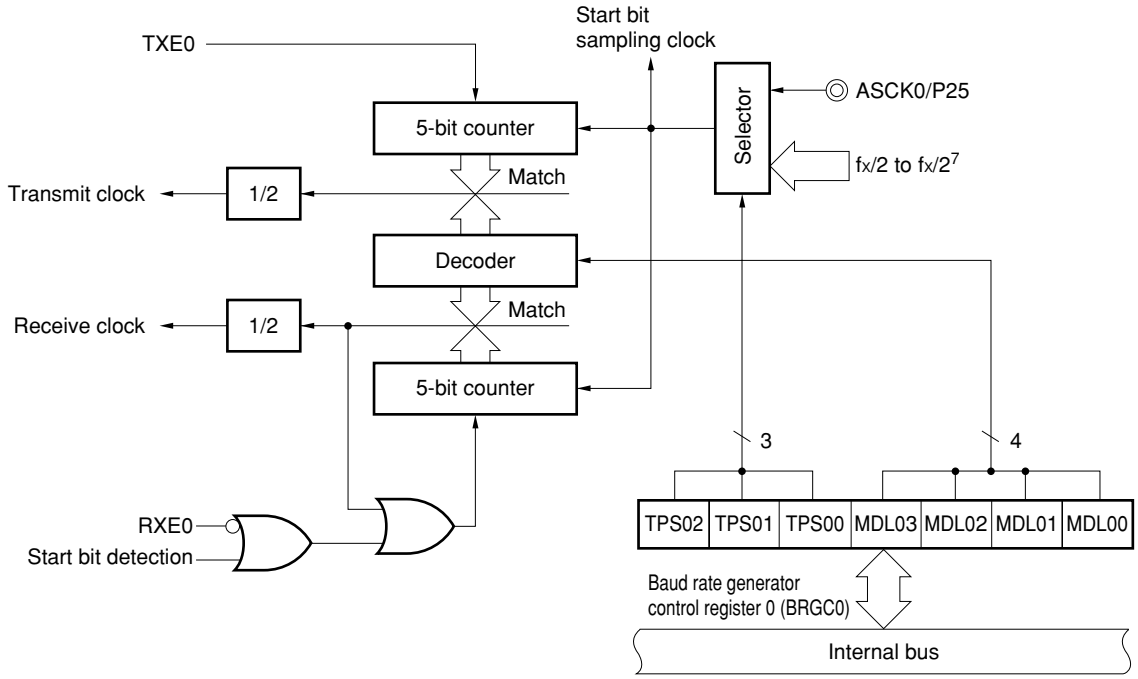
Figure 13-1 shows a block diagram of the serial interface (UART0).

Figure 13-1. Block Diagram of Serial Interface (UART0)



Note For the configuration of the baud rate generator, refer to **Figure 13-2**.

Figure 13-2. Block Diagram of Baud Rate Generator



Remark TXE0: Bit 7 of asynchronous serial interface mode register 0 (ASIM0)
RXE0: Bit 6 of asynchronous serial interface mode register 0 (ASIM0)

13.2 Configuration of Serial Interface

The serial interface (UART0) consists of the following hardware.

Table 13-1. Configuration of Serial Interface (UART0)

Item	Configuration
Registers	Transmit shift register 0 (TXS0) Receive shift register 0 (RX0) Receive buffer register 0 (RXB0)
Control registers	Asynchronous serial interface mode register 0 (ASIM0) Asynchronous serial interface status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0)

(1) Transmit shift register 0 (TXS0)

This is the register for setting transmit data. Data written to TXS0 is transmitted as serial data.

When the data length is set as 7 bits, bits 0 to 6 of the data written to TXS0 are transferred as transmit data.

Writing data to TXS0 starts the transmit operation.

TXS0 can be written by an 8-bit memory manipulation instruction. It cannot be read.

$\overline{\text{RESET}}$ input sets TXS0 to FFH.

Caution Do not write to TXS0 during a transmit operation.
The same address is assigned to TXS0 and the receive buffer register 0 (RXB0). A read operation reads values from RXB0.

(2) Receive shift register 0 (RX0)

This register converts serial data input via the RxD0 pin to parallel data. When one byte of data is received at this register, the receive data is transferred to the receive buffer register 0 (RXB0).

RX0 cannot be manipulated directly by a program.

(3) Receive buffer register 0 (RXB0)

This register is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred from the receive shift register (RX0).

When the data length is set as 7 bits, receive data is sent to bits 0 to 6 of RXB0. In this case, the MSB of RXB0 always becomes 0.

RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input sets RXB0 to FFH.

Caution The same address is assigned to RXB0 and the transmit shift register 0 (TXS0). During a write operation, values are written to TXS0.

(4) Transmit controller

The transmit controller controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to the transmit shift register 0 (TXS0), based on the values set to the asynchronous serial interface mode register 0 (ASIM0).

(5) Receive controller

The receive controller controls receive operations based on the values set to the asynchronous serial interface mode register 0 (ASIM0). During a receive operation, it performs error checking, such as for parity errors, and sets various values to the asynchronous serial interface status register 0 (ASIS0) according to the type of error that is detected.

13.3 Registers to Control Serial Interface

The serial interface (UART0) uses the following three types of registers for control functions.

- Asynchronous serial interface mode register 0 (ASIM0)
- Asynchronous serial interface status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)

(1) Asynchronous serial interface mode register 0 (ASIM0)

This is an 8-bit register that controls serial interface (UART0)'s serial transfer operations.

ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Figure 13-3 shows the format of ASIM0.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P23 (RXD0) to input mode (PM23 = 1)
- **During transmit operation**
  Set P24 (TXD0) to output mode (PM24 = 0)
- **During transmit/receive operation**
  Set P23 to input mode, and P24 to output mode

Figure 13-3. Format of Asynchronous Serial Interface Mode Register 0 (ASIM0)

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation mode	RxD0/P23 pin function	TxD0/P24 pin function
0	0	Operation stop	Port function (P23)	Port function (P24)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P23)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

IRDAM0	Operation specified for infrared data transfer mode[Note 1]
0	UART (transmit/receive) mode
1	Infrared data transfer (transmit/receive) mode[Note 2]

Notes 1. The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

2. When using infrared data transfer mode, be sure to set the baud rate generator control register 0 (BRGC0) to 10H.

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

(2) Asynchronous serial interface status register 0 (ASIS0)

When a receive error occurs during UART mode, this register indicates the type of error.

ASIS0 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS0 to 00H.

Figure 13-4. Format of Asynchronous Serial Interface Status Register 0 (ASIS0)

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Transmit data parity not matched)

FE0	Framing error flag
0	No framing error
1	Framing error**Note 1** (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error**Note 2** (Next receive operation was completed before data was read from receive buffer register 0 (RXB0))

Notes 1. Even if a stop bit length is set to 2 bits by setting bit 2 (SL0) in the asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of the receive buffer register 0 (RXB0) when an overrun error has occurred.

Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

(3) Baud rate generator control register 0 (BRGC0)

This register sets the serial clock for serial interface.

BRGC0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 00H.

Figure 13-5 shows the format of BRGC0.

Figure 13-5. Format of Baud Rate Generator Control Register 0 (BRGC0)

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(f_X = 8.38 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P25/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Cautions 1. Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

2. Set 10H to BRGC0 when using in infrared data transfer mode.

Remarks 1. f_{SCK}: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

13.4 Operations of Serial Interface

This section explains the three modes of the serial interface (UART0).

13.4.1 Operation stop mode

Because serial transfer is not performed during this mode, the power consumption can be reduced.
In addition, pins can be used as normal ports.

(1) Register settings

Operation stop mode is set by the asynchronous serial interface mode register 0 (ASIM0).
ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation mode	RxD0/P23 pin function	TxD0/P24 pin function
0	0	Operation stop	Port function (P23)	Port function (P24)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P23)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

13.4.2 Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted or received.

The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates.

The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(1) Register settings

UART mode settings are performed by the asynchronous serial interface mode register 0 (ASIM0), asynchronous serial interface status register 0 (ASIS0), and the baud rate generator control register 0 (BRGC0).

(a) Asynchronous serial interface mode register 0 (ASIM0)

ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P23 (RXD0) to input mode (PM23 = 1)
- **During transmit operation**
  Set P24 (TXD0) to output mode (PM24 = 0)
- **During transmit/receive operation**
  Set P23 to input mode, and P24 to output mode

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation mode	RxD0/P23 pin function	TxD0/P24 pin function
0	0	Operation stop	Port function (P23)	Port function (P24)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P23)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

IRDAM0	Operation specified for infrared data transfer mode[Note 1]
0	UART (transmit/receive) mode
1	Infrared data transfer (transmit/receive) mode[Note 2]

Notes 1. The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

2. When using infrared data transfer mode, be sure to set the baud rate generator control register 0 (BRGC0) to 10H.

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

(b) Asynchronous serial interface status register 0 (ASIS0)

ASIS0 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS0 to 00H.

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Transmit data parity not matched)

FE0	Framing error flag
0	No framing error
1	Framing error**Note 1** (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error**Note 2** (Next receive operation was completed before data was read from receive buffer register 0 (RXB0))

Notes 1. Even if a stop bit length is set to 2 bits by setting bit 2 (SL0) in the asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of the receive buffer register 0 (RXB0) when an overrun error has occurred.

Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

(c) Baud rate generator control register 0 (BRGC0)

BRGC0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 00H.

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(f_X = 8.38 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P25/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Cautions 1. Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

2. Set 10H to BRGC0 when using infrared data transfer mode.

Remarks 1. f_{SCK}: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the main system clock.

- Transmit/receive clock generation for baud rate by using main system clock
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated from the main system clock is determined according to the following formula.

$$[\text{Baud rate}] = \frac{fx}{2^{n+1}(k + 16)} \quad [\text{Hz}]$$

  fx: Main system clock oscillation frequency
  When ASCK0 is selected as the source clock of the 5-bit counter, substitute the input clock frequency to ASCK0 pin for fx in the above expression.
  n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)
  For details, see **Table 13-2**.
  k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

Table 13-2 shows the relationship between the 5-bit counter's source clock assigned to bits 4 to 6 (TPS00 to TPS02) of BRGC0 and the "n" value in the above formula. Table 13-3 shows the relationship between the main system clock and the baud rate.

Table 13-2. Relationship Between 5-Bit Counter's Source Clock and "n" Value

TPS02	TPS01	TPS00	5-Bit Counter's Source Clock Selected	n
0	0	0	P25/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

Remark fx: Main system clock oscillation frequency

Table 13-3. Relationship Between Main System Clock and Baud Rate

Baud Rate (bps)	f_X = 8.386 MHz		f_X = 8.000 MHz		f_X = 7.3728 MHz		f_X = 5.000 MHz		f_X = 4.1943 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)
600	–	–	–	–	–	–	–	–	7BH	1.14
1200	7BH	1.10	7AH	0.16	78H	0	70H	1.73	6BH	1.14
2400	6BH	1.10	6AH	0.16	68H	0	60H	1.73	5BH	1.14
4800	5BH	1.10	5AH	0.16	58H	0	50H	1.73	4BH	1.14
9600	4BH	1.10	4AH	0.16	48H	0	40H	1.73	3BH	1.14
19200	3BH	1.10	3AH	0.16	38H	0	30H	1.73	2BH	1.14
31250	31H	–1.3	30H	0	2DH	1.70	24H	0	21H	–1.3
38400	2BH	1.10	2AH	0.16	28H	0	20H	1.73	1BH	1.14
76800	1BH	1.10	1AH	0.16	18H	0	10H	1.73	–	–
115200	12H	1.10	11H	2.12	10H	0	–	–	–	–
Infrared data transfer mode[Note]	131031 bps		125000 bps		115200 bps		78125 bps		65536 bps	

Note The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.
When using the infrared data transfer mode, be sure to set the baud rate generator control register 0 (BRGC0) as follows.

- k = 0 (MDL0 to MDL3 = 0000)
- n = 1 (TPS00 to TPS02 = 001)

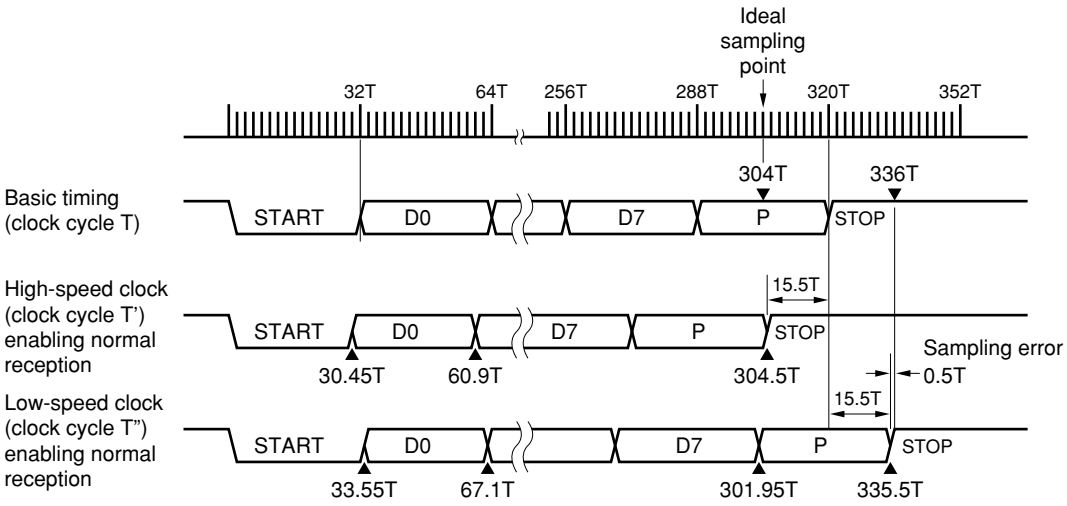
Remark f_X: Main system clock oscillation frequency
n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)
k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

- **Error tolerance range for baud rates**

  The tolerance range for baud rates depends on the number of bits per frame and the counter's division rate [1/(16 + k)].

  Figure 13-6 shows an example of a baud rate error tolerance range.

Figure 13-6. Baud Rate Error Tolerance (When k = 0), Including Sampling Errors



Remark T: 5-bit counter's source clock cycle

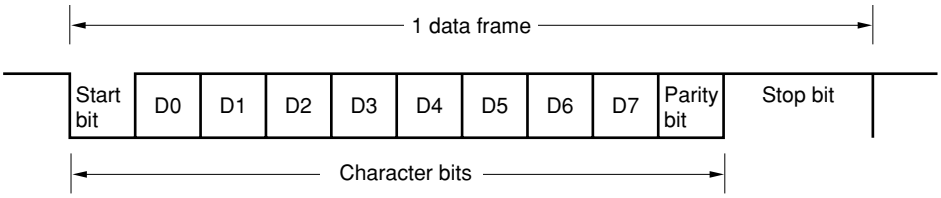
$$\text{Baud rate error tolerance (when k = 0)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

(2) Communication operations

(a) Data format

Figure 13-7 shows the format of the transmit/receive data.

Figure 13-7. Format of Transmit/Receive Data in Asynchronous Serial Interface



1 data frame consists of the following bits.

- Start bit ..... 1 bit
- Character bits ... 7 bits or 8 bits
- Parity bit ..... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) ..... 1 bit or 2 bits

The asynchronous serial interface mode register 0 (ASIM0) is used to set the character bit length, parity selection, and stop bit length within each data frame.
When “7 bits” is selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to “0”.
The ASIM0 and the baud rate generator control register 0 (BRGC0) are used to set the serial transfer rate.
If a receive error occurs, information about the receive error can be recognized by reading the asynchronous serial interface status register 0 (ASIS0).

(b) Parity types and operations

The parity bit is used to detect bit errors in communication data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

(i) Even parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there are an even number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: the parity bit is “1”
  If the transmit data contains an even number of bits whose value is 1: the parity bit is “0”

- During reception
  The number of bits whose value is 1 is counted among the receive data that include a parity bit, and a parity error occurs when the counted result is an odd number.

(ii) Odd parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there is an odd number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: the parity bit is “0”
  If the transmit data contains an even number of bits whose value is 1: the parity bit is “1”

- During reception
  The number of bits whose value is 1 is counted among the receive data that include a parity bit, and a parity error occurs when the counted result is an even number.

(iii) Zero parity

During transmission, the parity bit is set to “0” regardless of the transmit data.
During reception, the parity bit is not checked. Therefore, no parity errors will occur regardless of whether the parity bit is a “0” or a “1”.

(iv) No parity

No parity bit is added to the transmit data.
During reception, receive data is regarded as having no parity bit. Since there is no parity bit, no parity errors will occur.

(c) Transmission

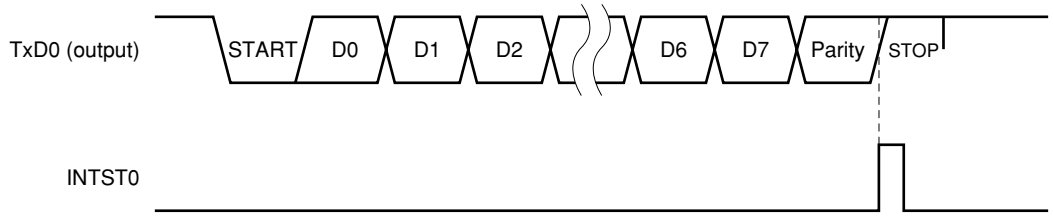
The transmit operation is enabled when bit 7 (TXE0) of the asynchronous serial interface mode register 0 (ASIM0) is set (1). The transmit operation is started when transmit data is written to the transmit shift register 0 (TXS0). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS0, thereby emptying TXS0, after which a transmit completion interrupt request (INTST0) is issued.

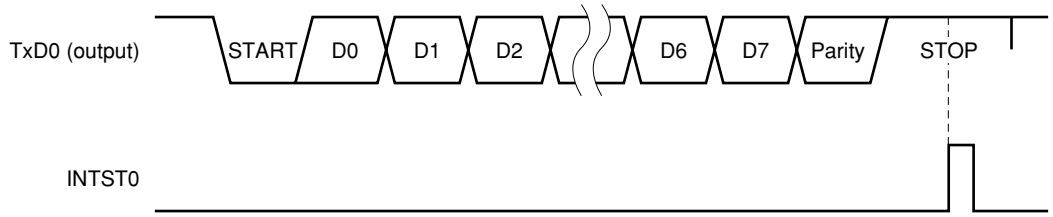
The timing of the transmit completion interrupt request is shown in Figure 13-8.

Figure 13-8. Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request

(i) Stop bit length: 1 bit



(ii) Stop bit length: 2 bits



Caution Do not rewrite to the asynchronous serial interface mode register 0 (ASIM0) during a transmit operation. Rewriting ASIM0 register during a transmit operation may disable further transmit operations (in such cases, input a $\overline{\text{RESET}}$ to restore normal operation). Whether or not a transmit operation is in progress can be determined via software using the transmit completion interrupt request (INTST0) or the interrupt request flag (STIF0) that is set by INTST0.

(d) Reception

The receive operation is enabled when bit 6 (RXE0) of the asynchronous serial interface mode register 0 (ASIM0) is set (1), and input via the RxD0 pin is sampled.

The serial clock specified by ASIM0 is used to sample the RxD0 pin.

When the RxD0 pin goes low, the 5-bit counter of the baud rate generator begins counting and the start timing signal for data sampling is output when half of the specified baud rate time has elapsed. If sampling the RxD0 pin input with this start timing signal yields a low-level result, a start bit is recognized, after which the 5-bit counter is initialized and starts counting and data sampling begins. After the start bit is recognized, the character data, parity bit, and one-bit stop bit are detected, at which point reception of one data frame is completed.

Once reception of one data frame is completed, the receive data in the shift register is transferred to the receive buffer register 0 (RXB0) and a receive completion interrupt request (INTSR0) occurs.

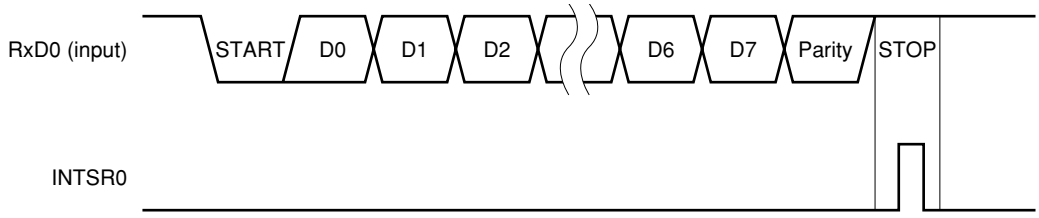
Even if an error has occurred, the receive data in which the error occurred is still transferred to RXB0. When ASIM0 bit 1 (ISRM0) is cleared (0) upon occurrence of an error, INTSR0 occurs (see **Figure 13-10**).

When ISRM0 bit is set (1), INTSR0 does not occur.

If the RXE0 bit is reset (0) during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXB0 and ASIS0 do not change, nor does INTSR0 or INTSER0 occur.

Figure 13-9 shows the timing of the asynchronous serial interface receive completion interrupt request.

Figure 13-9. Timing of Asynchronous Serial Interface Receive Completion Interrupt Request



Caution Be sure to read the contents of the receive buffer register 0 (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

(e) Receive errors

Three types of errors can occur during a receive operation: parity error, framing error, or overrun error. If, as the result of data reception, an error flag is set to the asynchronous serial interface status register 0 (ASIS0), a receive error interrupt request (INTSER0) will occur. Receive error interrupt requests are generated before receive completion interrupt request (INTSR0). Table 13-4 lists the causes behind receive errors.

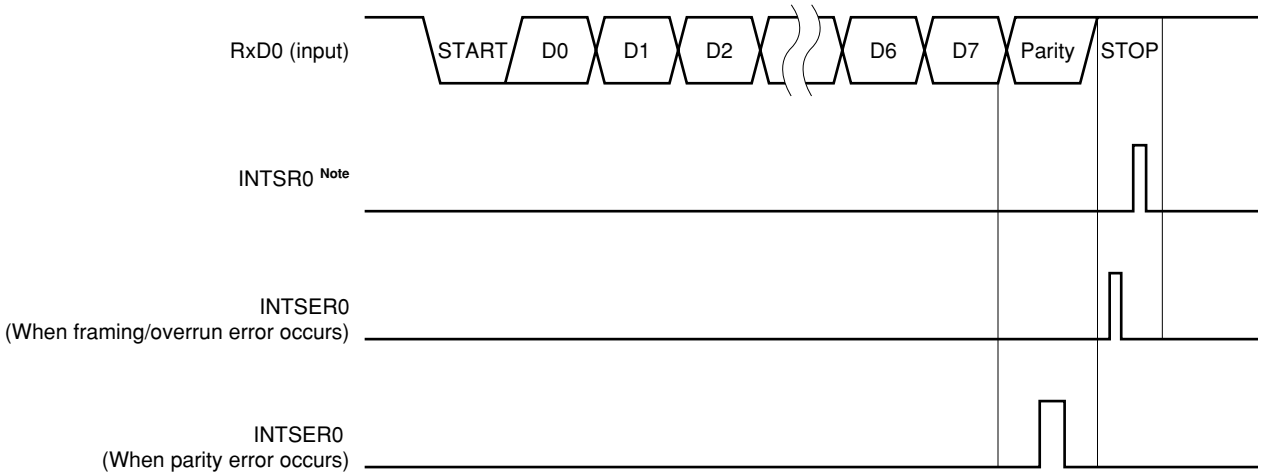
As part of receive error interrupt request (INTSER0) servicing, the contents of ASIS0 can be read to determine which type of error occurred during the receive operation (see **Table 13-4** and **Figure 13-10**).

The contents of ASIS0 are reset (0) when the receive buffer register 0 (RXB0) is read or when the next data is received (if the next data contains an error, its error flag will be set).

Table 13-4. Causes of Receive Errors

Receive Error	Cause	ASIS0 Value
Parity error	Parity specified during transmission does not match parity of receive data	04H
Framing error	Stop bit was not detected	02H
Overrun error	Reception of the next data was completed before data was read from the receive buffer register 0 (RXB0)	01H

Figure 13-10. Receive Error Timing



Note If a receive error occurs when ISRM0 bit has been set (1), INTSR0 does not occur.

Cautions 1. The contents of asynchronous serial interface status register 0 (ASIS0) are reset (0) when the receive buffer register 0 (RXB0) is read or when the next data is received. To obtain information about the error, be sure to read the contents of ASIS0 before reading RXB0.

2. Be sure to read the contents of the receive buffer register 0 (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

13.4.3 Infrared data transfer mode

In infrared data transfer mode, the following data format pulse output and pulse receiving are enabled. The relationship between the main system clock and baud rate is shown in Table 13-3.

(1) Data format

Figure 16-11 compares the data format used in UART mode with that used in infrared data transfer mode.

The IR (infrared) frame corresponds to the bit string of the UART frame, which consists of pulses – a start bit, eight data bits, and a stop bit.

The length of the electrical pulses that are used to transmit and receive in an IR frame is 3/16 the length of the cycle time for one bit (i.e., the “bit time”). This pulse (whose width is 3/16 the length of one bit time) rises from the middle of the bit time (see the figure below).

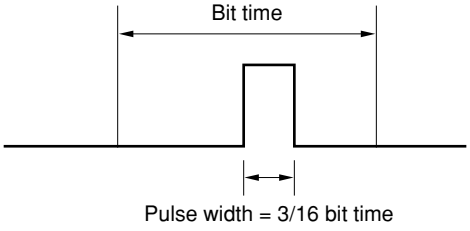
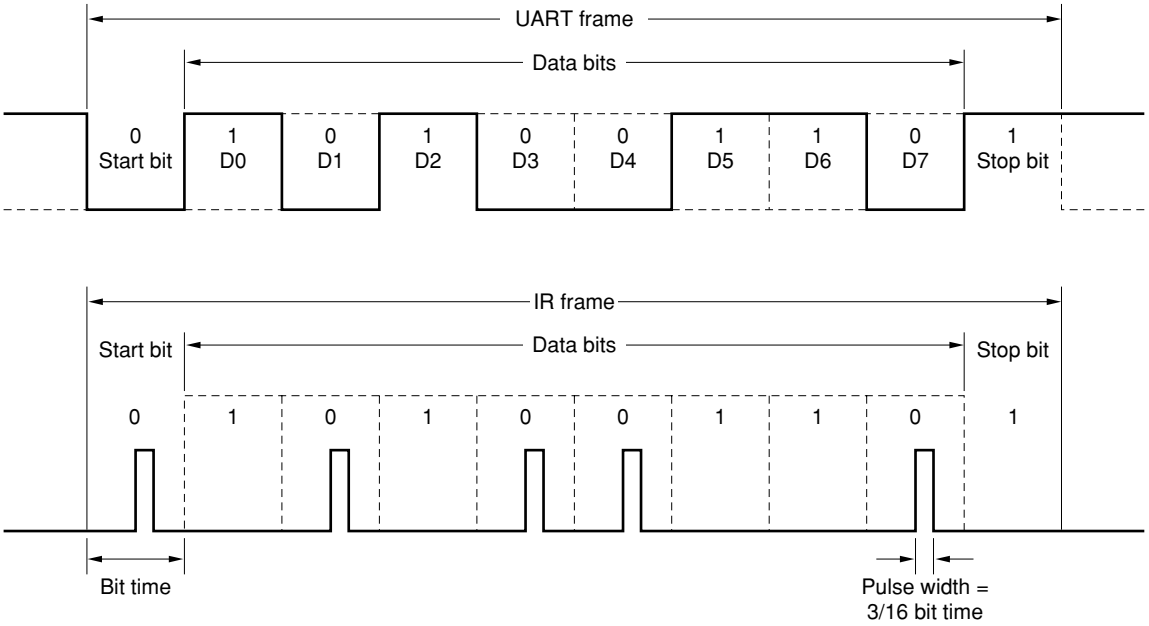


Figure 13-11. Data Format Comparison Between Infrared Data Transfer Mode and UART Mode



(2) Bit rate and pulse width

Table 13-5 lists bit rates, bit rate error tolerances, and pulse width values.

Table 13-5. Bit Rate and Pulse Width Values

Bit Rate (kbits/s)	Bit Rate Error Tolerance (% of bit rate)	Pulse Width Minimum Value (μs)[Note 2]	3/16 Pulse Width <Nominal Value> (μs)	Maximum Pulse Width (μs)
115.2[Note 1]	+/– 0.87	1.41	1.63	2.71

Notes 1. At the operation time with fx = 7.3728 MHz

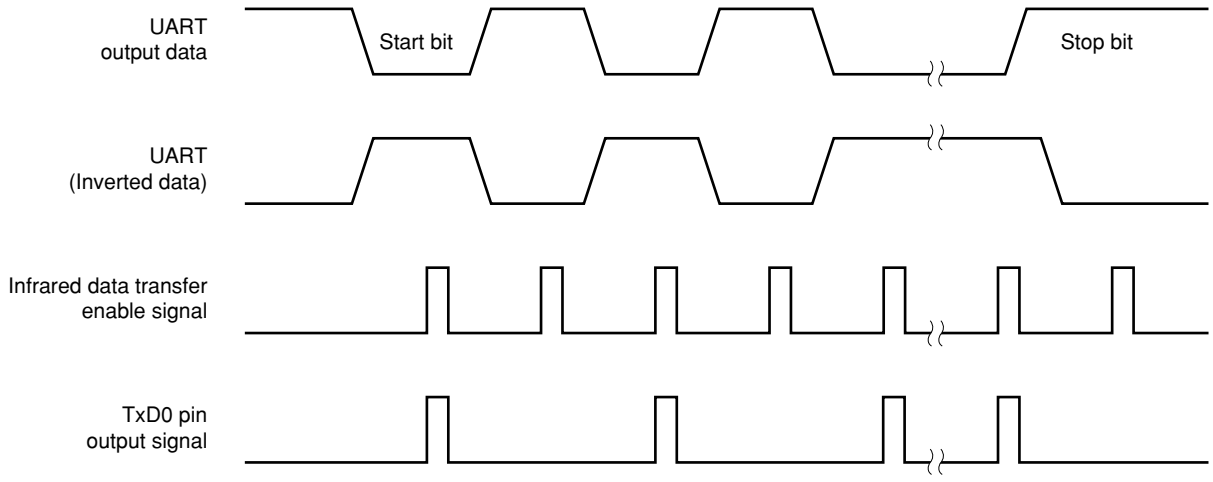
2. When a digital noise eliminator is used in a microcontroller operating at 1.41 MHz or above.

Caution When using in infrared data transfer mode, set 10H to the baud rate generator control register 0 (BRGC0).

Remark fx: Main system clock oscillation frequency

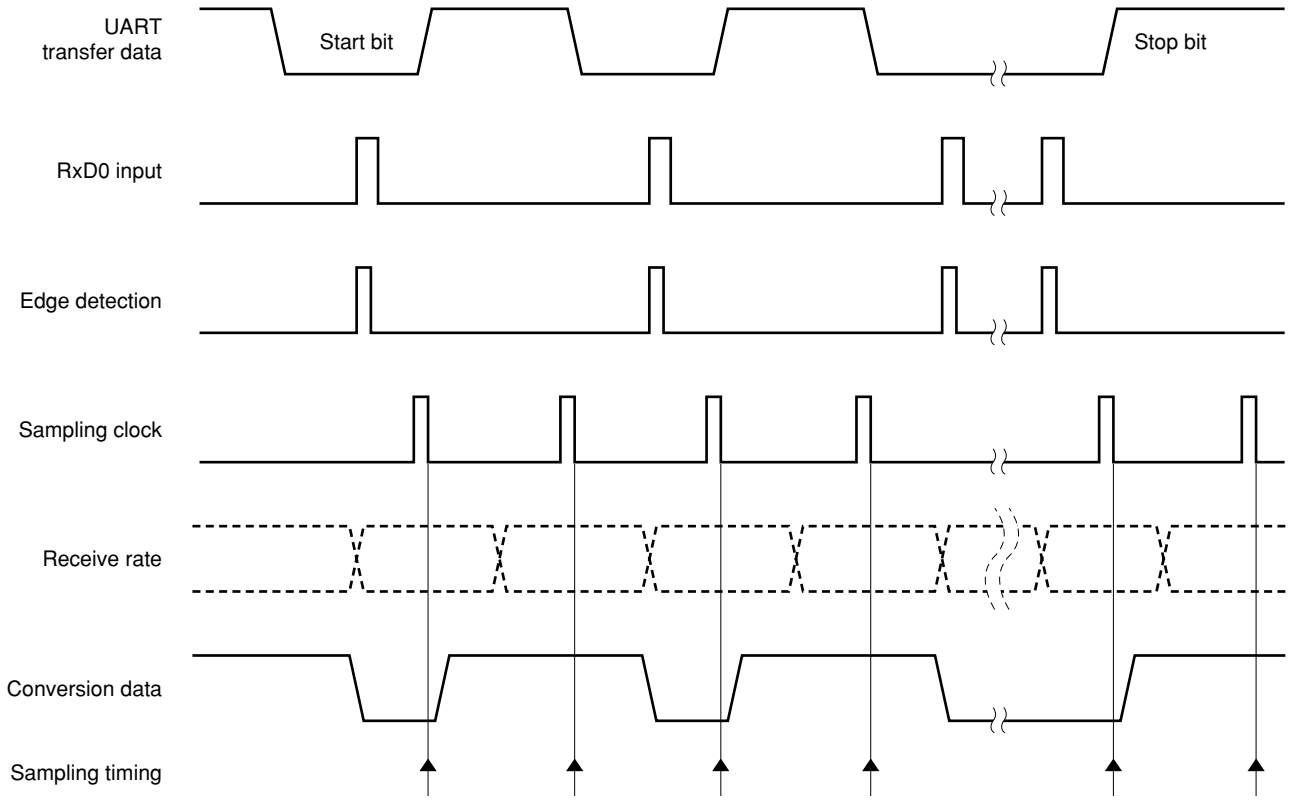
(3) Input data and internal signals

- **Transmit operation timing**



- **Receive operation timing**

  Data reception is delayed for one-half of the specified baud rate.



CHAPTER 14 SERIAL INTERFACE (SIO3)

The serial interface (SIO3) incorporates two 3-wire serial I/O mode channels (SIO30, SIO31).
These two channels have exactly the same functions.

14.1 Functions of Serial Interface

The serial interface (SIO3n) has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **14.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCK3n}}$), serial output line (SO3n), and serial input line (SI3n).

Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

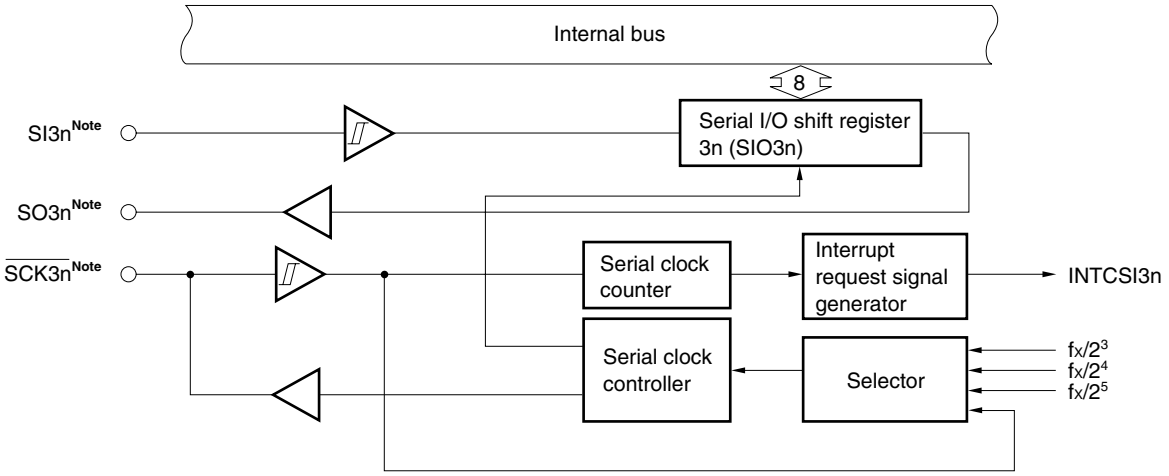
The first bit of the serial transferred 8-bit data is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clocked serial interface, or a display controller, etc. For details see **14.4.2 3-wire serial I/O mode**.

Figure 14-1 shows a block diagram of the serial interface (SIO3n).

Remark n = 0, 1

Figure 14-1. Block Diagram of Serial Interface (SIO3n)



Note SI30, SO30, and $\overline{\text{SCK30}}$ pins are shared with P20, P21, and P22 pins. SI31, SO31, and $\overline{\text{SCK31}}$ pins are shared with P34, P35, and P36 pins.

Remark n = 0, 1

14.2 Configuration of Serial Interface

The serial interface (SIO3n) consists of the following hardware.

Table 14-1. Configuration of Serial Interface (SIO3n)

Item	Configuration
Register	Serial I/O shift register 3n (SIO3n)
Control register	Serial operation mode register 3n (CSIM3n)

Remark n = 0, 1

(1) Serial I/O shift register 3n (SIO3n)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO3n is set by an 8-bit memory manipulation instruction.

When 1 is set to bit 7 (CSIE3n) of the serial operation mode register 3n (CSIM3n), a serial operation can be started by writing data to or reading data from SIO3n.

When transmitting, data written to SIO3n is output to the serial output (SO3n).

When receiving, data is read from the serial input (SI3n) and written to SIO3n.

$\overline{\text{RESET}}$ input makes SIO3n undefined.

Caution Do not access SIO3n during a transfer operation unless the access is triggered by a transfer start (read operation is disabled when MODEn = 0 and write operation is disabled when MODEn = 1).

Remark n = 0, 1

14.3 Registers to Control Serial Interface

The serial interface (SIO3n) is controlled by serial operation mode register 3n (CSIM3n).

(1) Serial operation mode register 30 (CSIM30)

This register is used to enable or disable SIO30's serial clock, operation modes, and specific operations.

CSIM30 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM30 to 00H.

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

<When SIO30 is used>

During serial clock output (master transmission or master reception)	PM22 = 0: Sets P22 ($\overline{\text{SCK30}}$) to output mode P22 = 0: Sets output latch of P22 to 0
During serial clock input (slave transmission or slave reception)	PM22 = 1: Sets P22 ($\overline{\text{SCK30}}$) to input mode
Transmit/receive mode	PM21 = 0: Sets P21 (SO30) to output mode P21 = 0: Sets output latch of P21 to 0
Receive mode	PM20 = 1: Sets P20 (SI30) to input mode

Figure 14-2. Format of Serial Operation Mode Register 30 (CSIM30)

Address: FFB0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	Enable/disable specification for SIO30		
	Shift register operation	Serial counter	Port
0	Operation disabled	Clear	Port function**Note 1**
1	Operation enabled	Count operation enabled	Serial function + port function**Note 2**

MODE0	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO30 output
0	Transmit/transmit and receive mode	Write to SIO30	Normal output
1	Receive-only mode	Read from SIO30	Fixed at low level

SCL301	SCL300	Clock selection
0	0	External clock input to $\overline{\text{SCK30}}$
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE30 = 0 (SIO30 operation stop status), the pins SI30, SO30, and $\overline{\text{SCK30}}$ can be used for port functions.

2. When CSIE30 = 1 (SIO30 operation enabled status), the SI30 pin can be used as a port pin if only the transmit function is used, and the SO30 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

(2) Serial operation mode register 31 (CSIM31)

This register is used to enable or disable SIO31's serial clock, operation modes, and specific operations.

CSIM31 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM31 to 00H.

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

<When SIO31 is used>

During serial clock output (master transmission or master reception)	PM36 = 0: Sets P36 ($\overline{\text{SCK31}}$) to output mode P36 = 0: Sets output latch of P36 to 0
During serial clock input (slave transmission or slave reception)	PM36 = 1: Sets P36 ($\overline{\text{SCK31}}$) to input mode
Transmit/receive mode	PM35 = 0: Sets P35 (SO31) to output mode P35 = 0: Sets output latch of P35 to 0
Receive mode	PM34 = 1: Sets P34 (SI31) to input mode

Figure 14-3. Format Serial Operation Mode Register 31 (CSIM31)

Address: FFB8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM31	CSIE31	0	0	0	0	MODE1	SCL311	SCL310

CSIE31	Enable/disable specification for SIO31		
	Shift register operation	Serial counter	Port
0	Operation disabled	Clear	Port function**Note 1**
1	Operation enabled	Count operation enabled	Serial function + port function**Note 2**

MODE1	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO31 output
0	Transmit/transmit and receive mode	Write to SIO31	Normal output
1	Receive-only mode	Read from SIO31	Fixed at low level

SCL311	SCL310	Clock selection
0	0	External clock input to $\overline{\text{SCK31}}$
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE31 = 0 (SIO31 operation stop status), the pins SI31, SO31, and $\overline{\text{SCK31}}$ can be used for port functions.

2. When CSIE31 = 1 (SIO31 operation enabled status), the SI31 pin can be used as a port pin if only the transmit function is used, and the SO31 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

14.4 Operations of Serial Interface

This section explains the two modes of the serial interface (SIO3n).

14.4.1 Operation stop mode

Because the serial transfer is not performed during this mode, the power consumption can be reduced.
In addition, pins can be used as normal I/O ports.

(1) Register settings

Operation stop mode is set by the serial operation mode register 3n (CSIM3n).
CSIM3n is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM3n to 00H.

Address: FFB0H (SIO30), FFB8H (SIO31) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3n	CSIE3n	0	0	0	0	MODEn	SCL3n1	SCL3n0

CSIE3n	Enable/disable specification for SIO3n		
	Shift register operation	Serial counter	Port
0	Operation disabled	Clear	Port function[Note 1]
1	Operation enabled	Count operation enabled	Serial function + port function[Note 2]

Notes 1. When CSIE3n = 0 (SIO3n operation stop status), the pins SI3n, SO3n, and $\overline{\text{SCK3n}}$ can be used for port functions.

2. When CSIE3n = 1 (SIO3n operation enabled status), the SI3n pin can be used as a port pin if only the transmit function is used, and the SO3n pin can be used as a port pin if only the receive-only mode is used.

Remark n = 0, 1

14.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clocked serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{SCK3n}$), serial output line (SO3n), and serial input line (SI3n).

(1) Register settings

3-wire serial I/O mode is set by the serial operation mode register 3n (CSIM3n).

CSIM3n is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM3n to 00H .

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

<When SIO30 is used>

During serial clock output (master transmission or master reception)	PM22 = 0: Sets P22 ($\overline{SCK30}$) to output mode P22 = 0: Sets output latch of P22 to 0
During serial clock input (slave transmission or slave reception)	PM22 = 1: Sets P22 ($\overline{SCK30}$) to input mode
Transmit/receive mode	PM21 = 0: Sets P21 (SO30) to output mode P21 = 0: Sets output latch of P21 to 0
Receive mode	PM20 = 1: Sets P20 (SI30) to input mode

<When SIO31 is used>

During serial clock output (master transmission or master reception)	PM36 = 0: Sets P36 ($\overline{SCK31}$) to output mode P36 = 0: Sets output latch of P36 to 0
During serial clock input (slave transmission or slave reception)	PM36 = 1: Sets P36 ($\overline{SCK31}$) to input mode
Transmit/receive mode	PM35 = 0: Sets P35 (SO31) to output mode P35 = 0: Sets output latch of P35 to 0
Receive mode	PM34 = 1: Sets P34 (SI31) to input mode

Address: FFB0H (SIO30), FFB8H (SIO31) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3n	CSIE3n	0	0	0	0	MODEn	SCL3n1	SCL3n0

CSIE3n	Enable/disable specification for SIO3n		
	Shift register operation	Serial counter	Port
0	Operation disabled	Clear	Port function[Note 1]
1	Operation enabled	Count operation enabled	Serial function + port function[Note 2]

MODEn	Transfer operation modes and flags		
	Operation Mode	Transfer Start Trigger	SO3n output
0	Transmit/transmit and receive mode	Write to SIO3n	Normal output
1	Receive-only mode	Read from SIO3n	Fixed at low level

SCL3n1	SCL3n0	Clock selection
0	0	External clock input to $\overline{\text{SCK3n}}$
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE3n = 0 (SIO3n operation stop status), the pins SI3n, SO3n, and $\overline{\text{SCK3n}}$ can be used for port functions.

2. When CSIE3n = 1 (SIO3n operation enabled status), the SI3n pin can be used as a port pin if only the transmit function is used, and the SO3n pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. n = 0, 1

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz.

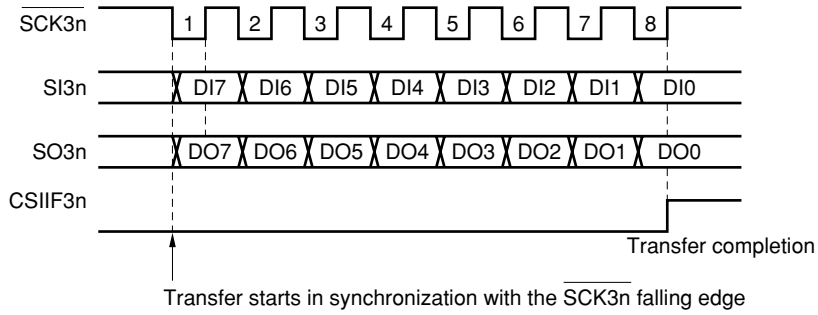
(2) Communication operations

In the 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

The serial I/O shift register 3n (SIO3n) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SO3n latch and is output from the SO3n pin. Data that is received via the SI3n pin in synchronization with the rising edge of the serial clock is latched to SIO3n.

Completion of an 8-bit transfer automatically stops operation of SIO3n and sets interrupt request flag (CSIIF3n).

Figure 14-4. Timing of 3-Wire Serial I/O Mode



Remark n = 0, 1

(3) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set (or read) to serial I/O shift register 3n (SIO3n).

- SIO3n operation control bit (CSIE3n) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or $\overline{\text{SCK3n}}$ is set to high level.
- Transmit/transmit and receive mode
  When CSIE3n = 1 and MODEn = 0, transfer starts when writing to SIO3n.
- Receive-only mode
  When CSIE3n = 1 and MODEn = 1, transfer starts when reading from SIO3n.

Caution After data has been written to SIO3n, transfer will not start even if the CSIE3n bit value is set to "1".

Completion of an 8-bit transfer automatically stops the serial transfer operation and interrupt request flag (CSIIF3n) is set.

Remark n = 0, 1

CHAPTER 15 INTERRUPT FUNCTIONS

15.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally even in an interrupt disabled state. It does not undergo priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

One interrupt request from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L).

Multiple high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 15-1**).

A standby release signal is generated.

Five external interrupt requests and 13 internal interrupt requests are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in an interrupt disabled state. The software interrupt does not undergo interrupt priority control.

15.2 Interrupt Sources and Configuration

A total of 20 interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 15-1**).

Remark Two watchdog timer interrupt sources (INTWDT): a non-maskable interrupt and a maskable interrupt (internal), are available, either of which can be selected.

Table 15-1. Interrupt Source List

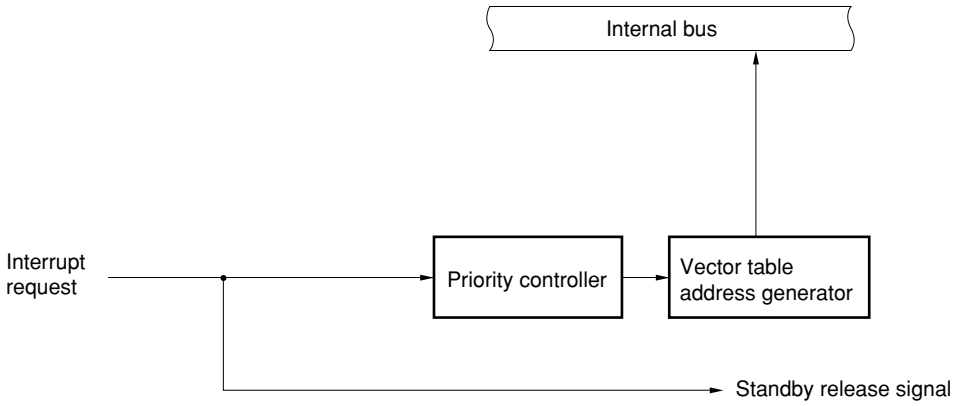
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTSER0	Serial interface UART0 reception error generation	Internal	000EH	(B)
	6	INTSR0	End of serial interface UART0 reception		0010H	
	7	INTST0	End of serial interface UART0 transmission		0012H	
	8	INTCSI30	End of serial interface SIO3 (SIO30) transfer		0014H	
	9	INTCSI31	End of serial interface SIO3 (SIO31) transfer		0016H	
	10	INTWTI	Reference time interval signal from watch timer		001AH	
	11	INTTM00	Match between TM0 and CR00 (when CR00 is specified as compare register) Detection of TI01 valid edge (when CR00 is specified as capture register)		001CH	
	12	INTTM01	Match between TM0 and CR01 (when CR01 is specified as compare register) Detection of TI00 valid edge (when CR01 is specified as capture register)		001EH	
	13	INTTM50	Match between TM50 and CR50		0020H	
	14	INTTM51	Match between TM51 and CR51		0022H	
	15	INTAD0	End of A/D converter conversion		0024H	
	16	INTWT	Watch timer overflow		0026H	
	17	INTKR	Port 4 falling edge detection	External	0028H	(D)
Software	—	BRK	BRK instruction execution	—	003EH	(E)

Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 17 is the lowest.

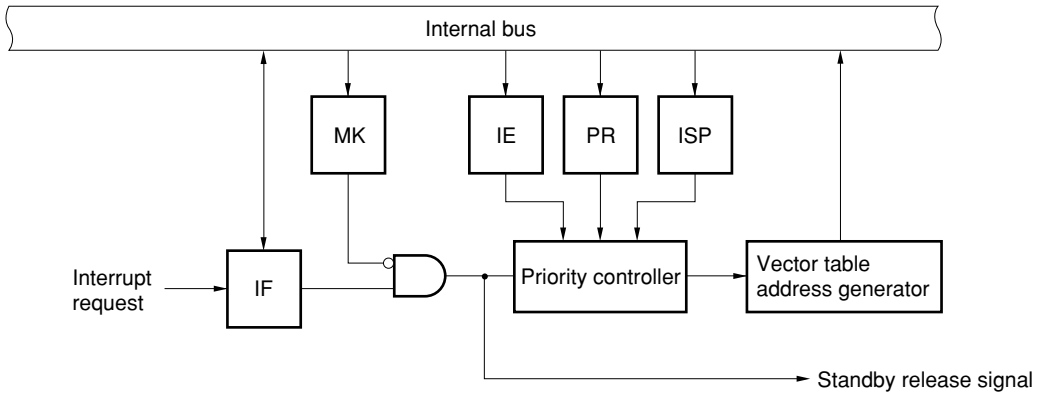
2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 15-1.

Figure 15-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0 to INTP3)

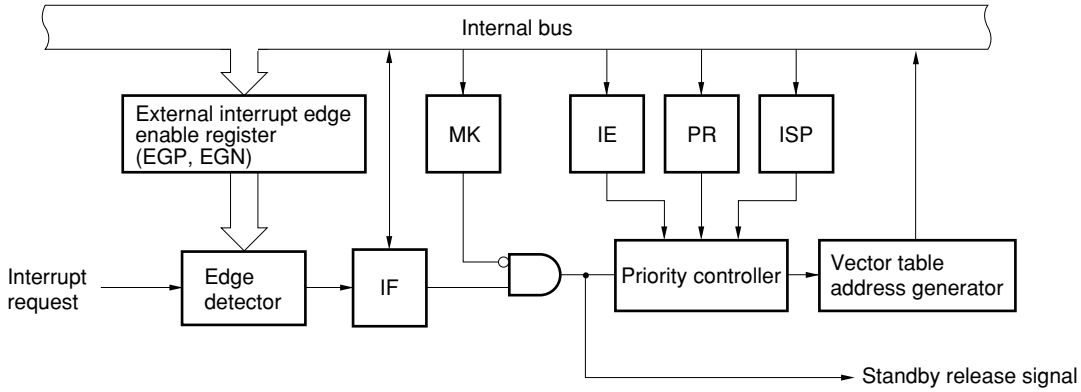
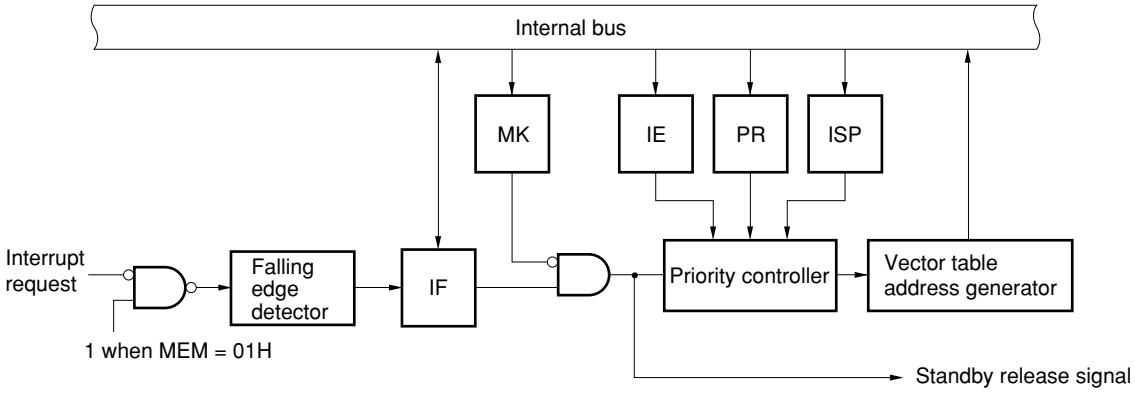
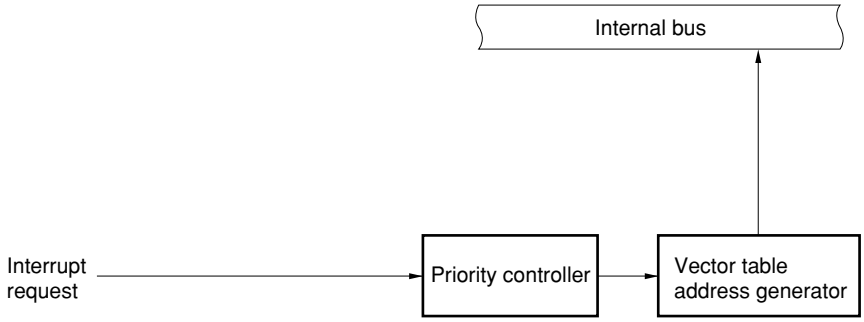


Figure 15-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (INTKR)



(E) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag
MEM: Memory expansion mode register

15.3 Registers to Control Interrupt Function

The following 7 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specification flag register (PR0L, PR0H, PR1L)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Memory expansion mode register (MEM)
- Program status word (PSW)

Table 15-2 gives a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 15-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	WDTIF[Note]	IF0L	WDTMK	MK0L	WDTPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTSER0	SERIF0		SERMK0		SERPR0	
INTSR0	SRIF0		SRMK0		SRPR0	
INTST0	STIF0		STMK0		STPR0	
INTCSI30	CSIIF30	IF0H	CSIMK30	MK0H	CSIPR30	PR0H
INTCSI31	CSIIF31		CSIMK31		CSIPR31	
INTWTI	WTIIF		WTIMK		WTIPR	
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTAD0	ADIF0	IF1L	ADMK0	MK1L	ADPR0	PR1L
INTWT	WTIF		WTMK		WTPR	
INTKR	KRIF		KRMK		KRPR	

Note Interrupt control flag when watchdog timer is used as interval timer

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are set by a 16-bit memory manipulation instruction.

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

Figure 15-2. Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L)

Address: FFE0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	STIF0	SRIF0	SERIF0	PIF3	PIF2	PIF1	PIF0	WDTIF

Address: FFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	TMIF51	TMIF50	TMIF01	TMIF00	WTIIF	0	CSIIF31	CSIIF30

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	0	0	0	0	0	KRIF	WTIF	ADIF0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated, interrupt request status

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as the interval timer. If watchdog timer mode 1 is used, set the WDTIF flag to 0.

2. Be sure to set bit 2 of IF0H and bits 3 to 7 of IF1L to 0.

3. When operating a timer, serial interface, or A/D converter after standby release, run it once after clearing an interrupt request flag. An interrupt request flag may be set by noise.

4. When an interrupt is acknowledged, the interrupt request flag is automatically cleared, and then processing of the interrupt routine is started.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt service.

MK0L, MK0H, and MK1L are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form a 16-bit register MK0, they are set by a 16-bit memory manipulation instruction.

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

Figure 15-3. Format of Interrupt Mask Flag Register (MK0L, MK0H, MK1L)

Address: FFE4H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	STMK0	SRMK0	SERMK0	PMK3	PMK2	PMK1	PMK0	WDTMK

Address: FFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	TMMK51	TMMK50	TMMK01	TMMK00	WTIMK	1	CSIMK31	CSIMK30

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	1	1	1	1	1	KRMK	WTMK	ADMK0

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the watchdog timer is used in watchdog timer mode 1, the contents of the WDTMK flag become undefined when read.

2. Because port 0 pins have an alternate function as external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

3. Be sure to set bit 2 of MK0H and bits 3 to 7 of MK1L to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L)

The priority specification flags are used to set the corresponding maskable interrupt priority orders.

PR0L, PR0H, and PR1L are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set by a 16-bit memory manipulation instruction.

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

Figure 15-4. Format of Priority Specification Flag Register (PR0L, PR0H, PR1L)

Address: FFE8H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0L	STPR0	SRPR0	SERPR0	PPR3	PPR2	PPR1	PPR0	WDTPR

Address: FFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0H	TMPR51	TMPR50	TMPR01	TMPR00	WTIPR	1	CSIPR31	CSIPR30

Address: FFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR1L	1	1	1	1	1	KRPR	WTPR	ADPR0

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Cautions 1. When the watchdog timer is used in the watchdog timer mode 1, set 1 in the WDTPR flag.

2. Be sure to set bit 2 of PR0H and bits 3 to 7 of PR1L to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP3.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 15-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 3)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

(5) Memory expansion mode register (MEM)

MEM sets the rising edge detection function of port 4.

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

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

Figure 15-6. Format of Memory Expansion Mode Register (MEM)

Address: FF47H After reset: 00H R/W

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

MM2	MM1	MM0	Single-chip/memory expansion mode selection
0	0	0	Single-chip mode
0	0	1	Port 4 falling edge detection mode
Other than above			Setting prohibited

Caution When using the falling edge detection function of port 4, be sure to set MEM to 01H.

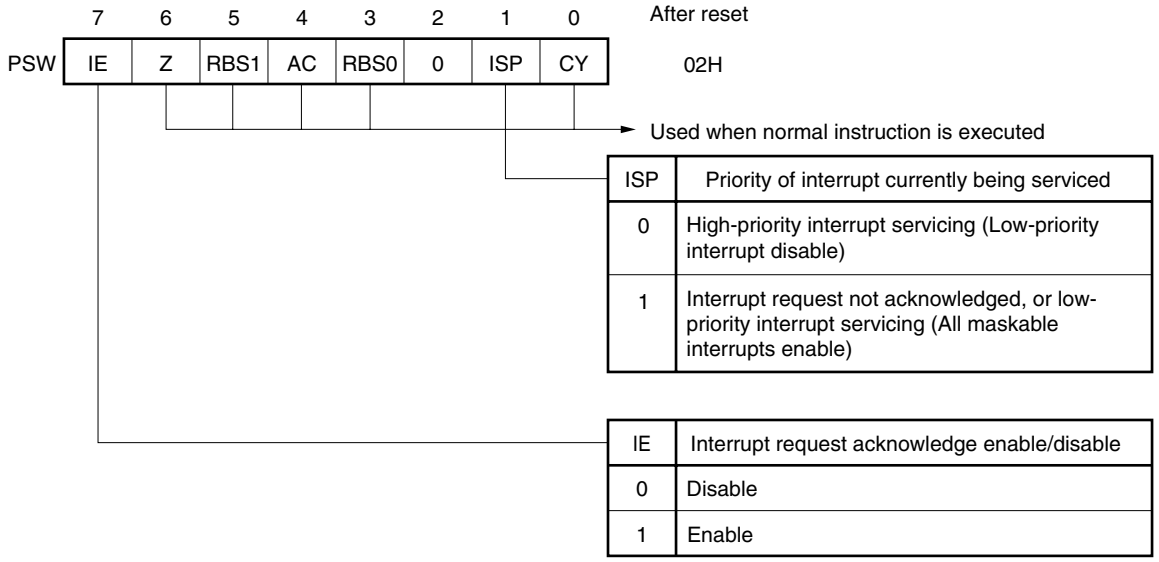
(6) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for an interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control nesting processing are mapped.

Besides 8-bit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are reset from the stack with the RETI, RETB, and POP PSW instructions.

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

Figure 15-7. Format of Program Status Word



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (Low-priority interrupt disable)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (All maskable interrupts enable)

IE	Interrupt request acknowledge enable/disable
0	Disable
1	Enable

15.4 Interrupt Servicing Operations

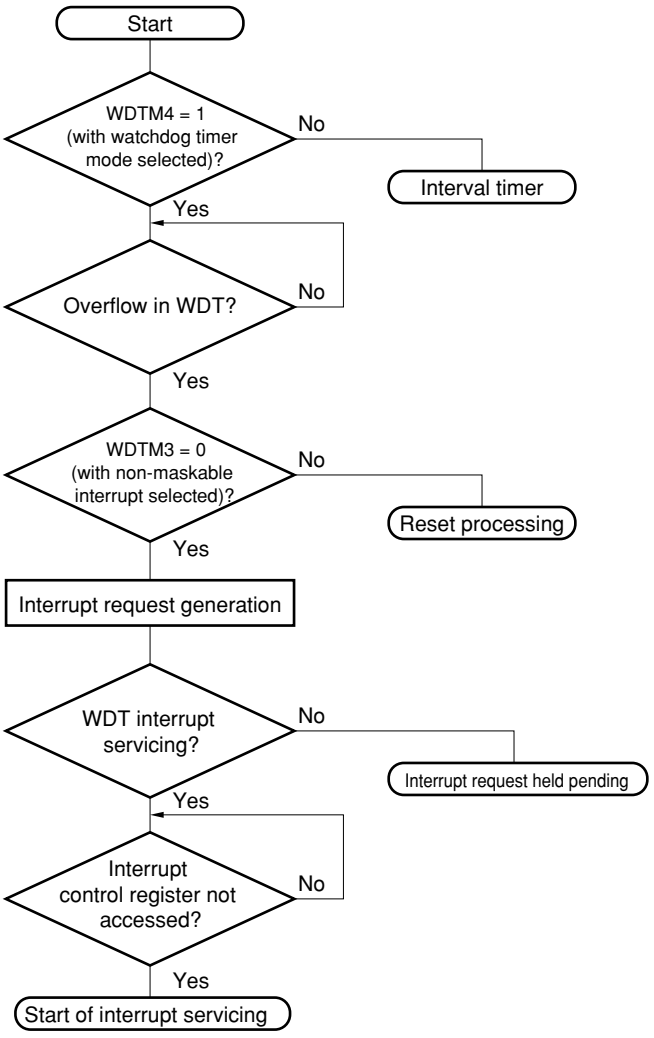
15.4.1 Non-maskable interrupt request acknowledge operation

A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag and ISP flag are reset (0), and the contents of the vector table are loaded into PC and branched.

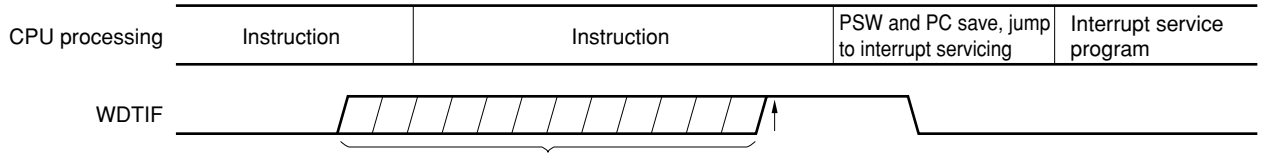
A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. However, if a new non-maskable interrupt request is generated twice or more during non-maskable interrupt servicing program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt servicing program execution. Figures 15-8, 15-9, and 15-10 show the flowchart of the non-maskable interrupt request generation through acknowledge, acknowledge timing of non-maskable interrupt request, and acknowledge operation at multiple non-maskable interrupt request generation, respectively.

Figure 15-8. Non-Maskable Interrupt Request Generation to Acknowledge Flowchart



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 15-9. Non-Maskable Interrupt Request Acknowledge Timing

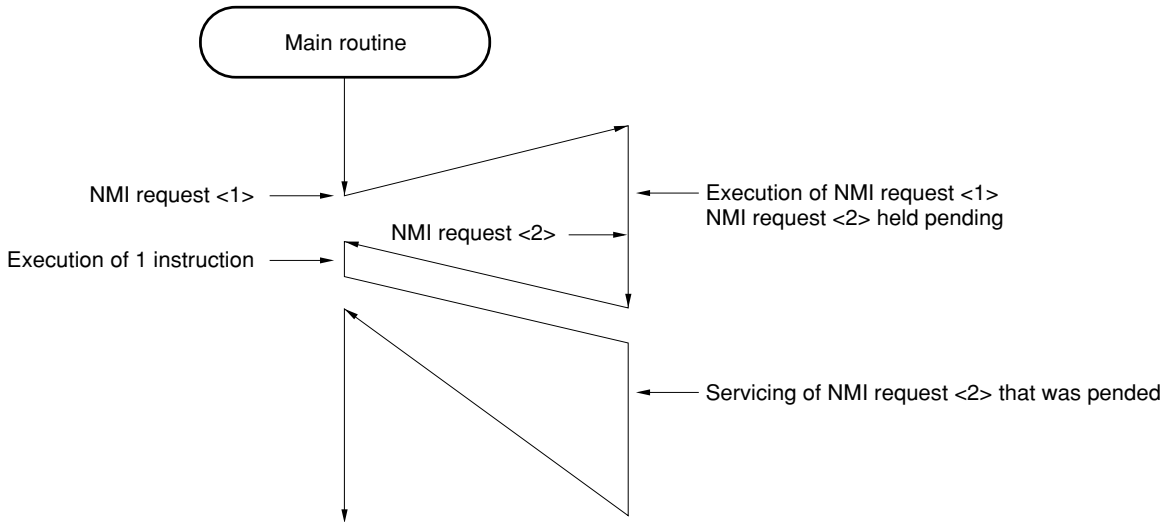


Interrupt request generated during this interval is acknowledged at ↑.

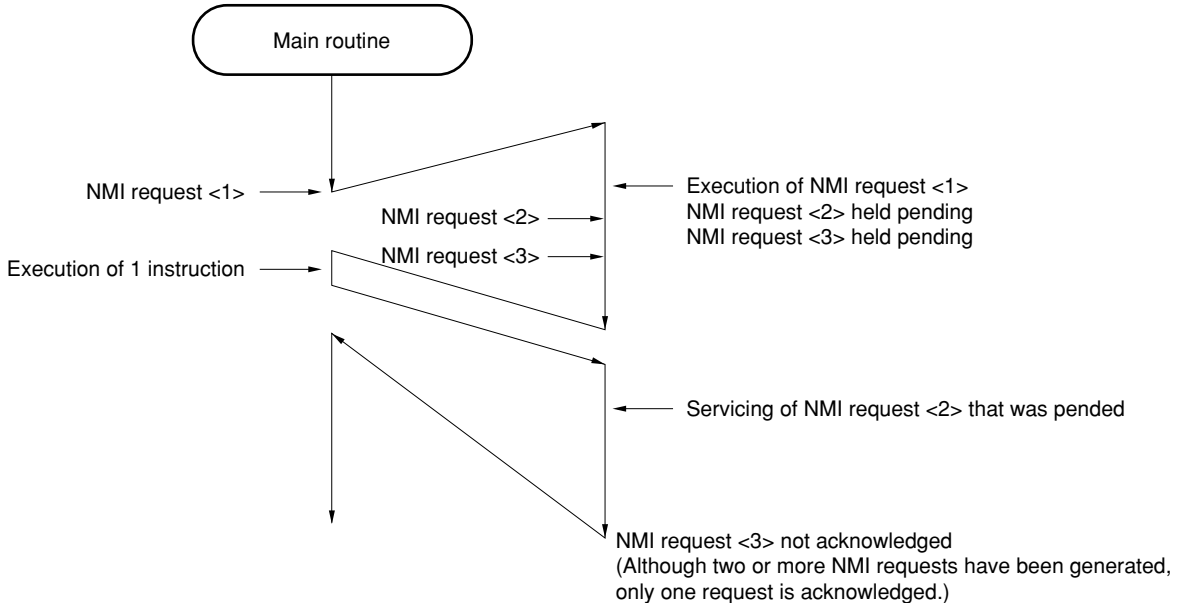
WDTIF: Watchdog timer interrupt request flag

Figure 15-10. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a non-maskable interrupt request is generated during non-maskable interrupt servicing program execution



(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution



15.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if in the interrupt enable state (when IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0). The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 15-3 below.

For the interrupt request acknowledge timing, see **Figures 15-12** and **15-13**.

Table 15-3. Times from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more maskable interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

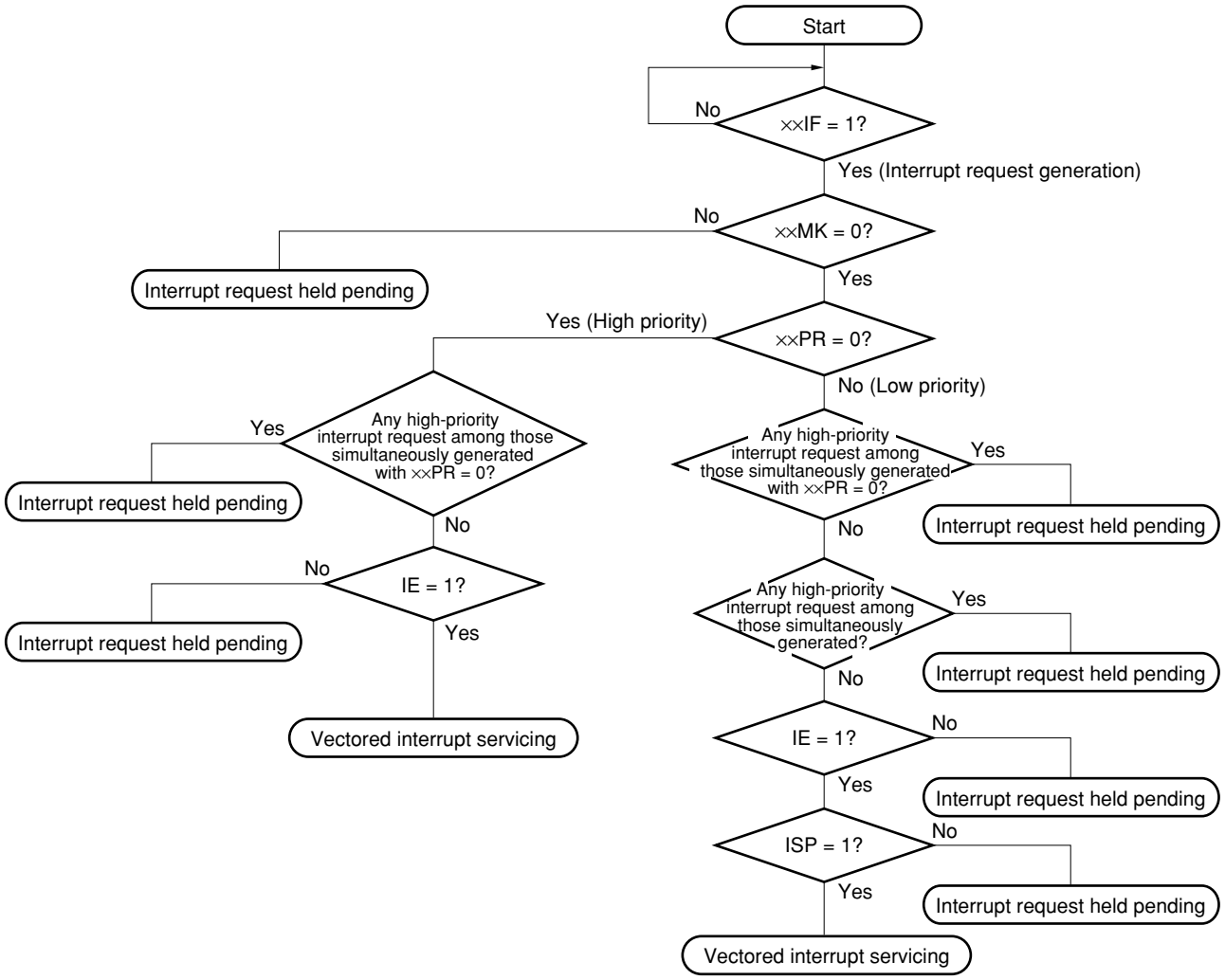
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 15-11 shows the interrupt request acknowledge algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

Return from an interrupt is possible with the RETI instruction.

Figure 15-11. Interrupt Request Acknowledge Processing Algorithm



××IF: Interrupt request flag

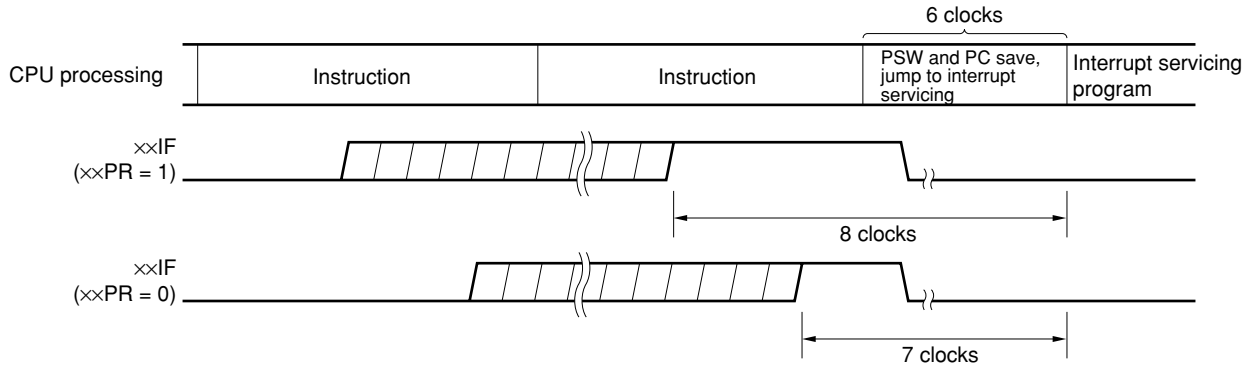
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledge of maskable interrupt request (1 = enable, 0 = disable)

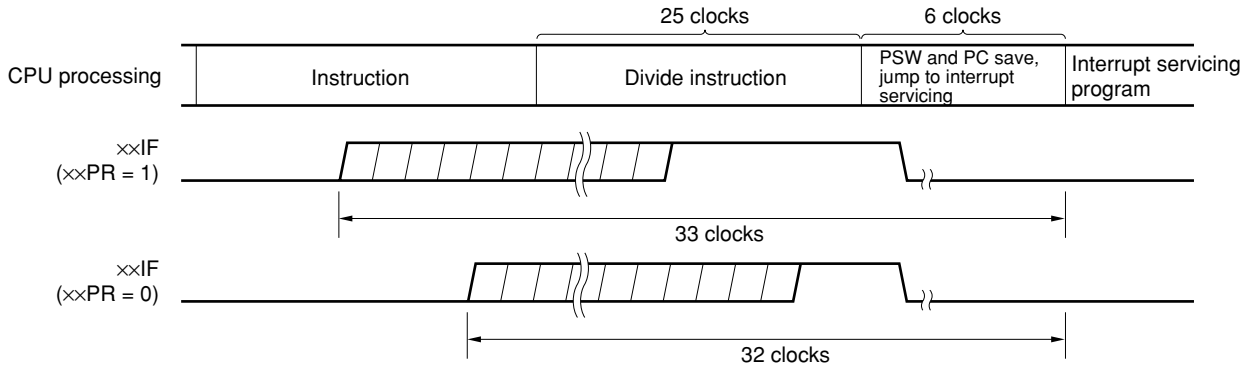
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = no interrupt request acknowledged, or low-priority interrupt servicing)

Figure 15-12. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 15-13. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

15.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (003EH, 003FH) are loaded into PC and branched.

Return from a software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

15.4.4 Nesting interrupt servicing

Nesting occurs when another interrupt request is acknowledged during execution of an interrupt.

Nesting does not occur unless the interrupt request acknowledge enable state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is acknowledged, interrupt request acknowledge becomes disabled (IE = 0). Therefore, to enable nesting, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledge.

Moreover, even if interrupts are enabled, nesting may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for nesting.

In the interrupt enable state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for nesting. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for nesting.

Interrupt requests that are not enabled because of the interrupt disable state or they have a lower priority are held pending. When servicing of the current interrupt ends, the pended interrupt request is acknowledged following execution of one main processing instruction execution.

Nesting is not possible during non-maskable interrupt servicing.

Table 15-4 shows interrupt requests enabled for nesting and Figure 15-14 shows nesting examples.

Table 15-4. Interrupt Request Enabled for Nesting During Interrupt Servicing

Nesting Request / Interrupt Being Serviced		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			PR = 0		PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		×	×	×	×	×
Maskable interrupt	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt		○	○	×	○	×

Remarks 1. ○: Nesting enabled

2. ×: Nesting disabled

3. ISP and IE are flags contained in PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledge is disabled.

IE = 1: Interrupt request acknowledge is enabled.

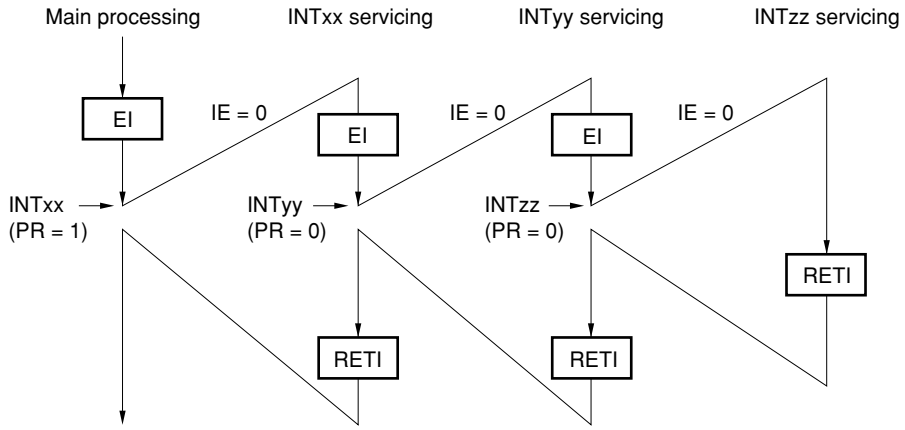
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level

PR = 1: Lower priority level

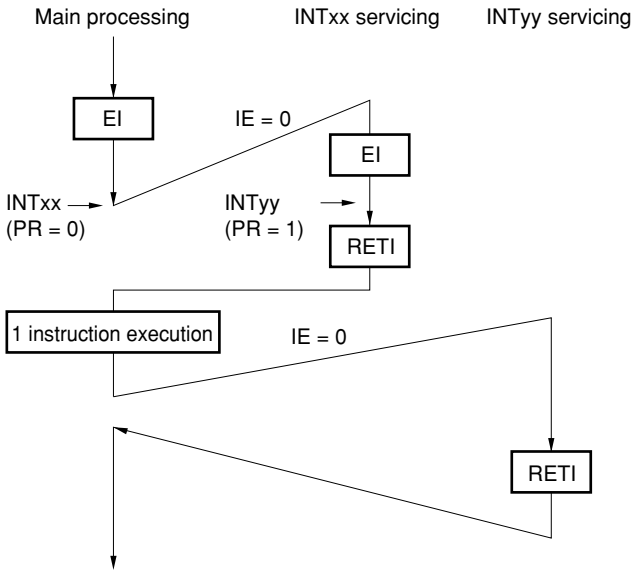
Figure 15-14. Nesting Examples (1/2)

Example 1. Nesting occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and nesting takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledge.

Example 2. Nesting does not occur due to priority control

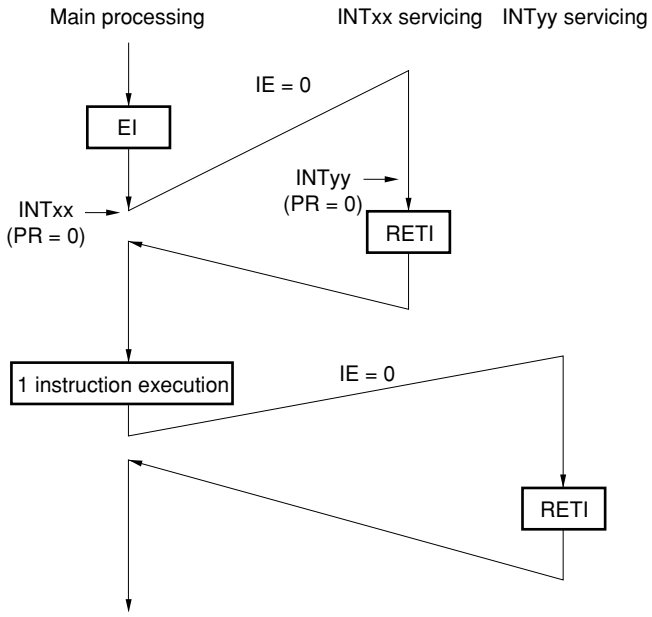


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and nesting does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledge disabled

Figure 15-14. Nesting Examples (2/2)

Example 3. Nesting does not occur because interrupt is not enabled



Interrupt is not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and nesting does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledge disabled

15.4.5 Interrupt request hold

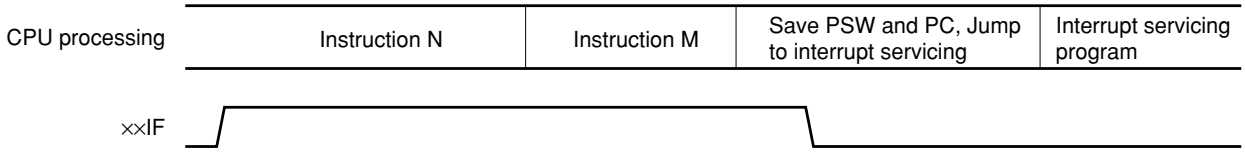
There are instructions where, even if an interrupt request is issued for them while another instruction is executed, request acknowledge is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, $addr16
- BF PSW. bit, $addr16
- BTCLR PSW. bit, $addr16
- EI
- DI
- Manipulate instructions for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, and PR1L registers

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared to 0. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

Figure 15-15 shows the timing with which interrupt requests are held pending.

Figure 15-15. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 16 STANDBY FUNCTION

16.1 Standby Function and Configuration

16.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. The system clock oscillator continues oscillating. In this mode, current consumption is not decreased as much as in the STOP mode. However, the HALT mode is effective to restart operation immediately upon interrupt request and to carry out intermittent operations such as watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU power consumption.

Data memory low-voltage hold (down to V_{DD} = 1.6 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption.

Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

2. When operation is transferred to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

16.1.2 Standby function control register

The wait time after the STOP mode is cleared upon interrupt request is controlled with the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H.

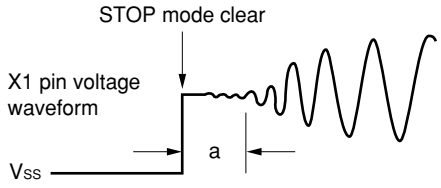
Figure 16-1. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFAH After reset: 04H R/W

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	2^{12}/fx (488 μs)
0	0	1	2^{14}/fx (1.95 ms)
0	1	0	2^{15}/fx (3.91 ms)
0	1	1	2^{16}/fx (7.81 ms)
1	0	0	2^{17}/fx (15.6 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is cleared does not include the time (see "a" in the illustration below) from STOP mode clear to clock oscillation start. The time is not included either by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 8.38 MHz.

16.2 Operations of Standby Function

16.2.1 HALT mode

(1) HALT mode setting and operating status

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating status in the HALT mode is described below.

Table 16-1. HALT Mode Operating Status

HALT Mode Setting / Item	During HALT Instruction Execution Using Main System Clock		During HALT Instruction Execution Using Subsystem Clock	
	Without Subsystem Clock[Note 1]	With Subsystem Clock[Note 2]	With Main System Clock Oscillation	With Main System Clock Oscillation Stopped
Clock generator	Both main system clock and subsystem clock can be oscillated. Clock supply to CPU stops.			
CPU	Operation stops.			
Port (output latch)	Status before HALT mode setting is held.			
16-bit timer/event counter	Operable			Operation stops.
8-bit timer/event counter	Operable			Operable when TI50, TI51 are selected as count clock.
Watch timer	Operable when $f_X/2^7$ is selected as count clock	Operable		Operable when f_{XT} is selected as count clock.
Watchdog timer	Operable		Operation stops.	
A/D converter	Operation stops.			
Serial interface	Operable			Operable during external SCK.
External interrupt	Operable			

Notes 1. Including case when external clock is not supplied.

2. Including case when external clock is supplied.

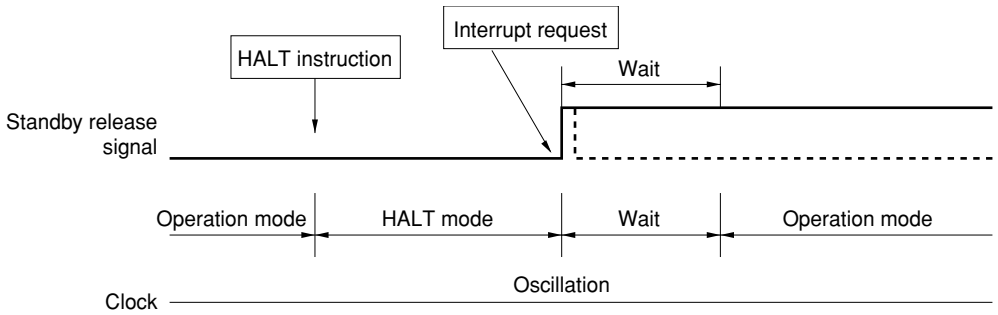
(2) HALT mode release

The HALT mode can be released with the following three types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 16-2. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has released the standby mode is acknowledged.

2. Wait times are as follows:

- When vectored interrupt service is carried out: 8 or 9 clocks
- When vectored interrupt service is not carried out: 2 or 3 clocks

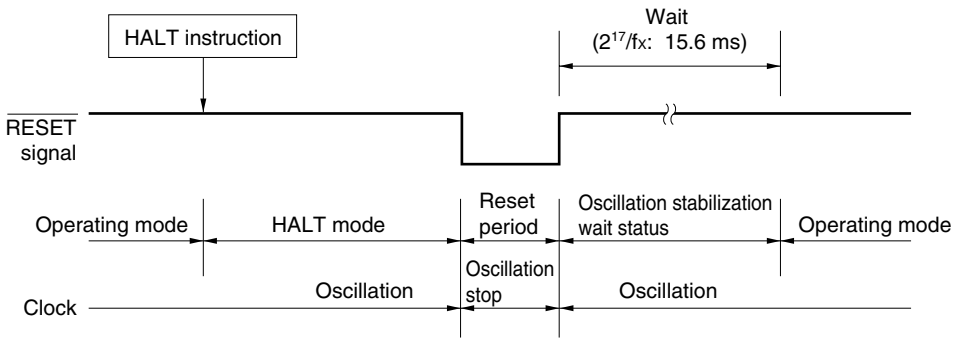
(b) Release by non-maskable interrupt request

When a non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

(c) Release by $\overline{\text{RESET}}$ input

When $\overline{\text{RESET}}$ signal is input, HALT mode is released. And, as in the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 16-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 8.38 MHz.

Table 16-2. Operation After HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	—	—	×	×	Interrupt service execution
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

16.2.2 STOP mode

(1) STOP mode setting and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD1} via a pull-up resistor to minimize the leakage current at the crystal oscillator. Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

2. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operating mode is set.

The operating status in the STOP mode is described below.

Table 16-3. STOP Mode Operating Status

STOP Mode Setting / Item		With Subsystem Clock	Without Subsystem Clock
Clock generator		Only main system clock oscillation is stopped.	
CPU		Operation stops.	
Port (output latch)		Status before STOP mode setting is held.	
16-bit timer/event counter		Operation stops.	
8-bit timer/event counter		Operable only when TI50, TI51 are selected as count clock.	
Watch timer		Operable when f_{XT} is selected as count clock.	Operation stops.
Watchdog timer		Operation stops.	
Clock output/buzzer output		PCL and BUZ at low level.	
A/D converter		Operation stops.	
Serial interface	Other than UART	Operable only when externally supplied input clock is specified as the serial clock.	
	UART	Operation stops (transmit shift register 0 (TXS0), receive shift register 0 (RX0), and receive buffer register 0 (RXB0) hold the value just before the clock stop).	
External interrupt		Operable	

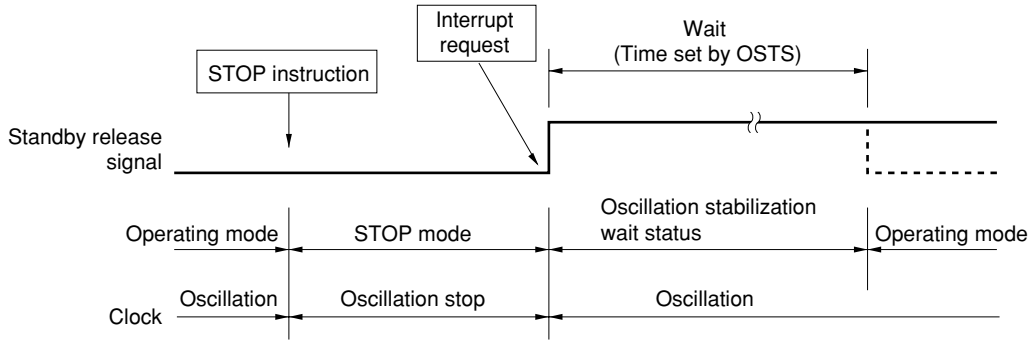
(2) STOP mode release

The STOP mode can be released by the following two types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 16-4. STOP Mode Release by Interrupt Request Generation

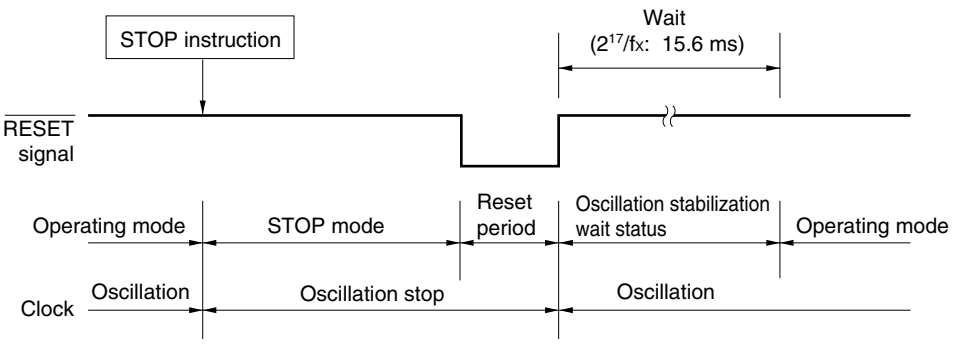


Remark The broken line indicates the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

The STOP mode is released when $\overline{\text{RESET}}$ signal is input, and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 16-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 8.38 MHz.

Table 16-4. Operation After STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

CHAPTER 17 RESET FUNCTION

17.1 Reset Function

The following two operations are available to generate the reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop time detection

External reset and internal reset 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 is set to the status shown in Table 17-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution starts after the lapse of oscillation stabilization time (2^{17}/fx). The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time (2^{17}/fx) (see **Figures 17-2** to **17-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.
3. When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 17-1. Block Diagram of Reset Function

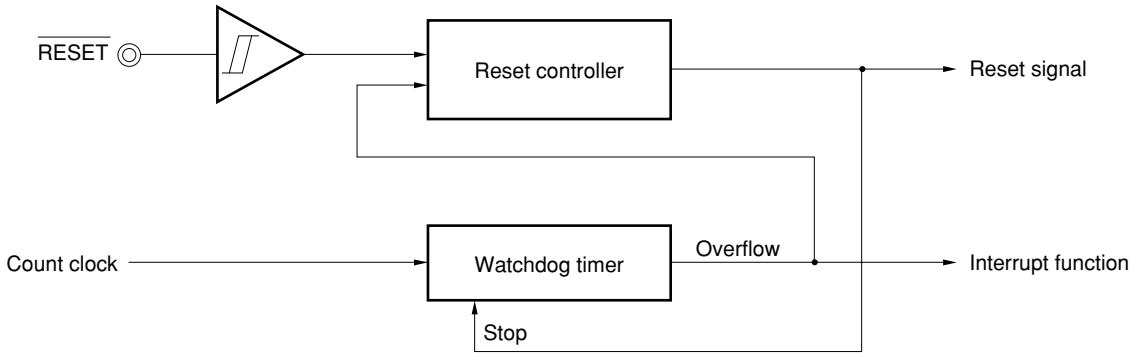


Figure 17-2. Timing of Reset by $\overline{\text{RESET}}$ Input

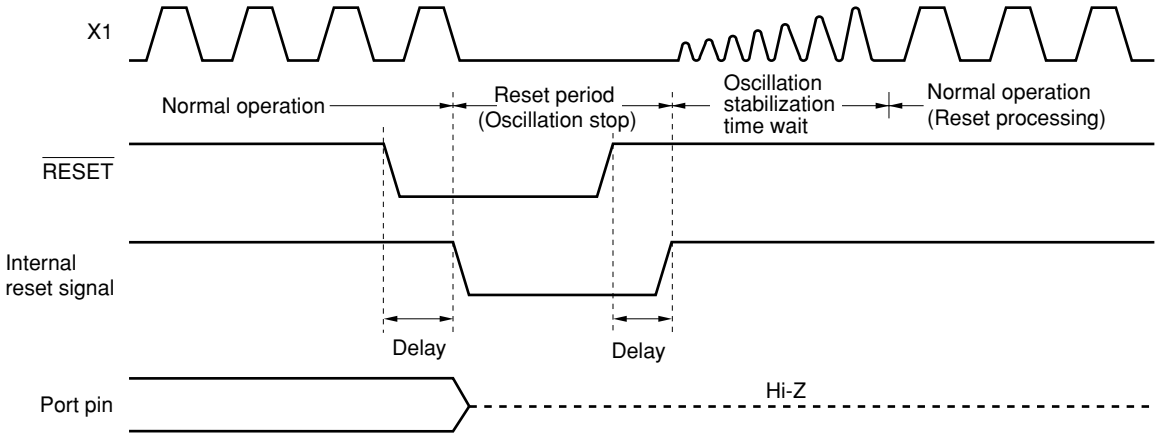


Figure 17-3. Timing of Reset Due to Watchdog Timer Overflow

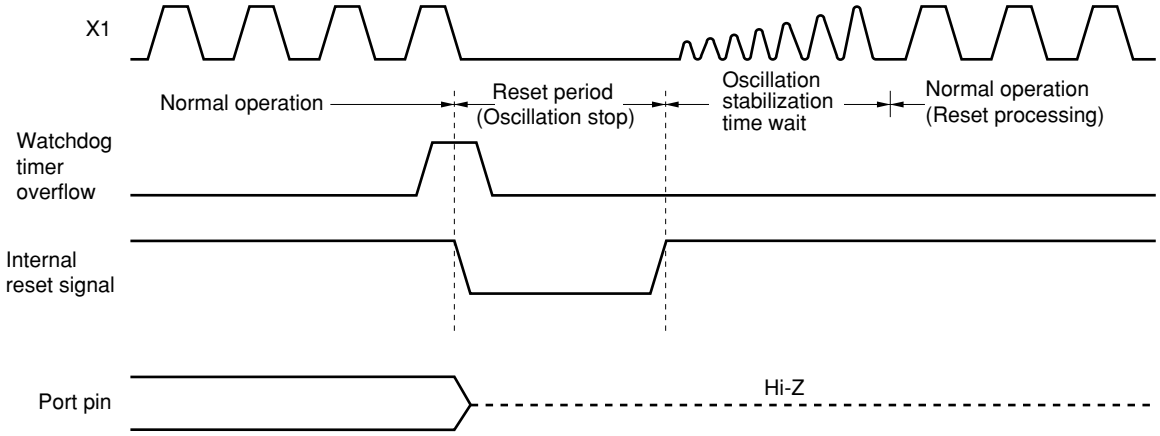


Figure 17-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

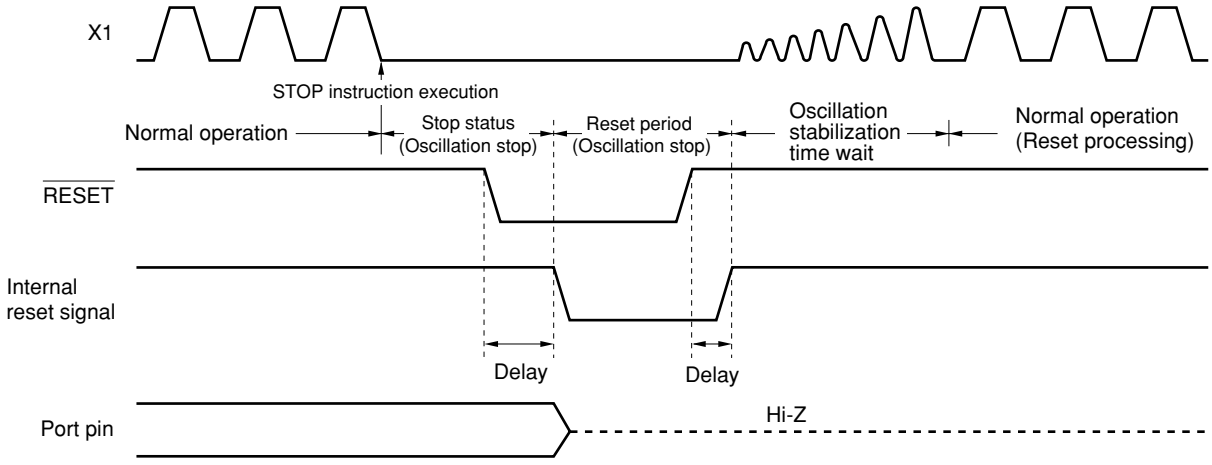


Table 17-1. Hardware Statuses After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Port (output latch)		00H
Port mode registers (PM0, PM2 to PM5, PM7)		FFH
Pull-up resistor option registers (PU0, PU2 to PU5, PU7)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		CFH[Note 3]
Memory expansion mode register (MEM)		00H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer counter (TM0)	0000H
	Capture/compare registers (CR00, CR01)	Undefined
	Prescaler mode register (PRM0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
8-bit timer/event counter	Timer counters (TM50, TM51)	00H
	Compare registers (CR50, CR51)	Undefined
	Clock select registers (TCL50, TCL51)	00H
	Mode control registers (TMC50, TMC51)	00H
Watch timer	Operation mode register (WTM)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. Although the initial value is CFH, use the following value to be set for each version.

μPD780021AS, 780031AS: 42H
μPD780022AS, 780032AS: 44H
μPD780023AS, 780033AS: C6H
μPD780024AS, 780034AS: C8H
μPD78F0034BS: Value for mask ROM versions

Table 17-1. Hardware Statuses After Reset (2/2)

Hardware		Status After Reset
Clock output/buzzer output controller	Clock output select register (CKS)	00H
A/D converter	Conversion result register (ADCR0)	00H
	Mode register (ADM0)	00H
	Analog input channel specification register (ADS0)	00H
Serial interface (UART0)	Asynchronous serial interface mode register (ASIM0)	00H
	Asynchronous serial interface status register (ASIS0)	00H
	Baud rate generator control register (BRGC0)	00H
	Transmit shift register (TXS0)	FFH
	Receive buffer register (RXB0)	
Serial interface (SIO3)	Shift registers (SIO30, SIO31)	Undefined
	Operating mode registers (CSIM30, CSIM31)	00H
Interrupt	Request flag registers (IF0L, IF0H, IF1L)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

CHAPTER 18 μPD78F0034BS

The μPD78F0034BS is provided as the flash memory version of the μPD780024AS, 780034AS Subseries.

The μPD78F0034BS replaces the internal mask ROM of the μPD780034BS with flash memory to which a program can be written, erased and overwritten while mounted on the board. Table 18-1 lists the differences among the μPD78F0034BS and the mask ROM versions.

Table 18-1. Differences Among μPD78F0034BS and Mask ROM Versions

Item	μPD78F0034BS	Mask ROM Versions	
		μPD780034AS Subseries	μPD780024AS Subseries
Internal ROM configuration	Flash memory	Mask ROM	
Internal ROM capacity	32 KB[Note]	μPD780031AS: 8 KB μPD780032AS: 16 KB μPD780033AS: 24 KB μPD780034AS: 32 KB	μPD780021AS: 8 KB μPD780022AS: 16 KB μPD780023AS: 24 KB μPD780024AS: 32 KB
Internal high-speed RAM capacity	1024 bytes[Note]	μPD780031AS: 512 bytes μPD780032AS: 512 bytes μPD780033AS: 1024 bytes μPD780034AS: 1024 bytes	μPD780021AS: 512 bytes μPD780022AS: 512 bytes μPD780023AS: 1024 bytes μPD780024AS: 1024 bytes
Resolution of A/D converter	10 bits		8 bits
IC pin	None	Available	
V_{PP} pin	Available	None	
Electrical specifications	Refer to data sheet of each product.		

Note The same capacity as the mask ROM versions can be specified by means of the memory size switching register (IMS).

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

18.1 Memory Size Switching Register

The μPD78F0034BS allows users to select the internal memory capacity using the memory size switching register (IMS) so that the same memory map as that of the μPD780021AS, 780022AS, 780023AS, 780024AS and μPD780031AS, 780032AS, 780033AS, 780034AS with a different size of internal memory capacity can be achieved.

IMS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to CFH.

Caution The initial value of IMS is "setting prohibited (CFH)". Be sure to set the value of the relevant mask ROM versions at initialization.

Figure 18-1. Format of Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
0	1	0	512 bytes
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	0	1	0	8 KB
0	1	0	0	16 KB
0	1	1	0	24 KB
1	0	0	0	32 KB
1	1	1	1	60 KB (setting prohibited)
Other than above				Setting prohibited

The IMS settings to obtain the same memory map as mask ROM versions are shown in Table 18-2.

Table 18-2. Memory Size Switching Register Settings

Target Mask ROM Versions	IMS Setting
μPD780021AS, 780031AS	42H
μPD780022AS, 780032AS	44H
μPD780023AS, 780033AS	C6H
μPD780024AS, 780034AS	C8H

Caution When using the mask ROM versions, be sure to set the value indicated in Table 18-2 to IMS.

18.2 Flash Memory Programming

On-board writing of flash memory (with device mounted on target system) is supported.

On-board writing is done after connecting a dedicated flash programmer (Flashpro III (FL-PR3, PG-FP3)) to the host machine and target system.

Moreover, writing to flash memory can also be performed using a flash memory writing adapter connected to Flashpro III.

Remark FL-PR3 is a product of Naito Densei Machida Mfg. Co., Ltd.

18.2.1 Selection of communication mode

Writing to flash memory is performed using Flashpro III and serial communication. Select the communication mode for writing from Table 18-3. For the selection of the communication mode, a format like the one shown in Figure 18-2 is used. The communication modes are selected with the V_{PP} pulse numbers shown in Table 18-3.

Table 18-3. Communication Mode List

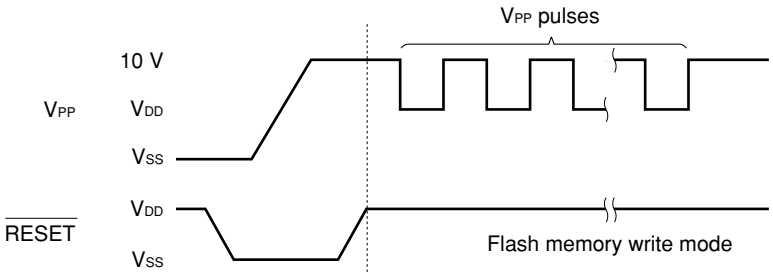
Communication Mode	Number of Channels	Pin Used[Note]	Number of V_{PP} Pulses
3-wire serial I/O	1	SI30/P20 SO30/P21 $\overline{SCK30}$/P22	0
		SI30/P20 SO30/P21 $\overline{SCK30}$/P22 HS/P25	3
3-wire serial I/O	1	SI31/P34 SO31/P35 $\overline{SCK31}$/P36	1
UART	1	RxD0/P23 TxD0/P24	8
Pseudo 3-wire serial I/O	1	P72/TI50/TO50 (Serial clock input) P71/TI01 (Serial data output) P70/TI00/TO0 (Serial data input)	12

Note When the flash memory programming mode is entered, all pins that are not used for flash memory programming become the same status as the status immediately after reset. Therefore, when the external device connected to each port does not acknowledge the port status immediately after reset, pin connections such as connecting to V_{DD0} or V_{DD1} via a resistor or connecting to V_{SS0} or V_{SS1} via a resistor are required.

Cautions 1. Be sure to select the number of V_{PP} pulses shown in Table 18-3 for the communication mode.

2. If performing write operations to flash memory with the UART communication mode, set the main system clock oscillation frequency to 3 MHz or higher.

Figure 18-2. Format of Communication Mode Selection



18.2.2 Flash memory programming function

Flash memory writing is performed through command and data transmit/receive operations using the selected communication mode. The main functions are listed in Table 18-4.

Table 18-4. Main Functions of Flash Memory Programming

Function	Description
Reset	Used to detect write stop and transmission synchronization.
Batch verify	Compares entire memory contents and input data.
Batch erase	Erases the entire memory contents.
Batch blank check	Checks the deletion status of the entire memory.
High-speed write	Performs writing to flash memory according to write start address and number of write data (bytes).
Continuous write	Performs continuous write operations using the data input with high-speed write operation.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Erase time setting	Inputs the memory erase time.
Baud rate setting	Sets the communication rate when the UART mode is used.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

18.2.3 Connection of Flashpro III

Connection of the Flashpro III and the μPD78F0034BS differs depending on communication mode (3-wire serial I/O and UART). Each type of connection is shown in Figures 18-3 to 18-6.

Figure 18-3. Connection of Flashpro III in 3-Wire Serial I/O Mode

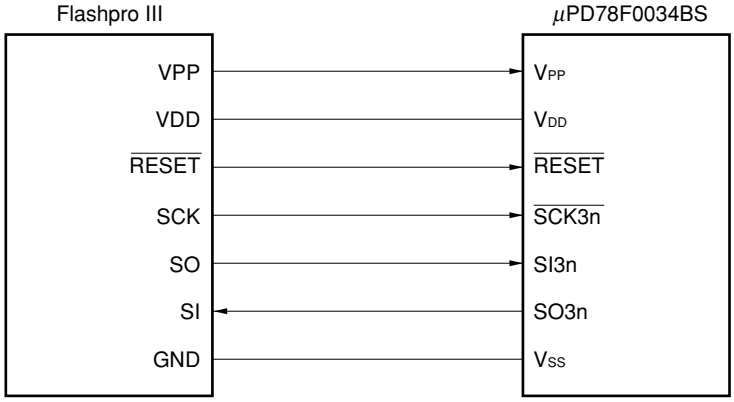


Figure 18-4. Connection of Flashpro III in 3-Wire Serial I/O Mode (Using Handshake)

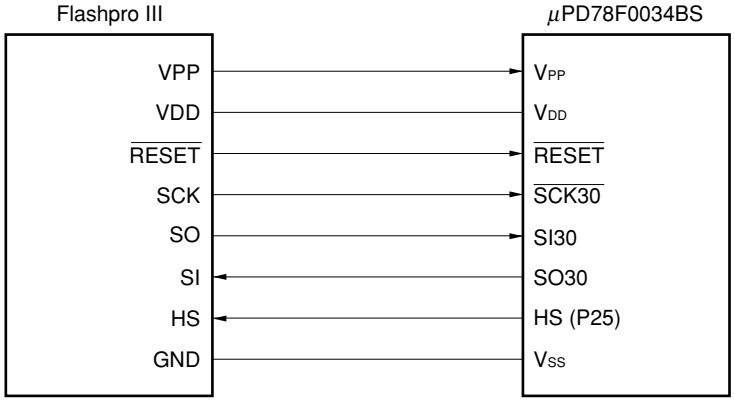


Figure 18-5. Connection of Flashpro III in UART Mode

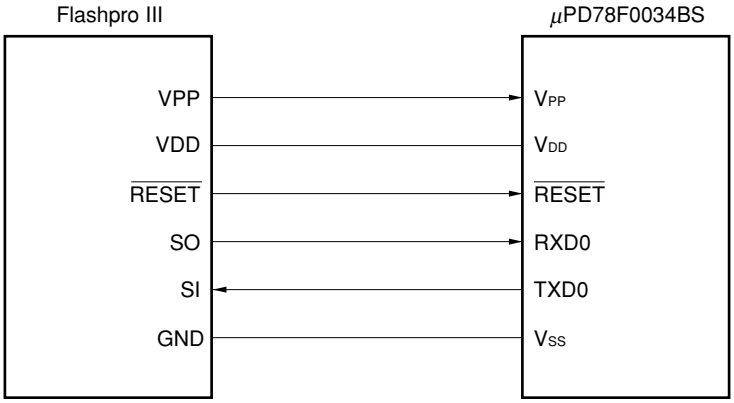
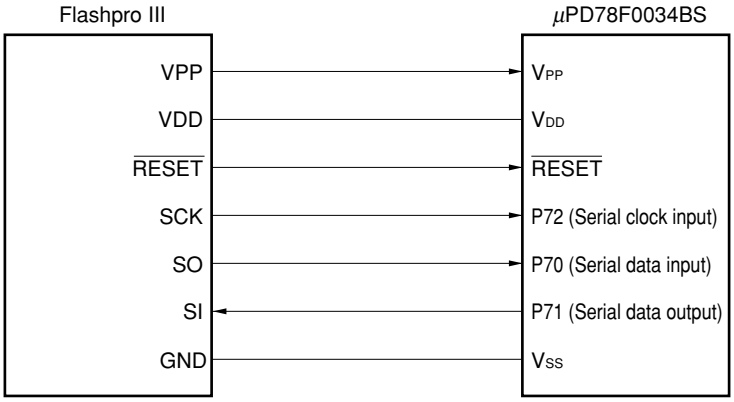


Figure 18-6. Connection of Flashpro III in Pseudo 3-Wire Serial I/O Mode



CHAPTER 19 INSTRUCTION SET

This chapter lists each instruction set of the μPD780024AS, 780034AS Subseries in table form. For details of its operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

19.1 Conventions

19.1.1 Operand identifiers and specification methods

Operands are written in “Operand” column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for specification.

Table 19-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol**Note**
sfrp	Special function register symbol (16-bit manipulatable register even addresses only)**Note**
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-5 Special Function Register List**.

19.1.2 Description of “operation” column

A: A register; 8-bit accumulator
X: X register
B: B register
C: C register
D: D register
E: E register
H: H register
L: L register
AX: AX register pair; 16-bit accumulator
BC: BC register pair
DE: DE register pair
HL: HL register pair
PC: Program counter
SP: Stack pointer
PSW: Program status word
CY: Carry flag
AC: Auxiliary carry flag
Z: Zero flag
RBS: Register bank select flag
IE: Interrupt request enable flag
NMIS: Non-maskable interrupt servicing flag
( ): Memory contents indicated by address or register contents in parentheses
X_H, X_L: Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$: Logical product (AND)
$\vee$: Logical sum (OR)
$\veebar$: Exclusive logical sum (exclusive OR)
$\overline{\quad}$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

19.1.3 Description of “flag operation” column

(Blank): Not affected
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is restored

19.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9 + n	A ← (addr16)	
		!addr16, A	3	8	9 + m	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5 + n	A ← (DE)	
		[DE], A	1	4	5 + m	(DE) ← A	
		A, [HL]	1	4	5 + n	A ← (HL)	
		[HL], A	1	4	5 + m	(HL) ← A	
		A, [HL + byte]	2	8	9 + n	A ← (HL + byte)	
		[HL + byte], A	2	8	9 + m	(HL + byte) ← A	
		A, [HL + B]	1	6	7 + n	A ← (HL + B)	
		[HL + B], A	1	6	7 + m	(HL + B) ← A	
		A, [HL + C]	1	6	7 + n	A ← (HL + C)	
		[HL + C], A	1	6	7 + m	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ (sfr)	
		A, !addr16	3	8	10 + n + m	A ↔ (addr16)	
		A, [DE]	1	4	6 + n + m	A ↔ (DE)	
		A, [HL]	1	4	6 + n + m	A ↔ (HL)	
		A, [HL + byte]	2	8	10 + n + m	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10 + n + m	A ↔ (HL + B)	
		A, [HL + C]	2	8	10 + n + m	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12 + 2n	AX ← (addr16)	
		!addr16, AX	3	10	12 + 2m	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r	× × ×
		r, A	2	4	–	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	× × ×
		r, A	2	4	–	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16) + CY	× × ×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9 + n	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5 + n	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9 + n	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9 + n	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9 + n	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A∨byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨byte	×
		A, r Note 3	2	4	–	A ← A∨r	×
		r, A	2	4	–	r ← r∨A	×
		A, saddr	2	4	5	A ← A∨(saddr)	×
		A, !addr16	3	8	9 + n	A ← A∨(addr16)	×
		A, [HL]	1	4	5 + n	A ← A∨(HL)	×
		A, [HL + byte]	2	8	9 + n	A ← A∨(HL + byte)	×
		A, [HL + B]	2	8	9 + n	A ← A∨(HL + B)	×
		A, [HL + C]	2	8	9 + n	A ← A∨(HL + C)	×
	XOR	A, #byte	2	4	–	A ← A⊻byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)⊻byte	×
		A, r Note 3	2	4	–	A ← A⊻r	×
		r, A	2	4	–	r ← r⊻A	×
		A, saddr	2	4	5	A ← A⊻(saddr)	×
		A, !addr16	3	8	9 + n	A ← A⊻(addr16)	×
		A, [HL]	1	4	5 + n	A ← A⊻(HL)	×
		A, [HL + byte]	2	8	9 + n	A ← A⊻(HL + byte)	×
		A, [HL + B]	2	8	9 + n	A ← A⊻(HL + B)	×
		A, [HL + C]	2	8	9 + n	A ← A⊻(HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9 + n	A – (addr16)	× × ×
		A, [HL]	1	4	5 + n	A – (HL)	× × ×
		A, [HL + byte]	2	8	9 + n	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9 + n	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9 + n	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROL	A, 1	1	2	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$ time			×
	RORC	A, 1	1	2	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROLC	A, 1	1	2	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$ time			×
	ROR4	[HL]	2	10	12 + n + m	$A_{3-0} \leftarrow (HL)_{3-0}$, $(HL)_{7-4} \leftarrow A_{3-0}$, $(HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12 + n + m	$A_{3-0} \leftarrow (HL)_{7-4}$, $(HL)_{3-0} \leftarrow A_{3-0}$, $(HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipu- late	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7 + n	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8 + n + m	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7 + n	CY ← CY ∧ (HL).bit	×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	6	7 + n	CY ← CY ∨ (HL).bit	×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ∀ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∀ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∀ A.bit	×
		CY, PSW. bit	3	–	7	CY ← CY ∀ PSW.bit	×
		CY, [HL].bit	2	6	7 + n	CY ← CY ∀ (HL).bit	×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1	
		sfr.bit	3	–	8	sfr.bit ← 1	
		A.bit	2	4	–	A.bit ← 1	
		PSW.bit	2	–	6	PSW.bit ← 1	× × ×
		[HL].bit	2	6	8 + n + m	(HL).bit ← 1	
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0	
		sfr.bit	3	–	8	sfr.bit ← 0	
		A.bit	2	4	–	A.bit ← 0	
		PSW.bit	2	–	6	PSW.bit ← 0	× × ×
		[HL].bit	2	6	8 + n + m	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$			
	CALLF	!addr11	2	5	–	$(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP-2$			
	CALLT	[addr5]	1	6	–	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$			
	BRK		1	6	–	$(SP-1) \leftarrow PSW$, $(SP-2) \leftarrow (PC+1)_H$, $(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (003FH), $PC_L \leftarrow$ (003EH), $SP \leftarrow SP-3$, $IE \leftarrow 0$			
	RET		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	RETI		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$, $NMIS \leftarrow 0$	R	R	R
	RETB		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$			
		rp	1	4	–	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$			
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R	R	R
		rp	1	4	–	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	MOVW	SP, #word	4	–	10	$SP \leftarrow$ word			
		SP, AX	2	–	8	$SP \leftarrow AX$			
		AX, SP	2	–	8	$AX \leftarrow SP$			
Unconditional branch	**BR**	!addr16	3	6	–	$PC \leftarrow$ addr16			
		$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8			
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
Conditional branch	**BC**	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Condi-tional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12 + n + m	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr. $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if(saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

19.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

First Operand \ Second Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand / First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

APPENDIX A DIFFERENCES BETWEEN μPD780024A, 780024AS, 780034A, AND 780034AS SUBSERIES

Table A-1 shows the major differences between μPD780024A, 780024AS, 780034A, and 780034AS Subseries.

Table A-1. Major Differences Between μPD780024A, 780024AS, 780034A, and 780034AS Subseries

Item \ Part Number		μPD780024A Subseries	μPD780034A Subseries	μPD780024AS Subseries	μPD780034AS Subseries
ROM capacity		8 KB/16 KB/24 KB/32 KB			
Internal high-speed RAM capacity		512 bytes/1,024 bytes			
Flash memory version		μPD78F0034B		μPD78F0034BS	
I^2C bus version		μPD780024AY Subseries	μPD780034AY Subseries	None	
Minimum instruction execution time		0.238 μs: V_{DD} = 4.5 to 5.5 V 0.400 μs: V_{DD} = 2.7 to 5.5 V 1.60 μs: V_{DD} = 1.8 to 5.5 V			
Operating voltage range		1.8 to 5.5 V			
I/O ports		Total: 51 Input: 8 I/O: 43 (5 V tolerance N-ch open drain: 4)		Total: 39 Input: 4 I/O: 35	
Timer/counter		16 bits × 1 ch 8 bits × 2 ch Watch timer × 1 ch Watchdog timer × 1 ch			
Serial interface		3-wire serial I/O mode × 2 ch UART mode × 1 ch			
A/D converter		8 bits × 8 ch	10 bits × 8 ch	8 bits × 4 ch	10 bits × 4 ch
Interrupt	Maskable	Internal: 13, external: 5			
	Non-maskable	1			
	Software	1			
External device expansion function		Time division method Expansion up to F7FFH is possible.		None	
Mask option		Pull-up resistor can be specified for P30 to P33		None	
Package		• 64-pin plastic SDIP • 64-pin plastic QFP • 64-pin plastic TQFP • 64-pin plastic LQFP		52-pin plastic LQFP	
Emulation board		IE-780034-NS-EM1			
Emulation probe (conversion adapter)		• NP-64CW • NP-64GC (EV-9200GC-64) NP-64GC-TQ (TGC-064SAP) • NP-64GK (TGK-064SBP) • NP-H64GB-TQ (TGB-064SDP)		NP-H52GB-TQ (TGB-052SBP) * A conversion board is required to connect the probe to the emulation board.	
Device file		DF780024	DF780034	DF780024	DF780034
Flash memory writing adapter		• FA-64CW • FA-64GC-8BS, FA64GC-AB8 • FA-64GK-9ET • FA-64GB-8EU		FA-52GB-8ET	

APPENDIX B DEVELOPMENT TOOLS

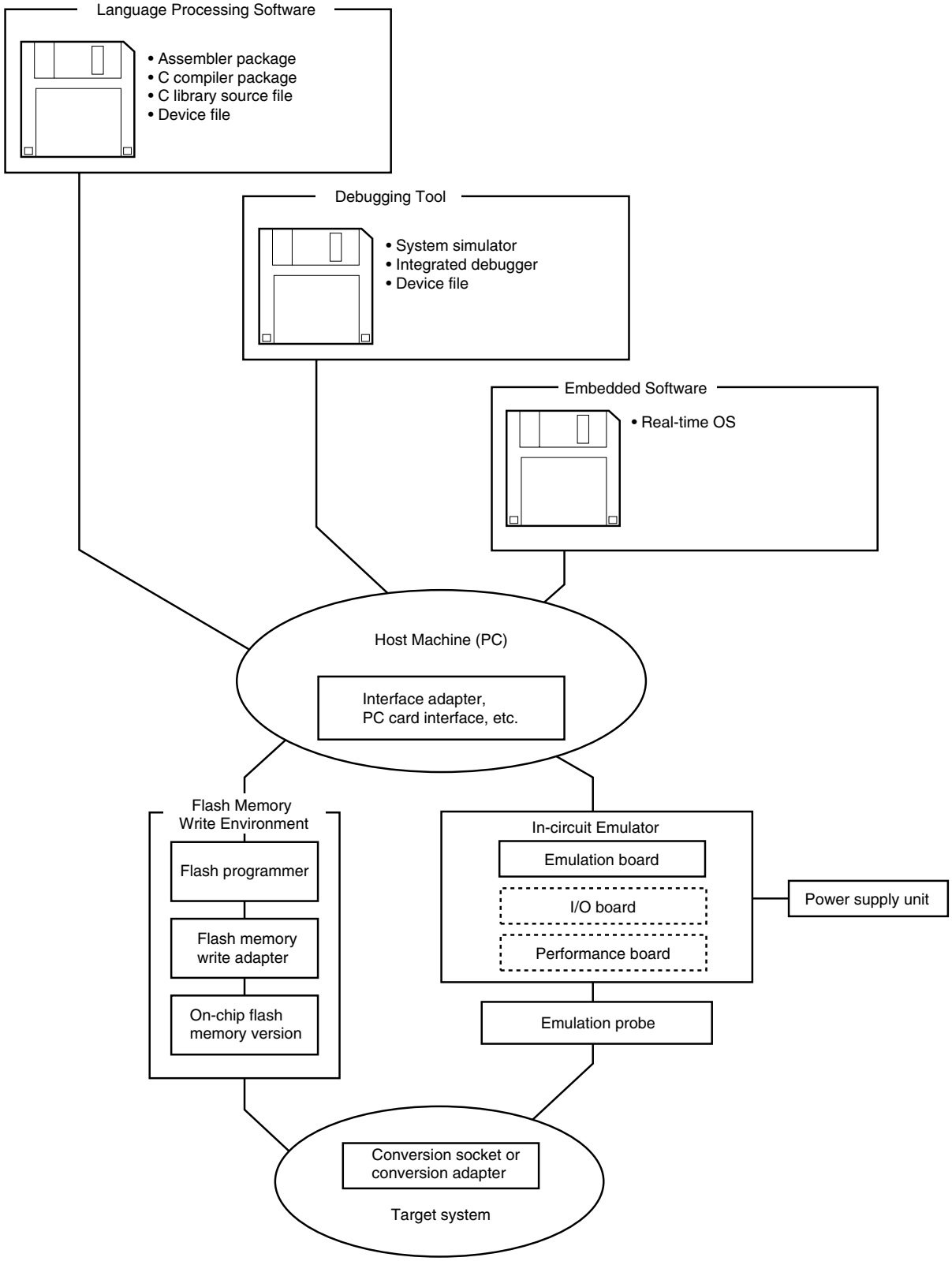
The following development tools are available for the development of systems that employ the μPD780024AS, 780034AS Subseries.

Figure B-1 shows the development tool configuration.

- **Support for PC98-NX series**
  Unless otherwise specified, products compatible with IBM PC/AT™ computers are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT computers.

- **Windows**
  Unless otherwise specified, “Windows” means the following OSs.
  - Windows 3.1
  - Windows 95, 98, 2000
  - Windows NT™ Ver 4.0

Figure B-1. Development Tool Configuration



Remark Items in broken line boxes differ according to the development environment. See **B.3.1 Hardware**.

B.1 Language Processing Software

SP78K0 78K/0 Series Software Package	This is a software package that includes the development tools common to the 78K/0 Series.
	Part number: μS××××SP78K0
RA78K0 Assembler Package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with an optional device file (DF780024 or DF780034). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××RA78K0
CC78K0 C Compiler Package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an optional assembler package and device file. **<Precaution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××CC78K0
DF780024**Note** DF780034**Note** Device File	This file contains information peculiar to the device. This device file should be used in combination with an optional tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0). Corresponding OS and host machine differ depending on the tool to be used with. • DF780024: for μPD780024A, 780024AY, 780024AS Subseries • DF780034: for μPD780034A, 780034AY, 780034AS Subseries
	Part number: μS××××DF780034
CC78K0-L C Library Source File	This is a source file of functions configuring the object library included in the C compiler package. This file is required to match the object library included in C compiler package to the customer's specifications. Operating environment for the source file is not dependent on the OS.
	Part number: μS××××CC78K0-L

Note The DF780024 and DF780034 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0.

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0

××××	Host machine	OS	Supply medium
AB17	PC-9800 Series, IBM PC/AT or compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

μS××××RA78K0
μS××××CC78K0

××××	Host machine	OS	Supply medium
AB13	PC-9800 Series, IBM PC/AT or compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 Series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF780024
μS××××DF780034
μS××××CC78K0-L

××××	Host machine	OS	Supply medium
AB13	PC-9800 Series, IBM PC/AT or compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
3P16	HP9000 Series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

B.2 Flash Memory Writing Tools

Flashpro III (part number: FL-PR3, PG-FP3) Flash Programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-52GB-8ET Flash Memory Writing Adapter	Flash memory writing adapter used connected to the Flashpro III. • FA-52GB-8ET: 52-pin plastic LQFP (GB-8ET type)

Remark FL-PR3 and FA-52GB-8ET are products of Naito Densei Machida Mfg. Co., Ltd.
Inquiry: Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191)

B.3 Debugging Tools

B.3.1 Hardware

IE-78K0-NS In-Circuit Emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0-NS). This emulator should be used in combination with power supply unit, emulation probe, and interface adapter which is required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance Board	This board is connected to the IE-78K0-NS to expand its functions. Adding this board adds a coverage function and enhances debugging functions such as tracer and timer functions.
IE-78K0-NS-A In-Circuit Emulator	This is an in-circuit emulator that combines IE-78K0-NS and IE-78K0-NS-PA.
IE-70000-MC-PS-B Power Supply Unit	This adapter is used for supplying power from a receptacle of 100 V to 240 V AC.
IE-70000-98-IF-C Interface Adapter	This adapter is required when using the PC-9800 series computer (except notebook type) as the host machine (C bus supported).
IE-70000-CD-IF-A PC Card Interface	This is PC card and interface cable required when using notebook-type computer as the host machine (PCMCIA socket supported).
IE-70000-PC-IF-C Interface Adapter	This adapter is required when using the IBM PC/AT compatible computers as the host machine (ISA bus supported).
IE-70000-PCI-IF-A Interface Adapter	This adapter is required when using a computer with PCI bus as the host machine.
IE-780034-NS-EM1 Emulation Board	This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
780034AS 52Pin Board Convension Board	This conversion board is used to connect the emulation probe to the emulation board.
NP-H52GB-TQ Emulation Probe	This probe is used to connect the in-circuit emulator to a target system and is designed for use with 52-pin plastic LQFP (GB-8ET type).
TGB-052SBP Conversion Adapter	This conversion socket connects the NP-H52GB-TQ to a target system board designed for a 52-pin plastic LQFP (GB-8ET type).

Remarks 1. NP-H52GB-TQ is a product of Naito Densei Machida Mfg. Co., Ltd.
Inquiry: Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191)

2. TGB-052SBP is a product of TOKYO ELETECH CORPORATION.
Inquiry: Daimaru Kogyo, Ltd.: Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

3. TGB-052SBP is sold in one units.

B.3.2 Software

SM78K0 System Simulator	This system simulator is used to perform debugging at C source level or assembler level while simulating the operation of the target system on a host machine. This simulator runs on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with the optional device file (DF780024 or DF780034).
	Part number: μS××××SM78K0
ID78K0-NS Integrated Debugger (supporting in-circuit emulator IE-78K0-NS(-A))	This debugger is a control program to debug 78K/0 Series microcontrollers. It adopts a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif™. It also has an enhanced debugging function for C language programs, and thus trace results can be displayed on screen in C-language level by using the windows integration function which links a trace result with its source program, disassembled display, and memory display. In addition, by incorporating function expansion modules such as task debugger and system performance analyzer, the efficiency of debugging programs, which run on real-time OSs can be improved. It should be used in combination with the optional device file.
	Part number: μS××××ID78K0-NS

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0
μS××××ID78K0-NS

××××	Host machine	OS	Supply medium
AB13	PC-9800 series, IBM PC/AT or compatibles	Japanese Windows	3.5-inch 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

APPENDIX C EMBEDDED SOFTWARE

For efficient program development and maintenance of the μPD780024AS, 780034AS Subseries, the following embedded software products are available.

Real-Time OS

RX78K0 Real-time OS	RX78K/0 is a real-time OS conforming to the μITRON specifications. Tool (configurator) for generating nucleus of RX78K0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K0) and device file (DF780024 or DF780034). **<Precaution when using RX78K/0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K0, fill in the purchase application form in advance and sign the user agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product outline	Maximum number for use in mass production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

××××	Host machine	OS	Supply medium
AA13	PC-9800 Series	Japanese Windows[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT or compatibles	Japanese Windows[Note]	3.5-inch 2HD FD
BB13		English Windows[Note]	
3P16	HP9000 Series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4),	3.5-inch 2HD FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT

Note Can also be operated in DOS environment.

APPENDIX D REGISTER INDEX

D.1 Register Name Index

[R]

[S]

[T]

[W]

D.2 Register Symbol Index

NEC

Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel. FAX

Address

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

P.R. China
NEC Electronics Shanghai, Ltd.
Fax: +86-21-6841-1137

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑