USER'S MANUAL

NEC

μPD780024, 780034, 780024Y, 780034Y SUBSERIES

8-BIT SINGLE-CHIP MICROCONTROLLER

(PRELIMINARY)

μPD780021
μPD780022
μPD780023
μPD780024
μPD780021(A)
μPD780022(A)
μPD780023(A)
μPD780024(A)

μPD780031
μPD780032
μPD780033
μPD780034
μPD780031(A)
μPD780032(A)
μPD780033(A)
μPD780034(A)
μPD78F0034

μPD780021Y
μPD780022Y
μPD780023Y
μPD780024Y
μPD780021Y(A)
μPD780022Y(A)
μPD780023Y(A)
μPD780024Y(A)

μPD780031Y
μPD780032Y
μPD780033Y
μPD780034Y
μPD780031Y(A)
μPD780032Y(A)
μPD780033Y(A)
μPD780034Y(A)
μPD78F0034Y

A

Document No. U12022EJ1V0UM00 (1st edition)
Date Published December 1996 N
Printed in Japan

The application circuits and their parameters are for reference only and are not intended for use in actual design-ins.

Purchase of NEC I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

M7 96.5

B

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.

6427525 0100250 584

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:

μPD78F0034CW, 78F0034GC-AB8, 78F0034GK-8A8
μPD78F0034YCW, 78F0034YGC-AB8, 78F0034YGK-8A8

The customer must judge the need for license:

μPD780021CW-xxx, 780022CW-xxx, 780023CW-xxx, 780024CW-xxx
μPD780021GC-xxx-AB8, 780022GC-xxx-AB8, 780023GC-xxx-AB8, 780024GC-xxx-AB8
μPD780021GK-xxx-8A8, 780022GK-xxx-8A8, 780023GK-xxx-8A8, 780024GK-xxx-8A8
μPD780021CW(A)-xxx, 780022CW(A)-xxx, 780023CW(A)-xxx, 780024CW(A)-xxx
μPD780021GC(A)-xxx-AB8, 780022GC(A)-xxx-AB8, 780023GC(A)-xxx-AB8, 780024GC(A)-xxx-AB8
μPD780021GK(A)-xxx-8A8, 780022GK(A)-xxx-8A8, 780023GK(A)-xxx-8A8, 780024GK(A)-xxx-8A8
μPD780021YCW-xxx, 780022YCW-xxx, 780023YCW-xxx, 780024YCW-xxx
μPD780021YGC-xxx-AB8, 780022YGC-xxx-AB8, 780023YGC-xxx-AB8, 780024YGC-xxx-AB8
μPD780021YGK-xxx-8A8, 780022YGK-xxx-8A8, 780023YGK-xxx-8A8, 780024YGK-xxx-8A8
μPD780021YCW(A)-xxx, 780022YCW(A)-xxx, 780023YCW(A)-xxx, 780024YCW(A)-xxx
μPD780021YGC(A)-xxx-AB8, 780022YGC(A)-xxx-AB8, 780023YGC(A)-xxx-AB8, 780024YGC(A)-xxx-AB8
μPD780021YGK(A)-xxx-8A8, 780022YGK(A)-xxx-8A8, 780023YGK(A)-xxx-8A8, 780024YGK(A)-xxx-8A8
μPD780031CW-xxx, 780032CW-xxx, 780033CW-xxx, 780034CW-xxx
μPD780031GC-xxx-AB8, 780032GC-xxx-AB8, 780033GC-xxx-AB8, 780034GC-xxx-AB8
μPD780031GK-xxx-8A8, 780032GK-xxx-8A8, 780033GK-xxx-8A8, 780034GK-xxx-8A8
μPD780031CW(A)-xxx, 780032CW(A)-xxx, 780033CW(A)-xxx, 780034CW(A)-xxx
μPD780031GC(A)-xxx-AB8, 780032GC(A)-xxx-AB8, 780033GC(A)-xxx-AB8, 780034GC(A)-xxx-AB8
μPD780031GK(A)-xxx-8A8, 780032GK(A)-xxx-8A8, 780033GK(A)-xxx-8A8, 780034GK(A)-xxx-8A8
μPD780031YCW-xxx, 780032YCW-xxx, 780033YCW-xxx, 780034YCW-xxx
μPD780031YGC-xxx-AB8, 780032YGC-xxx-AB8, 780033YGC-xxx-AB8, 780034YGC-xxx-AB8
μPD780031YGK-xxx-8A8, 780032YGK-xxx-8A8, 780033YGK-xxx-8A8, 780034YGK-xxx-8A8
μPD780031YCW(A)-xxx, 780032YCW(A)-xxx, 780033YCW(A)-xxx, 780034YCW(A)-xxx
μPD780031YGC(A)-xxx-AB8, 780032YGC(A)-xxx-AB8, 780033YGC(A)-xxx-AB8, 780034YGC(A)-xxx-AB8
μPD780031YGK(A)-xxx-8A8, 780032YGK(A)-xxx-8A8, 780033YGK(A)-xxx-8A8, 780034YGK(A)-xxx-8A8

D

INTRODUCTION

Readers

This manual has been prepared for user engineers who want to understand the functions of the μPD780024, 780034, 780024Y and 780034Y subseries and design and develop its application systems and programs

μPD780024 Subseries :	μPD780021, 780022, 780023, 780024
	μPD780021(A), 780022(A), 780023(A), 780024(A)
μPD780034 Subseries :	μPD780031, 780032, 780033, 780034, 78F0034
	μPD780031(A), 780032(A), 780033(A), 780034(A)
μPD780024Y Subseries:	μPD780021Y, 780022Y, 780023Y, 780024Y
	μPD780021Y(A), 780022Y(A), 780023Y(A), 780024Y(A)
μPD780034Y Subseries:	μPD780031Y, 780032Y, 780033Y, 780034Y, 78F0034Y
	μPD780031Y(A), 780032Y(A), 780033Y(A), 780034Y(A)

Purpose

This manual is intended for users to understand the functions described in the Organization below.

Organization

The μPD780024, 780034, and 780034Y subseries manual is separated into two parts: this manual and the instruction edition (common to the 78K/0 series).

μPD780024, 780034, 780024Y, 780034Y subseries This Manual	**78K/0 series User's Manual Instruction**
• Pin functions • Internal block functions • Interrupt • Other on-chip peripheral functions	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Maual

Before reading this manual, you should have general knowledge of electric and logic circuits and microcontrollers.

- When you want to understand the function in general:
  → 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 RA78K/0, AND IN CC78K/0, already defined in the header file named sfrbit.h.
- To make sure the details of the registers when you know the register name.
  → Refer to **Appendix C.**

Chapter Organization This manual devides the descriptions for the subseries into different chapters as shown below. Read only the chapters related to the device you use.

Chapter		μPD780024 Subseries	μPD780034 Subseries	μPD780024Y Subseries	μPD780034Y Subseries
Chapter 1	Outline (μPD780024, 780034 Subseries)	O	O	—	—
Chapter 2	Outline (μPD780024Y, 780034Y Subseries)	—	—	O	O
Chapter 3	Pin Function (μPD780024, 780034 Subseries)	O	O	—	—
Chapter 4	Pin Function (μPD780024Y, 780034Y Subseries)	—	—	O	O
Chapter 5	CPU Architecture	O	O	O	O
Chapter 6	Port Functions	O	O	O	O
Chapter 7	Clock Generator	O	O	O	O
Chapter 8	16-Bit Timer/Event Counter TM0	O	O	O	O
Chapter 9	8-Bit Timer/Event Counters TM5	O	O	O	O
Chapter 10	Watch Timer	O	O	O	O
Chapter 11	Watchdog Timer	O	O	O	O
Chapter 12	Clock Output/Buzzer output Control Circuit CKU	O	O	O	O
Chapter 13	8-Bit A/D Converter AD1 (μPD780024 ,780024Y Subseries)	O	—	O	—
Chapter 14	10-Bit A/D Converter AD0 (μPD780034, 780034Y Subseries)	—	O	—	O
Chapter 15	Serial Interface	O	O	O	O
Chapter 16	Serial Interface UART0	O	O	O	O
Chapter 17	Serial Interface SIO3	O	O	O	O
Chapter 18	Serial Interface IIC0 (μPD780024Y, 780034Y Subseries only)	—	—	O	O
Chapter 19	Interrupt Functions	O	O	O	O
Chapter 20	External Device Expansion Functions	O	O	O	O
Chapter 21	Standby Function	O	O	O	O
Chapter 22	Reset Function	O	O	O	O
Chapter 23	μPD78F0034, 78F0034Y	O	O	O	O
Chapter 24	Instruction Set	O	O	O	O

f

Differences between μPD780024, 780034 subseries and μPD780024Y, 780034Y subseries:

The μPD730024 780034 subseries and μPD780024Y, 780034Y subseries are different in the following functions of the serial interface channel 0.

Serial Interface	μPD780024, 780034 subseries	μPD780024Y, 780034Y subseries
3-wire serial I/O mode	2 ch (SIO30, SIO31)	1 ch (SIO30 only)
UART mode	1 ch	1 ch
I^2C bus mode	Not supported	1 ch

Legend

Data representation weight	:	High digits on the left and low digits on the right
Active low representations	:	$\overline{xxx}$ (line over the pin and signal names)
Note	:	Description of note in the text.
Caution	:	Information requiring particular attention
Remarks	:	Additional explanatory material
Numberal representations	:	Binary ··· xxxx or xxxxB
		Decimal ··· xxxx
		Hexadecimal ··· xxxxH

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

- **Related documents for μPD780024 subseries**

Document name	Document No.	
	Japanese	English
μPD780021, 780022, 780023, 780024, Preliminary Product Information	Planned	Planned
μPD780024, 780034, 780024Y, 780034Y Subseries User's Manual	U12022J	This manual
78K/0 Series User's Manual-Instruction	IEU-849	
78K/0 Series Instruction Table	U10903J	—
78K/0 Series Instruction Set	U10904J	—
μPD780034 Subseries Special Function Register Table	Planned	—

- **Related documents for μPD780024Y subseries**

Document name	Document No.	
	Japanese	English
μPD780021Y, 780022Y, 780023Y, 780024Y, Preliminary Product Information	Planned	Planned
μPD780024, 780034, 780024Y, 780034Y	U12022J	This manual
78K/0 Series User's Manual-Instruction	IEU-849	IEU-1372
78K/0 Series Instruction Table	U10903J	—
78K/0 Series Instruction Set	U10904J	—
μPD780034Y Subseries Special Function Register Table	Planned	—

Caution The above documents are subject to change without prior notice. Be sure to use the latest version document when starting design.

H

- **Related documents for μPD780034 subseries**

Document name	Document No.	
	Japanese	English
μPD780031, 780032, 780033, 780034, Preliminary Product Information	Planned	Planned
μPD78F0034 Preliminary Product Information	U11993J	
μPD780024, 780034, 780024Y, 780034Y Subseries User's Manual	U12022J	This manual
78K/0 Series User's Manual-Instruction	IEU-849	IEU-1372
78K/0 Series Instruction Table	U10903J	—
78K/0 Series Instruction Set	U10904J	—
μPD780034 Subseries Special Function Register Table	Planned	—

- **Related documents for μPD780034Y subseries**

Document name	Document No.	
	Japanese	English
μPD780031Y, 780032Y, 780033Y, 780034Y, Preliminary Product Information	Planned	Planned
μPD78F0034Y Preliminary Product Information	U11994J	Planned
μPD780024, 780034, 780024Y, 780034Y Subseries User's Manual	U12022J	This manual
78K/0 Series User's Manual-Instruction	IEU-849	IEU-1372
78K/0 Series Instruction Table	U10903J	—
78K/0 Series Instruction Set	U10904J	—
μPD780034Y Subseries Special Function Register Table	Planned	—

Caution The above documents are subject to change without prior notice. Besure to use the latest version document when starting design.

- **Related documents for development tool (User's Manuals)**

Document name		Document No.	
		Japanese	English
RA78K Series Assembler Package	Operation	EEU-809	EEU-1399
	Language	EEU-815	EEU-1404
RA78K Series Structured Assembler Preprocessor		EEU-817	EEU-1402
CC78K Series C Compiler	Operation	EEU-656	EEU-1280
	Language	EEU-655	EEU-1284
CC78K/0 C Compiler	Operation	U11517J	—
	Language	U11518J	—
CC78K/0 C Compiler Application Note	Programming Note	EEA-618	EEA-1208
CC78K Series Library Source File		EEU-777	—
IE-780000-SL		Planned	Planned
IE-78000-R-A		U10057J	U10057E
IE-78K0-SL-P01		Planned	Planned
IE-780034-SL-EM4		Planned	Planned
EP-64CW-SL		Planned	Planned
EP-64GC-SL		Planned	Planned
EP-64GK-SL		Planned	Planned
SM78K0 System Simulator Windows™ Base	Reference	U10181J	U10181E
SM78K0 Series System Simulator	External part user open Interface	U10092J	U10092E
ID78K0 Integrated Debugger EWS Base	Reference	U11151J	—
ID78K0 Integrated Debugger PC Base	Reference	U11539J	—
ID78K0 Integrated Debugger Windows Base	Guide	U11649J	—

Caution The above documents are subject to change without prior notice. Be sure to use the latest version document when starting design.

J

- **Related documents for embedded software (User's Manual)**

Document name		Document No.	
		Japanese	English
78K/0 Series Real-Time OS	Basics	U11537J	—
	Installation	U11536J	—
	Technicals	U11538J	—
78K/0 Series OS MX78K0	Basics	EEU-5010	—
Fuzzy Knowledge Data Creation Tool		EEU-829	EEU-1438
78K/0, 78IK/II, 87AD Series Fuzzy Inference Development Support System - Translator		EEU-862	EEU-1444
78K/0 Series Fuzzy Inference Development Support System - Fuzzy Inference Module		EEU-858	EEU-1441
78K/0 Series Fuzzy Inference Development Support System - Fuzzy Inference Debugger		EEU-921	EEU-1458

- **Other Documents**

Document name	Document No.	
	Japanese	English
IC PACKAGE MANUAL	C10943X	
Semiconductor Device Mounting Technology Manual	C10535J	C10535E
Quality Grade on NEC Semiconductor Devices	C11531J	C11531E
Reliabiilty Quality Control on NEC Semiconductor Devices	C10983J	C10983E
Electric Static Discharge (ESD) Test	MEM-539	—
Semiconductr Devices Quality Assurance Guide	MEI-603	MEI-1202
Microcontroller Related Product Guide - Third Party Manufacturers	U11416J	—

Caution The above documents are subject to change without prior notice. Be sure to use the latest version document when starting design.

CONTENTS

CONTENTS OF FIGURES (1/5)

CONTENTS OF FIGURES (2/5)

CONTENTS OF FIGURES (3/5)

CONTENTS OF FIGURES (4/5)

CONTENTS OF FIGURES (5/5)

CONTENTS OF TABLES (1/2)

CONTENTS OF TABLES (2/2)

CHAPTER 1 OUTLINE (μPD780024, 780034 SUBSERIES)

1.1 Features

- Internal Memory

Part Number \ Type	Program Memory (ROM/Flash memory)	Data Memory (High-Speed RAM)
μPD780021, 780031	8 Kbytes	512 bytes
μPD780022, 780032	16 Kbytes	
μPD780023, 780033	24 Kbytes	1024 bytes
μPD780024, 780034	32 Kbytes	
μPD78F0034	32 Kbytes Note	1024 bytes Note

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

- External Memory Expansion Space: 64 Kbytes
- Instruction execution time changeable from high speed (0.24 μs: In main system clock 8.38 MHz operation) to ultra-low speed (122 μs: In subsystem clock 32.768 kHz operation)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- Fifty-one I/O ports: (Four N-ch open-drain ports)
- 8-bit resolution A/D converter: 8 channels (μPD780024 Subseries only)
- 8-bit resolution D/A converter: 2 channels (μPD780034 Subseries only)
- Serial interface : 3 channels
  - 3-wireserial I/O mode: 2 channel
  - 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 interrupts: 22
- 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

μPD780021, 780022, 780023, 780024
μPD780031, 780032, 780033, 780034, 78F0034

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

μPD780021(A), 780022(A), 780023(A), 780024(A)
μPD780031(A), 780032(A), 780033(A), 780034(A), 78F0034

Controller of Transport Machines, Gas Detective Breaker, Safety Equipment, etc.

1.3 Ordering Information

(1) μPD780024 Subseries

Order Number	Package	Internal ROM
μPD780021CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780021GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780021GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780022CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780022GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780022GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780023CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780023GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780023GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780024CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780024GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780024GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780021CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780021GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780021GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780022CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780022GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780022GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780023CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780023GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780023GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780024CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780024GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780024GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM

Remark xxx indicates ROM code suffix.

(2) μPD78003 Subseries

Order Number	Package	Internal ROM
μPD780031CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780031GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780031GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780032CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780032GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780032GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780033CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780033GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780033GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780034CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780034GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780034GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780031CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780031GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780031GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780032CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780032GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780032GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780033CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780033GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780033GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780034CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780034GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780034GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78F0034CW	64-pin plastic shrink DIP (750 mil)	Flash memory
μPD78F0034GC-AB8	64-pin plastic QFP (14 × 14 mm)	Flash memory
μPD78F0034GK-8A8	64-pin plastic LQFP (12 × 12 mm)	Flash memory

Remark xxx indicates ROM code suffix.

1.4 Quality Grade

(1) μPD780024 Subseries

Order Number	Package	Package Quality Grade
μPD780021CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard General (electric appliances)
μPD780021GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard General (electric appliances)
μPD780021GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard General (electric appliances)
μPD780022CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard General (electric appliances)
μPD780022GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard General (electric appliances)
μPD780022GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard General (electric appliances)
μPD780023CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard General (electric appliances)
μPD780023GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard General (electric appliances)
μPD780023GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard General (electric appliances)
μPD780024CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard General (electric appliances)
μPD780024GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard General (electric appliances)
μPD780024GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard General (electric appliances)
μPD780021CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable)
μPD780021GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable)
μPD780021GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable)
μPD780022CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable)
μPD780022GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable)
μPD780022GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable)
μPD780023CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable)
μPD780023GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable)
μPD780023GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable)
μPD780024CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable)
μPD780024GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable)
μPD780024GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable)

Remark xxx indicates ROM code suffix.

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

(2) μPD780034 Subseries

Order Number	Package	Package Quality Grade
μPD780031CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (general electric appliances)
μPD780031GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (general electric appliances)
μPD780031GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (general electric appliances)
μPD780032CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (general electric appliances)
μPD780032GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (general electric appliances)
μPD780032GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (general electric appliances)
μPD780033CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (general electric appliances)
μPD780033GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (general electric appliances)
μPD780033GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (general electric appliances)
μPD780034CW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (general electric appliances)
μPD780034GC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (general electric appliances)
μPD780034GK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (general electric appliances)
μPD780031CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780031GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780031GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD780032CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780032GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780032GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD780033CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780033GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780033GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD780034CW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780034GC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780034GK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD78F0034CW	64-pin plastic shrink DIP (750 mil)	Standard (general electric appliances)
μPD78F0034GC-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (general electric appliances)
μPD78F0034GK-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (general electric appliances)

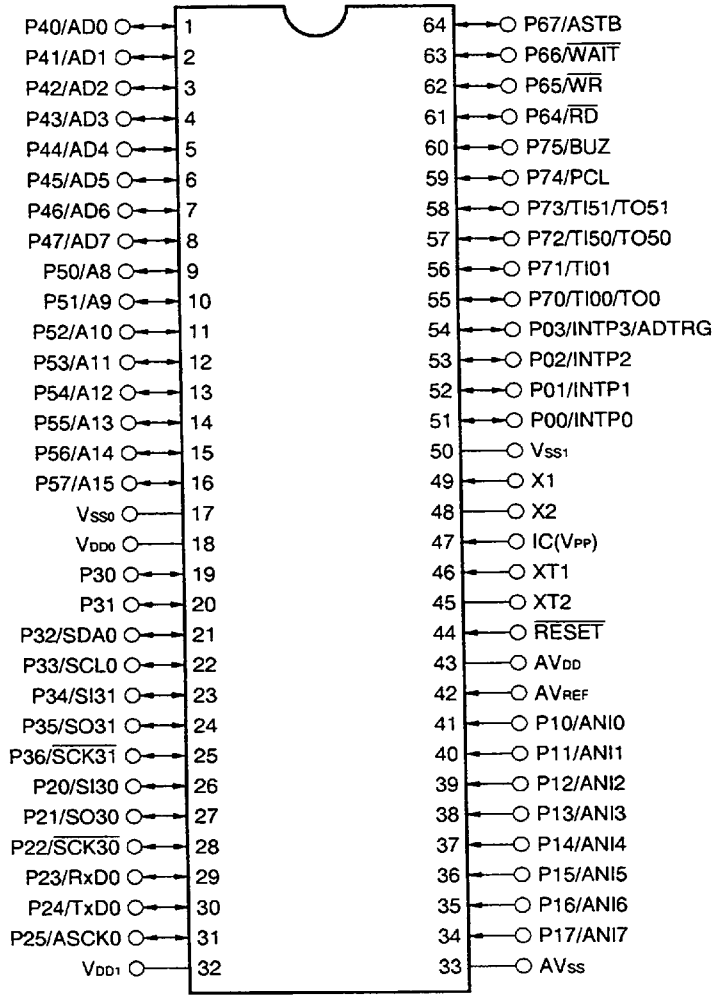
Remark xxx indicates ROM code suffix.

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

1.5 Pin Configuration (Top View)

- 64-pin plastic shrink DIP

μPD780021CW-xxx, 780022CW-xxx, 780023CW-xxx, 780024CW-xxx
μPD780021CW(A)-xxx, 780022CW(A)-xxx, 780023CW(A)-xxx, 780024CW(A)-xxx
μPD780031CW-xxx, 780032CW-xxx, 780033CW-xxx, 780034CW-xxx
μPD780031CW(A)-xxx, 780032CW(A)-xxx, 780033CW(A)-xxx, 780034CW(A)-xxx
μPD78F0034CW



Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS0} directly.

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

Remarks 1. When these are used in application required to reduce noises generated from a on-chip micro controller, noise measures are recommended that V_{DD0} and V_{DD1} are supplied independently, and that V_{SS0} and V_{SS1} are independently connected to ground lines and so on.

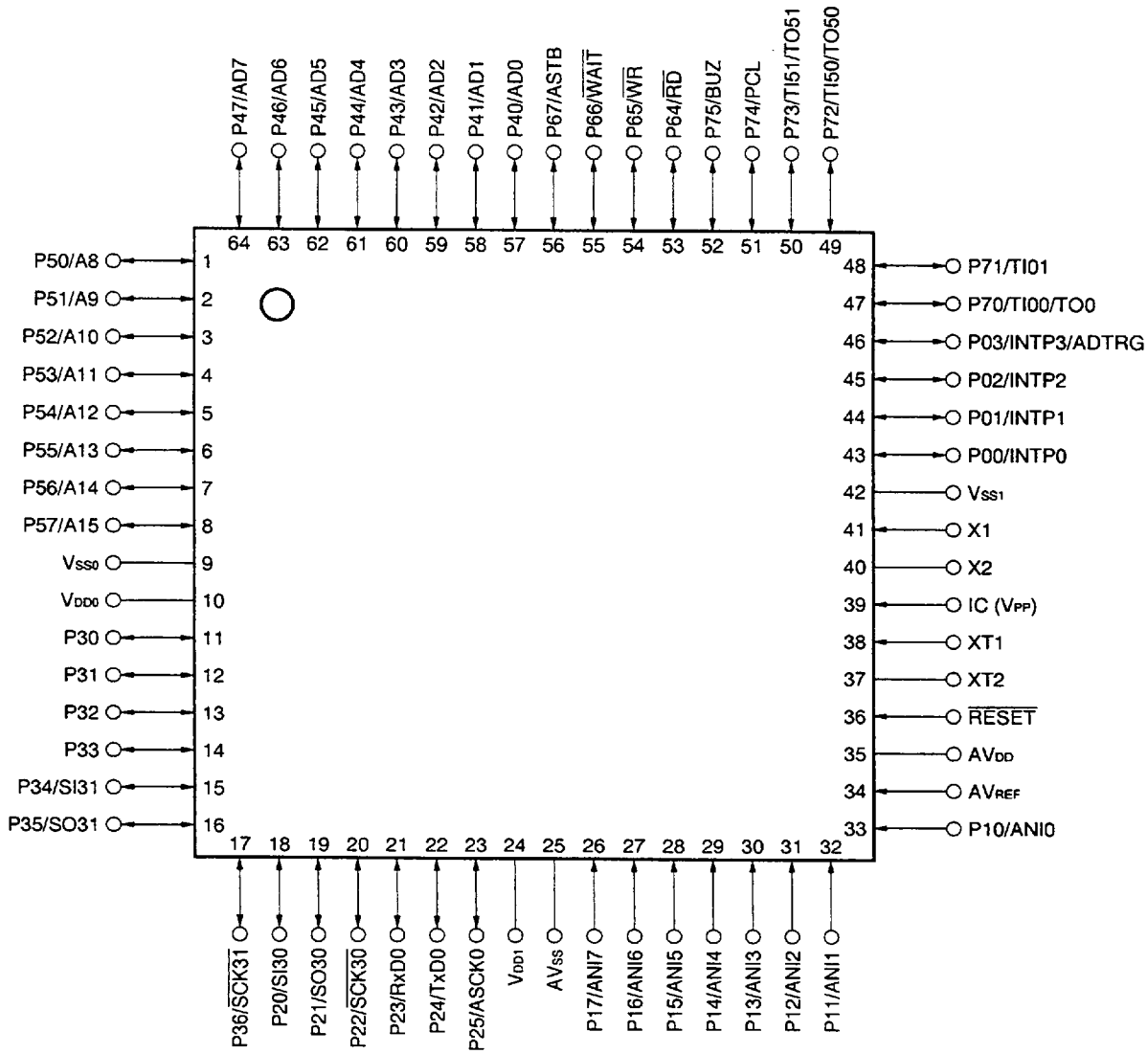
2. Pin connection in patentheses is intended for the μPD78F0034.

- 64-pin plastic QFP (14 × 14 mm)

  μPD780021GC-xxx-AB8, 780022GC-xxx-AB8, 780023GC-xxx-AB8, 780024GC-xxx-AB8
  μPD780021GC(A)-xxx-AB8, 780022GC(A)-xxx-AB8, 780023GC(A)-xxx-AB8, 780024GC(A)-xxx-AB8
  μPD780031GC-xxx-AB8, 780032GC-xxx-AB8, 780033GC-xxx-AB8, 780034GC-xxx-AB8
  μPD780031GC(A)-xxx-AB8, 780032GC(A)-xxx-AB8, 780033GC(A)-xxx-AB8, 780034GC(A)-xxx-AB8
  μPD78F0034GC-AB8

- 64-pin plastic LQFP (12 × 12 mm)

  μPD780021GK-xxx-8A8, 780022GK-xxx-8A8, 780023GK-xxx-8A8, 780024GK-xxx-8A8
  μPD780021GK(A)-xxx-8A8, 780022GK(A)-xxx-8A8, 780023GK(A)-xxx-8A8, 780024GK(A)-xxx-8A8
  μPD780031GK-xxx-8A8, 780032GK-xxx-8A8, 780033GK-xxx-8A8, 780034GK-xxx-8A8
  μPD780031GK(A)-xxx-8A8, 780032GK(A)-xxx-8A8, 780033GK(A)-xxx-8A8, 780034GK(A)-xxx-8A8
  μPD78F0034GK-8A8



Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS0} directly.

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

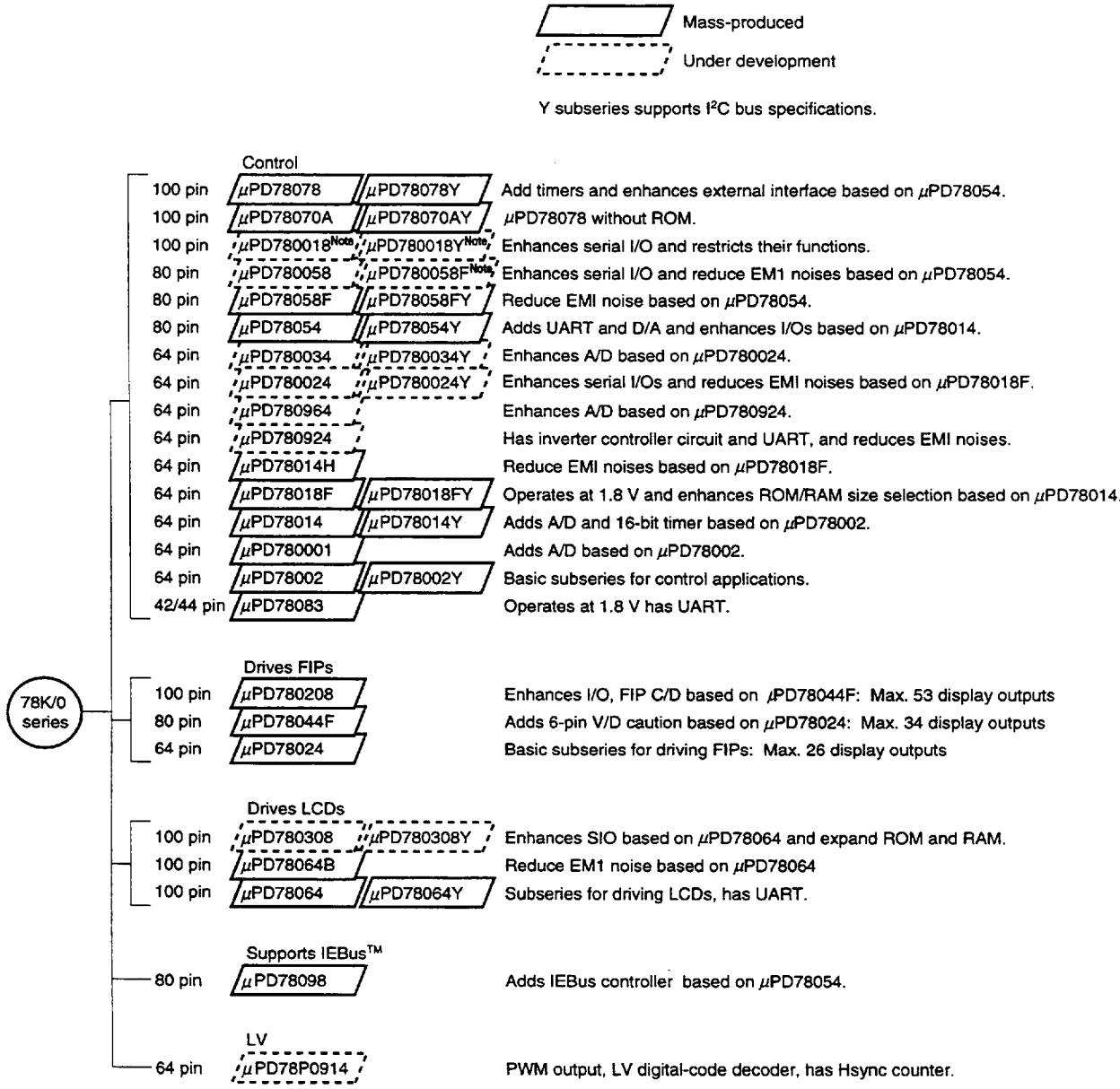
Remarks 1. When these are used in application required to reduce noises generated from a on-chip micro controller, noise measures are recommended that V_{DD0} and V_{DD1} are supplied independently, and that V_{SS0} and V_{SS1} are independently connected to ground lines and so on.

2. Pin connection in patentheses is intended for the μPD78F0034.

A8 - A15 : Address Bus
AD0 - AD7 : Address/Data Bus
ADTRG : AD Trigger Input
ANI0 - ANI7 : Analog Input
ASCK0 : Asynchronous Serial Clock
ASTB : Address Strobe
AV_{DD} : Analog Power Supply
AV_{REF} : Analog Reference Voltage
AV_{SS} : Analog Ground
BUZ : Buzzer Clock
IC : Internally Connected
INTP0 - INTP3 : Interrupt from Peripherals
P00 - P03 : Port0
P10 - P17 : Port1
P20 - P25 : Port2
P30 - P36 : Port3
P40 - P47 : Port4
P50 - P57 : Port5
P64 - P67 : Port6
P70 - P75 : Port7
PCL : Programmable Clock
$\overline{RD}$: Read Strobe
$\overline{RESET}$: Reset
RxD0 : Receive Data
$\overline{SCK30}$, $\overline{SCK31}$: 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
$\overline{WAIT}$: Wait
$\overline{WR}$: Write Strobe
X1, X2 : Crystal (Main System Clock)
XT1, XT2 : Crystal (Subsystem Clock)

1.6 78K/0 Series Expansion

78K/0 Series Expansion is shown below. The names in flames are Subseries.



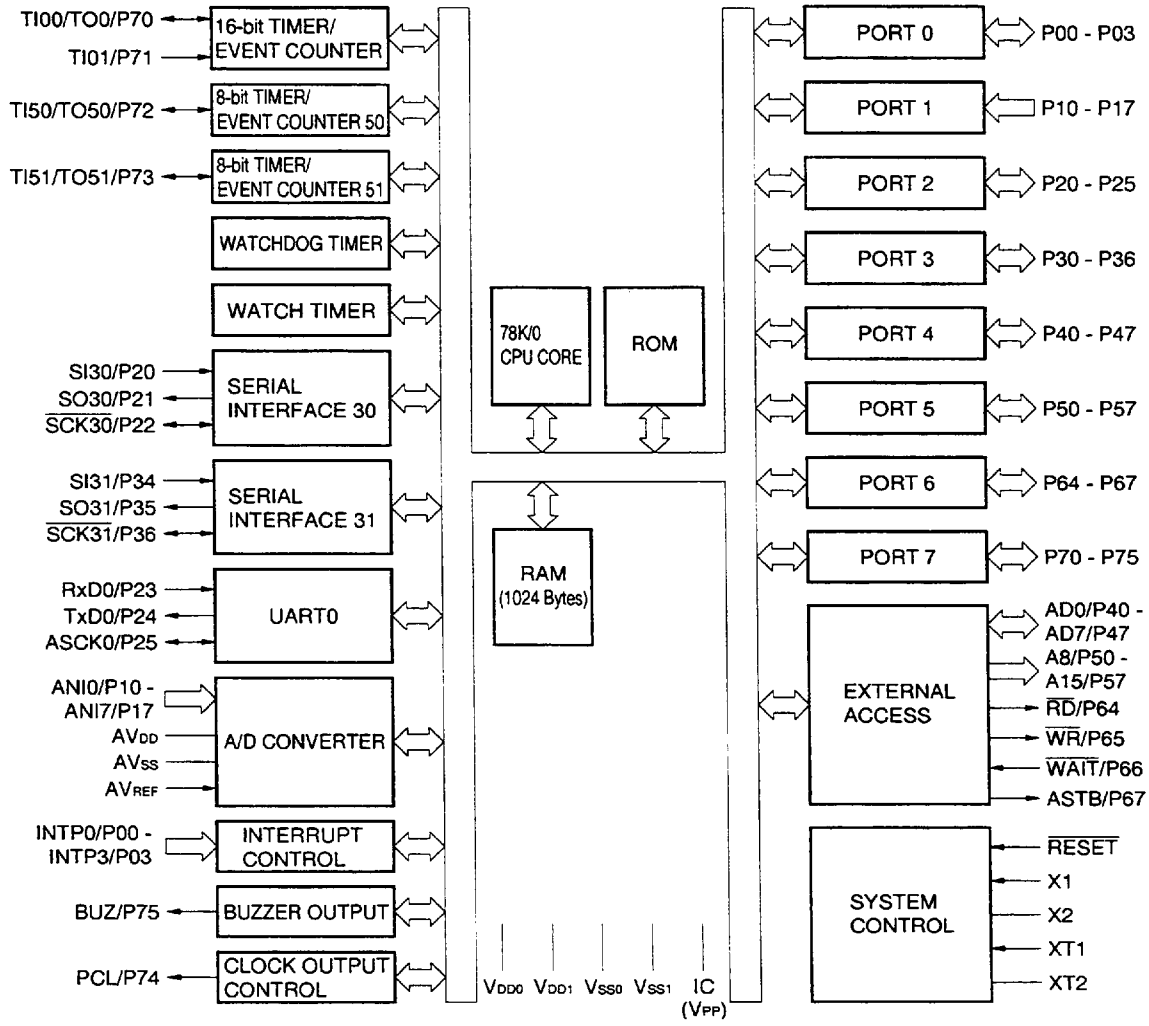
Note Under planning

Major differences among those subseries are tabulated below.

Subseries \ Function		ROM Capacity	Timer				8-bit A/D	10-bit A/D	8-bit D/A	Serial Interface	I/O	VDD MIN. value	External extension
			8-bit	16-bit	Watch	WDT							
Control	μPD78078	32K - 60K	4ch	1ch	1ch	1ch	8ch	—	2ch	3ch (UART: 1ch)	88	1.8 V	○
	μPD78070A	—									61	2.7 V	
	μPD780018	48K - 60K							—	2ch (time shared 3 wires: 1ch)	88		
	μPD780058	24K - 60K	2ch						2ch	3ch (time shared UART: 1ch)	68	1.8 V	
	μPD78058F	48K - 60K								3ch (UART: 1ch)	69	2.7 V	
	μPD78054	16K - 60K										2.0 V	
	μPD780034	8K - 32K					—	8ch	—	3ch (UART: 1ch, time shared 3 wires: 1ch)	51	1.8 V	
	μPD780024						8ch	—					
	μPD780964		3ch	Note	—		—	8ch		2ch (UART: 2ch)	47	2.7 V	
	μPD780924						8ch	—					
	μPD78014H		2ch	1ch	1ch					2ch	53	1.8 V	
	μPD78018F	8K - 60K											
	μPD78014	8K - 32K										2.7 V	
	μPD780001	8K		—	—					1ch	39		—
	μPD78002	8K - 16K			1ch		—				53		○
	μPD78083				—		8ch			1ch (UART: 1ch)	33	1.8 V	—
FIP drive	μPD780208	32K - 60K	2ch	1ch	1ch	1ch	8ch	—	—	2ch	74	2.7 V	—
	μPD78044F	16K - 40K									68		
	μPD78024	24K - 32K									54		
LCD drive	μPD780308	48K - 60K	2ch	1ch	1ch	1ch	8ch	—	—	3ch (time shared UART: 1ch)	57	1.8 V	—
	μPD78064B	32K								2ch (UART: 1ch)		2.0 V	
	μPD78064	16K - 32K											
IEBus support	μPD78098	32K - 60K	2ch	1ch	1ch	1ch	8ch	—	2ch	3ch (UART: 1ch)	69	2.7 V	○
LV support	μPD78P0914	32K	6ch	—	—	1ch	8ch	—	—	2ch	54	4.5 V	○

Note 10-bit timer: 1 channel

1.7 Block Diagram



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

2. Pin connection in parentheses is intended for the μPD78F0034.

1.8 Outline of Function

Item \ Part Number		μPD780021 μPD780031	μPD780022 μPD780032	μPD780023 μPD780033	μPD780024 μPD780034	μPD78F0034
Internal memory	ROM	8 Kbytes (Mask ROM)	16 Kbytes (Mask ROM)	24 Kbytes (Mask ROM)	32 Kbytes (Mask ROM)	32 Kbytes Note (Flash memory)
	High-speed RAM	512 bytes		1024 bytes		1024 bytes Note
Memory space		64 Kbytes				
General register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Instruction cycle		Instruction execution time changeable function				
	With main system clock selected	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (in operation at 8.38 MHz)				
	With subsystem clock selected	122 μs (in operation at 32.768 kHz)				
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 : 51 • CMOS input : 8 • CMOS I/O : 39 • N-ch open-drain I/O : 4				
A/D converter		• 8-bit resolution × 8 channels (μPD780021, 780022, 780023, 780024) • 10-bit resolution × 8 channels (μPD780031, 780032, 780033, 780034, 78F0034) • 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		• 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	Maskable	Internal: 13, External: 7				
	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		• 64-pin plastic shrink DIP (750 mil) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm)				

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

1.9 The Difference between the Standard and the Special

Standard : μPD780021, 780022, 780023, 780024
μPD780031, 780032, 780033, 780034, 78F0034

Special : μPD780021(A), 780022(A), 780023(A), 780024(A)
μPD780031(A), 780032(A), 780033(A), 780034(A)

There is the difference in only quality between the standard and the special.

1.10 Mask Options

The mask ROM versions (μPD780021, 780022, 780023, 780024, 780031, 780032, 780033, and 780034) provide pull-up resistor mask options which allow uses to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for the device production. Using the mask option when pull-up resistors are required reduces the number of components to add to the devise, resulting in board space saving.

The mask options provided in the μPD780024 and 780034 subseries are shown in Table 1-1.

Table 1-1. Mask Options of Mask ROM Versions

Pin names	Mask option
P30 - P33	Pull-up resistor connection can be specified in 1-bit units.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 2 OUTLINE (μPD780024Y, 780034Y SUBSERIES)

2.1 Features

- Internal Memory

Part Number \ Type	Program Memory (ROM)	Data Memory (High-Speed RAM)
μPD780021Y, 780031Y	8 Kbytes	512 bytes
μPD780022Y, 780032Y	16 Kbytes	
μPD780023Y, 780033Y	24 Kbytes	1024 bytes
μPD780024Y, 780034Y	32 Kbytes	
μPD78F0034Y	32 Kbytes 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.

- External Memory Expansion Space: 64 Kbytes
- Instruction execution time changeable from high speed (0.24 μs: In main system clock 8.38 MHz operation) to ultra-low speed (122 μs: In subsystem clock 32.768 kHz operation)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- Fifty-one I/O ports: (Four N-ch open-drain ports)
- 8-bit resolution A/D converter: 8 channels (μPD780024Y Subseries only)
- 10-bit resolution A/D converter: 8 channels (μPD780034Y Subseries only)
- Serial interface : 3 channels
  - I^2C mode : 1 channel
  - 3-wire serial mode: 1 channel
  - 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 interrupts: 22
- Two types of on-chip clock oscillators (main system clock and subsystem clock)
- Power supply voltage: V_{DD} =1.8 to 5.5 V

2.2 Applications

μPD780021Y, 780022Y, 780023Y, 780024Y
μPD780031Y, 780032Y, 780033Y, 780034Y, 78F0034Y

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

μPD780021Y(A), 780022Y(A), 780023Y(A), 780024Y(A), 78F0024Y
μPD780031Y(A), 780032Y(A), 780033Y(A), 780034Y(A), 78F0034Y

Controller of vending machines, gas detector breaker, safety arrangement, etc.

2.3 Ordering Information

(1) μPD780024Y Subseries

Part Number	Package	Internal ROM
μPD780021YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780021YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780021YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780022YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780022YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780022YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780023YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780023YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780023YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780024YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780024YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780024YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780021YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780021YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780021YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780022YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780022YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780022YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780023YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780023YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780023YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780024YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780024YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780024YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM

Remark xxx indicates ROM code number suffix.

(2) μPD780034Y Subseries

Part Number	Package	Internal ROM
μPD780031YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780031YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780031YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780032YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780032YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780032YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780033YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780033YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780033YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780034YCW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780034YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780034YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780031YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780031YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780031YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780032YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780032YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780032YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780033YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780033YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780033YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD780034YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD780034YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780034YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78F0034YCW	64-pin plastic shrink DIP (750 mil)	Flash memory
μPD78F0034YGC-AB8	64-pin plastic QFP (14 × 14 mm)	Flash memory
μPD78F0034YGK-8A8	64-pin plastic LQFP (12 × 12 mm)	Flash memory

Remark xxx indicates ROM code number suffix.

2.4 Quality Grade

(1) μPD780024Y Subseries

Part Number	Package	Internal ROM
μPD780021YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (electric appliances)
μPD780021YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (electric appliances)
μPD780021YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (electric appliances)
μPD780022YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (electric appliances)
μPD780022YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (electric appliances)
μPD780022YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (electric appliances)
μPD780023YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (electric appliances)
μPD780023YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (electric appliances)
μPD780023YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (electric appliances)
μPD780024YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (electric appliances)
μPD780024YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (electric appliances)
μPD780024YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (electric appliances)
μPD780021YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Extra (high-reliable electric appliances)
μPD780021YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Extra (high-reliable electric appliances)
μPD780021YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Extra (high-reliable electric appliances)
μPD780022YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Extra (high-reliable electric appliances)
μPD780022YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Extra (high-reliable electric appliances)
μPD780022YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Extra (high-reliable electric appliances)
μPD780023YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Extra (high-reliable electric appliances)
μPD780023YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Extra (high-reliable electric appliances)
μPD780023YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Extra (high-reliable electric appliances)
μPD780024YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Extra (high-reliable electric appliances)
μPD780024YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Extra (high-reliable electric appliances)
μPD780024YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Extra (high-reliable electric appliances)

Remark xxx indicates ROM code number suffix.

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

(2) μPD780034Y Subseries

Part Number	Package	Internal ROM
μPD780031YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (General electric appliances)
μPD780031YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (General electric appliances)
μPD780031YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (General electric appliances)
μPD780032YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (General electric appliances)
μPD780032YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (General electric appliances)
μPD780032YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (General electric appliances)
μPD780033YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (General electric appliances)
μPD780033YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (General electric appliances)
μPD780033YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (General electric appliances)
μPD780034YCW-xxx	64-pin plastic shrink DIP (750 mil)	Standard (General electric appliances)
μPD780034YGC-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (General electric appliances)
μPD780034YGK-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (General electric appliances)
μPD780031YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780031YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780031YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD780032YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780032YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780032YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD780033YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780033YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780033YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD780034YCW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Special (high-reliable electric appliances)
μPD780034YGC(A)-xxx-AB8	64-pin plastic QFP (14 × 14 mm)	Special (high-reliable electric appliances)
μPD780034YGK(A)-xxx-8A8	64-pin plastic LQFP (12 × 12 mm)	Special (high-reliable electric appliances)
μPD78F0034YCW	64-pin plastic shrink DIP (750 mil)	Standard (General electric appliances)
μPD78F0034YGC-AB8	64-pin plastic QFP (14 × 14 mm)	Standard (General electric appliances)
μPD78F0034YGK-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard (General electric appliances)

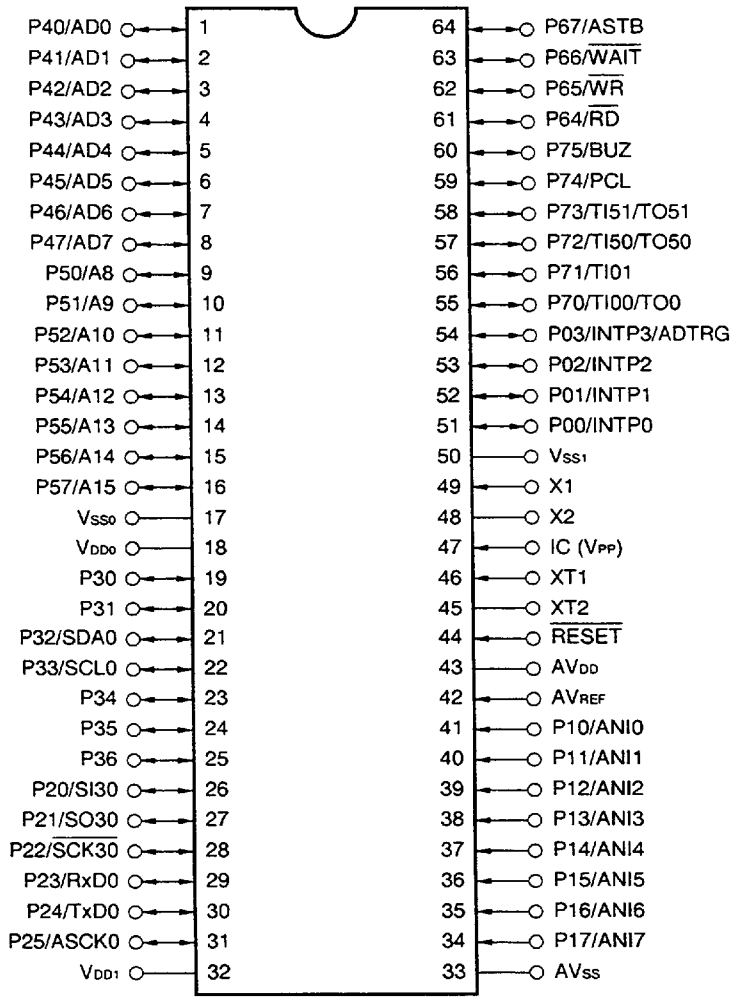
Remark xxx indicates ROM code number suffix.

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

2.5 Pin Configuration (Top View)

- 64-pin plastic shrink DIP (750 mil)

μPD780021YCW-xxx, 780022YCW-xxx, 780023YCW-xxx, 780024YCW-xxx
μPD780021YCW(A)-xxx, 780022YCW(A)-xxx, 780023YCW(A)-xxx, 780024YCW(A)-xxx
μPD780031YCW-xxx, 780032YCW-xxx, 780033YCW-xxx, 780034YCW-xxx
μPD780031YCW(A)-xxx, 780032YCW(A)-xxx, 780033YCW(A)-xxx, 780034YCW(A)-xxx
μPD78F0034YCW



Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS0} directly.
2. Connect AV_{SS} pin to V_{SS0}.

Remarks 1. When these are used in application required to reduce noises generated from a on-chip micro controller noise measures are recommended that V_{DD0} and V_{DD1} are supplied independently, and that V_{SS0} and V_{SS1} are independently connected to ground lines and so non.

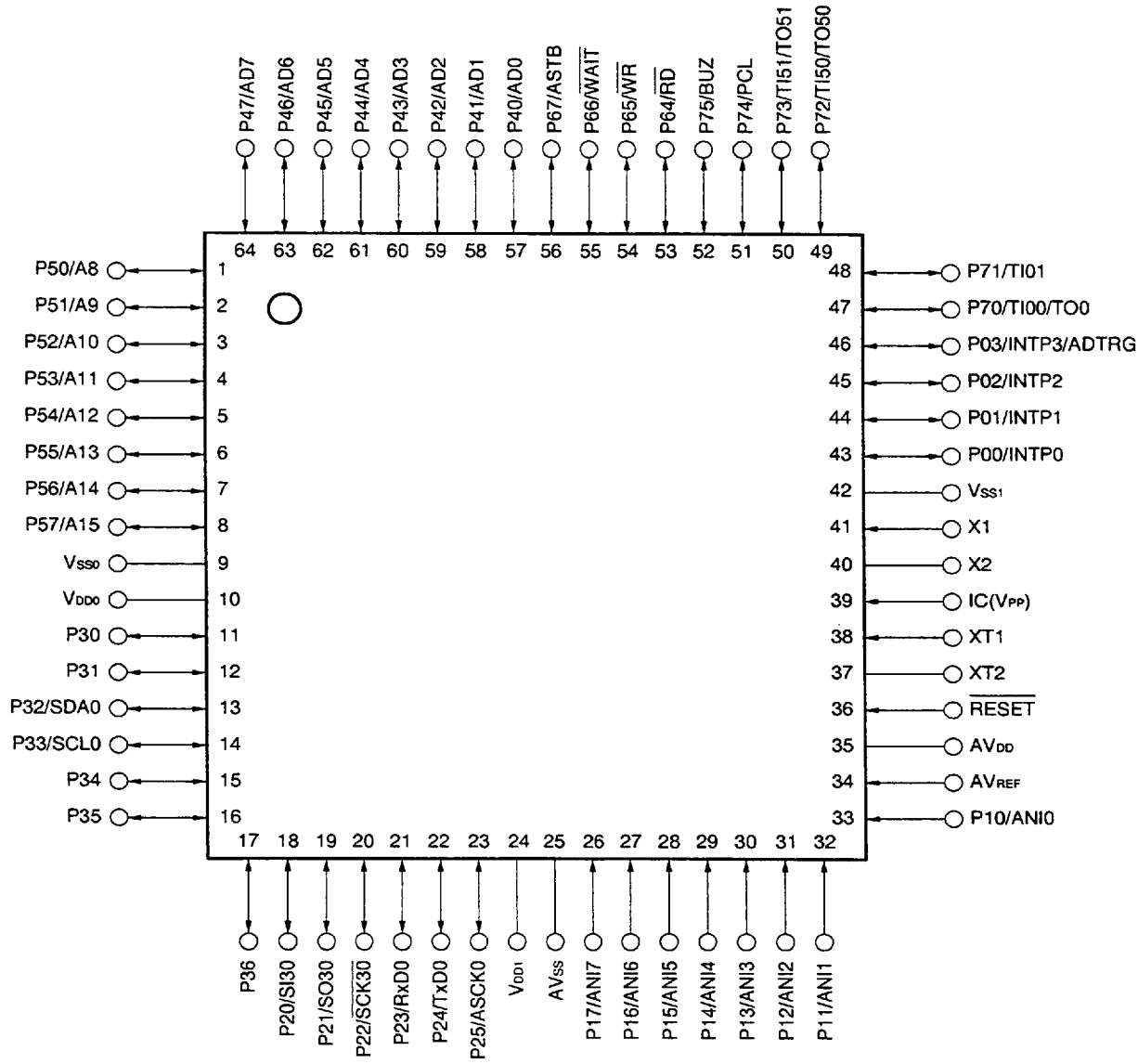
2. Pin connection in parenthesis is intended for the μPD78F0034.

- 64-pin plastic QFP (14 × 14 mm)

  μPD780021YGC-xxx-AB8, 780022YGC-xxx-AB8, 780023YGC-xxx-AB8, 780024YGC-xxx-AB8
  μPD780021YGC(A)-xxx-AB8, 780022YGC(A)-xxx-AB8, 780023YGC(A)-xxx-AB8, 780024YGC(A)-xxx-AB8
  μPD780031YGC-xxx-AB8, 780032YGC-xxx-AB8, 780033YGC-xxx-AB8, 780034YGC-xxx-AB8
  μPD780031YGC(A)-xxx-AB8, 780032YGC(A)-xxx-AB8, 780033YGC(A)-xxx-AB8, 780034YGC(A)-xxx-AB8
  μPD78F0034YGC-AB8

- 64-pin plastic LQFP (12 × 12 mm)

  μPD780021YGK-xxx-8A8, 780022YGK-xxx-8A8, 780023YGK-xxx-8A8, 780024YGK-xxx-8A8
  μPD780021YGK(A)-xxx-8A8, 780022YGK(A)-xxx-8A8, 780023YGK(A)-xxx-8A8, 780024YGK(A)-xxx-8A8
  μPD780031YGK-xxx-8A8, 780032YGK-xxx-8A8, 780033YGK-xxx-8A8, 780034YGK-xxx-8A8
  μPD780031YGK(A)-xxx-8A8, 780032YGK(A)-xxx-8A8, 780033YGK(A)-xxx-8A8, 780034YGK(A)-xxx-8A8
  μPD78F0034YGK-8A8



Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS0} directly.

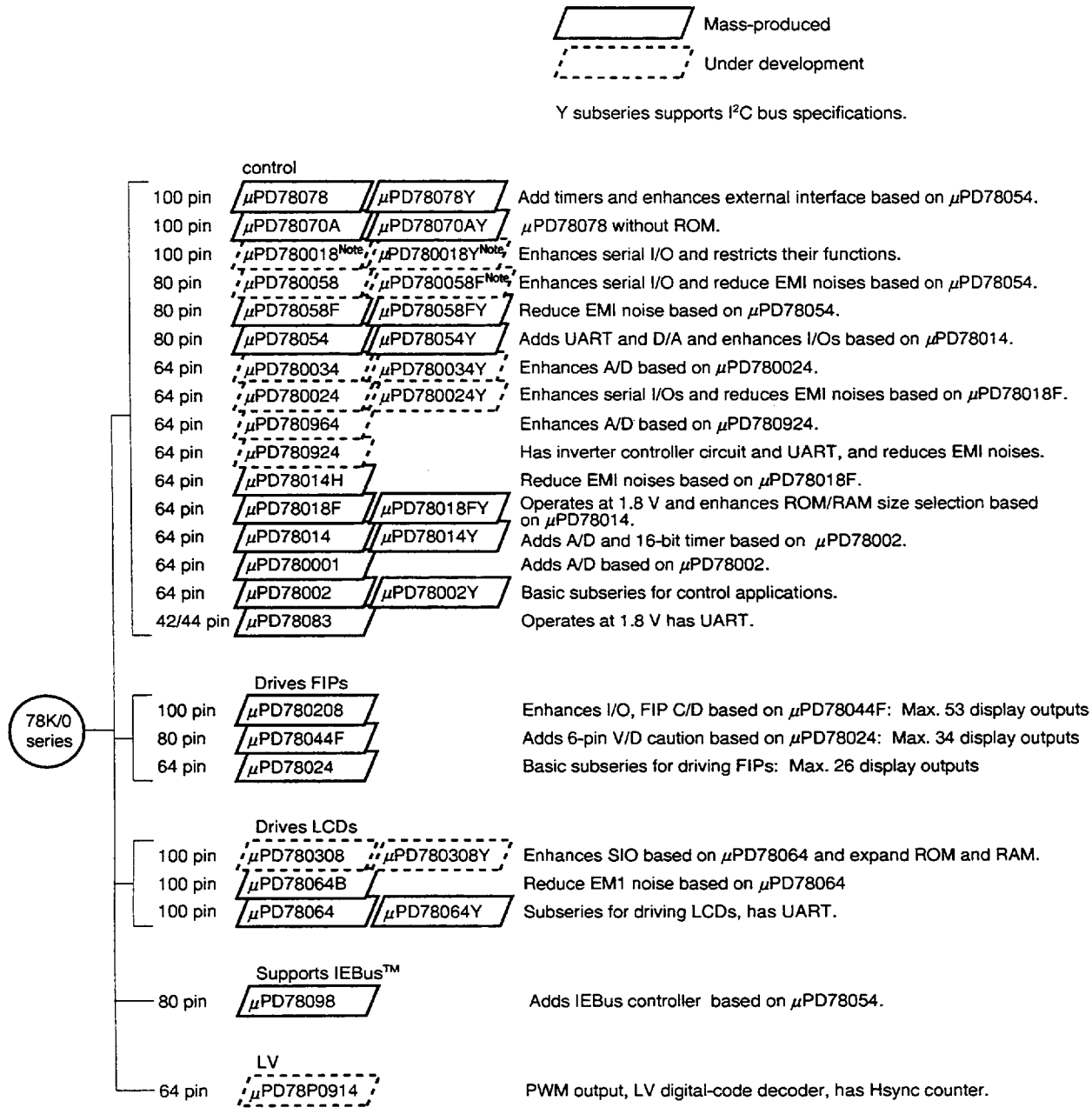
2. Connect AV_{SS} pin to V_{SS0}.

Remarks 1. When these are used in application required to reduce noises generated from a on-chip micro controller noise measures are recommended that V_{DD0} and V_{DD1} are supplied independently, and that V_{SS0} and V_{SS1} are independently connected to ground lines and so non.

2. Pin connection in parenthesis is intended for the μPD78F0034.

A8 - A15	:	Address Bus	PCL	:	Programmable Clock
AD0 - AD7	:	Address/Data Bus	$\overline{RD}$	:	Read Strobe
ADTRG	:	AD Trigger Input	$\overline{RESET}$	:	Reset
ANI0 - ANI7	:	Analog Input	RxD0	:	Receive Data
ASCK0	:	Asynchronous Serial Clock	$\overline{SCK30}$	:	Serial Clock
ASTB	:	Address Strobe	SCL0	:	Serial Clock
AV_{DD}	:	Analog Power Supply	SDA0	:	Serial Data
AV_{REF}	:	Analog Reference Voltage	SI30	:	Serial Input
AV_{SS}	:	Analog Ground	SO30	:	Serial Output
BUZ	:	Buzzer Clock	TI00, TI01, TI50, TI51	:	Timer Input
IC	:	Internally Connected	TO0, TO50, TO51	:	Timer Output
INTP0 - INTP3	:	Interrupt from Peripherals	TxD0	:	Transmit Data
P00 - P03	:	Port 0	V_{DD0}, V_{DD1}	:	Power Supply
P10 - P17	:	Port 1	V_{PP}	:	Programming Power Supply
P20 - P25	:	Port 2	V_{SS0}, V_{SS1}	:	Ground
P30 - P36	:	Port 3	$\overline{WAIT}$	:	Wait
P40 - P47	:	Port 4	$\overline{WR}$	:	Write Strobe
P50 - P57	:	Port 5	X1, X2	:	Crystal (Main System Clock)
P64 - P67	:	Port 6	XT1, XT2	:	Crystal (Subsystem Clock)
P70 - P75	:	Port 7			

2.6 78K/0 Series Expansion



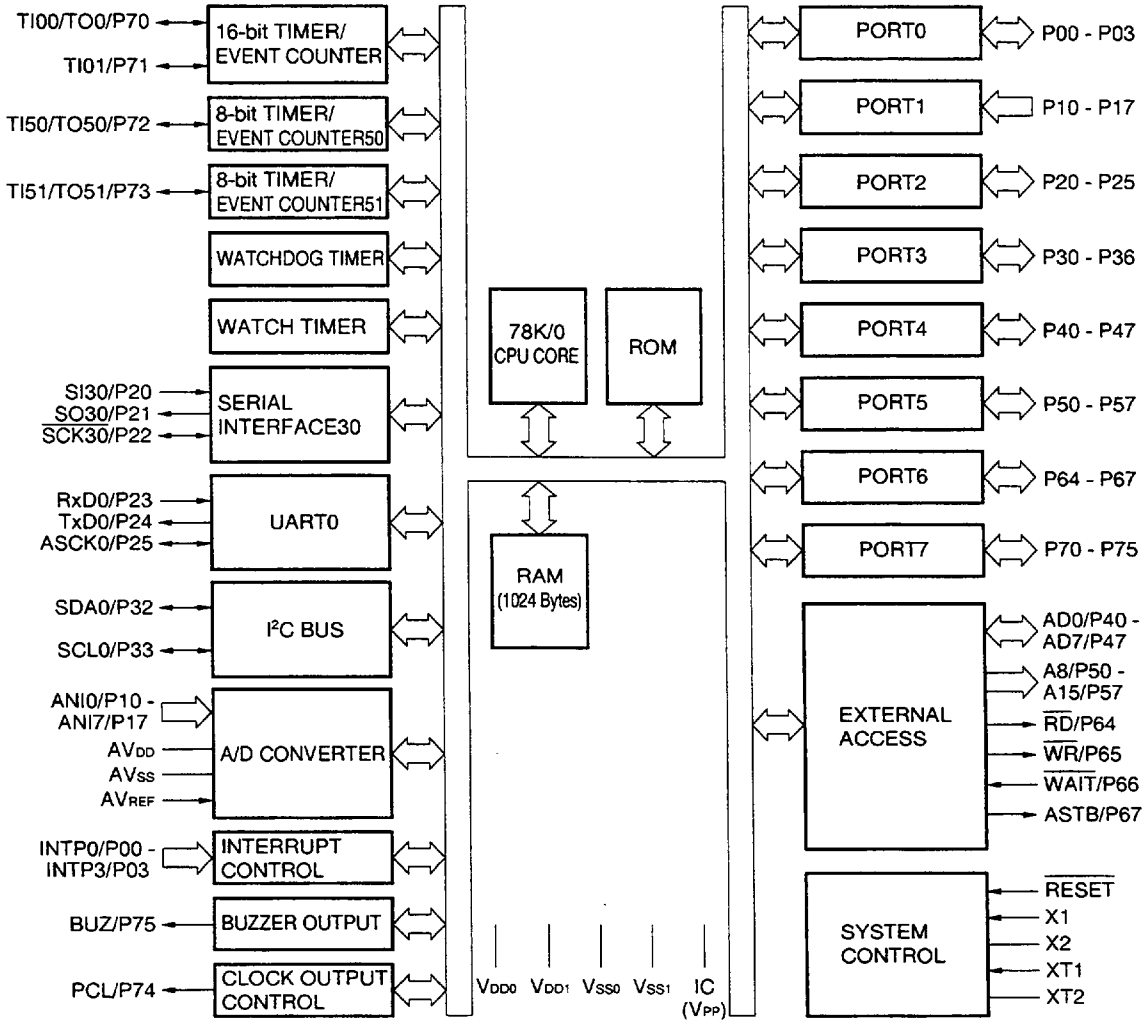
Note Planning

Major differences among these subseries are tabulated below.

Subseries \ Function		ROM Capacity	Serial Interface Configuration	I/O	V_{DD} MIN. value
Control	μPD78078Y	48K - 60K	3 wires/2 wires/I^2C : 1ch 3-wires with auto-transmit/receive : 1ch 3-wire/UART : 1ch	88	1.8 V
	μPD78070AY	—		61	2.7 V
	μPD780018Y	48K - 60K	3-wires with auto-transmit/receive : 1ch 3 wires/time shared : 1ch I^2C bus (Multi master support) : 1ch	88	
	μPD780058Y	24K - 60K	3 wires/2 wires/I^2C : 1ch 3-wires with auto-transmit/receive : 1ch 3-wire/UART timer shared : 1ch	68	1.8 V
	μPD78058FY	48K - 60K	3 wires/2 wires/I^2C : 1ch 3-wires with auto-transmit/receive : 1ch 3-wire/UART : 1ch	69	2.7 V
	μPD78054Y	16K - 60K			2.0 V
	μPD780034Y	8K - 32K	UART : 1ch 3-wire : 1ch I^2C bus (Multi master support) : 1ch	51	1.8 V
	μPD780024Y				
	μPD78018FY	8K - 60K	3 wires/2 wires/I^2C : 1ch 3-wires with auto-transmit/receive : 1ch	53	
	μPD78014Y	8K - 32K	3 wires/2 wires/SBI/I^2C : 1ch 3-wires with auto-transmit/receive : 1ch		2.7 V
	μPD78002Y	8K - 16K	3 wires/2 wires/SBI/I^2C : 1ch		
LCD Drive	μPD780308Y	48K - 60K	3 wires/2 wires/I^2C : 1ch 3-wire/UART timer shared : 1ch 3-wire : 1ch	57	1.8 V
	μPD78064Y	16K - 32K	3 wires/2 wires/I^2C : 1ch 3-wire/UART : 1ch		2.0 V

Remark Functions other than serial interface are the same as subseries without Y.

2.7 Block Diagram



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

2. Pin connection in parentheses is intended for the μPD78F0034Y.

2.8 Outline of Function

Item \ Part Number		μPD780021Y μPD780031Y	μPD780022Y μPD780032Y	μPD780023Y μPD780033Y	μPD780024Y μPD780034Y	μPD78F0034Y
Internal memory	ROM	8 Kbytes (Mask ROM)	16 Kbytes (Mask ROM)	24 Kbytes (Mask ROM)	32 Kbytes (Mask ROM)	32 Kbytes[Note] (Flash memory)
	high-speed RAM	512 bytes		1024 bytes		1024 bytes[Note]
Memory space		64 Kbytes				
General register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Instruction cycle		Instruction execution time changeable function				
	With main system clock selected	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (in operation at 8.38 MHz)				
	With subsystem clock selected	122 μs (in operation at 32.768 kHz)				
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 : 51 • CMOS input : 8 • CMOS I/O : 39 • N-ch open-drain I/O : 4				
A/D converter		• 8-bit resolution × 8 channels (μPD780021Y, 780022Y, 780023Y, 780024Y) • 10-bit resolution × 8 channels (μPD780031Y, 780032Y, 780033Y, 780034Y, 78F0034Y) • Low-voltage operation: AV_{DD} = 1.8 to 5.5 V				
Serial interface		• 3-wire serial I/O mode : 1 channel • UART mode : 1 channel • I^2C bus 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		• 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	Maskable interrupt	Internal: 13, External: 7				
	Non-maskable interrupt	Internal: 1				
	Software interrupt	1				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = −40 to +85 °C				
Package		• 64-pin plastic shrink DIP (750 mil) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm)				

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

2.9 The Difference between Standard Versions and Special Versions

Standard : μPD780021Y, 780022Y, 780023Y, 780024Y
μPD780031Y, 780032Y, 780033Y, 780034Y, 78F0034Y

Special : μPD780021Y(A), 780022Y(A), 780023Y(A), 780024Y(A)
μPD780031Y(A), 780032Y(A), 780033Y(A), 780034Y(A)

2.10 Mask Options

The mask ROM versions (μPD780021Y, 780022Y, 780023Y, 780024Y, 780031Y, 780032Y, 780033Y, 780034Y) provide pull-up resistor mask options which allow users to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for the device production. Using this mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD780024Y, 780034Y subseries are shown in Table 2-1.

Table 2-1. Mask Options of Mask ROM Versions

Pin names	Mask option
P30 - P33	Pull-up resistor connection can be specified in 1-bit units.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

3.1 Pin Function List

(1) Port Pins (1/2)

Pin Name	Input/Output	Function		After Reset	Alternative Function
P00	Input/Out put	Port 0 4-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		Input	INTP0
P01					INTP1
P02					INTP2
0P3					INTP3/ADTRG
P10 - P17	Input	Port 1 8-bit input only port.		Input	AN10 - AN17
P20	Input/Output	Port 2 6-bit Input/output port Input/output mode can be specified bit-wise If used as an input port, an on-chip pull-up resistor can be used by software		Input	SI30
P21					SO30
P22					$\overline{\text{SCK30}}$
P23					RxD0
P24					TxD0
P25					ASCK0
P30	Input/Output	Port 3 7-bit Input/output port Input/output mode can be specified bit-wise	N-ch open-drain Input/output port On-chip pull-up resistor can be specified by mask option (Mask version only) LEDs can be driven directly	Input	—
P31					
P32					
P33					
P34			If used as an input port, an on-chip pull-up resistor can be used by software		SI31
P35					SO31
P36					$\overline{\text{SCK31}}$
P40 - P47	Input/Output	Port 4 8-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software Interrupt request flag (KRIF) is set to 1 by falling edge detection		Input	AD0 - AD7
P50 - P57	Input/Output	Port 5 8-bit Input/Output port LED can be driven directly Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		Input	A8 - A15
P64	Input/Output	Port 6 4-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		Input	$\overline{\text{RD}}$
P65					$\overline{\text{WR}}$
P66					$\overline{\text{WAIT}}$
P67					ASTB

(1) Port Pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternative Function
P70	Input/Out put	Port 7 6-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software	Input	TI00/TO0
P71				TI01
P72				TI50/TO50
P73				TI51/TO51
P74				PCL
P75				BUZ

(2) Pins other than port pins (1/2)

Pin Name	Input/Output	Function	After Reset	Alternative Function
INTP0	Input	External interrupt 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
$\overline{SCK30}$	Input/Output	Serial Interface serial clock input/output	Input	P22
$\overline{SCK31}$				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 (TM0)	Input	P70/TO0
TI01				P71
TI50		External count clock input to 8-bit timer (TM50)		P72/TO50
TI51		External count clock input to 8-bit timer (TM51)		P73/TO51
TO0	Output	16-bit timer TM0 output	Input	P70/TI00
TO50		8-bit timer (TM50) output (also used for 8-bit PWM output)	Input	P72/TI50
TO51		8-bit timer (TM51) 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
AD0 - AD7	Input/Output	Lower-order address/data bus when expanding external memory	Input	P40 - P47
A8 - A15	Output	High-order address/data bus when expanding external memory	Input	P50 - P57
$\overline{RD}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{WR}$		Strobe signal output for write operation from external memory		P65
$\overline{WAIT}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to port 4,5 to access external memory	Input	P67

(2) Pins other than port pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternative Function
AN10 - AN17	Input	A/D converter analog input	Input	P10 - P17
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}	—	—
$\overline{RESET}$	Input	System Reset 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	—	—
V_{SS0}	—	Ground potential	—	—
V_{DD1}	—	Positive power supply other than port	—	—
V_{SS1}	—	Ground potential other than port	—	—
IC	—	Internal connection. Connect directly to V_{SS}	—	—
V_{PP}	—	High-voltage application for program write/verify Connect directly to V_{SS0} in normal operating mode.	—	—

3.2 Description of Pin Functions

3.2.1 P00 to P03 (Port 0)

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

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bits input/output ports.

P00 to P03 can be specified for input or output ports bit-wise with a port mode register 0 (PM0). When they are used as input ports, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register 0(PU0).

(2) Control mode

In this mode, these ports function as an external interrupt 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

Pin for external count clock input to 16-bit timer/event counter

A/D converter external trigger input

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

3.2.2 P10 to P17 (Port 1)

These are 8-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 bit-wise.

(1) Port mode

These ports function as 8-bit input only ports.

(2) Control mode

These ports function as A/D converter analog input pins (ANI0-ANI7).

3.2.3 P20 to P25 (Port 2)

These are 6-bit input/output ports. Besides serving as input/output ports, they function as data input/output to/from the serial interface and clock input/output, functions.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 6-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 2 (PM2). When they are used as input ports, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register 2 (PV2).

(2) Control mode

These ports function as serial interface data input/output and clock input/output functions.

(a) SI30 and SO30

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pins

(c) ASCK0

Asynchronous serial interface serial clock input pins

3.2.4 P30 to P36 (Port 3)

These are 7-bit input/output ports. Beside serving as input/output ports, they function as serial interface data input/output and clock input/output

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 7-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 3 (PM3). P30 to P33 are N-ch open claim input/output port. On-chip pull-up resistor can be used by mask option. (Mask ROM version only) When P34 to P36 are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 3 (PU3).

(2) Control mode

These ports function as serial interface data input/output and clock input/output.

(a) SI31 and SO31

Serial interface serial data input/output pins.

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

Serial interface serial clock input/output pins.

3.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address/data bus.
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.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified in 1-bit units for input or output ports by using the port mode register 4 (PM4). When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 4 (PU4).

(2) Control mode

These ports function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode. When pins are used as an address/data bus, the pull-up resistor is automatically disabled.

3.2.6 P50 to P57 (Port 5)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus.
Port 5 can drive LEDs directly.
The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5 (PM5). When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 5 (PU5).

(2) Control mode

These ports function as high-order address bus pins (A8 to A15) in external memory expansion mode. When pins are used as an address bus, the pull-up resistor is automatically disabled.

3.2.7 P64 to P67 (Port 6)

These are 4-bit input/output ports. Besides serving as input/output ports, they are used for control in external memory expansion mode.
The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 6(PM6).
When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 6(PU6).

(2) Control mode

These ports function as control signal output pins ($\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB) in external memory expansion mode. When a pin is used as a control signal output, the pull-up resistor is automatically disabled.

Caution When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

3.2.8 P70 to P75 (Port 7)

This is a 6-bit input/output port. In addition to its use as an input/output port, it also has timer input/output, clock output and buzzer output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 7 functions as a 6-bit input/output port. Bit-wise specification as an input port or output port to is possible by means of port mode register 7 (PM7). When used as input ports, on-chip pull-up resistors can be used by defining the 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 input/output, clock output and buzzer output.

(a) TI00

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

(b) TI01

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

(c) TI50 and TI51

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

(d) TO0, TO50 and TO51

Timer output pin.

(e) PCL

Clock output pins.

(f) BUZ

Buzzer output pin.

3.2.9 AV_{REF}

These are A/D converter reference voltage input pins.

When A/D converter is not used, connect this pin to V_{SS0}.

3.2.10 AV_{DD}

Analog power supply pin of A/D converter. Always use the same voltage as that of the V_{DD0} pin even when A/D converter is not used.

3.2.11 AV_{SS}

This is a ground voltage pin of A/D converter. Always use the same voltage as that of the V_{SS0} pin even when A/D converter is not used.

3.2.12 $\overline{\text{RESET}}$

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

3.2.13 X1 and X2

Crystal resonator connect pins for main system clock oscillation. For external clock supply, input it to X1 and its inverted signal to X2.

3.2.14 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

3.2.15 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.

3.2.16 V_{SS0} and V_{SS1}

V_{SS0} is a ground potential port pin.

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

3.2.17 V_{PP} (Flash memory versions only)

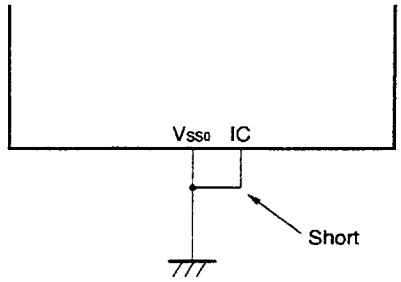
High-voltage apply pin for Flash memory programming mode setting and program write/verify. Connect directly to Vss in normal operating mode.

3.2.18 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780024, 780034 subseries at delivery. Connect it directly to the V_{SS0} with the shortest possible wire in the normal operating mode.

When a voltage difference is produced between the IC pin and V_{SS0} 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 run normally.

- Connect IC pins to V_{SS0} pins directly.



3.3 Input/output Circuits and Recommended Connection of Unused Pins

Table 3-1 shows the input/output circuit types of pins and the recommended conditions for unused pins. Refer to Figure 3-1 for the configuration of the input/output circuit of each type.

Table 3-1. Pin Input/Output Circuit Types (1/2)

Pin Name	Input/output circuit type	Input/output	Recommended connection of unused pins
P00/INTP0	8-C	Input/output	Connect independently via a resistor to V_{SS0}
P01/INTP1			
P02/INTP2			
P03/INTP3			
P10/ANI0 - P17/ANI7	25	Input	Connect independently via a resistor to V_{DD0} or V_{SS0}
P20/SI30	8-C	Input/output	
P21/SO30	5-H		
P22/$\overline{SCK30}$	8-C		
P23/RxD0			
P24/TxD0	5-H		
P25/ASCK0	8-C		
P30, P31 (Mask ROM product)	13-Q	Input/output	Connect independently via a resistor to V_{DD0}
P30, P31 (Flash memory product)	13-P		
P32, P33 (Mask ROM product)	13-S		
P32, P33 (Flash memory product)	13-R		
P34/SI31	8-C		Connect independently via a resistor to V_{DD0} or V_{SS0}
P35/SO31	5-H		
P36/$\overline{SCK31}$	8-C		
P40/AD0 - P47/AD7	5-H	Input/output	Connect independently via a resistor to V_{DD0}

Table 3-1. Pin Input/Output Circuit Types (2/2)

Pin Name	Input/output circuit type	Input/output	Recommended connection of unused pins
P50/A8 - P57/A15	5-H	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}
P64/$\overline{RD}$		Input/output	
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
P70/TI00/TO0	8-C		
P71/TI01			
P72/TI50/TO50			
P73/TI51/TO51			
P74/PCL	5-H		
P75/BUZ			
$\overline{RESET}$	2	Input	—
XT1	16		Connect to V_{DD0}
XT2		—	Open
AV_{DD}	—		Connect to V_{DD0}
AV_{REF}			Connect to V_{SS0}
AV_{SS}			
IC (Mask ROM product)			
V_{PP} (Flash memory product)			

Figure 3-1 Pin Input/Output Circuit of List (1/2)

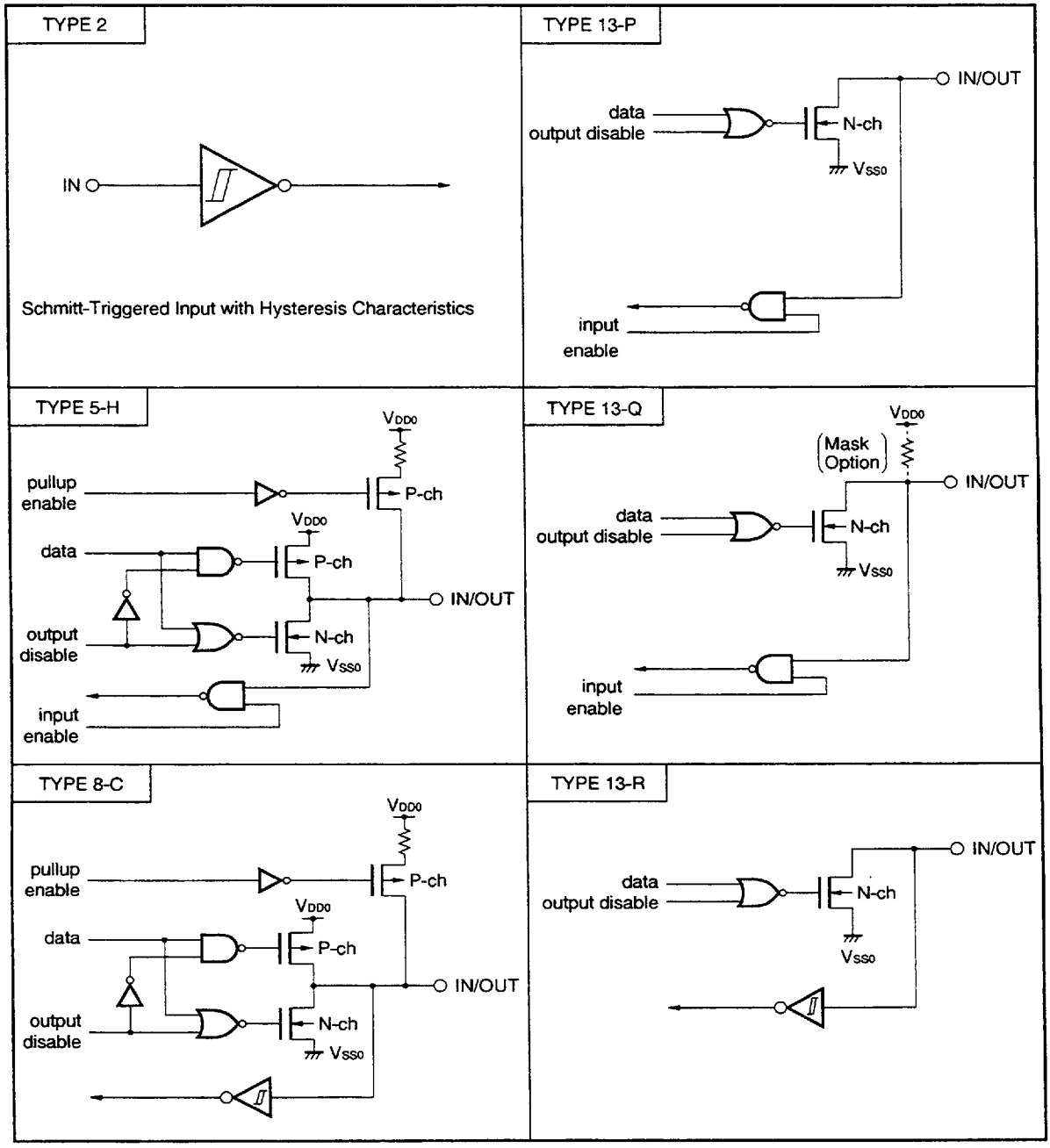
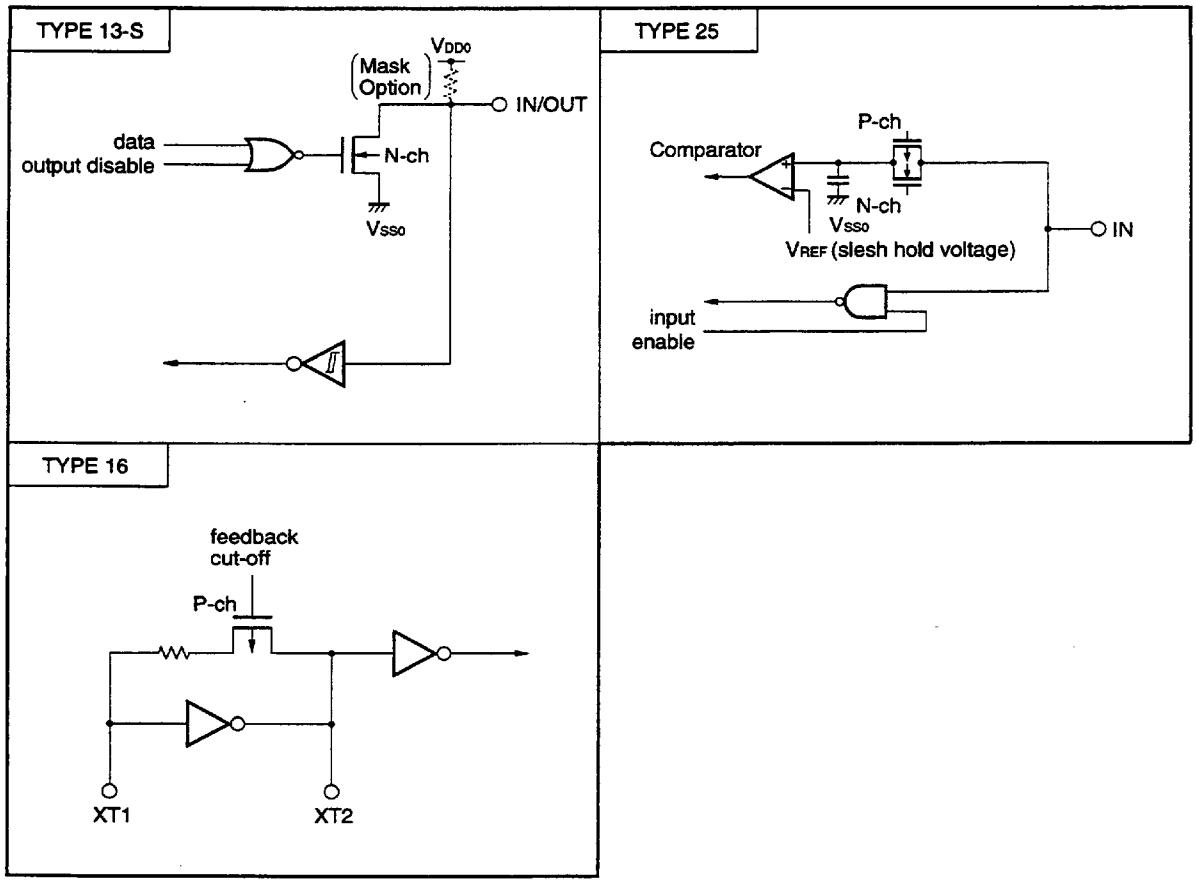


Figure 3-1 Pin Input/Output Circuit of List (2/2)



4.1 Pin Function List

(1) Port Pins (1/2)

Pin Name	Input/Output	Function		After Reset	Alternative Function
P00	Input/Out put	Port 0 4-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		Input	INTP0
P01					INTP1
P02					INTP2
0P3					INTP3/ADTRG
P10 - P17	Input	Port 1 8-bit input only port.		Input	AN10 - AN17
P20	Input/Output	Port 2 6-bit Input/output port Input/output mode can be specified bit-wise If used as an input port, an on-chip pull-up resistor can be used by software		Input	SI30
P21					SO30
P22					$\overline{\text{SCK30}}$
P23					RxD0
P24					TxD0
P25					ASCK0
P30	Input/Output	Port 3 7-bit Input/output port Input/output mode can be specified bit-wise	N-ch open-drain Input/output port On-chip pull-up resistor can be used by mask option (Mask version only) LEDs can be driven directly	Input	—
P31					SDA0
P32					SCL0
P33					—
P34			If used as an input port, an on-chip pull-up resistor can be used by software		
P35					
P36					
P40 - P47	Input/Output	Port 4 8-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software Interrupt request flag (KRIF) is set to 1 by falling edge detection		Input	AD0 - AD7
P50 - P57	Input/Output	Port 5 8-bit Input/Output port LED can be driven directly Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		Input	A8 - A15
P64	Input/Output	Port 6 4-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		Input	$\overline{\text{RD}}$
P65					$\overline{\text{WR}}$
P66					$\overline{\text{WAIT}}$
P67					ASTB

(1) Port Pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternative Function
P70	Input/Out put	Port 7 6-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software	Input	TI00/TO0
P71				TI01
P72				TI50/TO50
P73				TI51/TO51
P74				PCL
P75				BUZ

(2) Pins other than port pins (1/2)

Pin Name	Input/Output	Function	After Reset	Alternative Function
INTP0	Input	External interrupt 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
SO30	Output	Serial Interface Serial data output	Input	P21
SDA0	Input/output	Serial Interface Serial data Input/Output	Input	P32
$\overline{SCK30}$	Input/Output	Serial Interface serial clock input/output	Input	P22
SCL0				P33
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 (TM0)	Input	P70/TO0
TI01				P71
TI50		External count clock input to 8-bit timer (TM50)		P72/TO50
TI51		External count clock input to 8-bit timer (TM51)		P73/TO51
TO0	Output	16-bit timer TM0 output	Input	P70/TI00
TO50		8-bit timer (TM50) output (also used for 8-bit PWM output)	Input	P72/TI50
TO51		8-bit timer (TM51) output (also used for 8-bit PWM output)		P72/TI51
PCL	Output	clock output (for main system clock and subsystem clock trimming)	Input	P74
BUZ	Output	Buzzer output	Input	P75
AD0 - AD7	Input/Output	Lower-order address/data bus when expanding external memory	Input	P40 - P47
A8 - A15	Output	High-order address/data bus when expanding external memory	Input	P50 - P57
$\overline{RD}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{WR}$		Strobe signal output for write operation from external memory		P65
$\overline{WAIT}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to port 4,5 to access external memory	Input	P67

(2) Pins other than port 1 pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternative Function
AN10 - AN17	Input	A/D converter analog input	Input	P10 - P17
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}	—	—
$\overline{RESET}$	Input	System Reset 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	—	—
V_{SS0}	—	Ground potential	—	—
V_{DD1}	—	Positive power supply other than port	—	—
V_{SS1}	—	Ground potential other than port	—	—
IC	—	Internal connection. Connect directly to V_{SS}	—	—
V_{PP}	—	High-voltage application for program write/verify Connect directly to V_{SS0} in normal operating mode.	—	—

4.2 Description of Pin Functions

4.2.1 P00 to P03 (Port 0)

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

The following operating modes can be specified bit-wise.

(1) Port mode

In this mode, these ports function as 4-bit input/output ports.

P00 to P03 can be specified for input or output ports bit-wise with a port mode register 0 (PM0). When they are used as input ports, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, these ports function as an external interrupt input, and A/D converter external trigger input pins..

(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 pins.

Caution When the P03 is used as A/D converter external trigger input, specify in the bit 1, 2 (EGA00, EGA01) of A/D converter mode register (ADMC) the valid edges and set the interrupt mask flog (PMK3) to 1.

4.2.2 P10 to P17 (Port 1)

These are 8-bit input only ports. Besides serving as input/output ports, they function as an A/D converter analog input.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input only ports.

(2) Control mode

These ports function as A/D converter analog input pins (ANI0-ANI7).

4.2.3 P20 to P25 (Port 2)

These are 6-bit input/output ports. Besides serving as input/output ports, they function as data input/output to/from the serial interface and clock input/output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 6-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 2 (PM2). When they are used as input ports, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register 2 (PU2).

(2) Control mode

These ports function as serial interface data input/output and clock input/output.

(a) SI30 and SO30

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pins

(c) ASCK0

Asynchronous serial interface serial clock input/output pins

4.2.4 P30 to P36 (Port 3)

These are 7-bit input/output ports. Beside serving as input/output ports, they function as serial interface data input/output and clock input/output.

P30 to P33 (Port 3) can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 7-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 3 (PM3). P30 to P33 are N-ch open drain output. Mask ROM version can contain pull-up resistors with the mask option. When P64 to P67 are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 3 (PU3).

(2) Control mode

These ports function as serial interface serial data input/output and clock input/output.

(a) SDA0

Serial interface serial data input/output pins.

(b) SCL0

Serial interface serial clock input/output pins.

4.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address/data bus.
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.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise for input or output ports by using the port mode register 4 (PM4). When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 4 (PU4).

(2) Control mode

These ports function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode. When pins are used as an address/data bus, the pull-up resistor is automatically disabled.

4.2.6 P50 to P57 (Port 5)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus.
Port 5 can drive LEDs directly.
The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5 (PM5). When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 5 (PU5).

(2) Control mode

These ports function as high-order address bus pins (A8 to A15) in external memory expansion mode. When pins are used as an address bus, the pull-up resistor is automatically disabled.

4.2.7 P64 to P67 (Port 6)

These are 4-bit input/output ports. Besides serving as input/output ports, they are used for control in external memory expansion mode.
The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 6 (PM6).
When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 6 (PU6).

(2) Control mode

These ports function as control signal output pins ($\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, $\overline{ASTB}$) in external memory expansion mode. When a pin is used as a control signal output, the pull-up resistor is automatically disabled.

Caution When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

4.2.8 P70 to P75 (Port 7)

This is a 6-bit input/output port. In addition to its use as an input/output port, it also has timer input/output, clock output and buzzer output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 7 functions as a 6-bit input/output port. Bit-wise specification as an input port or output port is possible by means of port mode register 7 (PM7). When used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register 7(PU7). P70 P71 also becomes a 16-bit timer/event counter capture trigger signal input pin with a valid edge input.

(2) Control mode

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

(a) TI00

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

(b) TI01

Pin for capture trigger signal input to the 16-bit timer/event counter capture resister (CR00).

(c) TI50, TI51

Pin for external count clock input to the 8-bit timer/event counter.

(d) TO0, TO50, TO51

Timer output pin.

(e) PCL

Clock output pin.

(f) BUZ

Buzzer output pin.

4.2.9 AV_{REF}

A/D converter reference voltage input pin.

When A/D converter is not used, connect this pin to V_{SS0}.

4.2.10 AV_{DD}

Analog power supply pin of A/D converter. Always use the same voltage as that of the V_{DD0} pin even when A/D converter is not used.

4.2.11 AV_{SS}

This is a ground voltage pin of A/D converter. Always use the same voltage as that of the V_{SS0} pin even when A/D converter is not used.

4.2.12 $\overline{\text{RESET}}$

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

4.2.13 X1 and X2

Crystal resonator connect pins for main system clock oscillation. For external clock supply, input it to X1 and its inverted signal to X2.

4.2.14 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

4.2.15 V_{DD0}, V_{DD1}

V_{DD0} is a positive power supply pin

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

4.2.16 V_{SS0}, V_{SS1}

V_{SS0} is a ground potential port pin

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

4.2.17 V_{PP} (Flash memory versions only)

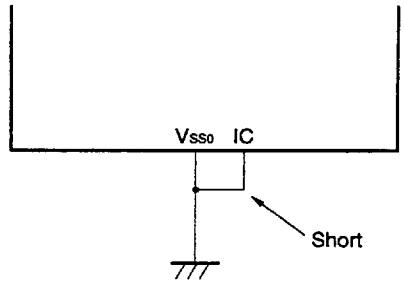
High-voltage apply pin for Flash memory programming mode setting and program write/verify. Connect directly to V_{SS0} in normal operating mode.

4.2.18 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780024, 780034Y subseries at delivery. Connect it directly to the V_{SS0} with the shortest possible wire in the normal operating mode.

When a voltage difference is produced between the IC pin and V_{SS0} 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 run normally.

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



4.3 Input/output Circuits and Recommended Connection of Unused Pins

Table 4-1 shows the input/output circuit types of pins and the recommended conditions for unused pins. Refer to Figure 4-1 for the configuration of the input/output circuit of each type.

Table 4-1. Pin Input/Output Circuit Types (1/2)

Pin Name	Input/output circuit type	Input/output	Recommended connection of unused pins
P00/INTP0	8-C	Input/output	Connect independently via a resistor to V_{SS0}
P01/INTP1			
P02/INTP2			
P03/INTP3			
P10/ANI0 - P17/ANI7	25	Input	Connect independently via a resistor to V_{DD0} or V_{SS0}
P20/SI30	8-C	Input/output	
P21/SO30	5-H		
P22/$\overline{SCK30}$	8-C		
P23/RxD0			
P24/TxD0	5-H		
P25/ASCK0	8-C		
P30, P31 (Mask ROM product)	13-Q	Input/output	Connect independently via a resistor to V_{DD0}
P30, P31 (Flash memory product)	13-P		
P32/SDA0 (Mask ROM product)	13-S		
P32/SDA0 (Flash memory product)	13-R		
P33/SCL0 (Mask ROM product)	13-S		
P33/SCL0 (Flash memory product)	13-R		
P34	8-C		Connect independently via a resistor to V_{DD0} or V_{SS0}
P35	5-H		
P36	8-C		
P40/AD0 - P47/AD7	5-H	Input/output	Connect independently via a resistor to V_{DD0}

Table 4-1. Pin Input/Output Circuit Types (2/2)

Pin Name	Input/output circuit type	Input/output	Recommended connection of unused pins
P50/A8 - P57/A15	5-H	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}
P64/$\overline{RD}$		Input/output	
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
P70/TI00/TO0	8-C		
P71/TI01			
P72/TI50/TO50			
P73/TI51/TO51			
P74/PCL	5-H		
P75/BUZ			
$\overline{RESET}$	2	Input	—
XT1	16		Connect to V_{DD0}
XT2		—	Open
AV_{DD}	—		Connect to V_{DD0}
AV_{REF}			Connect to V_{SS0}
AV_{SS}			
IC (Mask ROM product)			
V_{PP} (Flash memory product)			

Figure 4-1 Pin Input/Output Circuit of List (1/2)

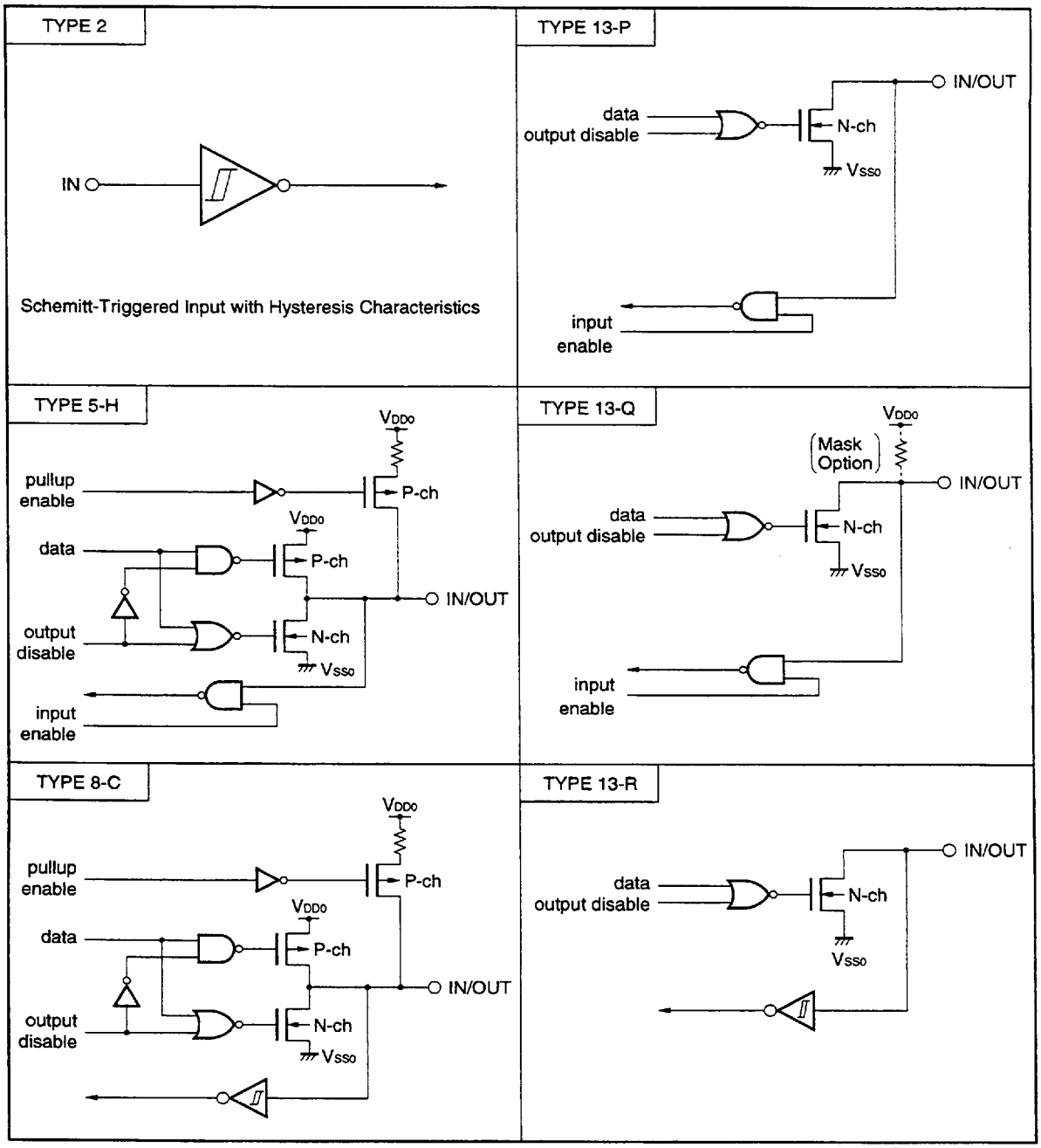
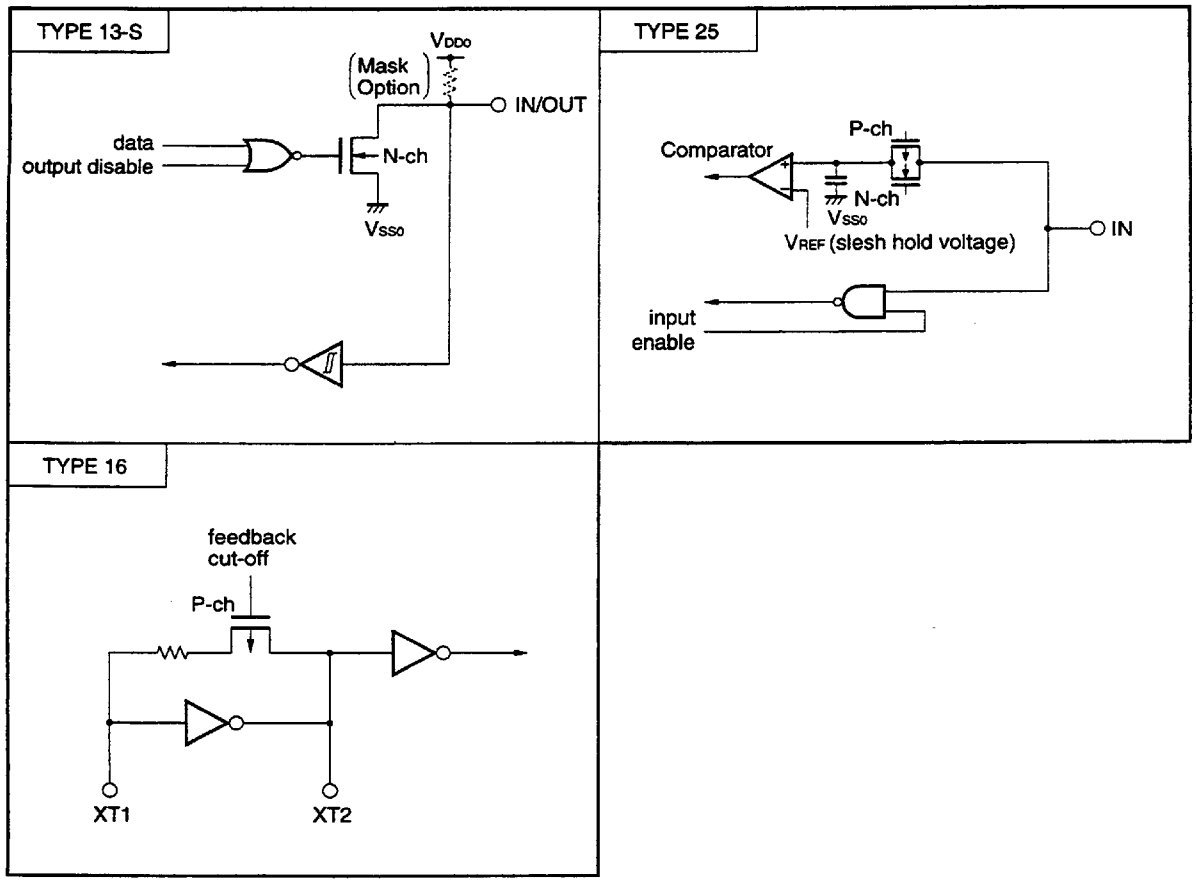


Figure 4-1 Pin Input/Output Circuit of List (2/2)



5.1 Memory Spaces

Figures 5-1 to 5-5 shows memory maps.

Figure 5-1. Memory Map (μPD780021, 780031, 780021Y, 780031Y)

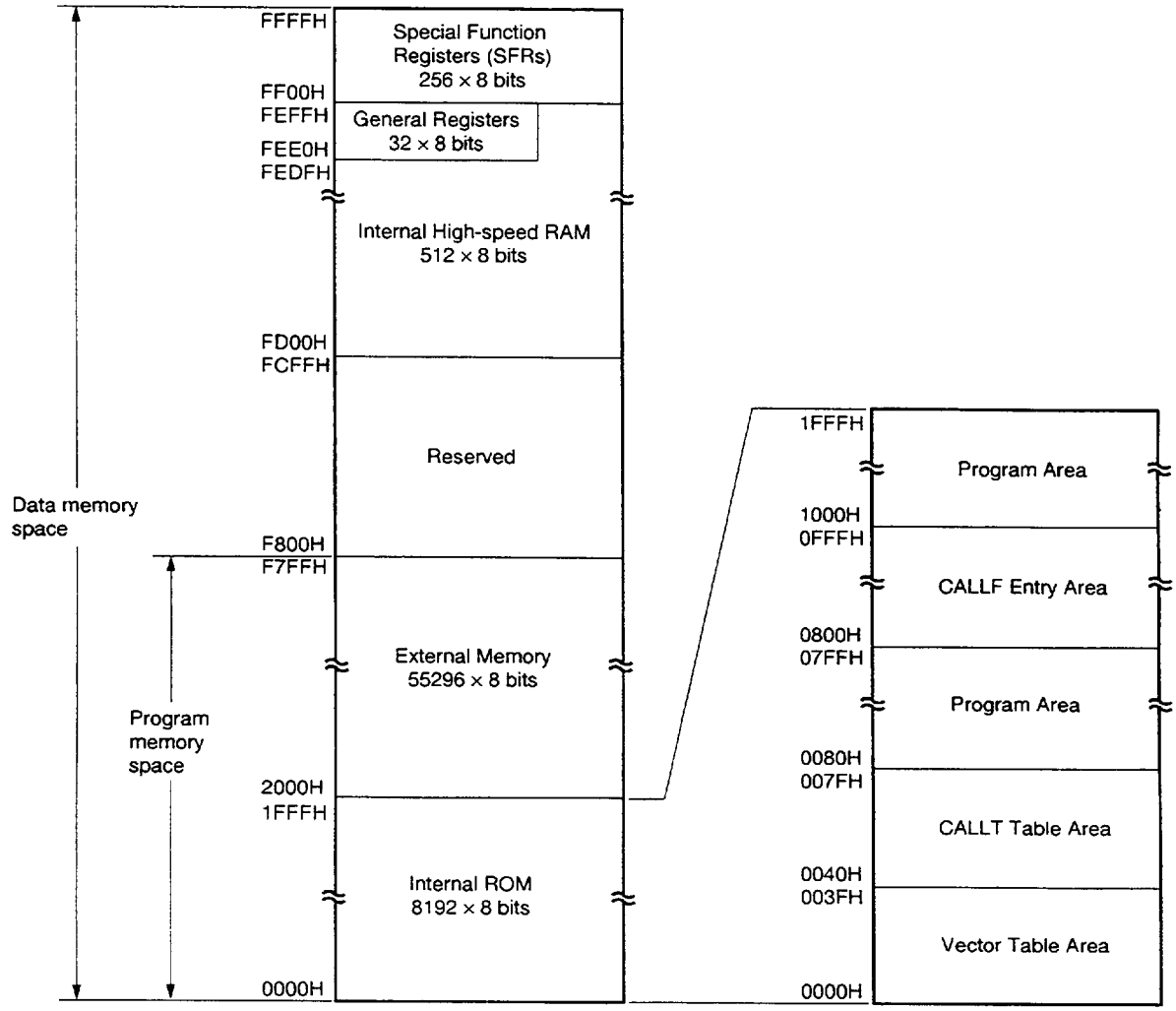


Figure 5-2. Memory Map (μPD780022, 780032, 780022Y, 780032Y)

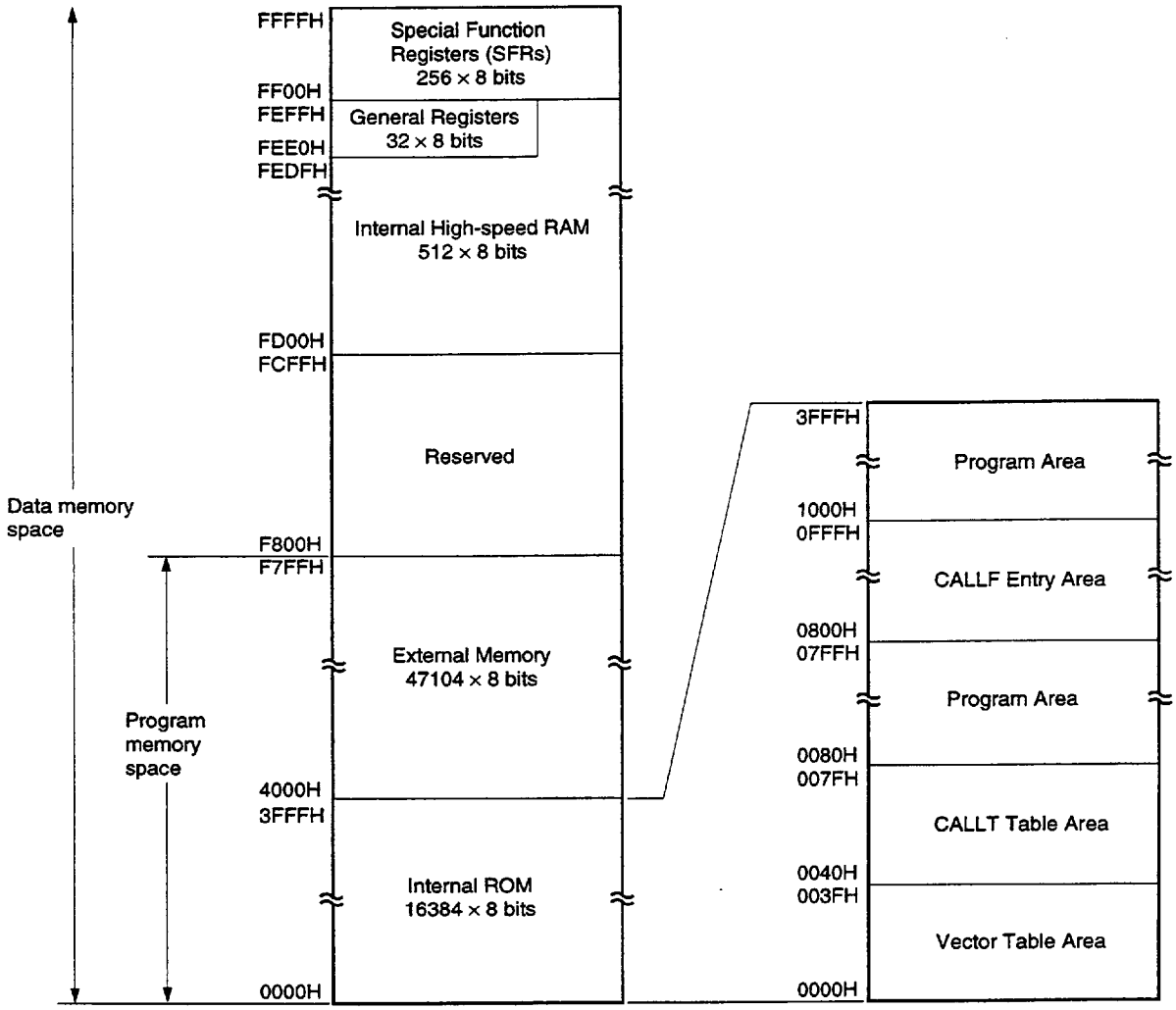


Figure 5-3. Memory Map (μPD780023, 780033, 780023Y, 780033Y)

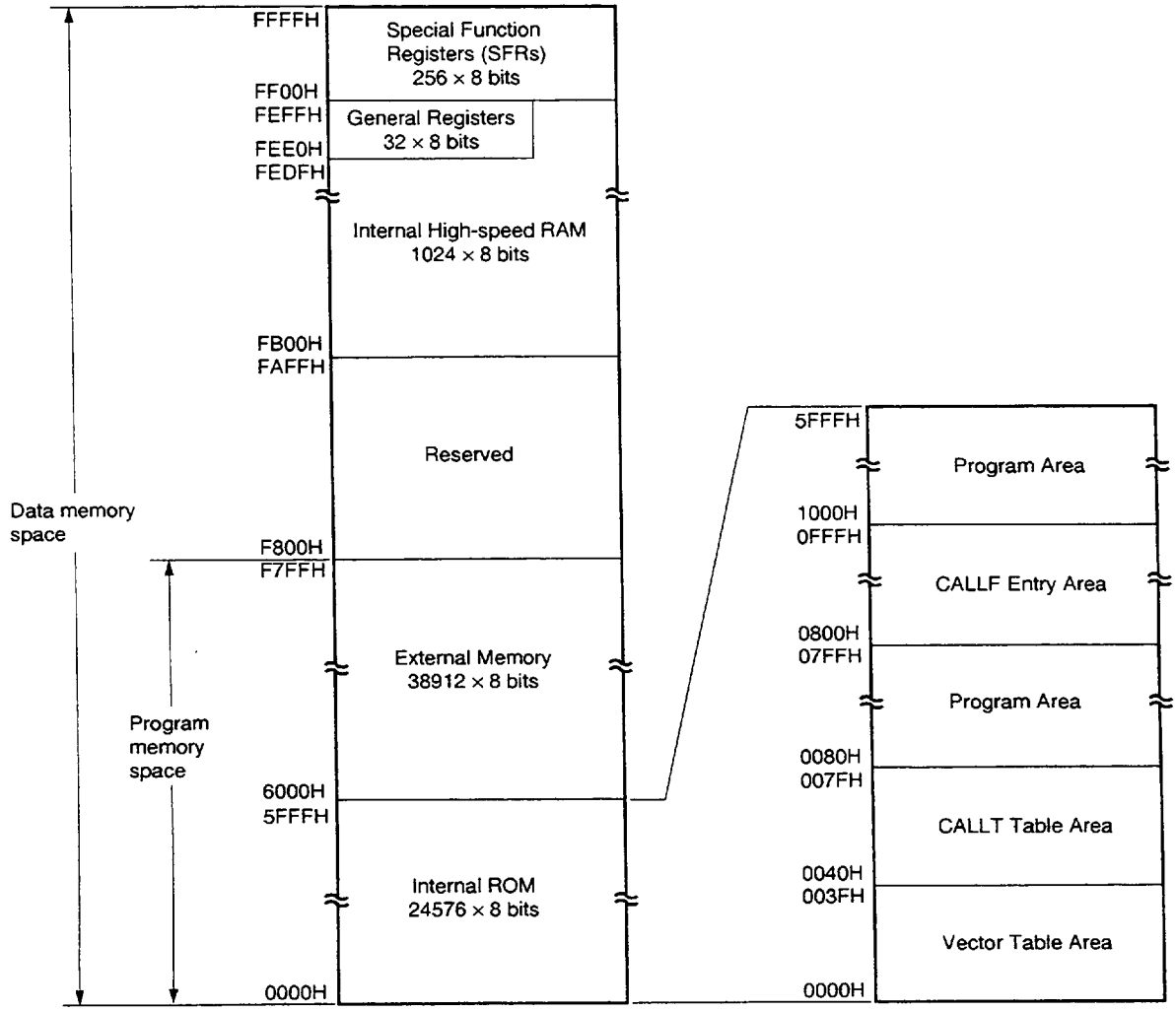


Figure 5-4. Memory Map (μPD780024, 780034, 780024Y, 780034Y)

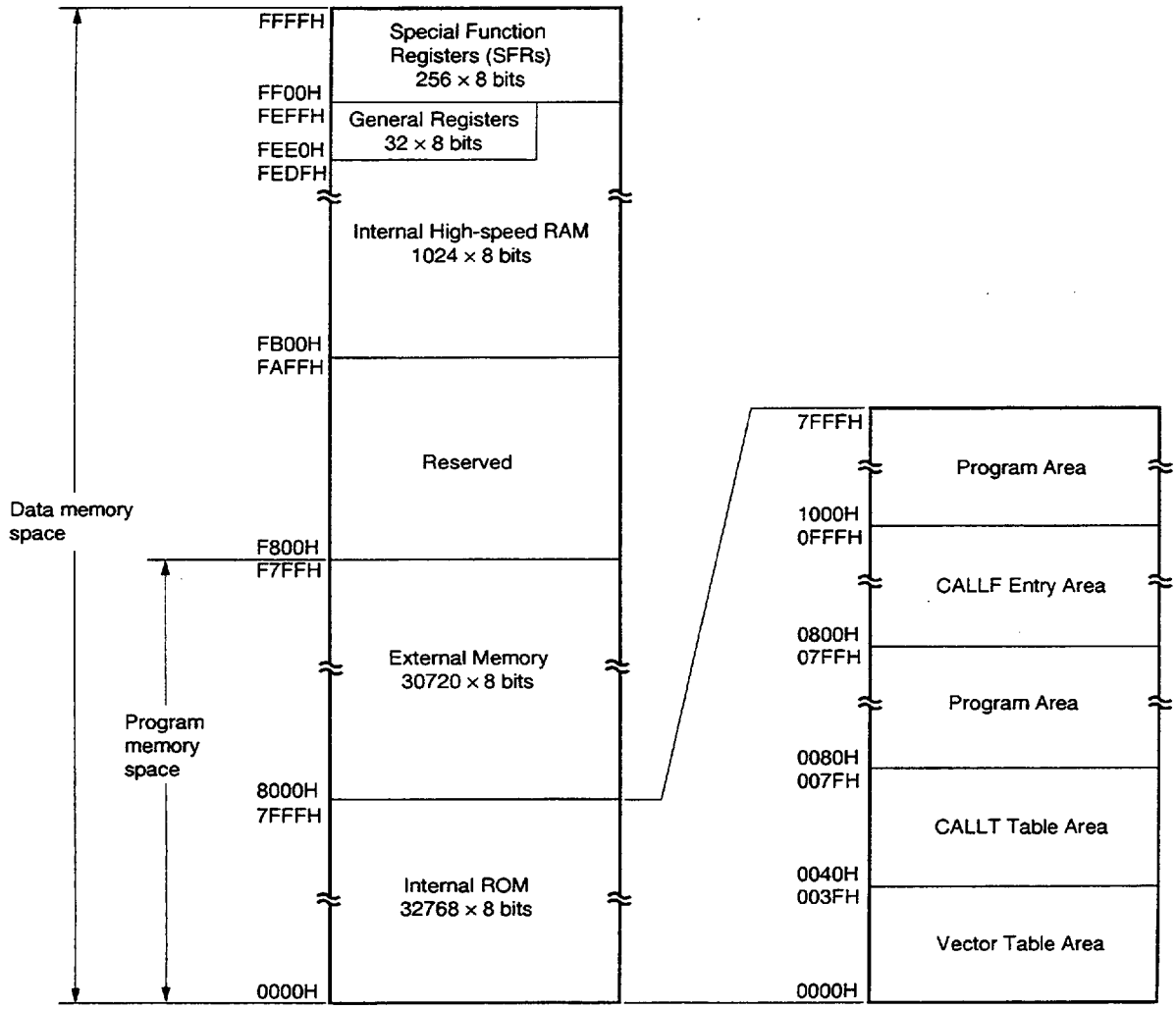
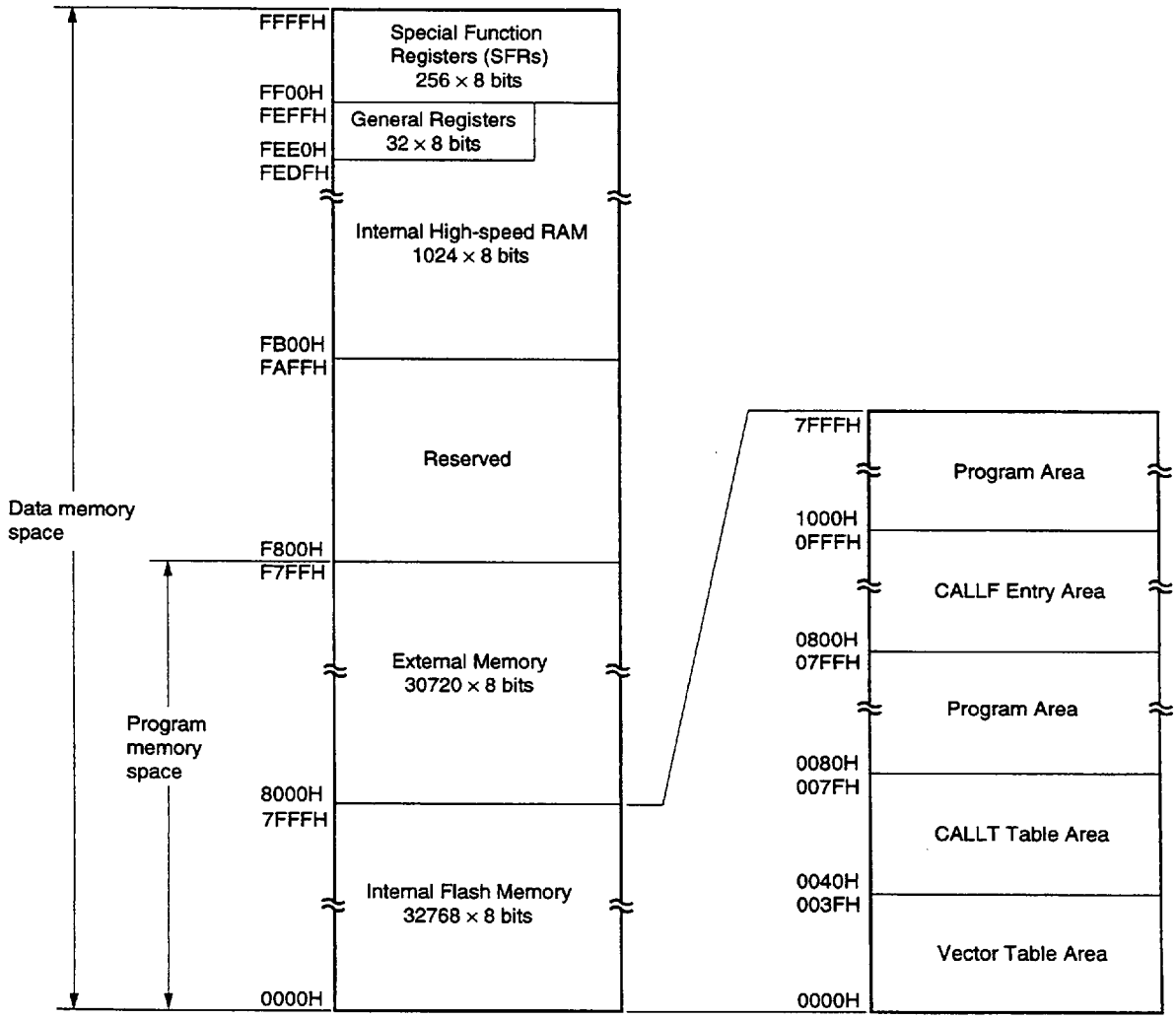


Figure 5-5. Memory Map (μPD78F0034, 78F0034Y)



5.1.1 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 products of the μPD780024, 780034, 780024Y, ane 780034Y Subseries incorate on-chip ROM (or flash memory), as listed below.

Table 5-1. Internal ROM Capacity

Part number	Internal ROM	
	Type	Capacity
μPD780021, 780031, 780021Y, 780031Y	Mask ROM	8192 × 8 bits (0000H to 1FFFH)
μPD780022, 780032, 780022Y, 780032Y		16384 × 8 bits (0000H to 3FFFH)
μPD780023, 780033, 780023Y, 780033Y		24576 × 8 bits (0000H to 5FFFH)
μPD780024, 780034, 780024Y, 780034Y		32768 × 8 bits (0000H to 7FFFH)
μPD78F0034, 78F0034Y	Flash Memory	32768 × 8 bits (0000H to 7FFFH)

The internal program memory 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, low-order 8 bits are stored at even addresses and high-order 8 bits are stored at odd addresses.

Table 5-2. Vector Table

Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTSER0
0010H	INTSR0
0012H	INTST0
0014H	INTCSI30
0016H	INTCSI31 Note 1
0018H	INTIIC0 Note 2
001AH	INTWTI
001CH	INTTM00
001EH	INTTM01
0020H	INTTM50
0022H	INTTM51
0024H	INTAD0
0026H	INTWT
0028H	INTKR
003EH	BRK

Notes 1. μPD780024, 780034 subseries only
2. μPD780024Y, 780034Y subseries only

(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).

5.1.2 Internal data memory space

The μPD780024, 780034, 780024Y, and 780034Y Subseries have on-chip high-speed RAM, as listed below.

Table 5-3. Internal High-Speed RAM Capacity

Product	Internal high-speed RAM
μPD780021, 780031, 780021Y, 780031Y	512 × 8 bits (FD00H to FEFFH)
μPD780022, 780032, 780022Y, 780032Y	
μPD780023, 780033, 780023Y 780033Y	1024 × 8 bits (FB00H to FEFFH)
μPD780024, 780034, 780024Y, K780034Y	
μPD78F0034, 78F0034Y	

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 be used as stack memory.

5.1.3 Special Function Register (SFR) area

An on-chip peripheral hardware special-function register (SFR) is allocated in the area FF00H to FFFFH. (Refer to **5.2.3 Special Function Register (SFR) Table 5-5. Special Function Register List.**)

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

5.1.4 External memory space

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

5.1.5 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 5.3 Instruction Address Addressing).

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780024, 780034, 780024Y, and 780034Y 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 5-9 to 5-10. For the details of each addressing mode, see **5.4 Operand Address Addressing**.

Figure 5-6. Data Memory Addressing (μPD780021, 780031, 780021Y, 780031Y)

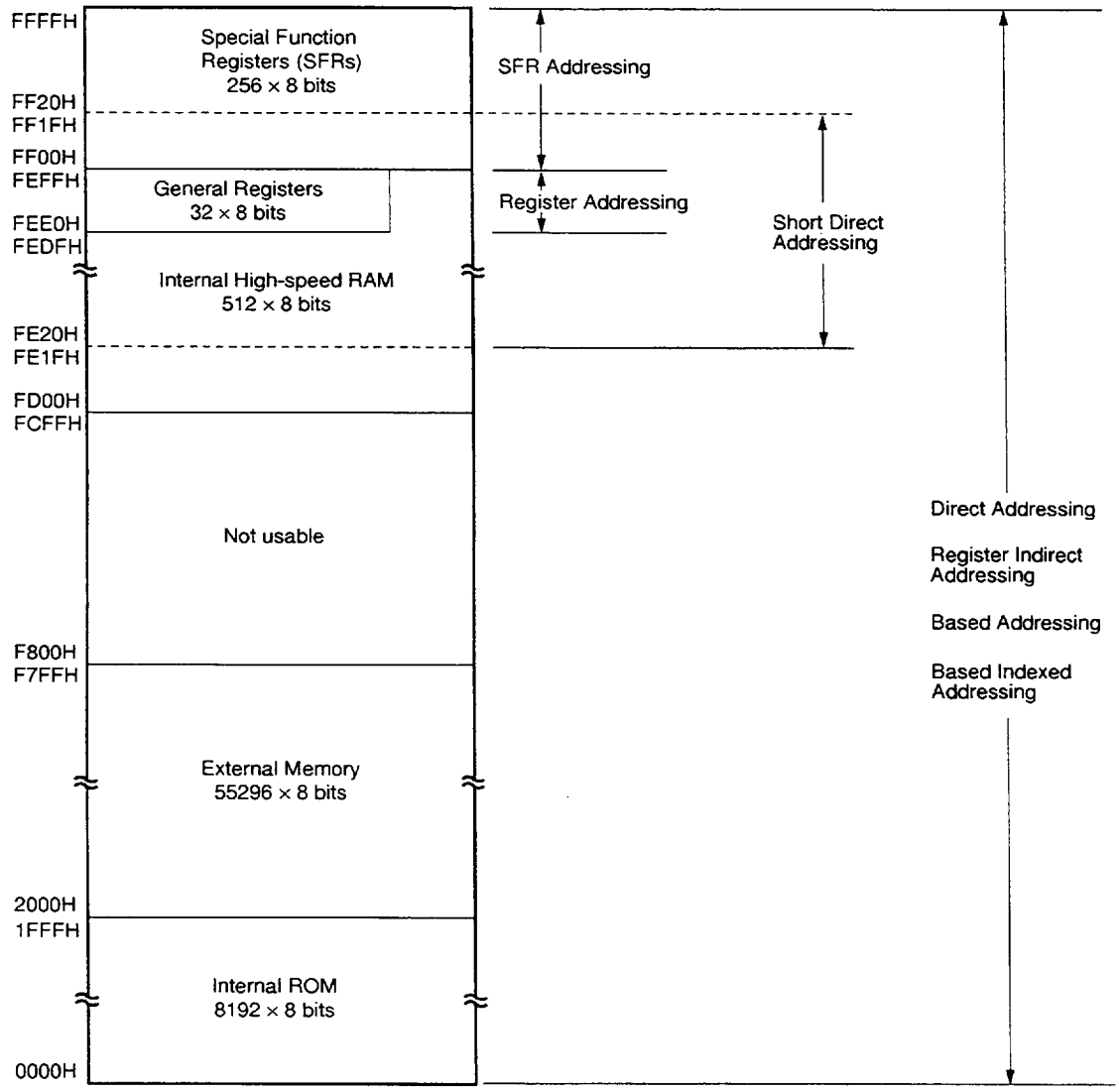


Figure 5-7. Data Memory Addressing (μPD780022, 780032, 780022Y, 780032Y)

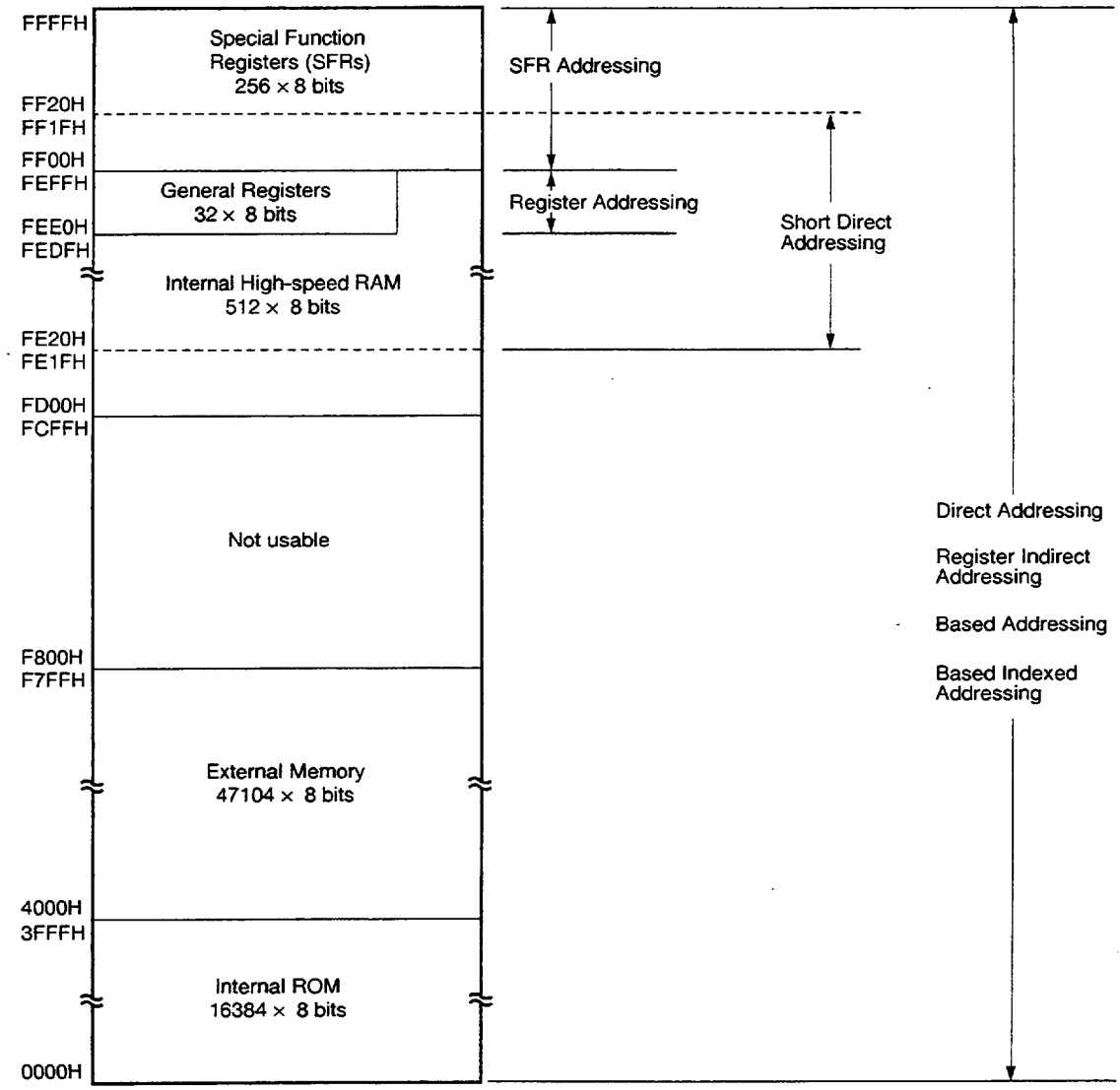


Figure 5-8. Data Memory Addressing (μPD780023, 780033, 780023Y, 780033Y)

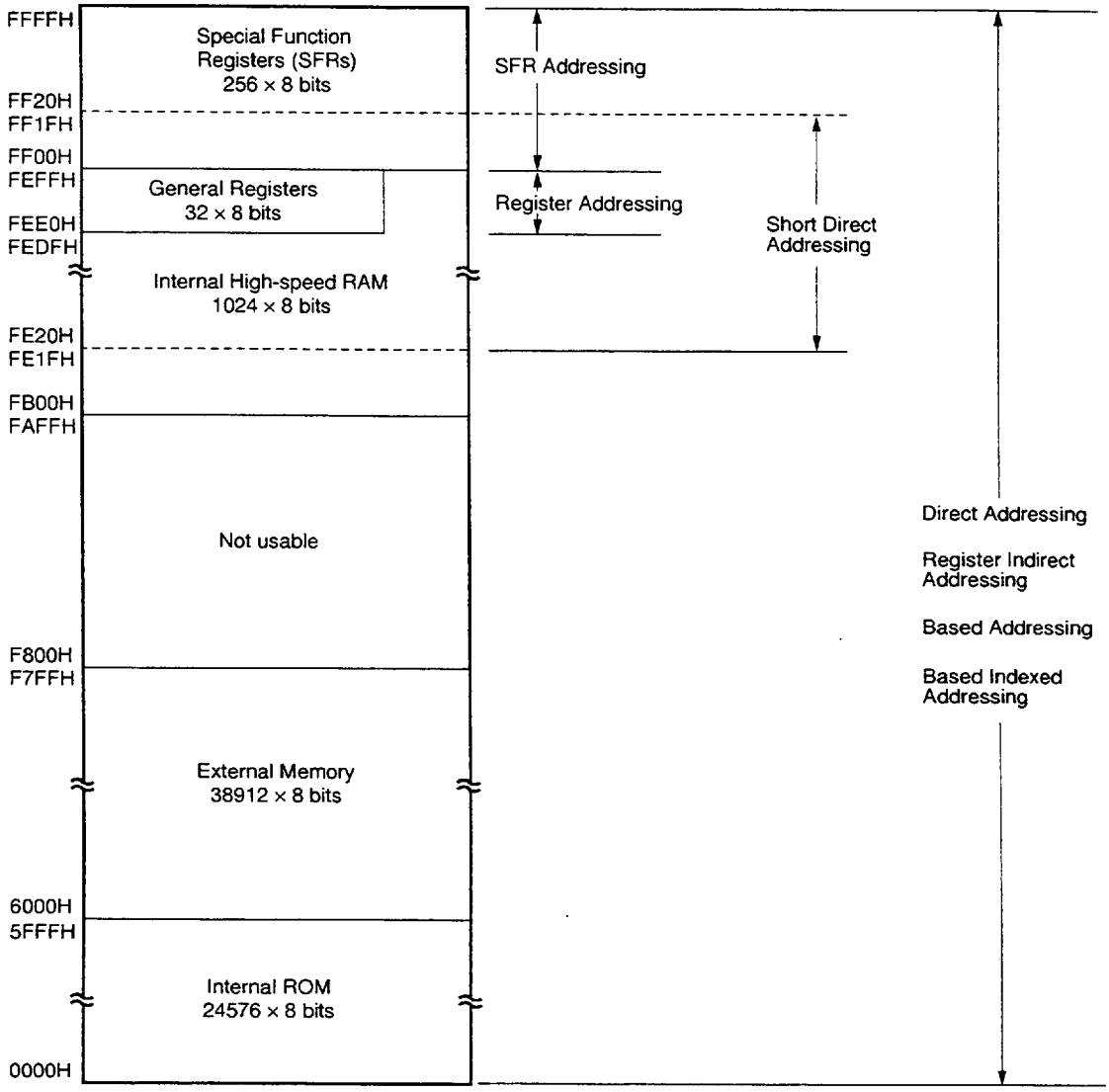


Figure 5-9. Data Memory Addressing (μPD780024, 780034, 780024Y, 780034Y)

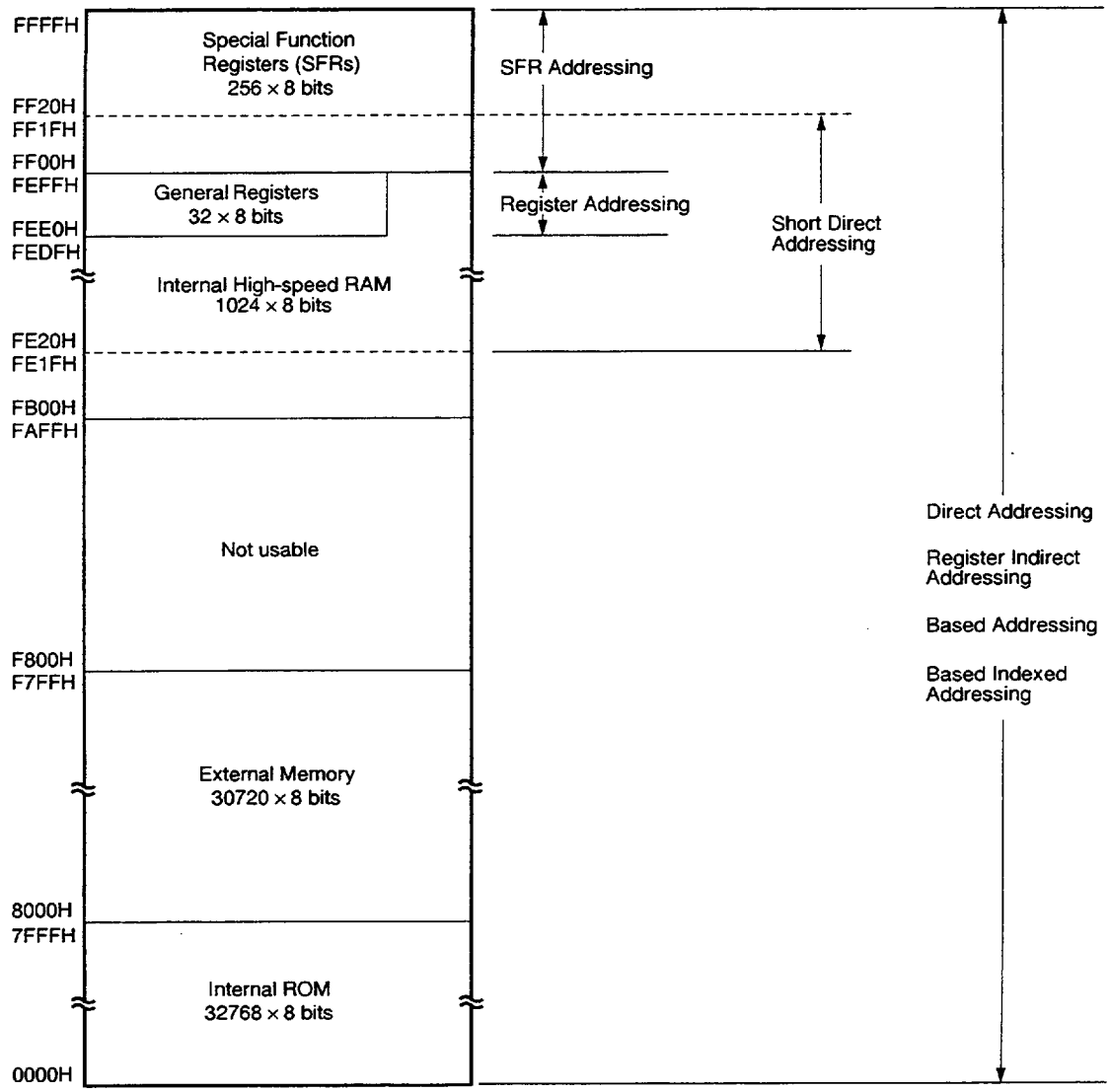
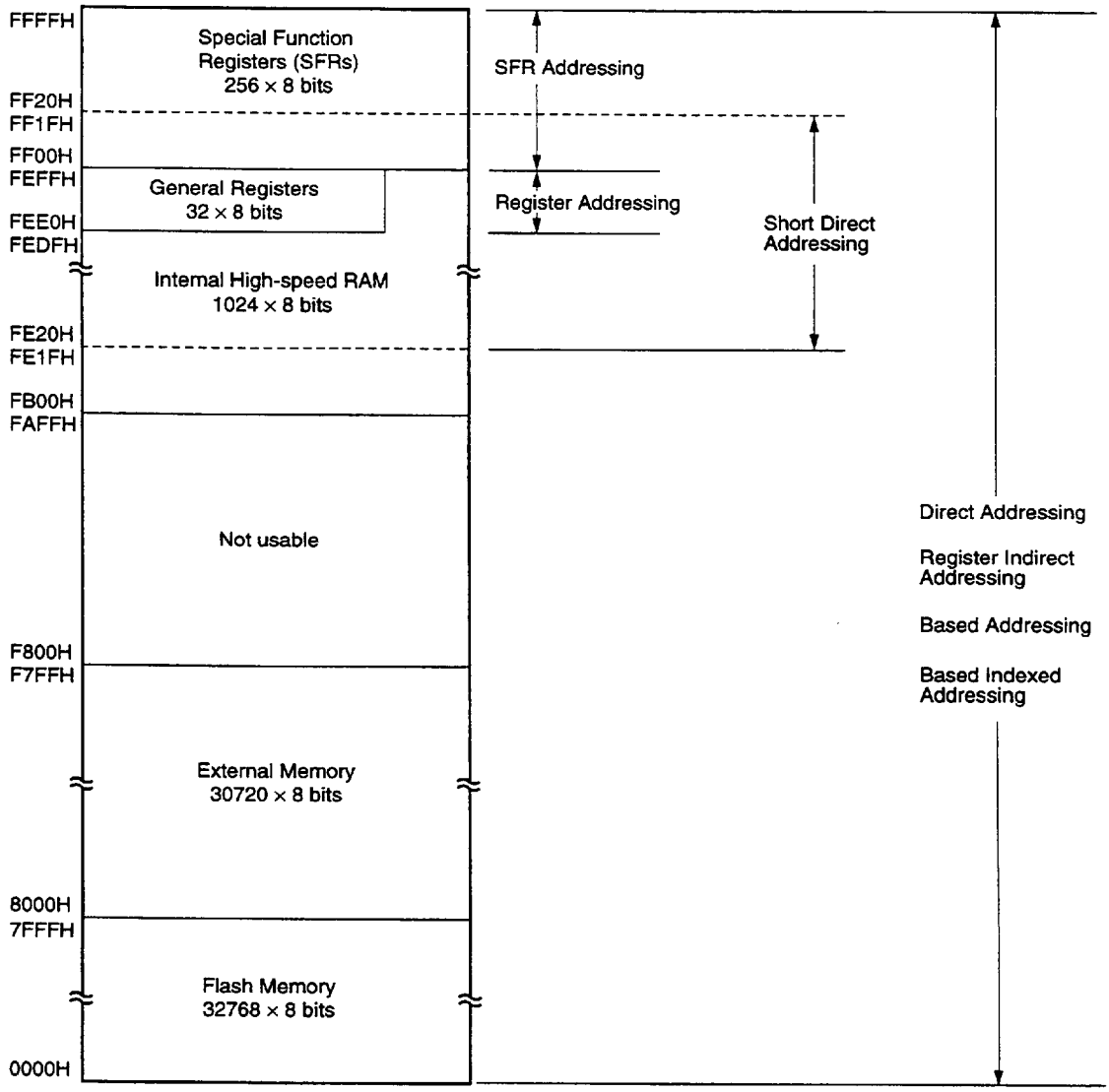


Figure 5-10. Data Memory Addressing (μPD78F0034, 78F0034Y)



5.2 Processor Registers

The μPD780024, 780034, 780024Y, 780034Y subseries units incorporate the following processor registers.

5.2.1 Control registers

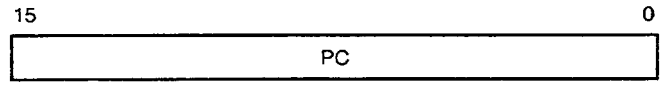
The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter, a program status word and a stack pointer.

(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 5-11. Program Counter Configuration



(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 5-12. Program Status Word Configuration

7							0
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 DI, and only non-maskable interrupt request becomes acknowledgeable. Other interrupt requests are all disabled. When 1, the IE is set to EI and interrupt request acknowledge enable is controlled with an inservice priority flag (ISP), an interrupt mask flag for various interrupt sources and a priority specification flag.

The IE is reset to (0) upon DI instruction execution or interrupt acknowledgement and is set to (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 interrupts specified with a priority specify flag register (PR0L, PR0H, PR1L) (refer to **19.3 (3) Priority Specify Flag Register (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. When it is 1, all interrupts are acknowledgeable. Actual 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.

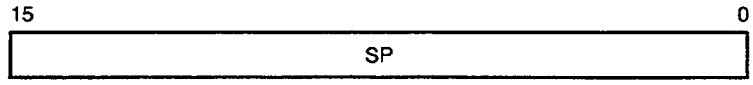
(3) Stack pointer (SP)

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 5-4. Internal High-Speed RAM Area

Product	Internal high-speed RAM area
μPD780021, 780031, 780021Y, 780031Y μPD780022, 780032, 780022Y, 780032Y	FD00H to FEFFH
μPD780023, 780033, 780023Y, 780033Y μPD780024, 780034, 780024Y, 780034Y μPD78F0034, 78F0034Y	FB00H to FEFFH

Figure 5-13. Stack Pointer Configuration



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 5-14 and 5-15.

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

Figure 5-14. Data to be Saved to Stack Memory

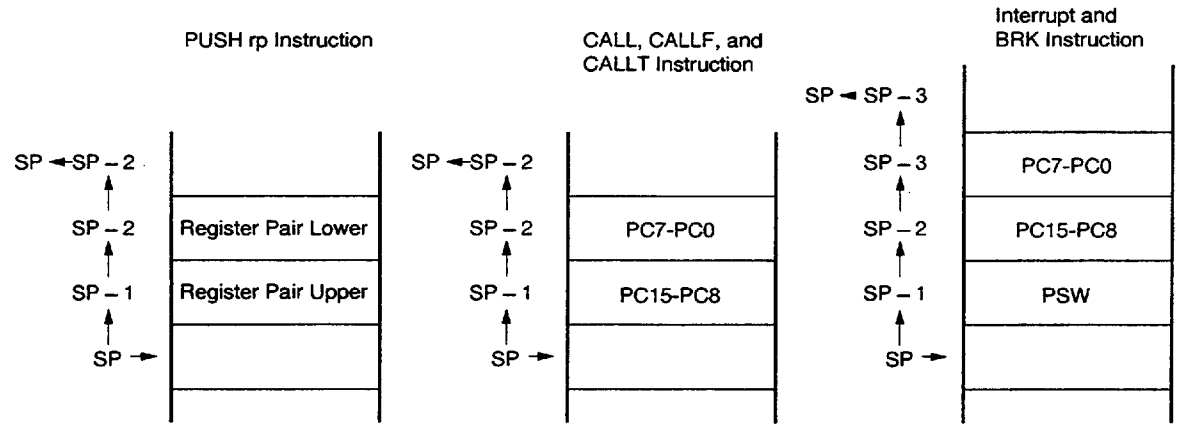
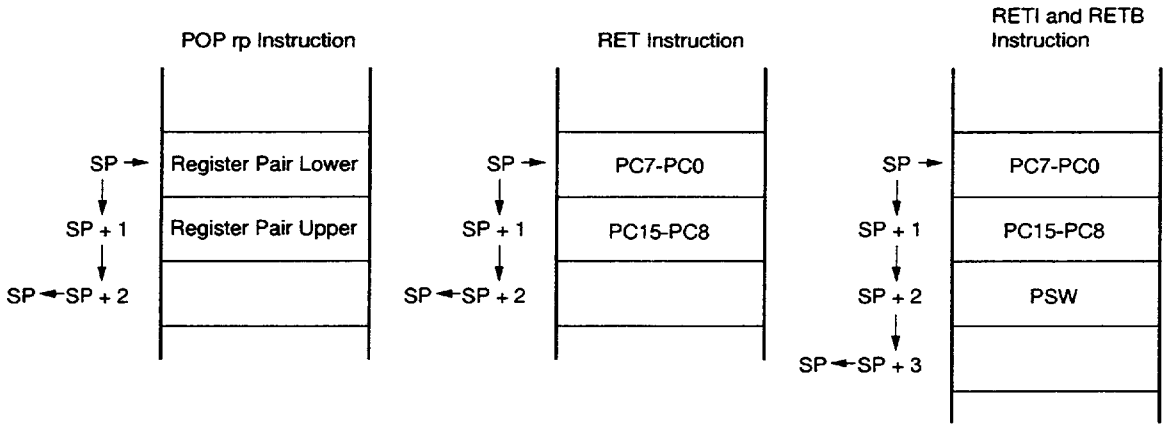


Figure 5-15. Data to be Reset from Stack Memory



5.2.2 General registers

A general 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).

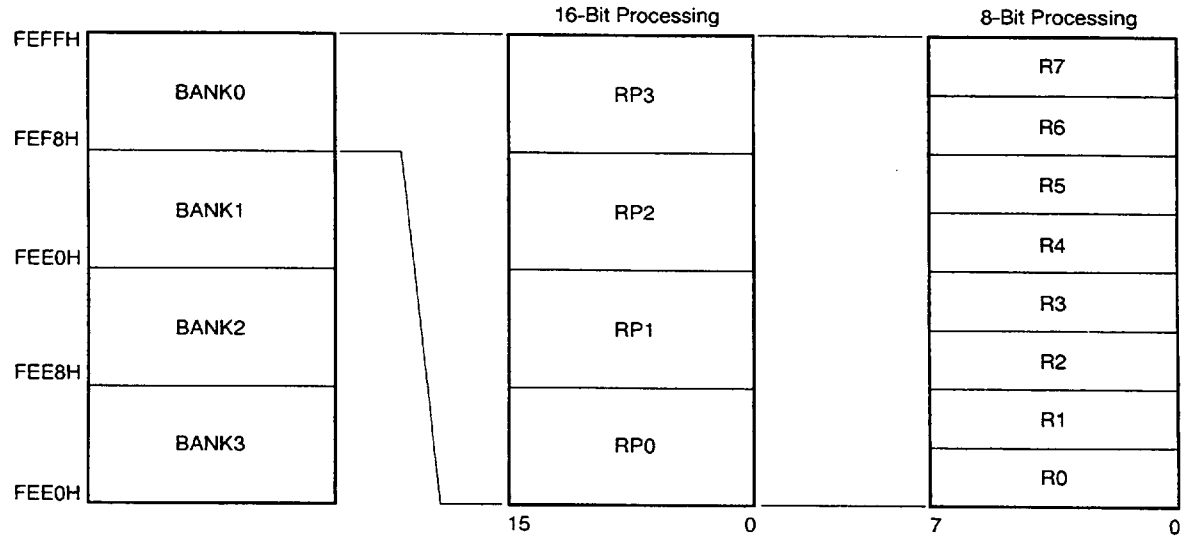
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE and HL).

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).

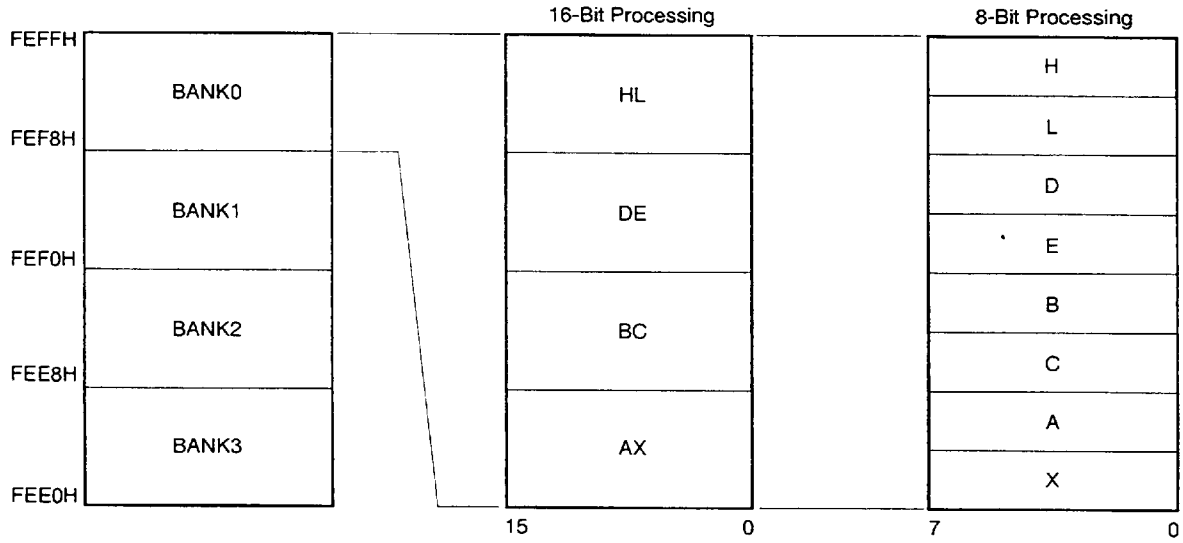
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interruption for each bank.

Figure 5-16. General Register Configuration

(a) Absolute Name



(b) Function Name



5.2.3 Special Function Register (SFR)

Unlike a general 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 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 5-5 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 RA78/0, and is defined via the header file "sfrbit.h" in the CC78K/0.
- 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 5-5. Special Function Register List (1/3)

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 bit	8 bits	16 bits	
FF00H	Port0	P0	R/W	○	○	—	00H
FF01H	Port1	P1	R	○	○	—	
FF02H	Port2	P2	R/W	○	○	—	
FF03H	Port3	P3		○	○	—	
FF04H	Port4	P4		○	○	—	
FF05H	Port5	P5		○	○	—	
FF06H	Port6	P6		○	○	—	
FF07H	Port7	P7		○	○	—	
FF0AH	Capture/compare register 00	CR00		—	—	○	Undefined
FF0BH							
FF0CH	Capture/compare register 01	CR01		—	—	○	
FF0DH							
FF0EH	16-bit timer register	TM0	R	—	—	○	0000H
FF0FH							
FF10H	8-bit compare register 50	CR50	R/W	○	○	—	Undefined
FF11H	8-bit compare register 51	CR51		○	○	—	
FF12H	8-bit counter 50	TM5 TM50	R	—	○	○	00H
FF13H	8-bit counter 51	TM51		—	○		
FF16H	A/D conversion result register 0	ADCR0		—	○Note 1	○Note 2	
FF17H							
FF18H	Transmit shift register	TXS0 W	—	○	—	FFH	
	Receive buffer register	RXB0 R	—	○	—		
FF1AH	Serial I/O shift register 30	SIO30 R/W	—	○	—	Undefined	
FF1BH	Serial I/O shift register 31Note 3	SIO31	—	○	—		
FF1FH	IIC shift registerNote 4	IIC0	—	○	—	00H	

Notes 1. μPD780024, 780024Y Subseries only

2. μPD780034, 780034Y Subseries only, 16-bit access possible

3. μPD780024, 780034 Subseries only

4. μPD780024Y, 780034Y Subseries only

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

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 bit	8 bits	16 bits	
FF20H	Port mode register 0	PM0	R/W	○	○	—	FFH
FF22H	Port mode register 2	PM2		○	○	—	
FF23H	Port mode register 3	PM3		○	○	—	
FF24H	Port mode register 4	PM4		○	○	—	
FF25H	Port mode register 5	PM5		○	○	—	
FF26H	Port mode register 6	PM6		○	○	—	
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		○	○	—	
FF36H	Pull-up resistor option register 6	PU6		○	○	—	
FF37H	Pull-up resistor option register 7	PU7		○	○	—	
FF40H	Clock output selection register	CKS		○	○	—	
FF41H	Watch timer mode control register	WTM		○	○	—	
FF42H	Watchdog timer clock selection register	WDCS		—	○	—	
FF47H	Memory expansion mode register	MEM		○	○	—	
FF48H	External interrupt rising edge enable register	EGP		○	○	—	
FF49H	External interrupt falling edge enable register	EGN		○	○	—	
FF60H	16-bit timer mode control register	TMC0		○	○	—	
FF61H	Prescaler mode register	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		○	○	—	04H
FF71H	Timer clock selection register 50	TCL50		—	○	—	00H
FF78H	8-bit timer mode control register 51	TMC51		○	○	—	04H
FF79H	Timer clock selection register 51	TCL51		—	○	—	00H
FF80H	A/D converter mode register	ADM0		○	○	—	
FF81H	Analog input channel specification register	ADS0		—	○	—	
FFA0H	Asynchronous serial interface mode register	ASIM0		○	○	—	
FFA1H	Asynchronous serial interface status register	ASIS0	R	—	○	—	
FFA2H	Baud rate generator control register	BRGC0	R/W	—	○	—	

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

Address	Special-Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 bit	8 bits	16 bits	
FFA8H	IIC control register[Note 1]	IICC0		R/W	O	O	—	00H
FFA9H	IIC status register[Note 1]	IICS0		R	O	O	—	
FFAAH	IIC clock selection register[Note 1]	IICCL0		R/W	O	O	—	
FFABH	Slave address register[Note 1]	SVA0			—	O	—	
FFB0H	Serial operation mode register 30	CSIM30			O	O	—	
FFB8H	Serial operation mode register 31[Note 2]	CSIM31			O	O	—	
FFD0H FFDFH	External access areaNote 3				O	O	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		O	O	O	00H
FFE1H	Interrupt request flag register 0H		IF0H		O	O		
FFE2H	Interrupt request flag register 1L	IF1L			O	O	—	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		O	O	O	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		O	O		
FFE6H	Interrupt mask flag register 1L	MK1L			O	O	—	
FFE8H	Priority level specification flag register 0L	PR0	PR0L		O	O	O	
FFE9H	Priority level specification flag register 0H		PR0H		O	O		
FFEAH	Priority level specification flag register 1L	PR1L			O	O	—	
FFF0H	Memory size switching register	IMS			—	O	—	CFH[Note 4]
FFF8H	Memory expansion wait setting register	MM		R/W	O	O	—	10H
FFF9H	Watchdog timer mode register	WDTM			O	O	—	00H
FFFAH	Oscillation stabilization time selection register	OSTS			—	O	—	04H
FFFBH	Processor clock control register	PCC			O	O	—	

Notes 1. μPD780024Y, 780034Y Subseries only

2. μPD780024, 780034 Subseries only

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

4. The default is CFH, but set the value corresponding to each respective product as indicated below.
μPD780021, 780031, 780021Y, 780031Y: 42H
μPD780022, 780032, 780022Y, 780032Y: 44H
μPD780023, 780033, 780023Y, 780033Y: C6H
μPD780024, 780034, 780024Y, 780034Y: C8H
μPD78F0034, 780034Y: Value for mask ROM version

5.3 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 USER'S MANUAL: Instruction (IEU-1372)**.

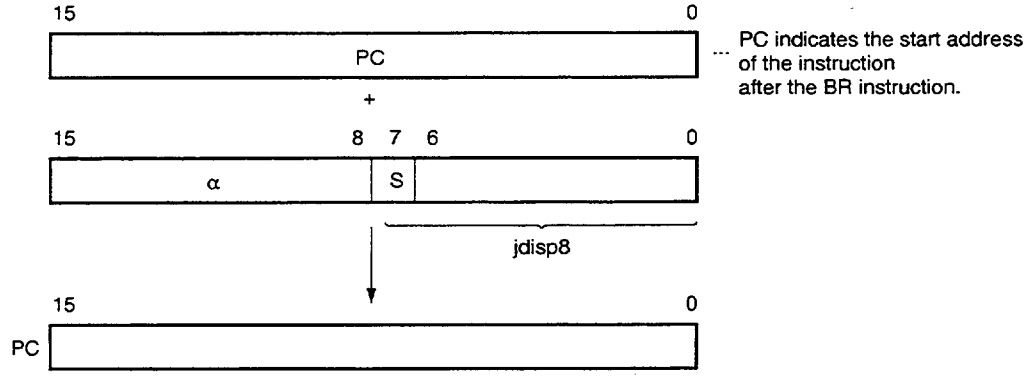
5.3.1 Relative Addressing

[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.

5.3.2 Immediate addressing

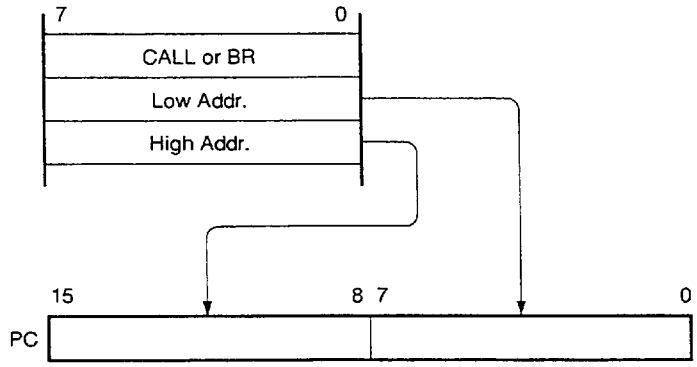
[Function]

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

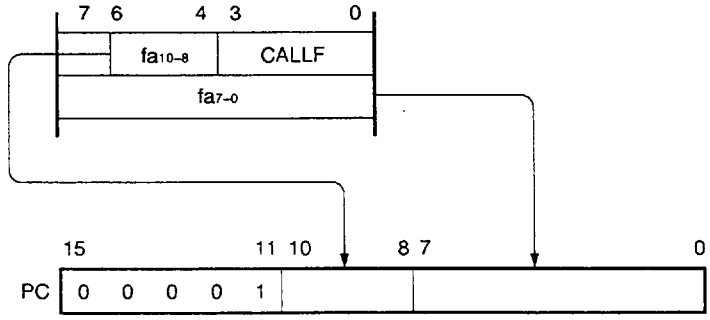
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



5.3.3 Table indirect addressing

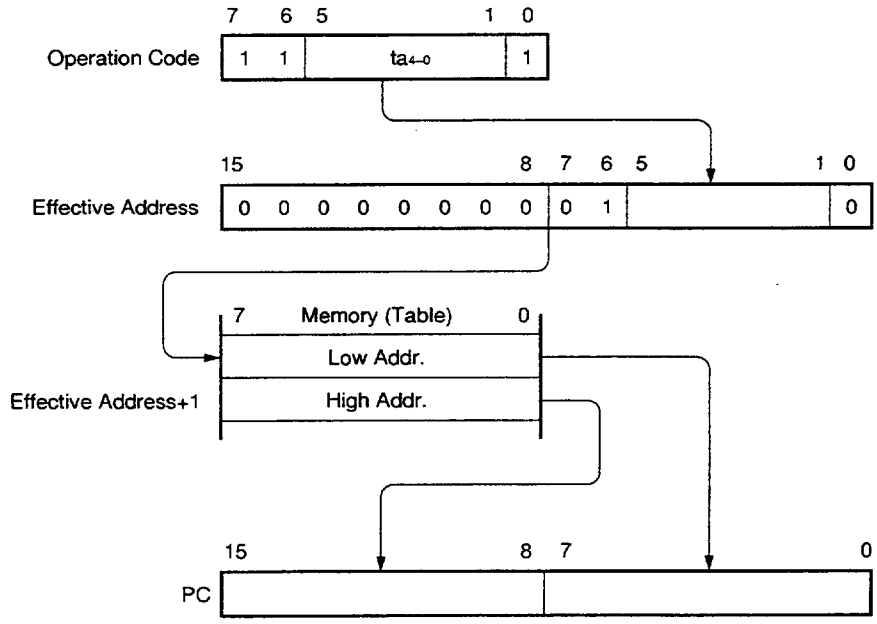
[Function]

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]



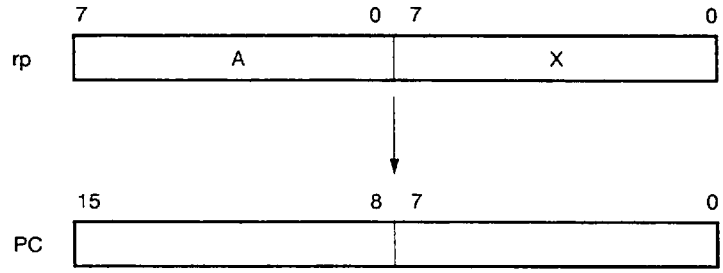
5.3.4 Register addressing

[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]



5.4 Operand Address Addressing

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

5.4.1 Implied addressing

[Function]

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

Of the μPD780024, 780034, 780024Y, 780034Y 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.

5.4.2 Register addressing

[Function]

The general 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 to 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.

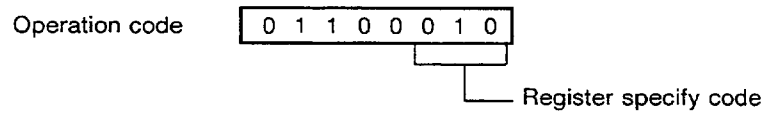
[Operand format]

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

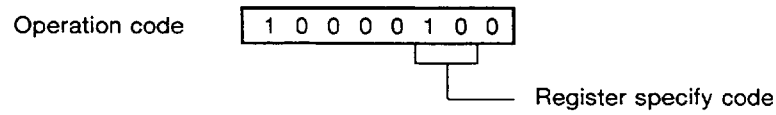
'r' and 'rp' can be described with function names (X, A, C, B, E, D, L, H, AX, BC, DE and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



5.4.3 Direct addressing

[Function]

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

[Operand format]

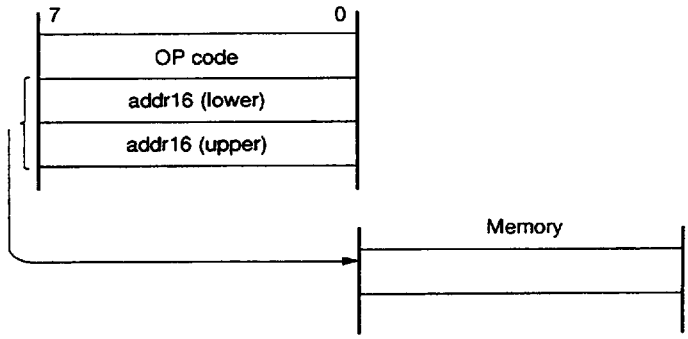
Identifier	Description
addr16	Label or 16-bit immediate data

[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]



5.4.4 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]

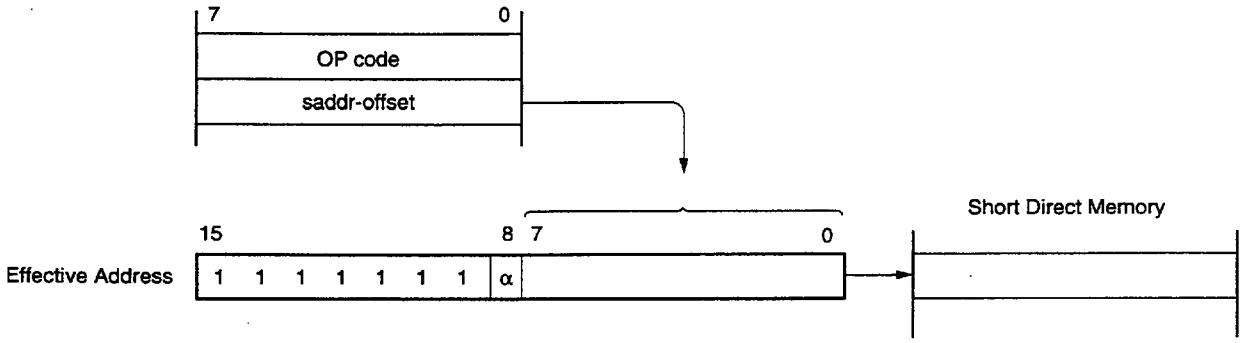
Identifier	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Description example]

MOV FE30H, #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, α = 0
When 8-bit immediate data is 00H to 1FH, α = 1

5.4.5 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)

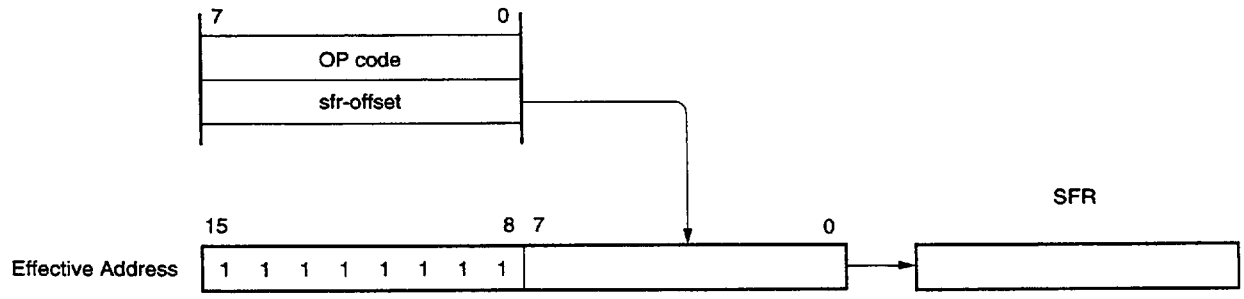
[Description example]

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]



5.4.6 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]

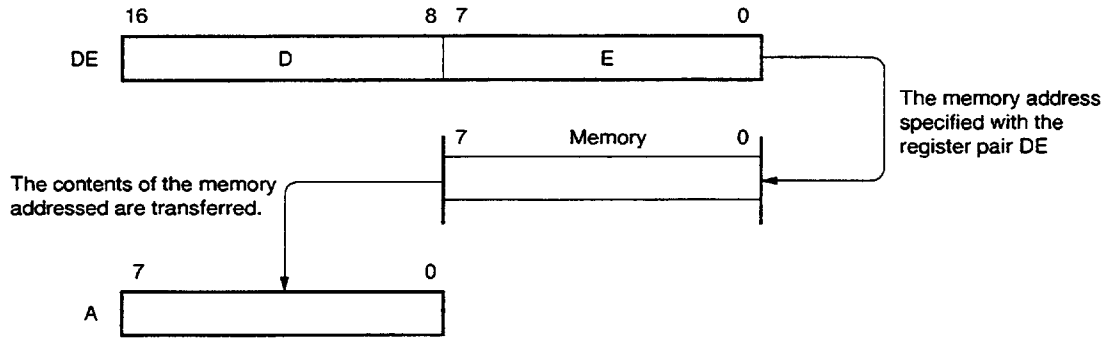
Identifier	Description
—	[DE], [HL]

[Description example]

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

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

[Illustration]



5.4.7 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

5.4.8 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 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 + B], [HL + C]

[Description example]

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

Operation code	1	0	1	0	1	0	1	1

5.4.9 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 6 PORT FUNCTIONS

6.1 Port Functions

The μPD780024, 780034, 780024Y and 780034Y subseries units incorporate eight input ports and forty-three input/output ports. Figure 6-1 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 input/output pins.

Figure 6-1. Port Types

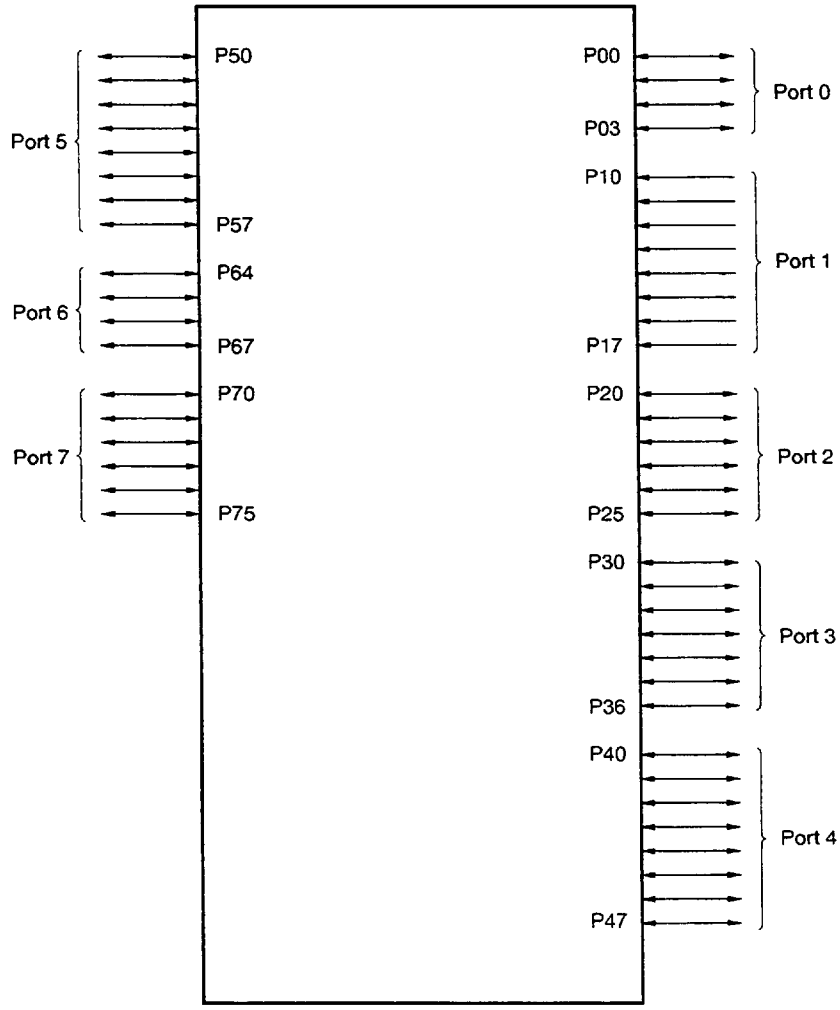


Table 6-1. Port Functions (μPD780024, 780034 subseries)

Pin Name	Function		Alternative Function
P00	Port 0 4-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		INTP0
P01			INTP1
P02			INTP2
0P3			INTP3/ADTRG
P10 - P17	Port 1 8-bit input only port.		AN10 - AN17
P20	Port 2 6-bit Input/output port Input/output mode can be specified bit-wise If used as an input port, an on-chip pull-up resistor can be used by software		SI30
P21			SO30
P22			$\overline{\text{SCK30}}$
P23			RxD0
P24			TxD0
P25			ASCK0
P30	Port 3 7-bit Input/output port Input/output mode can be specified bit-wise	N-ch open-drain Input/output port On-chip pull-up resistor can be specified by mask option (Mask version only) LEDs can be driven directly	—
P31			
P32			
P33			
P34		If used as an input port, an on-chip pull-up resistor can be specified by software	SI31
P35			SO31
P36			$\overline{\text{SCK31}}$
P40 - P47	Port 4 8-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be specified by software Interrupt request flag (KRIF) is set to 1 by falling edge detection		AD0-AD7
P50 - P57	Port 5 8-bit Input/Output port LED can be driven directly Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		A8 - A15
P64	Port 6 4-bit Input/Output port Input/output mode can be specified If used as an input port, an on-chip pull-up resistor can be used by software		$\overline{\text{RD}}$
P65			$\overline{\text{WR}}$
P66			$\overline{\text{WAIT}}$
P67			ASTB
P70	Port 7 6-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		TI00/TO0
P71			TI01
P72			TI50/TO50
P73			TI51/TO51
P74			PCL
P75			BUZ

Table 6-2. Port Functions (μPD780024Y, 780034Y subseries)

Pin Name	Function		Alternative Function
P00	Port 0 4-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software		INTP0
P01			INTP1
P02			INTP2
0P3			INTP3/ADTRG
P10 - P17	Port 1 8-bit input only port.		AN10 - AN17
P20	Port 2 6-bit Input/output port Input/output mode can be specified bit-wise If used as an input port, an on-chip pull-up resistor can be used by software		SI30
P21			SO30
P22			$\overline{SCK30}$
P23			RxD0
P24			TxD0
P25			ASCK0
P30	Port 3 7-bit Input/output port Input/output mode can be specified bit-wise	N-ch open-drain Input/output port On-chip pull-up resistor can be specified by mask option (Mask version only) LEDs can be driven directly	—
P31			
P32			SDA0
P33			SCL0
P34		If used as an input port, pull-up resistor can be used by software	—
P35			
P36			
P40 - P47	Port 4 8-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, pull-up resistor can be used by software Interrupt request flag (KRIF) is set to 1 by falling edge detection		AD0 - AD7
P50 - P57	Port 5 8-bit Input/Output port LED can be driven directly Input/output mode can be specified bit-wise. If used as an input port, pull-up resistor can be used by software		A8 - A15
P64	Port 6 4-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, pull-up resistor can be used by software		$\overline{RD}$
P65			$\overline{WR}$
P66			$\overline{WAIT}$
P67			ASTB
P70	Port 7 6-bit Input/Output port Input/output mode can be specified bit-wise. If used as an input port, a pull-up resistor can be connected by software		TI00/TO0
P71			TI01
P72			TI50/TO50
P73			TI51/TO51
P74			PCL
P75			BUZ

6.2 Port Configuration

A port consists of the following hardware:

Table 6-3. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0, 2 to 7) Pull-up resistor option register (PUOm,: m = 0, 2 to 7)
Port	Total: 51 ports (8 inputs, 43 inputs/outputs)
Pull-up resistor	• Mask ROM version Total: 43 (software specifiable: 39, mask option: 4) • Flash memory version Total: 39

6.2.1 Port 0

Port 0 is an 4-bit input/output 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). When P00 to P03 pins are used as input ports, an on-chip pull-up resistor can be used to them in 6-bit units with a pull-up resistor option register0 (PU0).

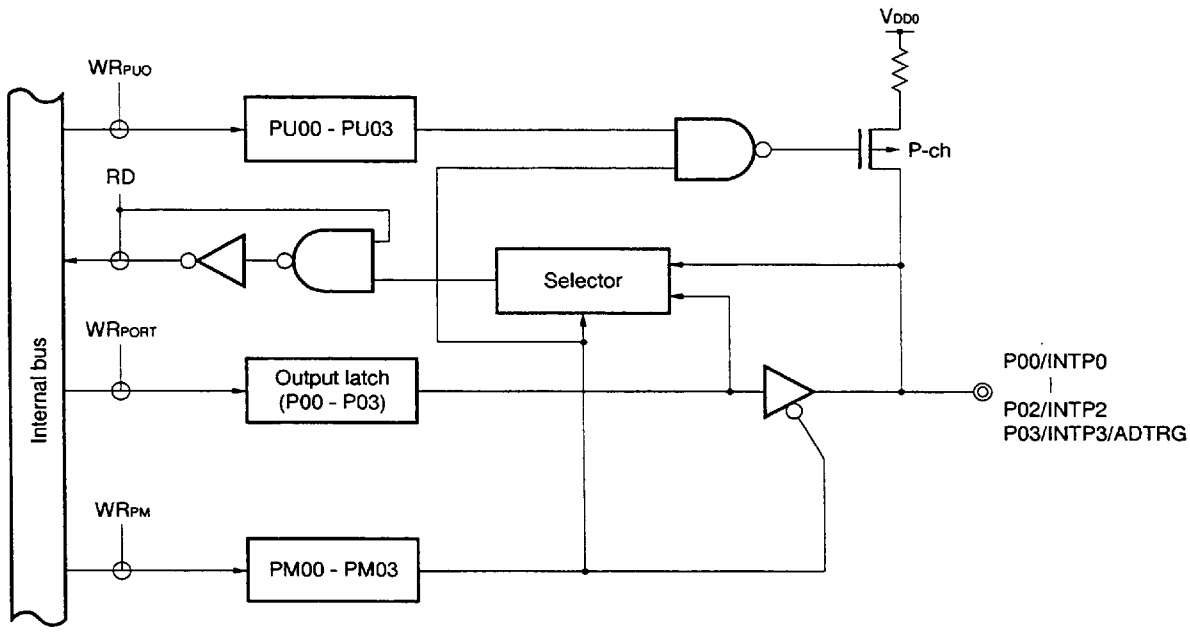
Dual-functions include external interrupt input, and A/D converter external trigger input.

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

Figures 6-2 show block diagrams of port0.

Caution Because port 0 also serves for external interrupt 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 6-2. P00 and P03 Configurations

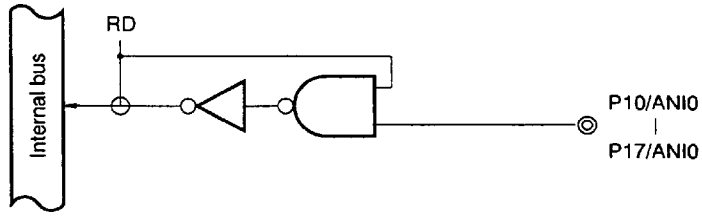


PU : Pull-up resistor option register
PM : Port mode register
RD : Port 0 read signal
WR : Port 0 write signal

6.2.2 Port 1

Port 1 is an 8-bit input only port.
Dual-functions include an A/D converter analog input.
Figure 6-3 shows a block diagram of port 1.

Figure 6-3. P10 to P17 Configurations



RD : Port 1 read signal

6.2.3 Port 2

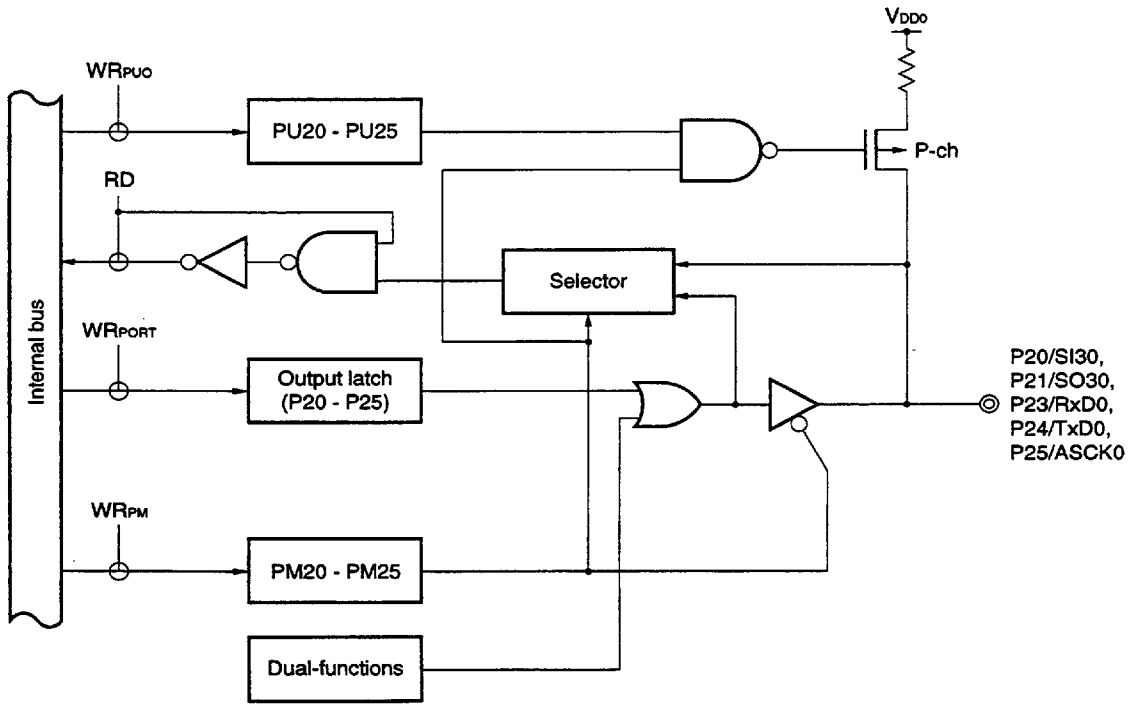
Port 2 is an 6-bit input/output port with output latch. P20 to P25 pins can specify the input mode/output mode in 1-bit units with the port mode register 2 (PM2). When P20 to P25 pins are used as input ports, an on-chip pull-up resistor can be used to them in 1-bit units with a pull-up resistor option register 2 (PU2).

Dual-functions include serial interface data input/output and clock input/output.

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

Figures 6-4 show a block diagram of port 2.

Figure 6-4. P20 to P25 Configurations



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

6.2.4 Port 3 (μPD780024, 780034 Subseries)

Port 2 is an 7-bit input/output port with output latch. P30 to P36 pins can specify the input mode/output mode in 1-bit units with the port mode register 3.

This port has the following functions for pull-up resistor. These functions depend on. High 36-bits/low 4 bits MASK ROM product/Flash memory product.

Table 6-4. Pull-up resistor of port 3 (μPD780024, 780034 subseries)

	High 3 bits (34 to P36 pins)	Low 4 bits (30 - P33 pins)
Mask ROM products	An on-chip pull-up resistor can be used bit-wise by PU 3	on-chip pull-up resistor can be specified bit-wise by mask option
Flash memory products		On-chip pull-up resistor is not be specified

PU3: Pull-up resistor option register 3

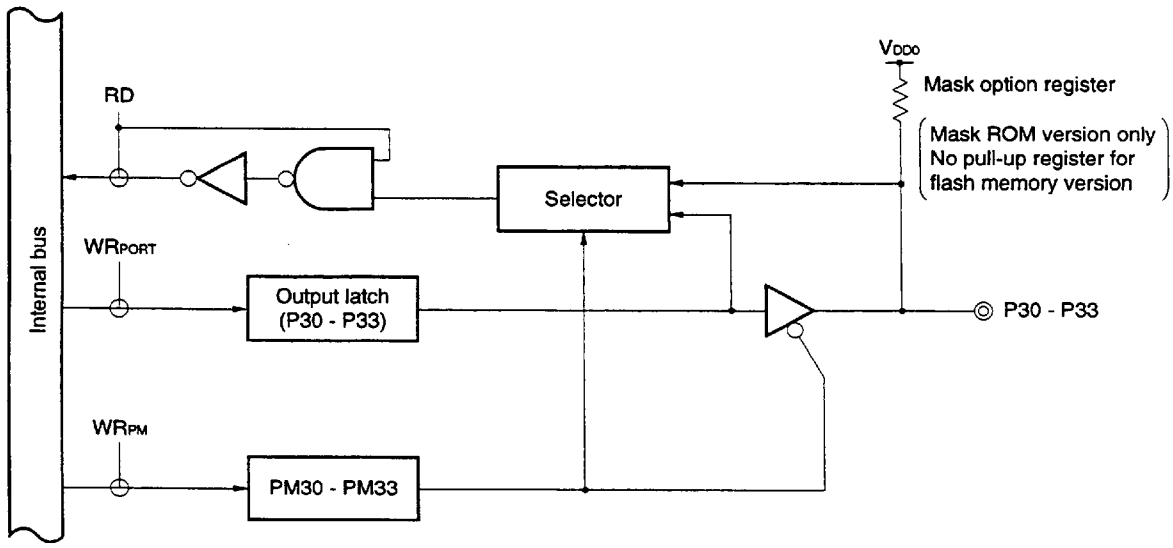
P30 to P33 pins can drive LED directly.

Dual-functions for P34 to P36 pins include serial interface data input/output and clock input/output.

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

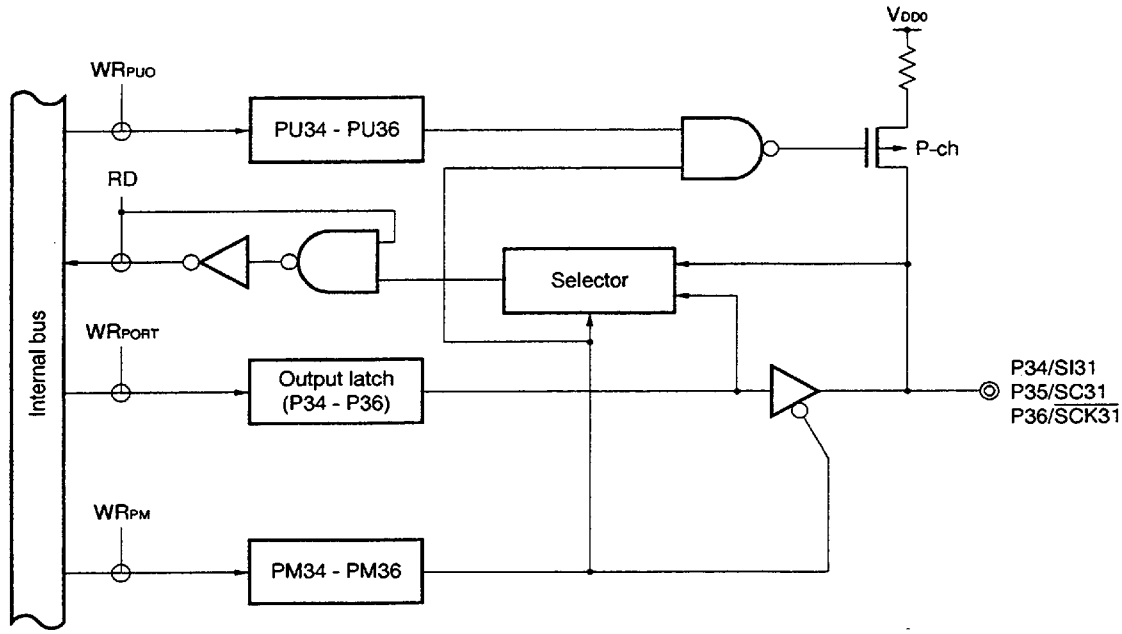
Figures 6-5 and 6-6 show a block diagram of port 3.

Figure 6-5. P30 to P33 Configurations (μPD780024, 780034 Subseries)



PM : Port mode register
RD : Port 3 read signal
WR : Port 3 write signal

Figure 6-6. P34 to P36 Configurations (μPD780024, 780034 Subseries)



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

6.2.5 Port 3 (μPD780024Y, 780034Y Subseries)

Port 3 is an 7-bit input/output port with output latch. P30 to P36 pins can specify the input mode/output mode in 1-bit units with the port mode register 3 (PM3).

This port has following functions about pull-up resistor.

These functions differ depending on high 3bits/low 4 bits and Mask ROM version/Flash memory version.

Table 6-5. Pull-up resistor of port 3 (μPD780024, 780034 Subseries)

	high 3 bits (34 to P36 pins)	low 4 bits (30 to P33 pins)
Mask ROM products	A pull-up resistor can be connected bit-wise by P03	On-chip pull-up resistor can be specified bit-wise by mask option
Flash memory version		On-chip pull-up resistor don't be specified

PU3: Pull-up resistor option register 3

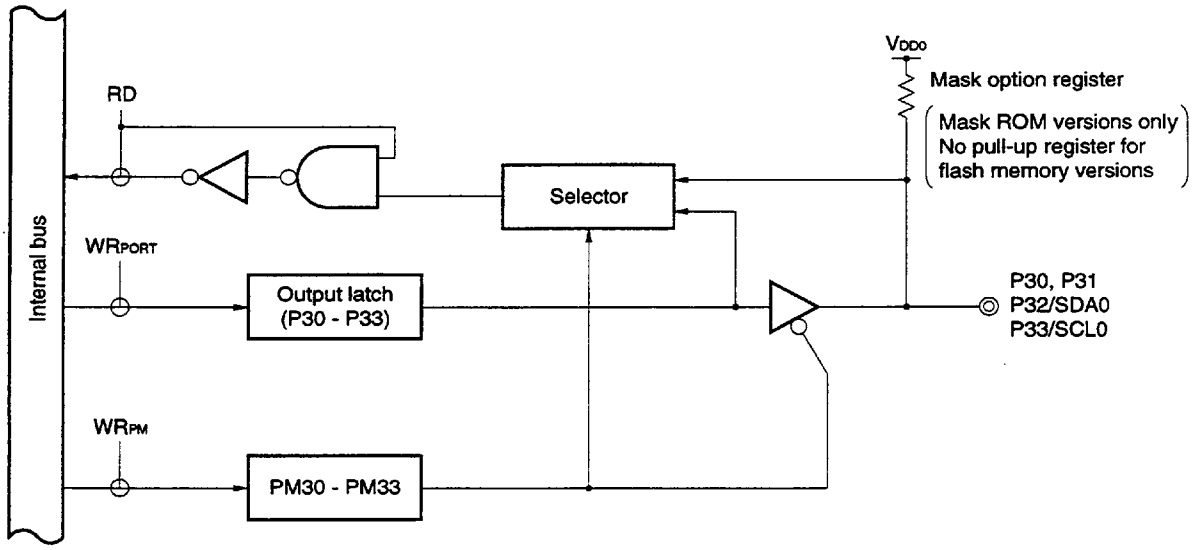
P30 to P33 pins can drive LED directly.

Dual-functions for P32 and P33 pins include interface data input/output and clock input/output.

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

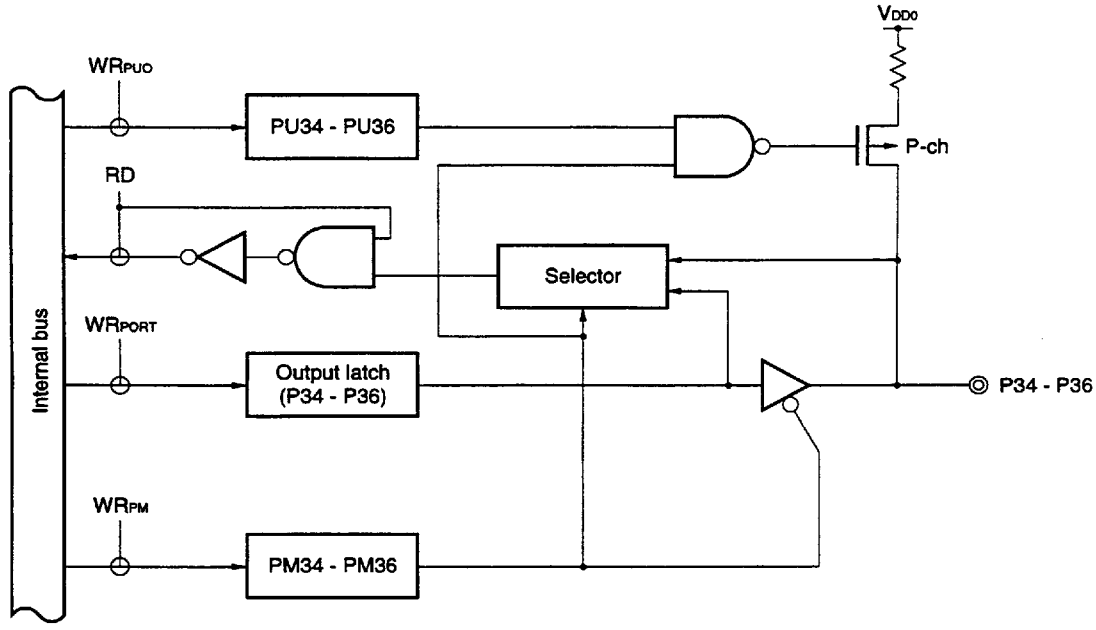
Figure 7,8 shows a block diagram of port 3.

Figure 6-7. P30 to P33 Configurations (μPD780024Y, 780034Y Subseries)



PM : Port mode register
RD : Port 3 read signal
WR : Port 3 write signal

Figure 6-8. P34 to P36 Configurations (μPD780024Y, 780034Y, Subseries)



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

6.2.6 Port 4

Port 4 is an 8-bit input/output port with output latch. P40 to P47 pins can specify the input mode/output mode in 1-bit units with the port mode register 4 (PM4). When P40 to P47 pins are used as input ports, a pull-up resistor can be connected to them in 1-bit units with pull-up resistor option register 4 (PU4).

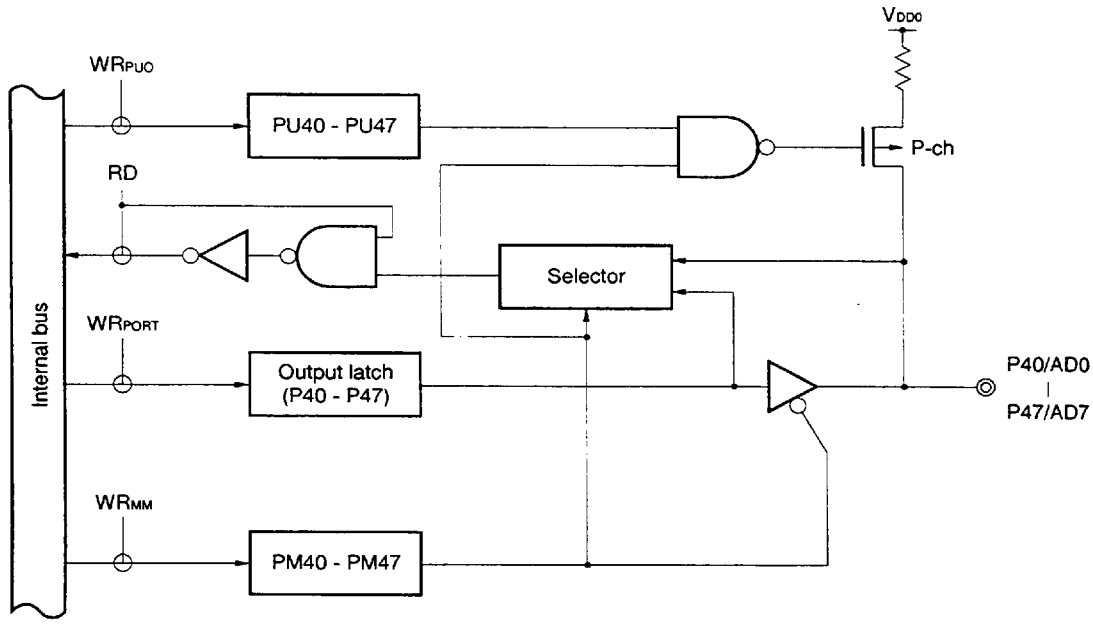
The interrupt request flag (KRIF) can be set to 1 by detecting falling edges.

Dual-functions also include address/data bus function in external memory expansion mode.

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

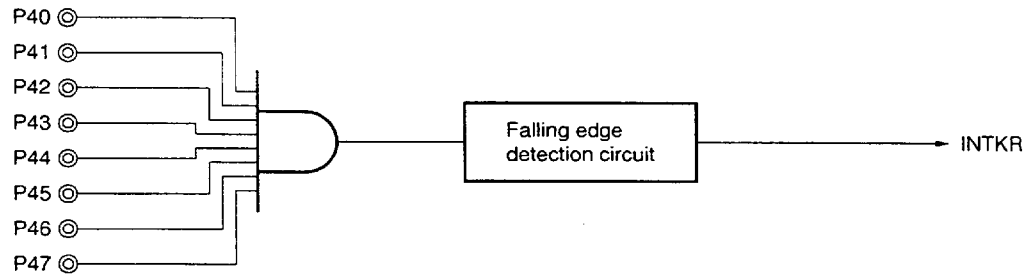
Figures 6-9 and 6-10 show a block diagram of port 4 and block diagram of falling edge detection circuit, respectively.

Figure 6-9. P40 to P47 Configurations



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

Figure 6-10. Block Diagram of Falling Edge Detection Circuit



6.2.7 Port 5

Port 5 is an 8-bit input/output port with output latch. P50 to P57 pins can specify the input mode/output mode in 1-bit units with the port mode register 5 (PM5). When P50 to P57 pins are used as input ports, an on-chip pull-up resistor can be used to them in 1-bit units with a pull-up resistor option register 5 (PU5).

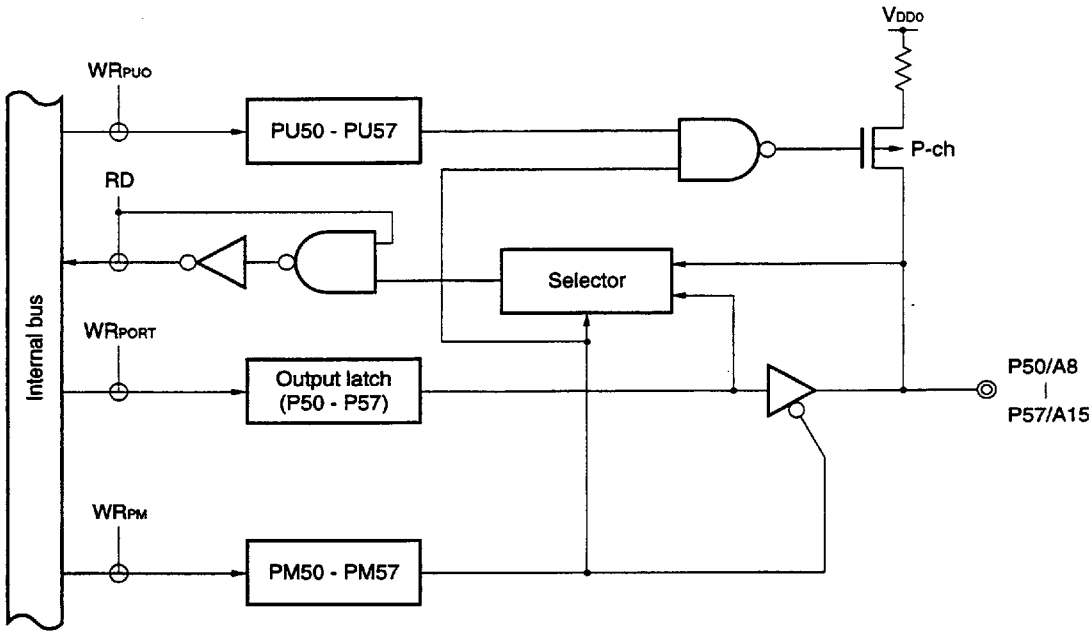
Port 5 can drive LEDs directly.

Dual-functions include address bus function in external memory expansion mode.

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

Figure 6-11 shows a block diagram of port 5.

Figure 6-11. P50 to P57 Configurations



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

6.2.8 Port 6

Port 6 is an 4-bit input/output port with output latch. P64 to P67 pins can specify the input mode/output mode in 1-bit units with the port mode register 6 (PM6).

When pins P64 to P67 are used as input ports, an on-chip pull-up resistor can be used to them in 1-bit units with pull-up resistor option register 6 (PU6). Mask ROM versions can have pull-up resistors set to pins P60 to P63 by the mask option. The PROM versions have no mask option and therefore specifying pull-up resistors to the P00 through P63 pins is impossible.

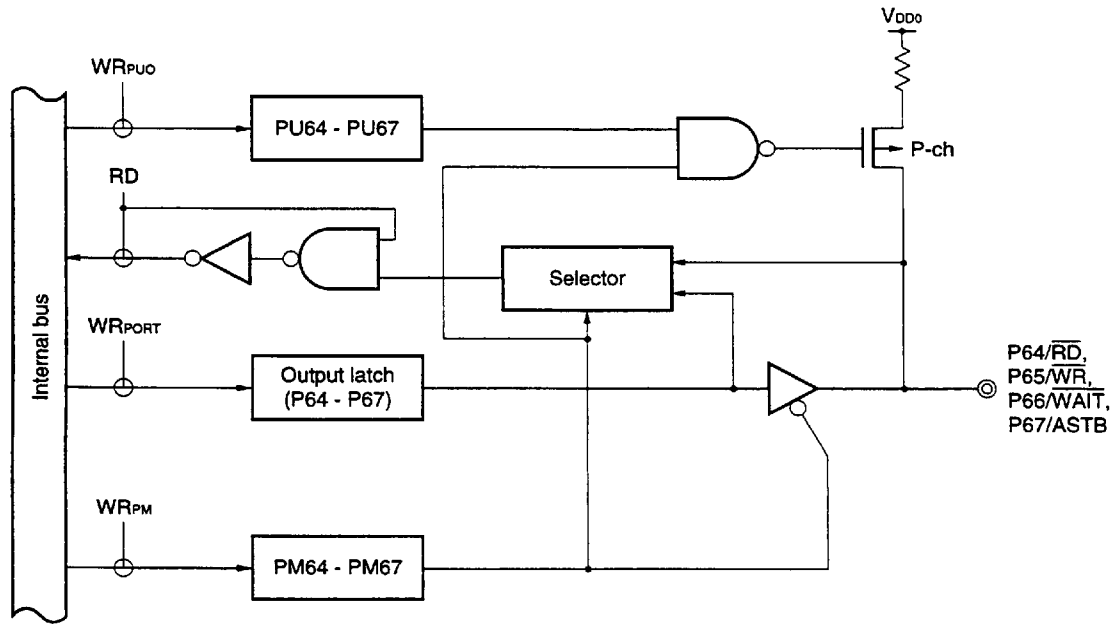
Dual-functions include the control signal output function in external memory expansion mode.

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

Figures 6-12 shows block diagrams of port 6.

Caution When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

Figure 6-12. P64 to P67 Configurations



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

6.2.9 Port 7

This is a 6-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 7 (PM7). When pins P70 to P75 are used as input port pins, an on-chip pull-up resistor can be used as a 1-bit unit by means of pull-up resistor option register 7 (PU7).

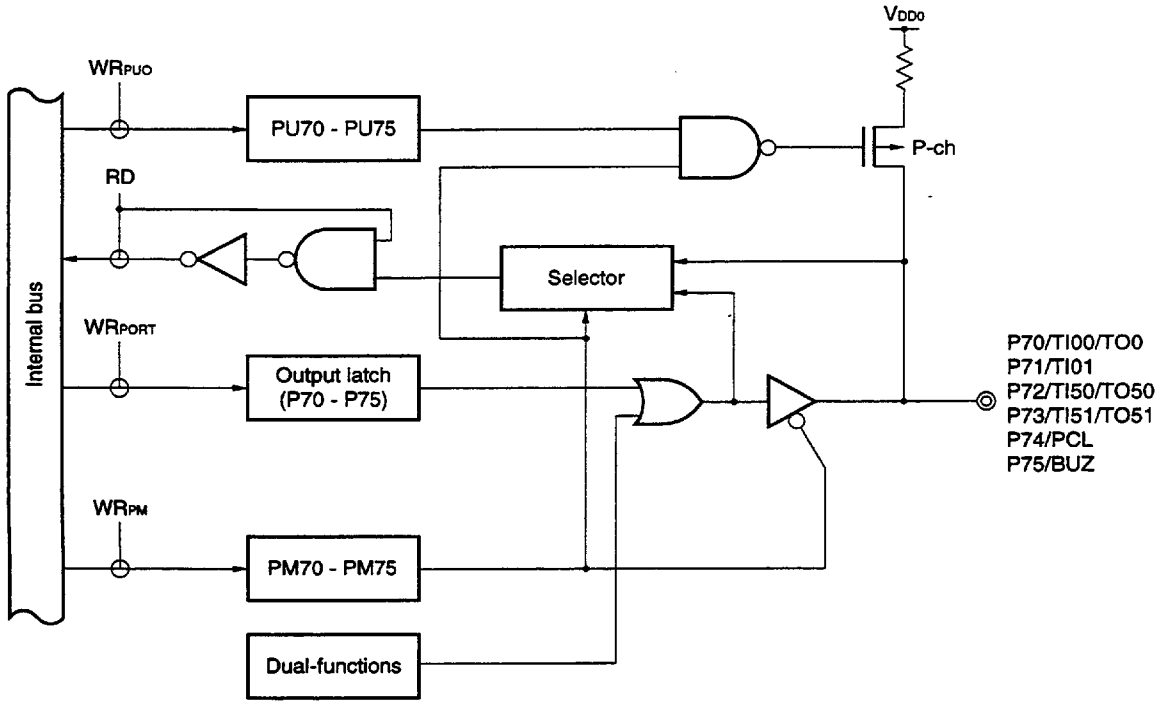
Dual-functions also include timer input/output, clock output and buzzer output.

$\overline{\text{RESET}}$ input sets the input mode.

Port 7 block diagrams are shown in Figures 6-13.

Caution Because P70, P71 also serves for external interrupt 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 6-13. P70 to P75 Configurations



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

6.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM2 to PM7)
- Pull-up resistor option register (PU0, PU 2 to PU7)

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

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

PM0 and PM2 to PM7 are independently set with a 1-bit or 8-bit memory manipulation instruction

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

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

2. As port 0, P70 and P71 has a dual function as external interrupt input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. When the output mode is used, therefore, the interrupt mask flag should be set to 1 beforehand.

Figure 6-14. Port Mode Register (PM0, PM2 to PM7) Format

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	PM33	PM32	PM31	PM30

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: FF26H After Reset : FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	1	1	1	1

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 Input/output Mode Select (m = 0, 2 to 7 ; n = 0 to 7)
0	Output Mode (Output buffer on)
1	Input Mode (Output buffer off)

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

This register is used to set whether to use an internal pull-up resistor at each port or not. A pull-up resistor is internally used at bits which are set to the input mode at a port where pull-up resistor use has been specified with PU0, PU2 to PU7. No pull-up resistors can be used to the bits set to the output mode irrespective of PU0 or PU2 to PU7 setting.

PU0 and PU2 to PU7 are set with a 1-bit or 8-bit memory manipulation instruction.

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

Cautions 1. P10 and P17 pins do not incorporate a pull-up resistor.

2. When ports 4 and 6 pins are used as dual-function pins, a pull-up resistor cannot be used even if 1 is set in PUmn (m = 0, 2 to 7:n = 0 to 7).

3. Pins P30 to P33 can be used with pull-up resistor by mask option only for mask ROM version.

Figure 6-15. Pull-Up Resistor Option Register (PU0, PU2 to PU7) Format

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: FF36H After Reset : 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU6	PU67	PU66	PU65	PU64	0	0	0	0

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

PMmn	Pmn pin Internal pull-up resistor select (m = 0, 2 to 7 ; n = to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

6.4 Port Function Operations

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

6.4.1 Writing to input/output 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 except for the manipulated bit.

6.4.2 Reading from input/output 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.

6.4.3 Operations on input/output 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.

6.5 Selection of Mask Option

The following mask option is provided in mask ROM version. The Flash memory versions have no mask options.

Table 6-6. Comparison between Mask ROM Version and Flash memory Version

Pin Name	Mask ROM Version	Flash memory Version
Mask option for pins P30 to P33	Bit-wise-selectable on-chip pull-up resistors	No on-chip pull-up resistor

CHAPTER 7 CLOCK GENERATOR

7.1 Clock Generator Functions

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, not using the internal feedback resistance (PCC) can be set by the processor clock control register. This enables to decrease power consumption in the STOP mode.

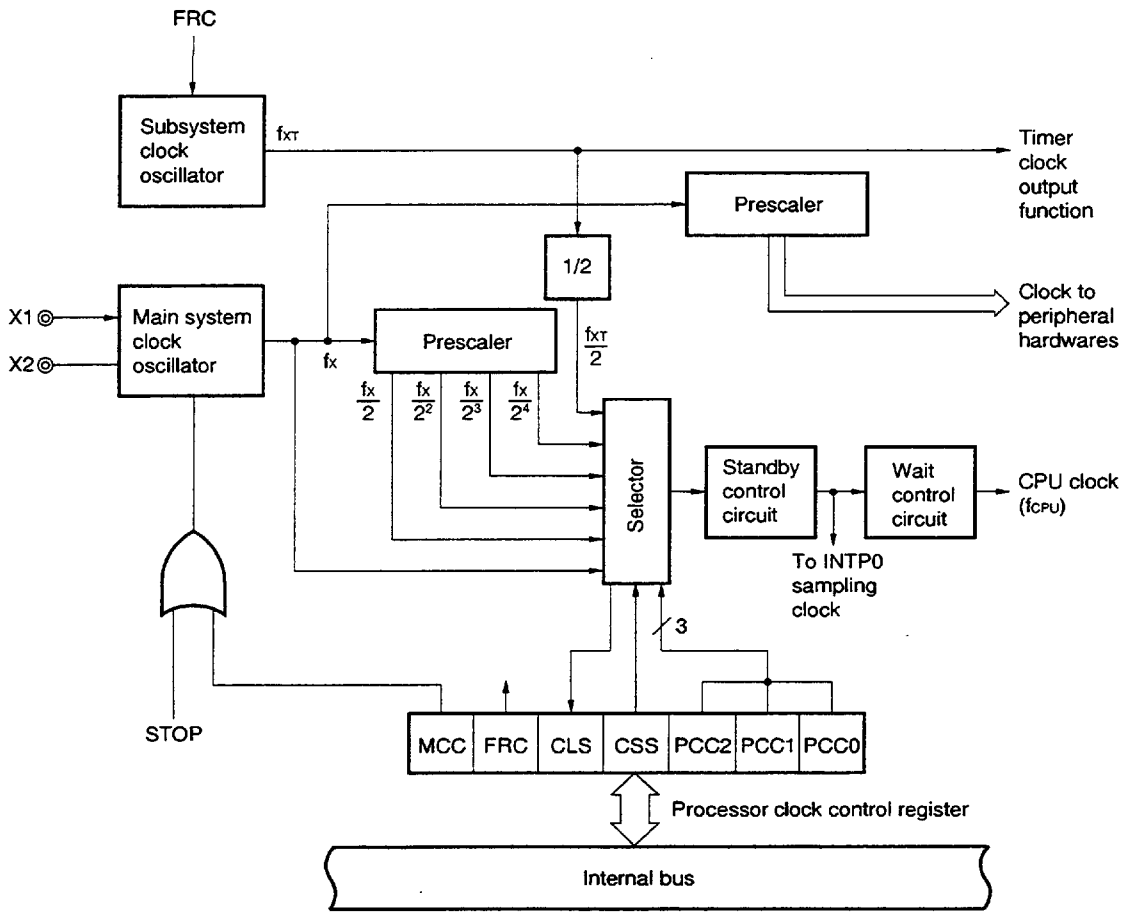
7.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 7-1. Clock Generator Configuration

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

Figure 7-1. Block Diagram of Clock Generator



7.3 Clock Generator Control Register

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 with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-2. Subsystem Clock Feedback Resistor

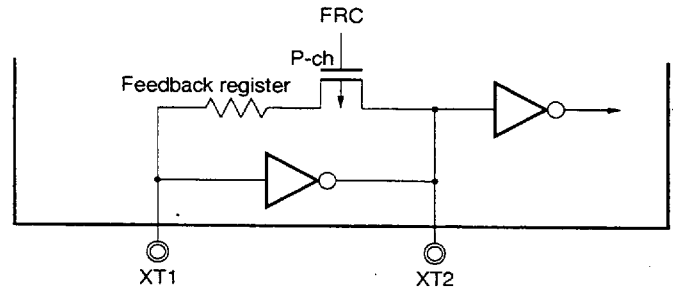


Figure 7-3. Processor Clock Control Register (PCC) Format

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 select
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 select
0	0	0	0	f_X (0.24 μs)
	0	0	1	$f_X/2$ (0.48 μs)
	0	1	0	$f_X/2^2$ (0.95 μs)
	0	1	1	$f_X/2^3$ (1.91 μs)
	1	0	0	$f_X/2^4$ (3.81 μs)
1	0	0	0	f_{XT} (122 μs)
	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.

Caution Bit 3 must be set to 0.

Remarks 1. f_X : Main system clock oscillator frequency

2. f_{XT}: Subsystem clock oscillator frequency

3. Figures in parentheses indicate minimum instruction execution time: $2/f_{CPU}$ when operating at f_X = 8.38 MHz or f_{XT} = 32.768 kHz.

7.4 System Clock Oscillator

7.4.1 Main system clock oscillator

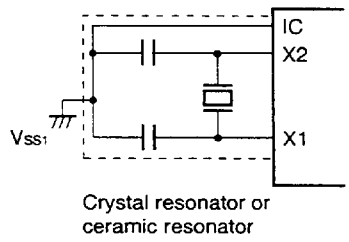
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 8.38 MHz) connected to the X1 and X2 pins.

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

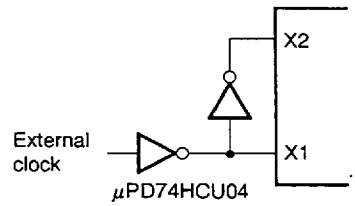
Figure 7-4 shows an external circuit of the main system clock oscillator.

Figure 7-4. External Circuit of Main System Clock Oscillator

(a) Crystal and ceramic oscillation



(b) External clock



Caution Do not execute the STOP instruction or do not set MCC [bit 7 of processor clock control register (PCC)] to 1 if an external clock is used. This is because the X2 pin is connected to VDD, via a pull-up register.

7.4.2 Subsystem clock oscillator

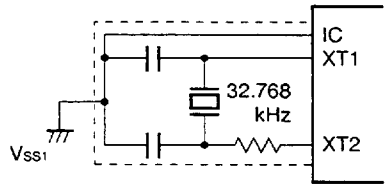
The subsystem clock oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

External clocks can be input to the main system clock oscillator. In this case, input a clock signal to the XT1 pin and an antiphase clock signal to the XT2 pin.

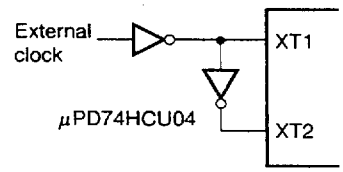
Figure 7-5 shows an external circuit of the subsystem clock oscillator.

Figure 7-5. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



Cautions 1. When using a main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken line area in Figures 7-4 and 7-5 to prevent any effects from wiring capacities.

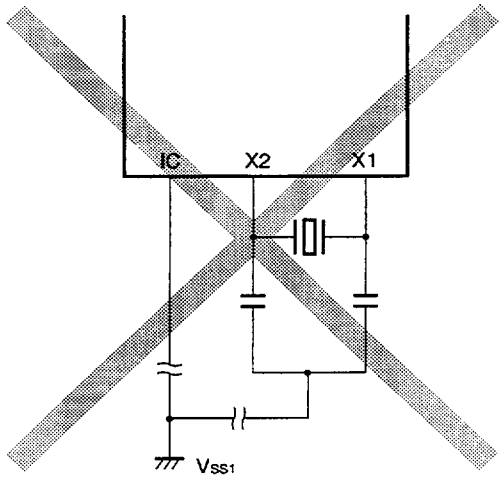
- **Minimize the wiring length.**
- **Do not allow wiring to intersect with other signal conductors. Do not allow wiring to come near changing high current.**
- **Set the potential of the grounding position of the oscillator capacitor to that of V_{SS}. Do not ground to any ground pattern where high current is present.**
- **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.

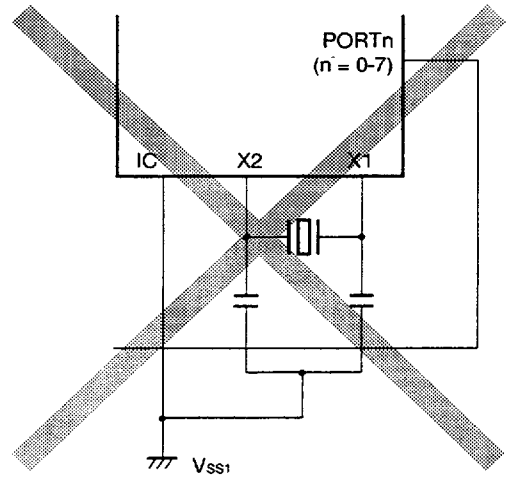
Figure 7-6 shows examples of oscillator having bad connection.

Figure 7-6. Examples of Oscillator with Bad Connection (1/2)

(a) Wiring of connection circuits is too long



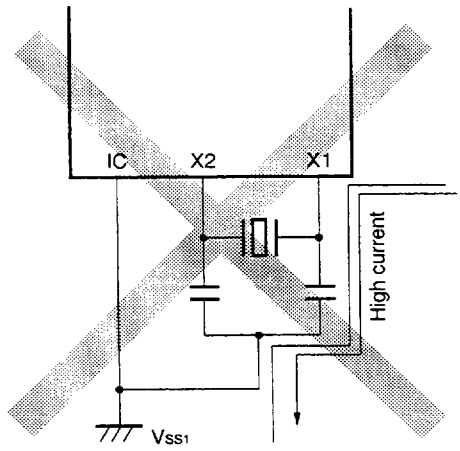
(b) Signal conductors intersect each other



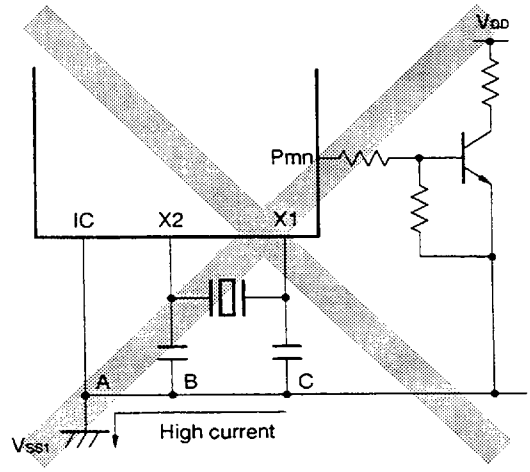
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Figure 7-6. Examples of Oscillator with Bad Connection (2/2)

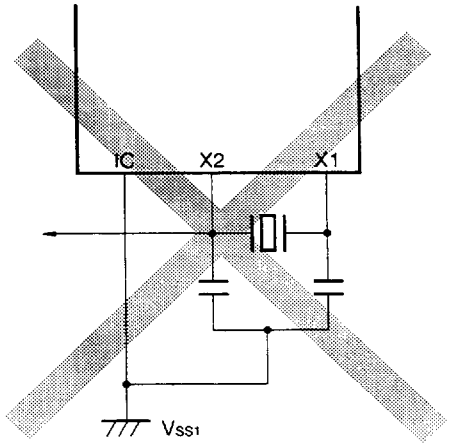
(c) Changing high current is too near a signal conductor



(d) Current flows through the grounding line of the oscillator (potential at points A, B, and C fluctuate)



(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.

Caution 2. When X2 and XT1 are wired in parallel, the cross-talk 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 Vss.

7.4.3 Scaler

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

7.4.4 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 resistance 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.

7.5 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 when operated at 8.38 MHz) 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, 1.9 μs, 3.8 μs, @ 8.38 MHz) 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 decrease 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 when operated at 32.768 kHz).

(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 16-bit timer/event counter, the watch timer, and clock output functions only. Thus, 16-bit timer/event counter (when selecting watch timer output for count clock operating with subsystem clock), 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)

7.5.1 Main system clock operations

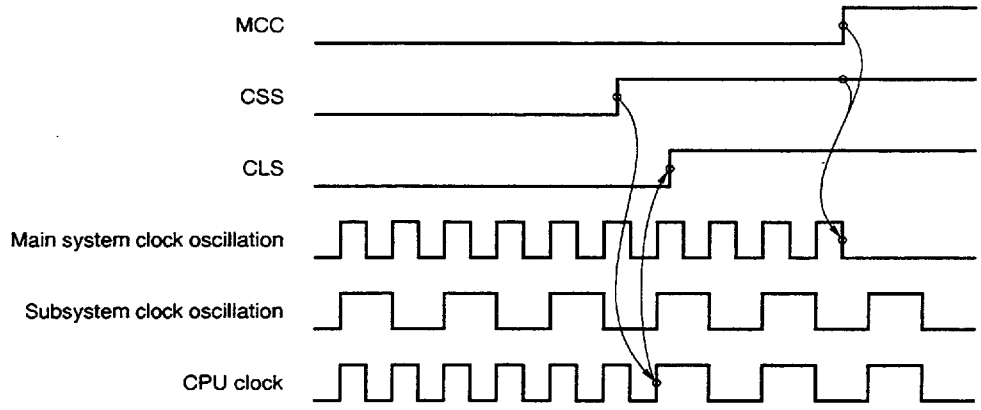
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

(a) Because the operation guarantee instruction execution speed depends on the power supply voltage, the 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 7-7**).

Figure 7-7. 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

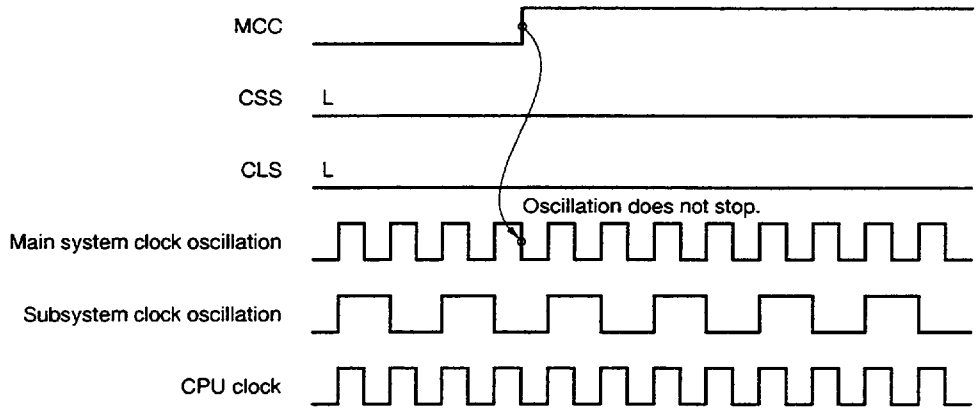
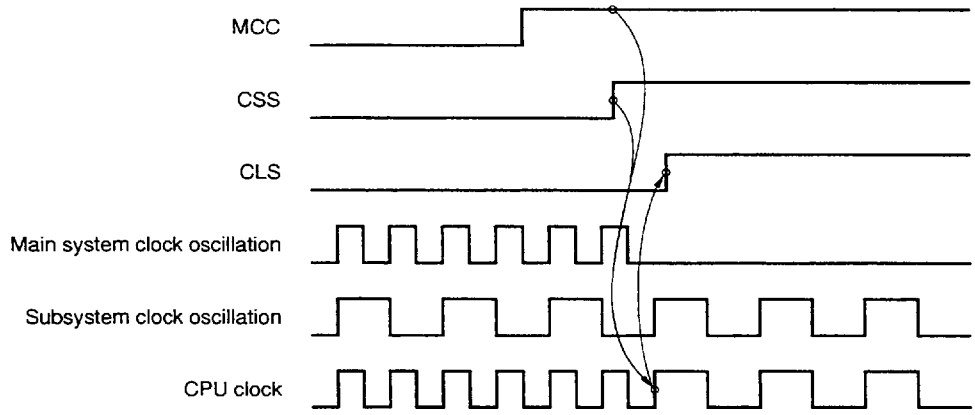


Figure 7-7. Main System Clock Stop Function (2/2)

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



7.5.2 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 instruction execution time remains constant (122 μs when operated at 32.768 kHz) 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.

7.6 Changing System Clock and CPU Clock Settings

7.6.1 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 7-2).

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 7-2. 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 instruction				16 instruction				16 instruction				16 instruction				fx/2fxT instruction (77 instruction)			
	0	0	1	8 instruction								8 instruction				8 instruction				8 instruction				fx/4fxT instruction (39 instruction)			
	0	1	0	4 instruction				4 instruction								4 instruction				4 instruction				fx/8fxT instruction (20 instruction)			
	0	1	1	2 instruction				2 instruction				2 instruction								2 instruction				fx/16fxT instruction (10 instruction)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								fx/32fxT instruction (5 instruction)			
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 paretheses apply to operation with fx = 8.38 MHz and fxr = 32.768 kHz.

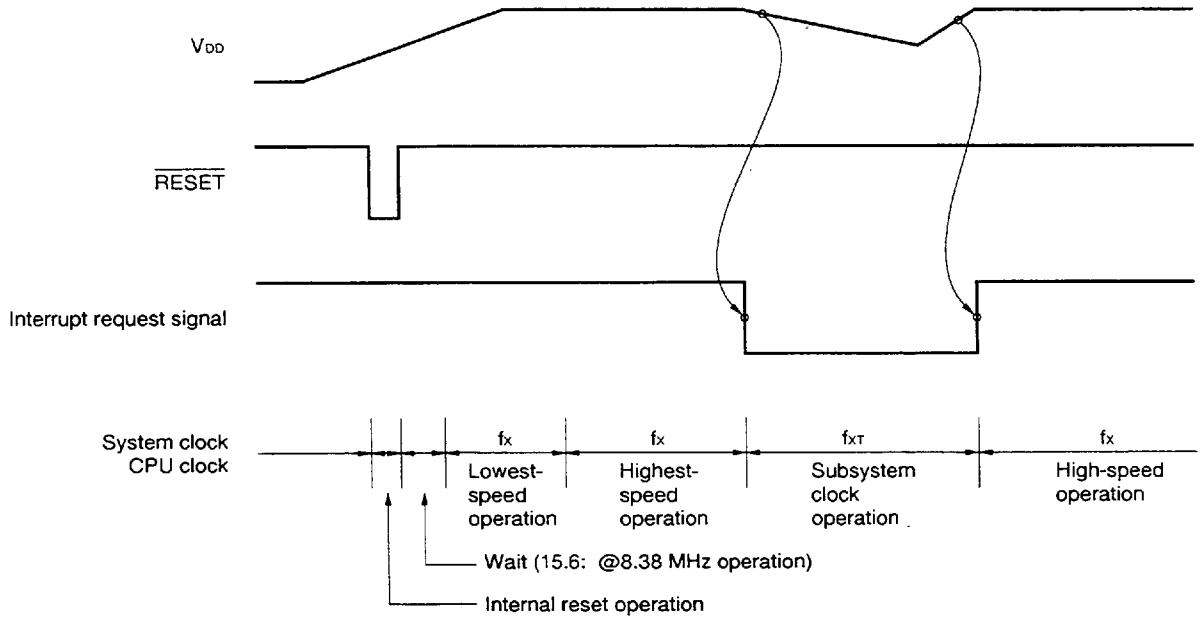
Caution Selection of the CPU clock cycle scaling factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be performed simultaneously.

Simultaneous setting is possible, however, for selection of the CPU clock cycle scaling factor (PCC0 to PCC2) and switch over from the subsystem clock to the main system clock (changing CSS from 1 to 0).

7.6.2 System clock and CPU clock switching procedure

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

Figure 7-8. 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 when operated at 8.38 MHz).

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

<3> Upon detection of a decrease of the VDD 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 VDD voltage reset due to an interrupt, 0 is set to the MCC 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 main system clock was stopped, if switching to the main system clock is made again, be sure to switch after securing oscillation stable time by software.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 8 16-BIT TIMER/EVENT COUNTER TM0

The timers incorporated into the μPD780024, 780024Y, 780034, and 780034Y subseries are outlined below.

(1) 16-bit timer/event counter TM0

The TM0 can be used for an interval timer, PWM output, pulse widths measurement (infrared ray remote control receive function), external event counter, square wave output of any frequency or one-shot pulse output.

(2) 8-bit timers/event counters TM5

TM5 can be used to serve as an interval timer and an external event counter and to output square waves with any selected frequency. Two 8-bit timer/event counters can be used as one 16-bit timer/event counter (See **CHAPTER 9 8-BIT TIMER/EVENT COUNTERS TM5**).

(3) Watch timer (WT)

This timer can set a flag every 0.5 sec. and simultaneously generates interrupt request at the preset time intervals (See **CHAPTER 10 WATCH TIMER**).

(4) Watchdog timer (WDT)

WDTM can perform the watchdog timer function or generate non-maskable interrupt request, maskable interrupt request and $\overline{\text{RESET}}$ at the preset time intervals (See **CHAPTER 11 WATCHDOG TIMER**).

(5) Clock output,/buzzer output control circuit CKU

Clock output supplies other devices with the divided main system clock and the subsystem clock, and buzzer output supplies the buzzer frequency with the divided main system clock (See **CHAPTER 12 CLOCK OUTPUT/ BUZZER OUTPUT CONTROL CIRCUIT**).

Table 8-1. Timer/Event Counter Types and Functions

		16-bit Timer/event event Counter TM0	8-bit Timer/event event Counter TM0	Watch Timer	Watchdog Timer
Type	Interval timer	2 channels[Note3]	2 channel	1 channel[Note1]	1 channel[Note2]
	External event counter	○	○	–	–
Function	Timer output	○	○	–	–
	PWM output	○	○	–	–
	Pulse width measurement	○	–	–	–
	Square-wave output	○	○	–	–
	One-shot pulse output	○	–	–	–
	Interrupt request	○	○	○	○
	Test input	–	–	○	–

Notes 1. Watch time 4 can perform both watch timer and interval timer functions at the same time.
2. WDTM can perform either the watchdog timer function or the interval timer function.
3. When capture/compare registers 00, 01 (CR00, CR01) are specified as compare registers.

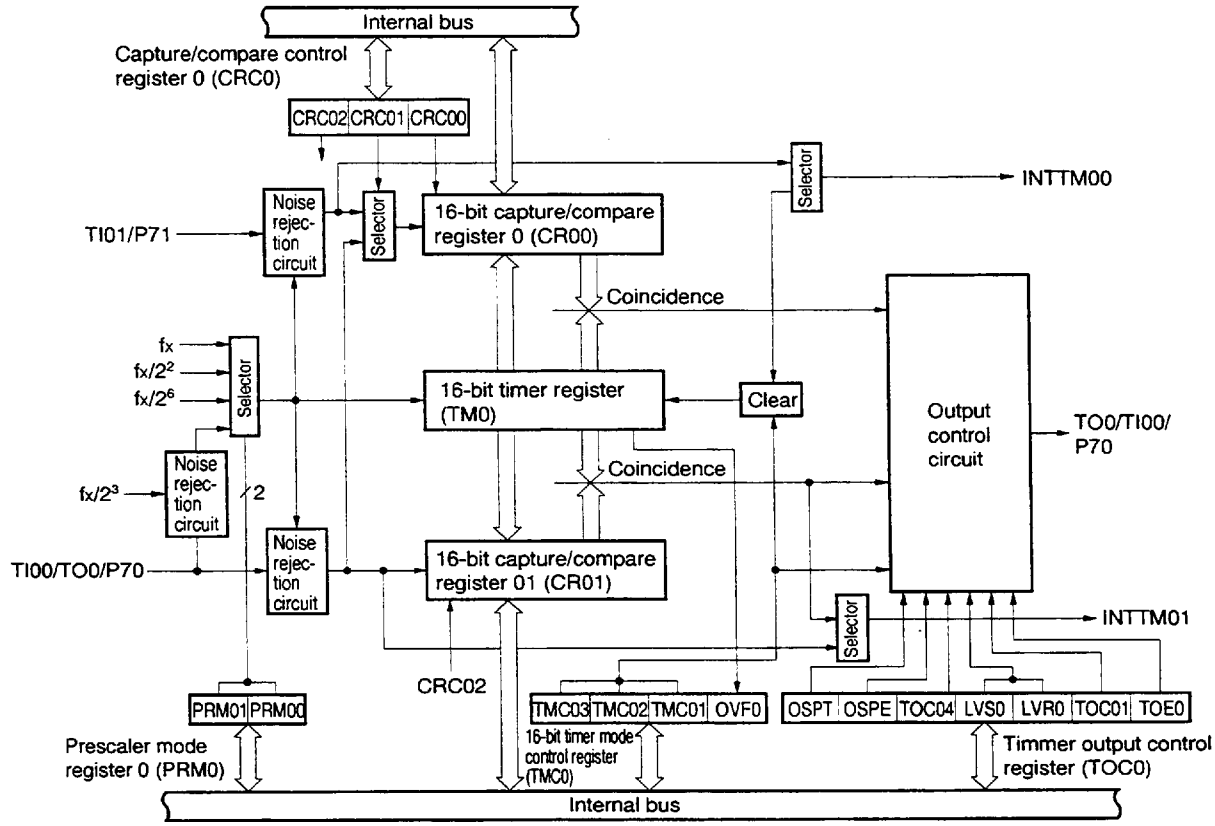
8.1 Functions

The 16-bit timer/event counter (TM0) has the following functions.

- Interval timer
- Pulse width measurement
- External event counter
- Square-wave output
- One-shot pulse output

Figure 8-1 shows 16-Bit Timer/Evant Counter Block Diagram.

Figure 8-1. Timer 0 (TM0) Block Diagram



(1) Interval timer

TM0 generates interrupt request at the preset time interval.

(2) Pulse width measurement

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

(3) External event counter

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

(4) Square-wave output

TM0 can output a square wave with any selected frequency.

(5) One-shot pulse output

TM0 is able to output one-shot pulse which can set any width of output pulse.

8.2 Configuration

Timer 0 consists of the following hardware.

Table 8-2. Timer 0 Configuration

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

(1) 16-bit timer register (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 coincide with each other in the clear & start mode on coincidence between TM0 and CR00

<5> If OSPT is set or if the valid edge of TI00 is input in the one-shot pulse output mode

(2) 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.

When CR00 is used as a compare register, the value set in the CR00 is constantly compared with the 16-bit timer register (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, and as the register which sets the pulse width in the PWM operating mode.

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 the table 8-3.

Table 8-3. TI00/TO0/P70 Pin Valid Edge and CR00 Capture Trigger Valid Edge

ES11	ES10	TI00/TO0/P70 Pin Valid Edge	Capture Trigger Valid Edge	
			CR00	CR01
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

CR00 is set by a 16-bit memory manipulation instruction.

After $\overline{\text{RESET}}$ input, the value of CR00 is undefined.

(3) 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.

When CR01 is used as a compare register, the value set in the CR01 is constantly compared with the 16-bit timer register (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. Setting of the TI00/TO0/P70 valid edge is performed by means of prescaler mode register 0 (PRM0).

CR01 is set with a 16-bit memory manipulation instruction.

After $\overline{\text{RESET}}$ input, the value of CR01 is undefined.

8.3 Timer 0 Control Registers

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

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

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

This register sets the 16-bit timer operating mode, the 16-bit timer register clear mode and output timing, and detects an overflow.

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

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

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

Figure 8-2. 16-Bit Timer Mode Control Register Format

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 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, and 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	Match between TM0 and CR00 or match between TM0 and CR01	
1	0	1		Match between TM0 and CR00, match between TM0 and CR01or TI00 valid edge	
1	1	0	Clear & start on match between TI00 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 CR01or TI00 valid edge	

OVF0	16-bit Timer Register Overflow Detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Switch the clear mode and the TO0 output timing after stopping the timer operation (by setting TMC02 to TMC03 to 0, 0).

2. Set the valid edge of the TI00/TO0/P70 pin with 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.

Remarks 1. TO0 : 16-bit timer/event counter output pin
2. TI00 : 16-bit timer/event counter input pin
3. TM0 : 16-bit timer register
4. CR00 : Compare register 00
5. CR01 : Compare register 01

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

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

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

$\overline{\text{RESET}}$ input sets CRC0 value to 04H.

Figure 8-3. Capture/Compare Control Register 0 (CRC0) Format

Address: FF62H After Reset: 04H 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 TI01n
1	Cpatures on valid edge of TI00n by reverse phase

CRC00	CR00n 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 (TMC0), CR00 should not be specified as a capture register.

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

This register controls the operation of the timer 0 output control circuit. It sets R-S type flip-flop (LV0) setting/resetting, the active level in PWM mode, inversion enabling/disabling in modes other than PWM mode, timer 0 output enabling/disabling, one-shot pulse output operation enabling/disabling, and output trigger for a one-shop pulse by software.

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

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

Figure 8-4. 16-Bit Timer Output Control Register L (TOC0) Format

Address: FF63H After Reset: 00H R/W

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

OSPT	Control of One-Shot Pulse Output Trigger by Software
0	One-shot pulse trigger not used
1	One-shot pulse trigger used

OSPE	One-shot pulse Output Control
0	Continuous pulse output
1	One-shot pulse output

TOC04	Timer output F/F control by match of CR01 and TM0n
0	Inversion operation disabled
1	Inversion operation enabled

LVS0	LVR0	Timer 0 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	Timer ouptut control
0	Ouput 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.

3. OSPT is cleared automatically after data setting, and will therefore be 0 if read.

(4) Prescaler mode register 0 (PRM0)

This register is used to set 16-bit timer (TM0) count clodck and T00 valid edges.

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

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

Figure 8-5. Prescaler Mode Register 0 (PRM0) Format

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	Counter clock Selection
0	0	fx (8.38 MHz)
0	1	$fx/2^2$ (2.09 MHz)
1	0	$fx/2^6$ (131 kHz)
1	1	TI00 valid edge

Caution When TI00 valid edge is set to counter clock, clear/start mode and capture trigger should not be set by TI00 valid edge.

Remarks 1. fx: Main system clock oscillation frequency

2. TI00, TI01: 16-bit timer/event counter input pin

3. Figures in parenthesis indicates fx=8.38 operation

(5) Port mode register 7 (PM7)

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

When using the P70/T00/TI00 pin for timer output, set PM70 and output latch of P70 to 0.

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

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

Figure 8-6. Port Mode Register 7 (PM7) Format

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 input/output mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

8.4 Operations

8.4.1 Interval timer (16-bit) operations

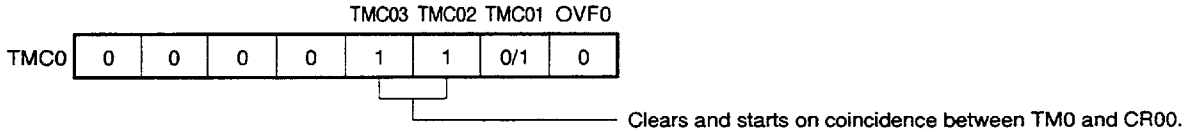
Setting the 16-bit timer mode control register (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 8-7 allows operation as an interval timer. Interrupt request is generated repeatedly using the count value set in 16-bit capture/compare register 00 (CR00) beforehand as the interval.

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

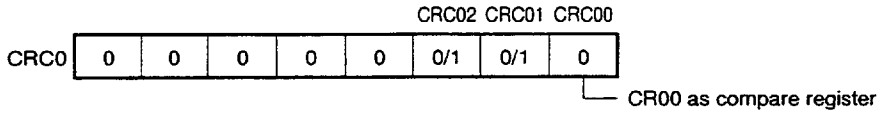
Count clock of the 16-bit timer/event counter can be selected with bits 0 to 1 (PRM00, PRM01) of the prescaler mode register 0 (PRM0).

Figure 8-7. Control Register Settings for Interval Timer Operation

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

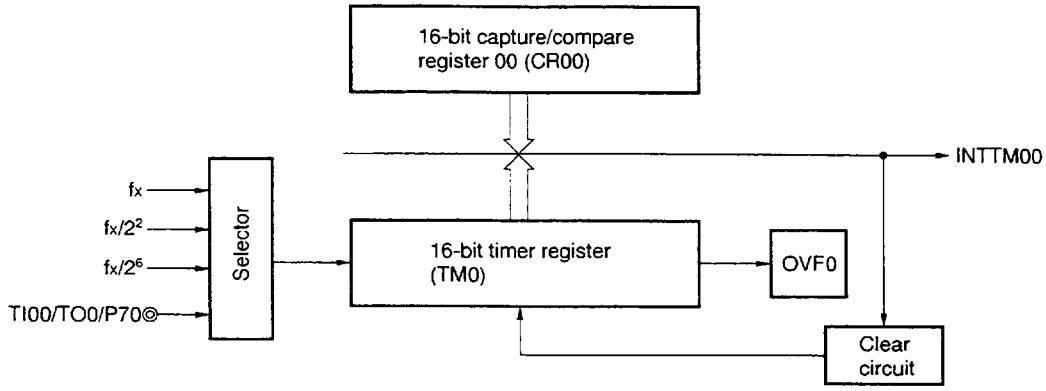


(b) Capture/compare control register 0 (CRC0)



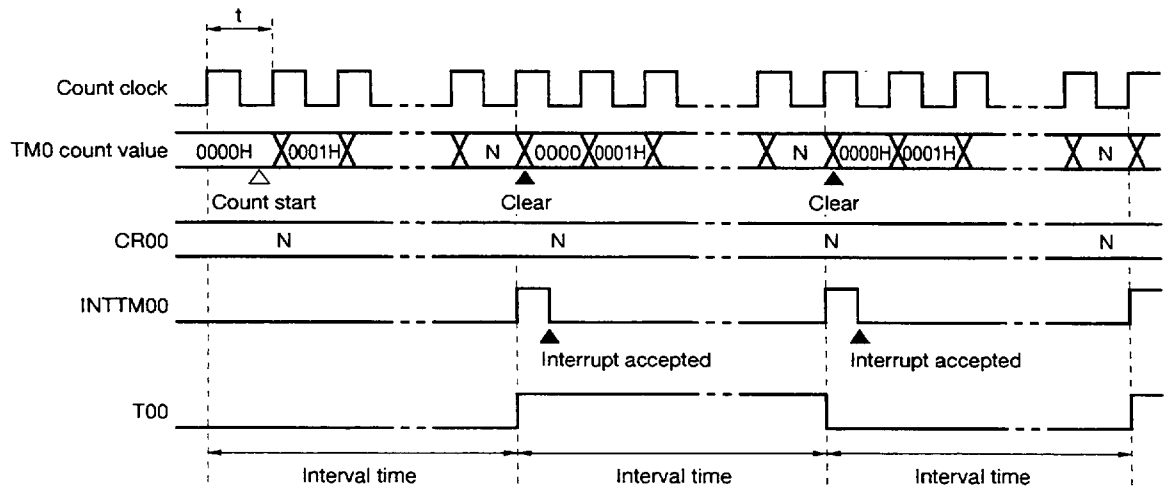
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See the figures 8-2 and 8-3.

Figure 8-8. Interval Timer Configuration Diagram



Remark " ◎—— " indicates the signals that can be directly connected to ports.

Figure 8-9. Timing of Interval Timer Operation



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

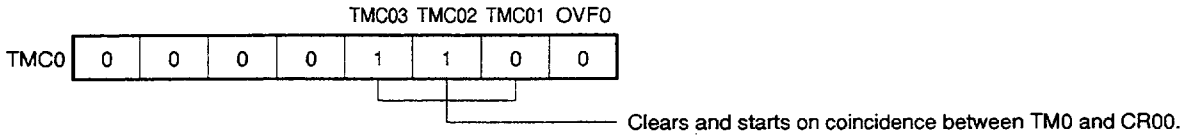
8.4.2 PPG output operations

Setting the 16-bit timer mode control register (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 8-10 allows operation as PPG (Programmable Pulse Generator) output.

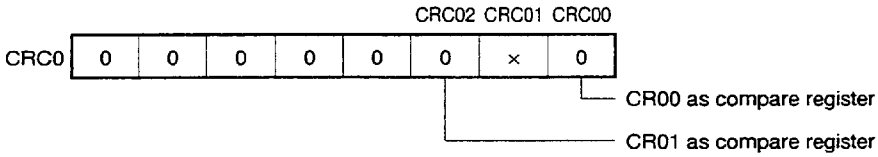
In the PPG output operation, square waves are output from the TO0/P30 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit capture/compare register 01 (CR01) and in 16-bit capture/compare register 00 (CR00), respectively.

Figure 8-10. Control Register Settings for PPG Output Operation

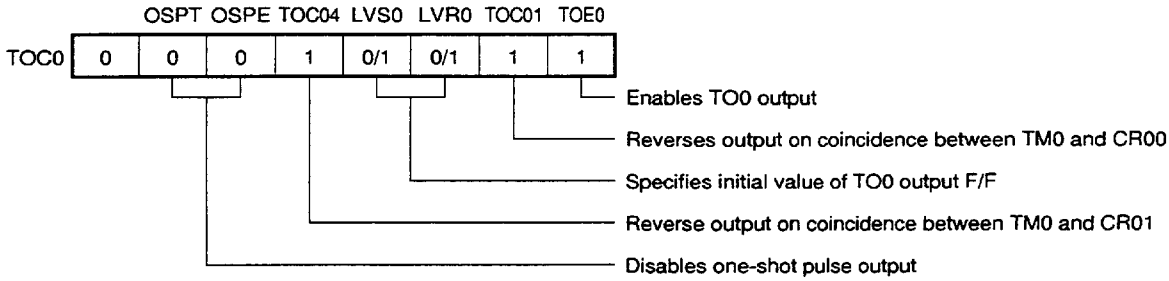
(a) 16-bit timer mode control register (TMC0)



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



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



Remark ×: Don't care

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

8.4.3 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 register (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.

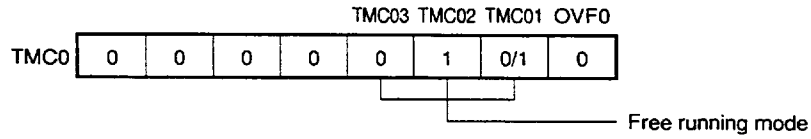
(1) Pulse width measurement with free-running counter and one capture register

When the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-11), 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 capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set. Any of three edge specifications can be selected—rising, falling, or both edges—by means of bits 6 and 7 (ES10 and ES11) of PRM0.

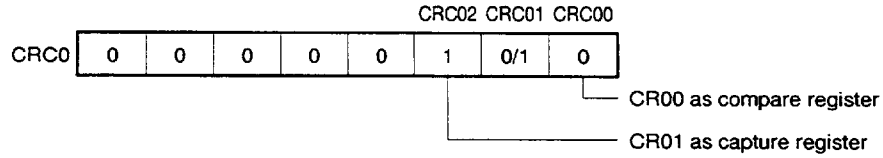
For valid edge detection, sampling is performed at the count clock selected by PRM0), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 8-11. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control register (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 Figures 8-2 and 8-3.

Figure 8-12. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

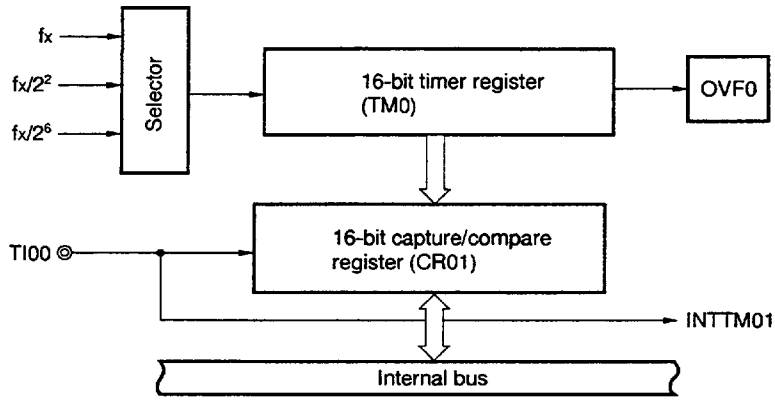
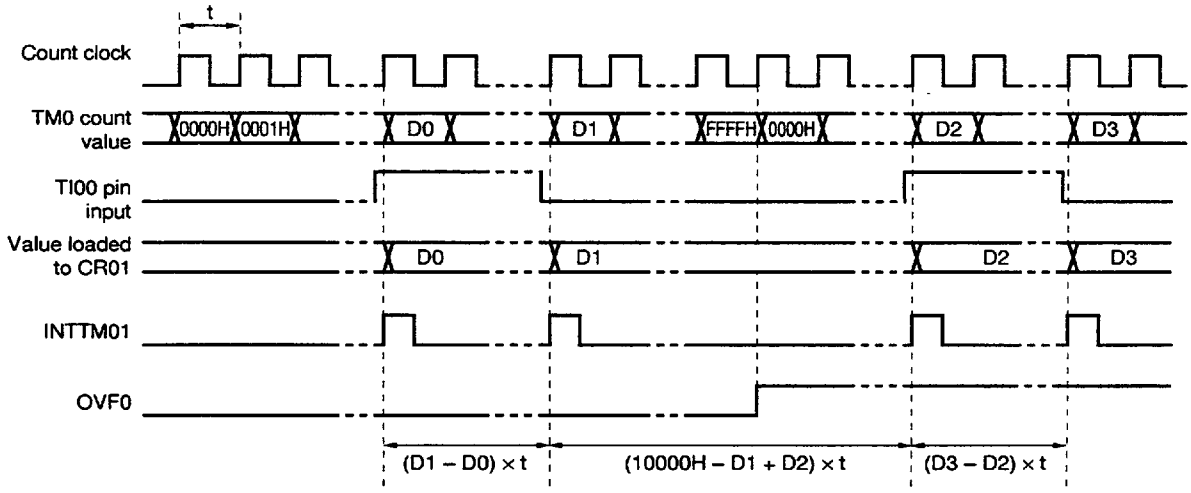


Figure 8-13. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



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

When the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-14), 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 capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

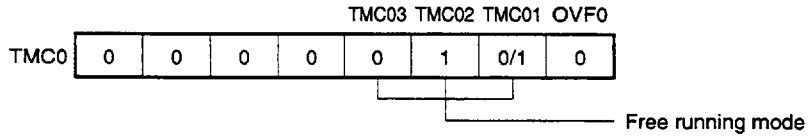
Also, when the edge specified by bits 6 and 7 (ES10 and ES11) of PRM0 is input to the TI01/P71 pin, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00) and an external interrupt request signal (INTTM00) is set.

Any of three edge specifications can be selected—rising, falling, or both edges—as the valid edges for the TI00/TO0/P70 pin and the TI01/P71 pin by means of bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of INTM0, respectively.

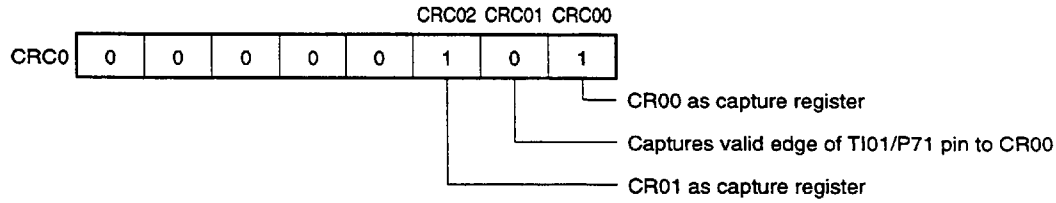
For TI00/TO0/P70 pin valid edge detection, sampling is performed at the interval selected by means of the prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 8-14. Control Register Settings for Measurement of Two Pulse Widths with Free Running Counter

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



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



Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the pulse width measurement function. For details, refer to figures 8-2 and 8-3.

- **Capture operation (Free-Running mode)**
  Capture register operation in capture trigger input is shown.

Figure 8-15. Capture Operation with Rising edge specified

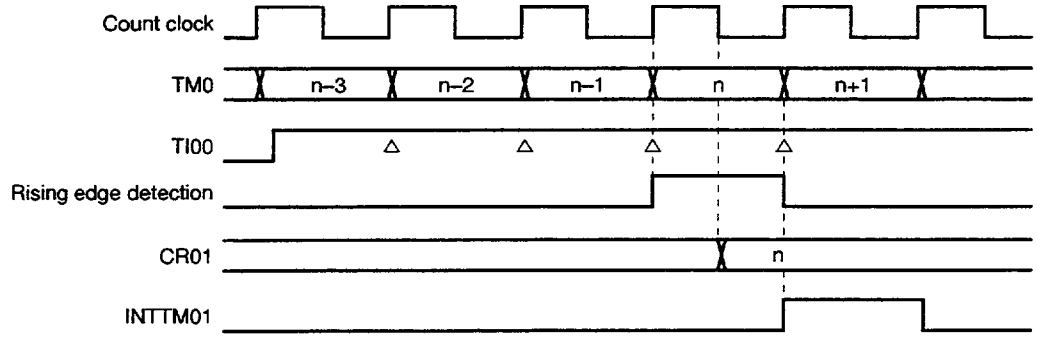
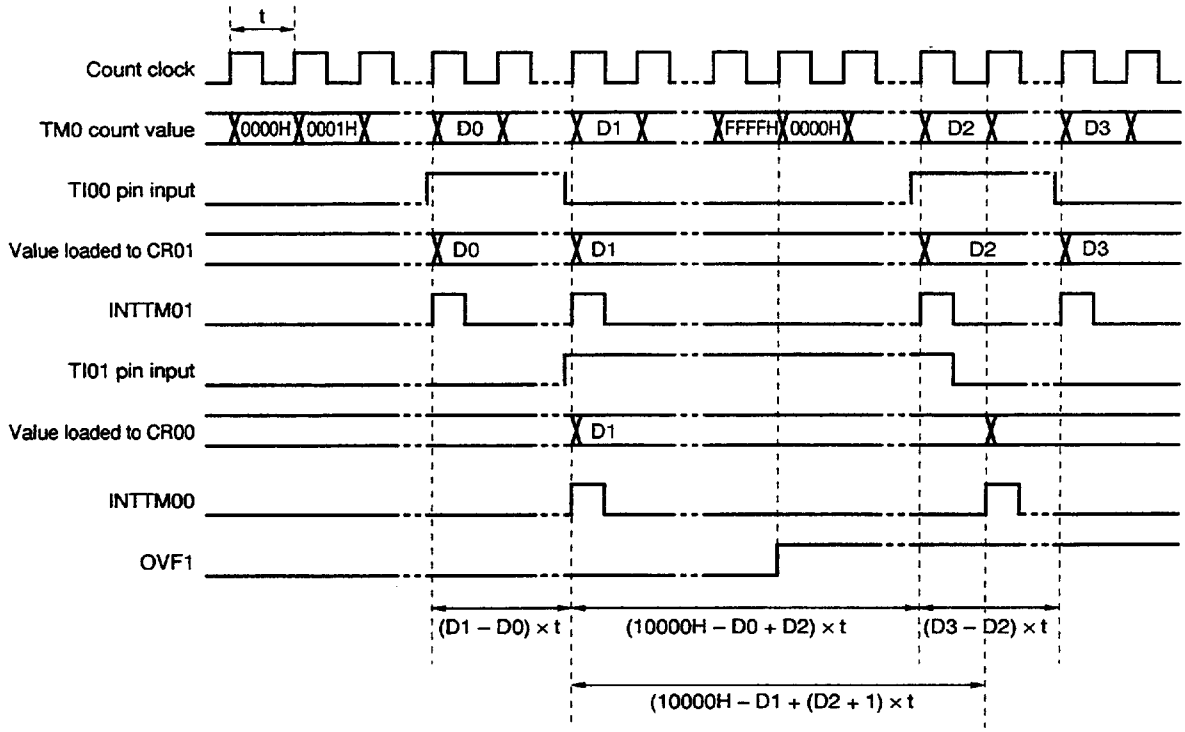


Figure 8-16. Timing of Pulse width Measurement operation with Free-Running Counter (with Both Edge specified)



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

When the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-17), 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 capture/compare register 01 (CR01) and an external 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 capture/compare register 00 (CR00).

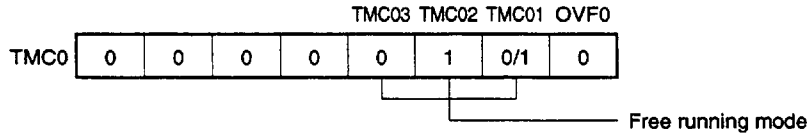
Either of two edge specifications can be selected—rising or falling—as the valid edges for the TI00/TO0/P70 pin by means of bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

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

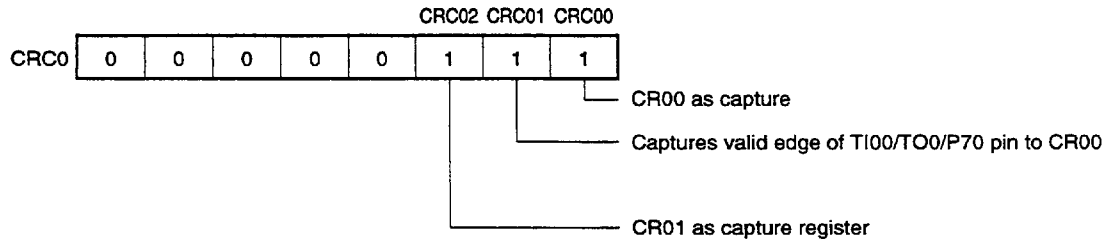
Caution If the valid edge of TI00//TO0/P70 is specified to be both rising and falling edge, capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 8-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

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

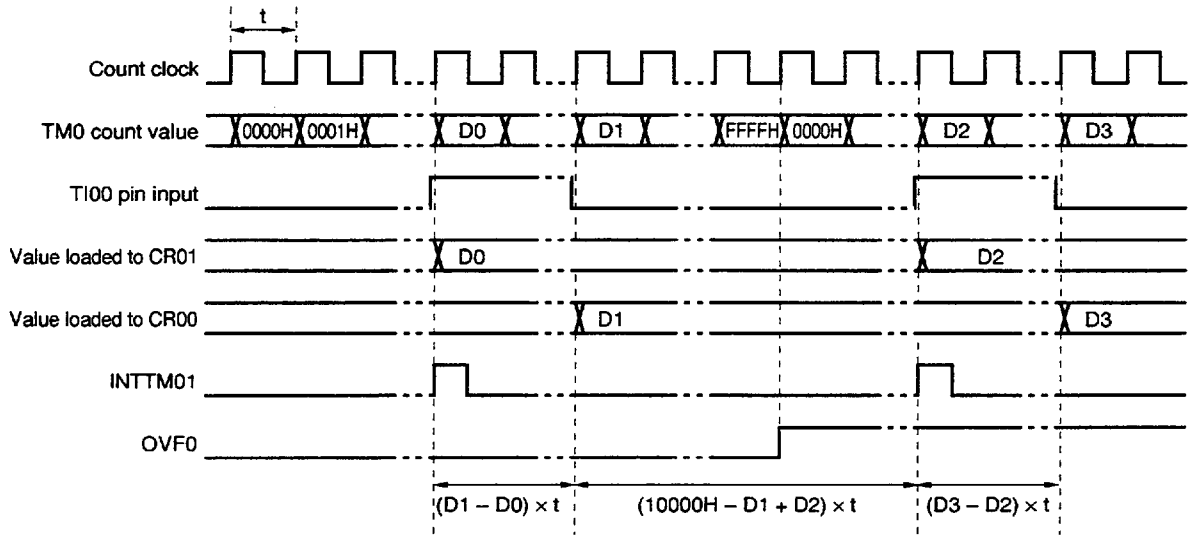


(b) Capture/compare control register 0 (CRC0)



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 8-18. 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 register (TM0) is taken into 16-bit 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 8-19).

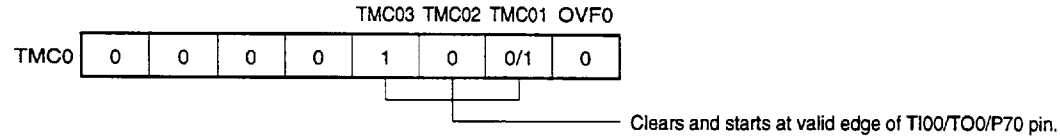
The edge specification can be selected from two types, rising and falling edges by bits 4 and 5 (ES00 and ES01) of the prescaler mode resister 0 (PRM0)

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

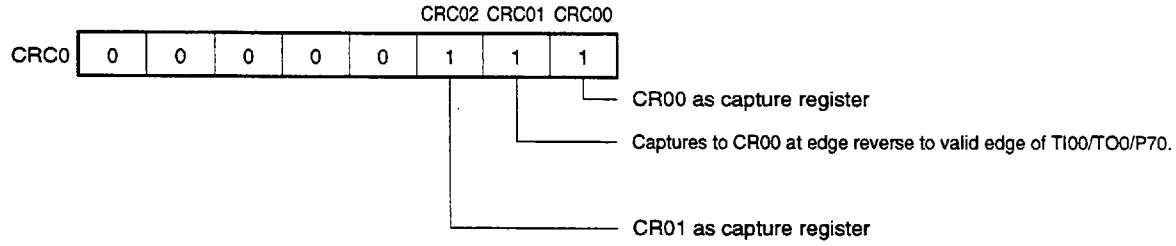
Caution If the valid edge of TI00/TO0/P70 is specified to be both rising and falling edge, the 16-bit capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 8-19. Control Register Settings for Pulse Width Measurement by Means of Restart

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

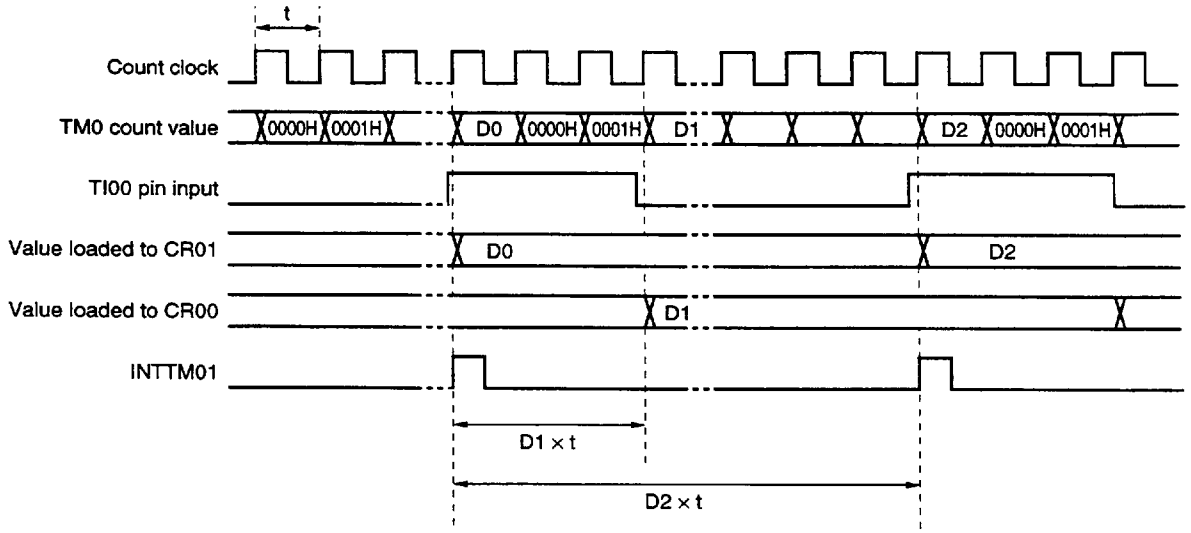


(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See figure 8-2 and 8-3.

Figure 8-20. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



8.4.4 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 register (TM0).

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

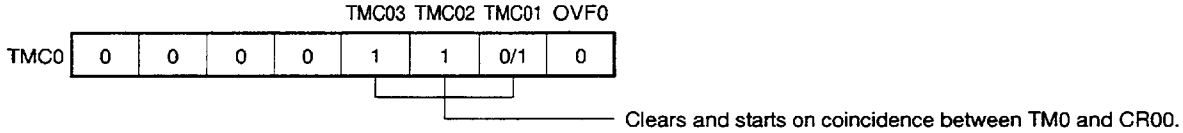
When the TM0 counted value matches the 16-bit capture/compare register 00 (CR00) value, TM0 is cleared to 0 and the interrupt request signal (INTTM00) is generated.

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).

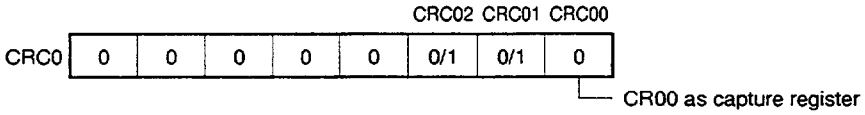
Because operation is carried out only after the valid edge is detected twice by sampling at the interval selected with the prescaler mode register 0 (PRM0), noise with short pulse widths can be removed.

Figure 8-21. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control register (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 the figures 8-2 and 8-3.

Figure 8-22. External Event Counter Configuration Diagram

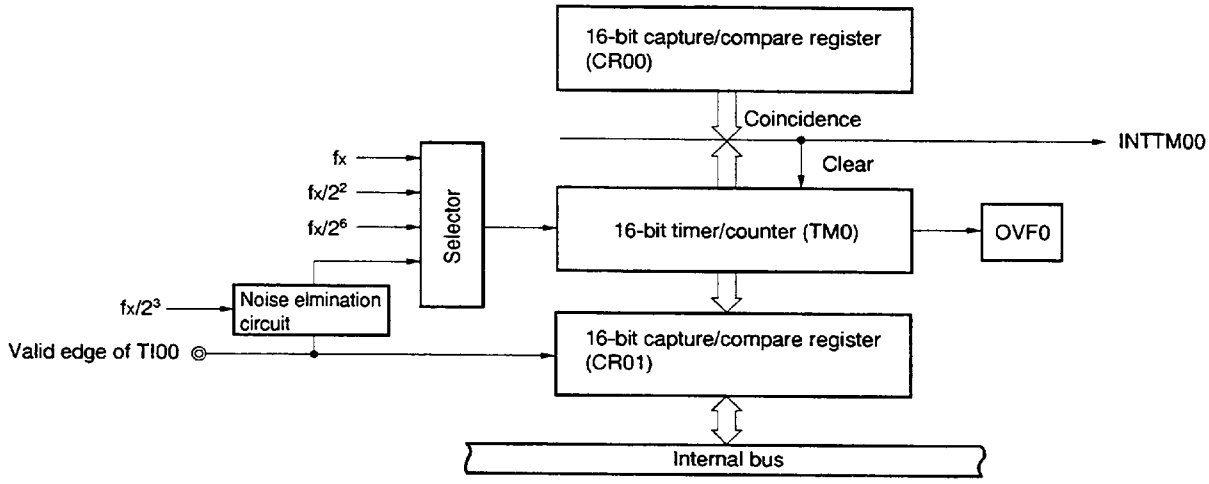
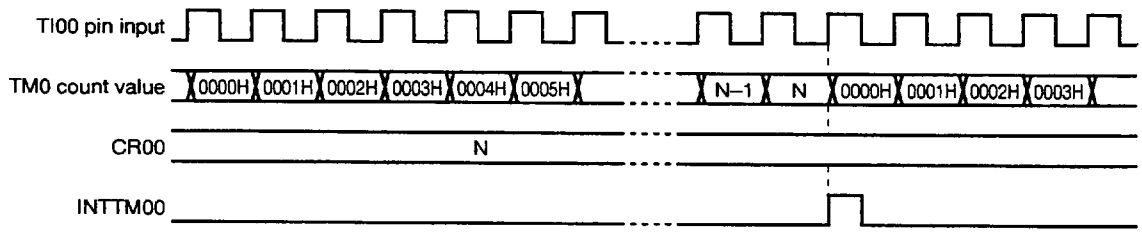


Figure 8-23. External Event Counter Operation Timings (with Rising Edge Specified)



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

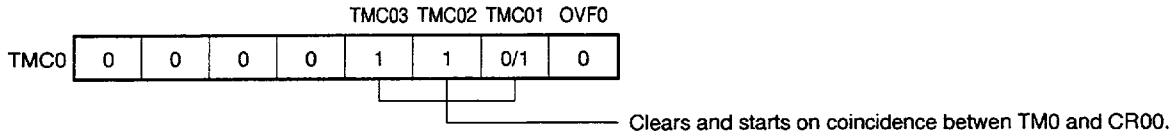
8.4.5 Square-wave output operation

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

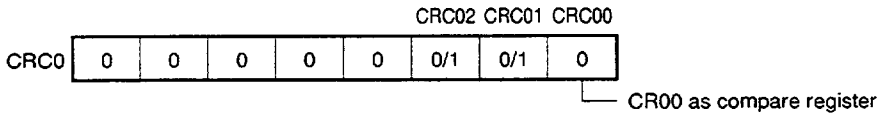
The TO0 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of the 16-bit timer output control register to 1. This enables a square wave with any selected frequency to be output.

Figure 8-24. Control Register Settings in Square-Wave Output Mode

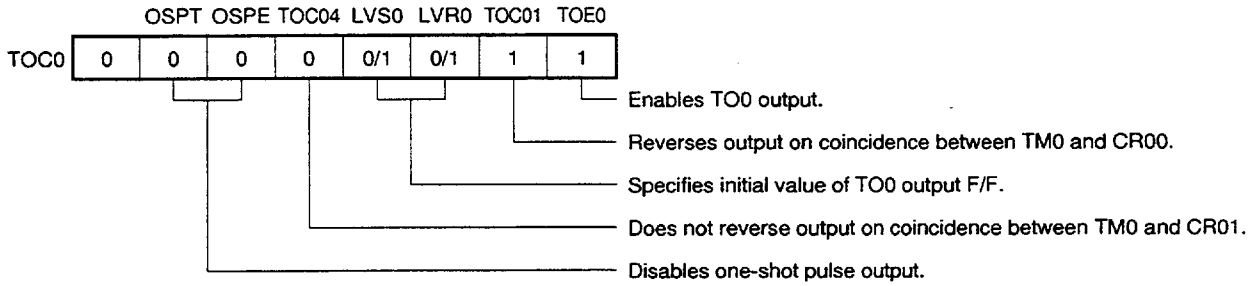
(a) 16-bit timer mode control register (TMC0)



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

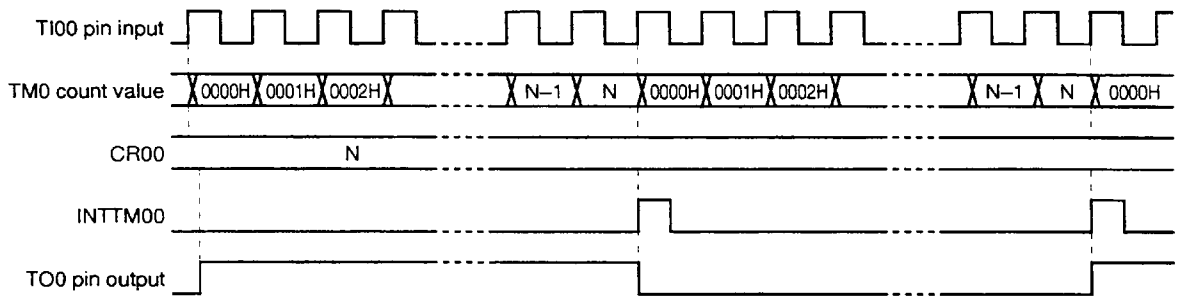


(c) 16-bit timer output control register (TOC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See the figures 8-2, 8-3 and 8-4.

Figure 8-25. Square-Wave Output Operation Timing



8.4.6 One-shot pulse output operation

It is possible to output one-shot pulses synchronized with a software trigger or an external trigger (TI00/TO0/P70 pin input).

(1) One-shot pulse output using software trigger

If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Figure 8-26, and 1 is set in bit 6 (OSPT) of TOC0 by software, a one-shot pulse is output from the TO0/TI00/P70 pin.

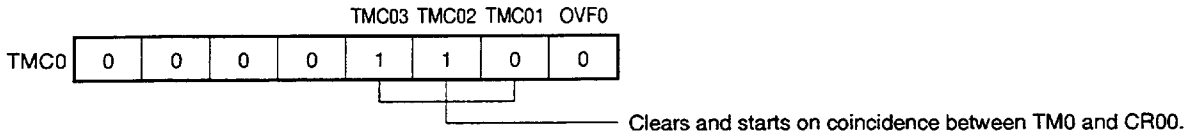
By setting 1 in OSPT, the 16-bit timer/event counter is cleared and started, and output is activated by the count value set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

TM0 continues to operate after one-shot pulse is output. To stop TM0, 00H must be set to TMC0.

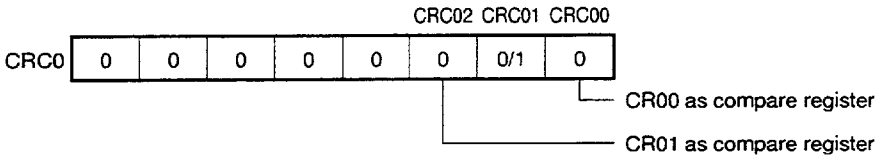
Caution When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, execute after the INTTM00, or interrupt match signal with CR00, is generated.

Figure 8-26. Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger

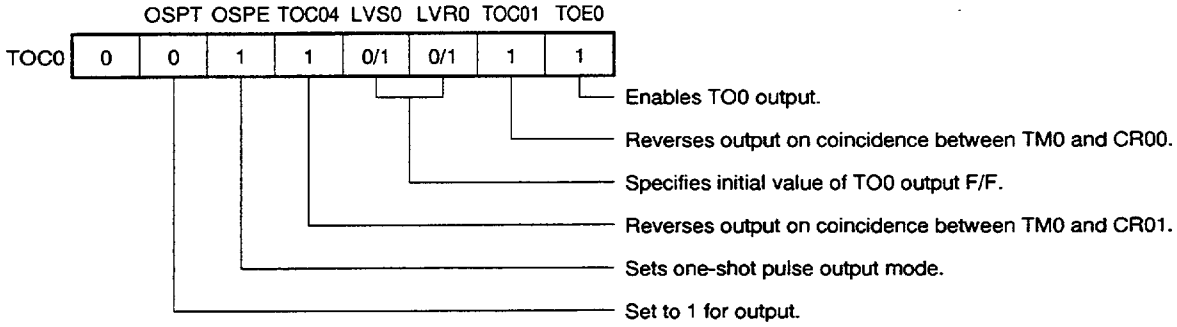
(a) 16-bit timer mode control register (TMC0)



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



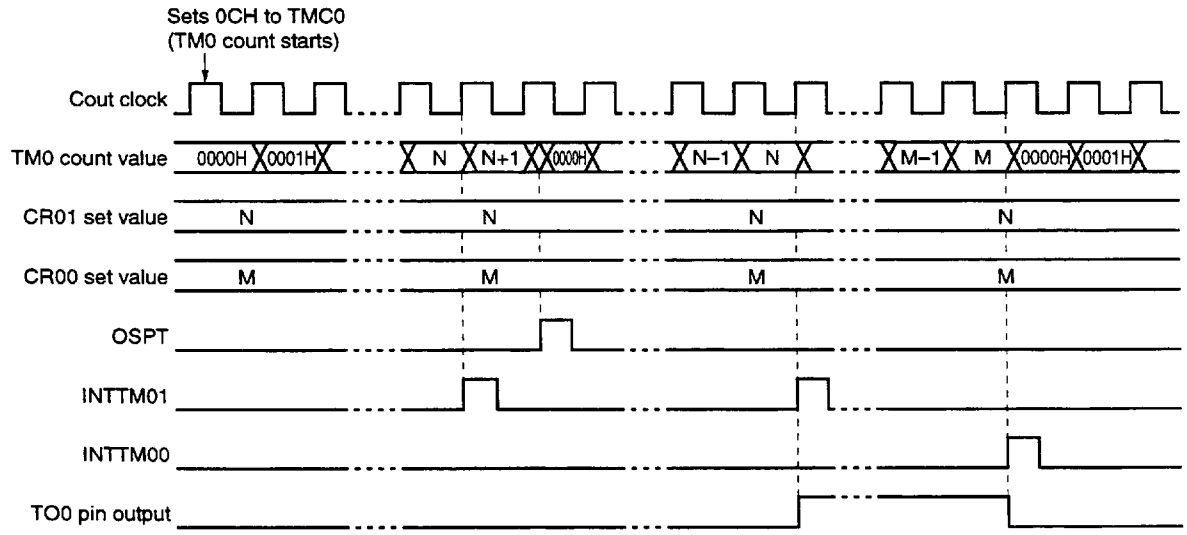
(c) 16-bit timer output control register (TOC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with one-shot pulse output. See the figures 8-2, 8-3 and 8-4.

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

Figure 8-27. Timing of One-Shot Pulse Output Operation Using Software Trigger



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

(2) One-shot pulse output using external trigger

If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Figure 8-28, a one-shot pulse is output from the TO0/TI00/P70 pin with a TI00/TO0/P70 valid edge as an external trigger.

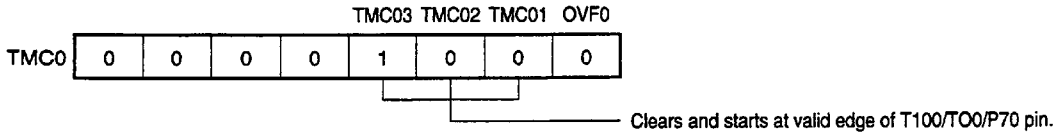
Any of three edge specifications can be selected—rising, falling, or both edges — as the valid edges for the TI00/TO0/P70 pin by means of bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

When a valid edge is input to the TI00/TO0/P70 pin, the 16-bit timer/event counter is cleared and started, and output is activated by the count values set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

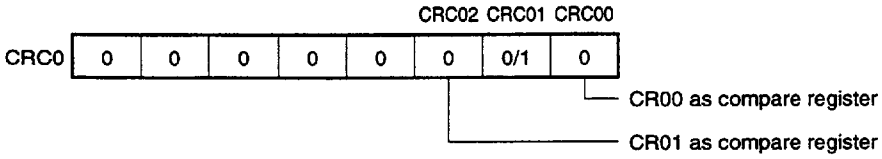
Caution When outputting one-shot pulses, external trigger is ignored if generated again.

Figure 8-28. Control Register Settings for One-Shot Pulse Output Operation Using External Trigger

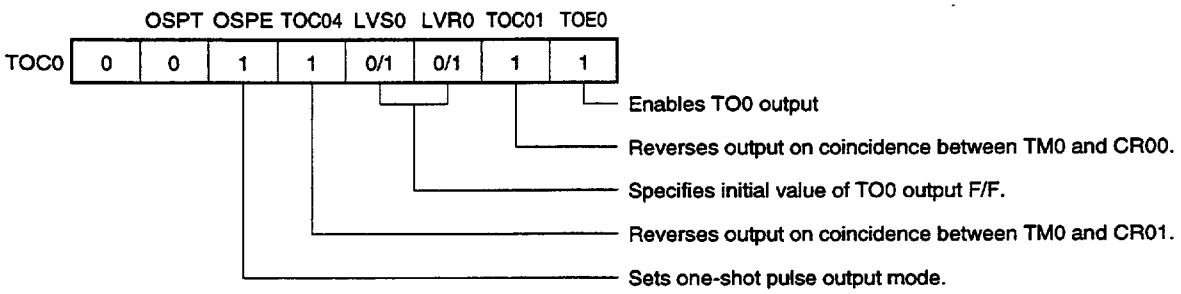
(a) 16-bit timer mode control register (TMC0)



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



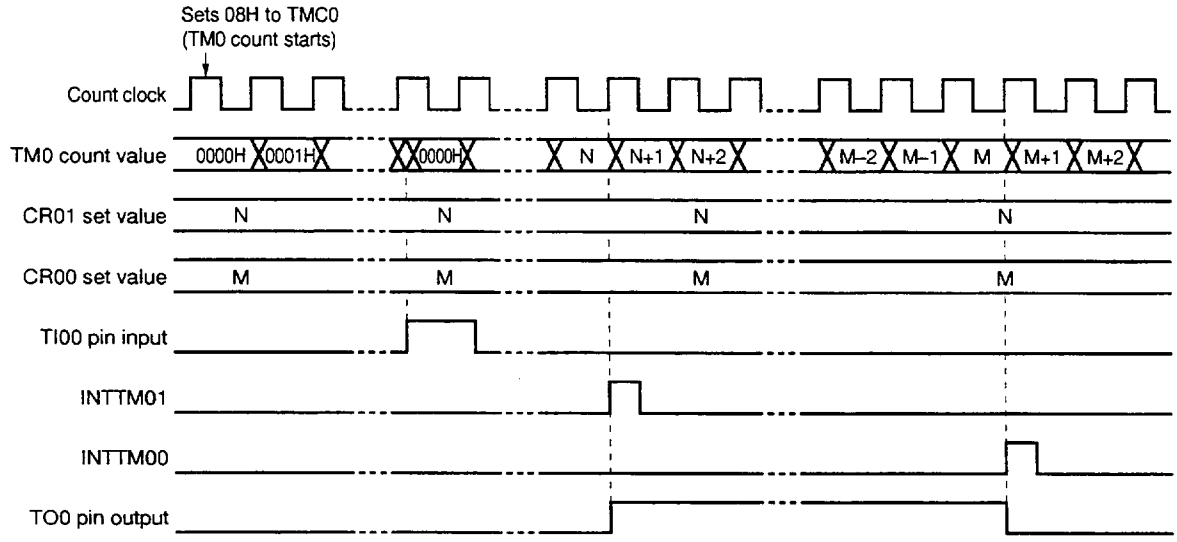
(c) 16-bit timer output control register (TOC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with one-shot pulse output. See the figures 8-2, 8-3 and 8-4 for details.

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

Figure 8-29. Timing of One-Shot Pulse Output Operation Using External Trigger (With Rising Edge Specified)



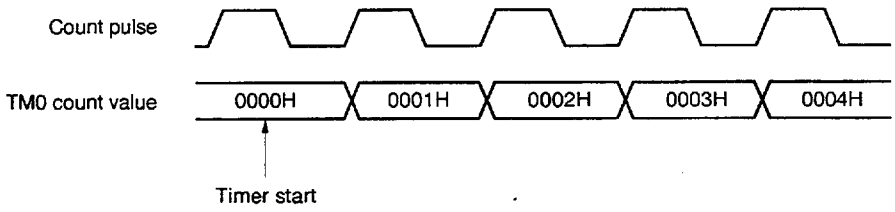
Caution The 16-bit timer register starts operation at the moment a value other than 0, 0 (operation stop mode) is set to TMC02 to TMC03, respectively.

8.5 Operating Precautions

(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 register (TM0) is started asynchronously with the count pulse.

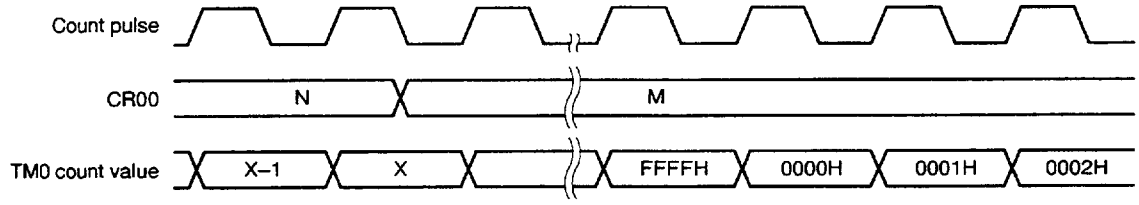
Figure 8-30. 16-Bit Timer Register Start Timing



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

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

Figure 8-31. Timings After Change of Compare Register During Timer Count Operation

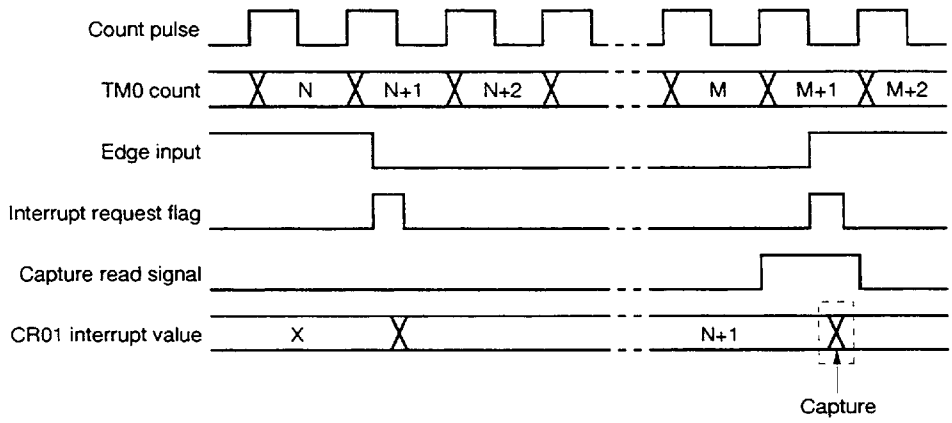


Remark N > X > M

(3) Capture register data retention timings

If the valid edge of the TI00/TO0/P70 pin is input during 16-bit capture/compare register 01 (CR01) read, CR01 holds data without carrying out capture operation. However, the interrupt request flag (INTTM01) is set upon detection of the valid edge.

Figure 8-32. Capture Register Data Retention Timing



(4) Valid edge setting

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

(5) Re-trigger of one-shot pulse

(a) One-shot pulse output using software

When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, execute it after the INTTM00, or interrupt request match signal with CR00, is generated.

(b) One-shot pulse output using external trigger

When outputting one-shot pulses, external trigger is ignored if generated again.

(6) Operation of OVF0 flag

OFV0 flag is set to 1 in the following case.

The clear & start mode on match between TM0 and CR00 is selected.

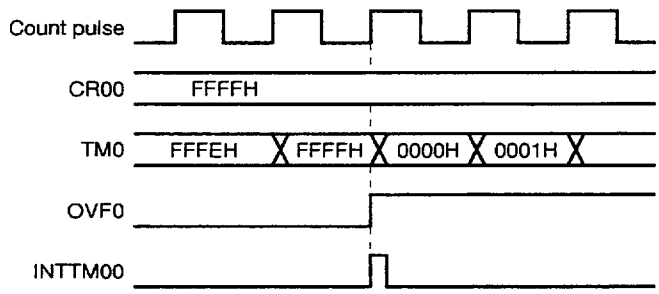
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 8-33. Operation Timing of OVF0 Flag



CHAPTER 9 8-BIT TIMER/EVENT COUNTERS TM5

9.1 Functions

Timer 5 has the following two modes.

- Mode using Timer 5 alone (individual mode)
- Mode using the cascade connection (16-bit resolution: cascade connection mode)

These two modes are described next.

(1) Mode using Timer 5 alone (individual mode)

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

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

The figure 9-1 shows TM50 block diagram, and the figure 9-2 shows TM51 block diagram.

Figure 9-1. Block Diagram of TM50

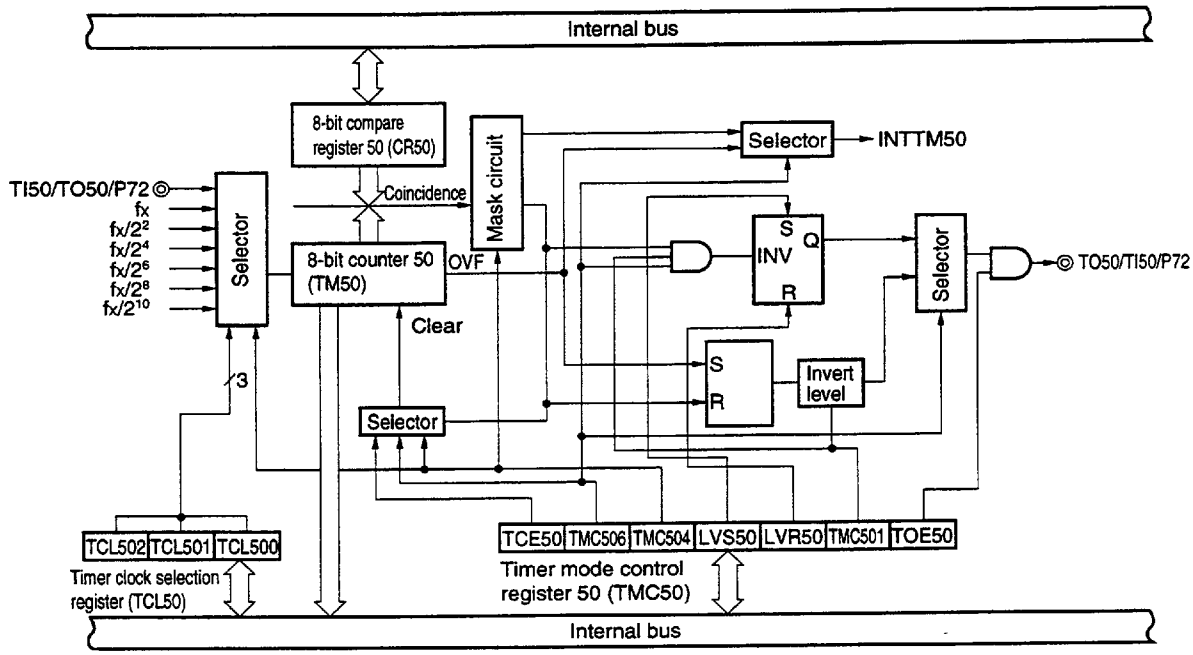
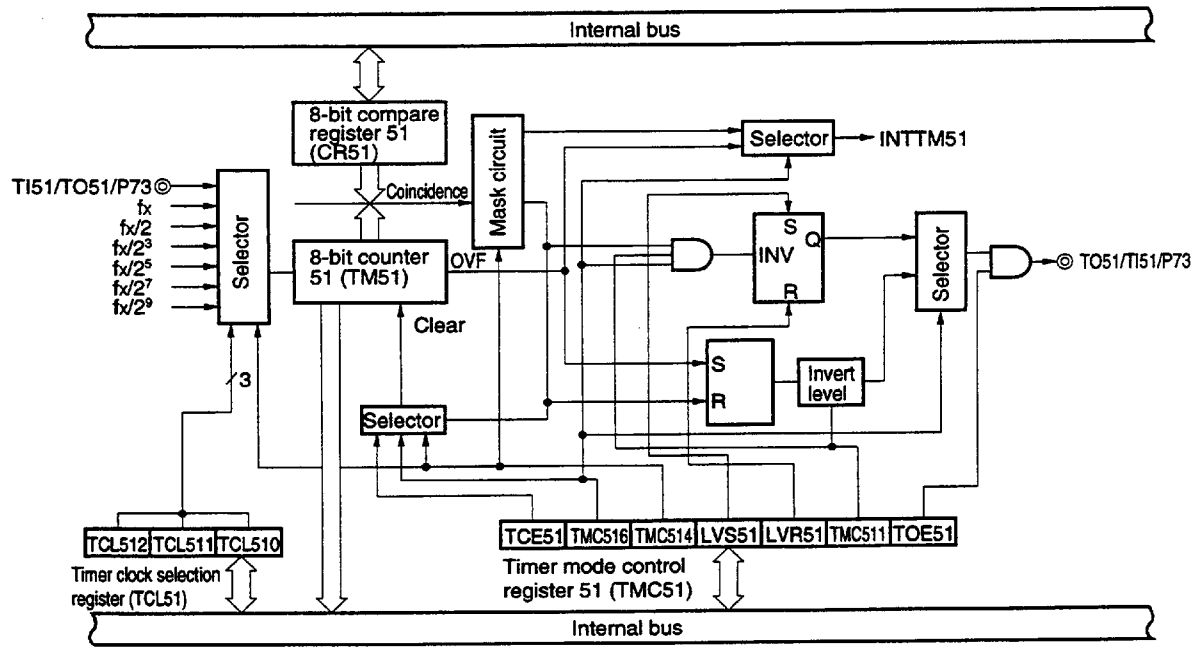


Figure 9-2. Block Diagram of TM51



9.2 Configurations

Timer 5 consists of the following hardware.

Table 9-1. Timer 5 Configurations

Item	Configuration
Timer register	8 bits counter 5n (TM5n)
Register	8 bits compare register: 5n (CR5n)
Timer output	2 (TO5n)
Control register	Timer clock select register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n)

Remark n = 0, 1

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

TM5n is a 8-bit register which counts the count pulses.

When count clock starts, a counter is incremented. When count value is read during operation, count clock input is temporary stopped, and them count valve is read.

<1> $\overline{\text{RESET}}$ input In the following situations, count valve is set to 00H.

<2> Clear TCE5n

<3> Match between TM5n and CR5n in clear and start made with match between TM5n and CR5n.

Caution In cascade connection mode, when TCE5n, the lowest timer, is cleared, it sets to 00H.

Remark n = 0, 1

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

When CR5n is used as a compare resistor, the valve set in the CR5n is constantly compared with the 8-bit counter (TM5n) count valve, and an interrupt request (INTTM5n) is generated if they match. (Except PWM mode).

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

Caution When setting data in cascade connection mode, stop the timer operation beforehand.

Remark n = 0, 1

9.3 Timer 5 Registers

The following three types of registers are used to control the timer 5.

- 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 timer 5.

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

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

Figure 9-3 Timer Clock Select Register 50 (TCL50) Format

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	fx (8.38 MHz)
0	1	1	$fx/2^2$ (2.09 MHz)
1	0	0	$fx/2^4$ (523 kHz)
1	0	1	$fx/2^6$ (131 kHz)
1	1	0	$fx/2^8$ (32.7 kHz)
1	1	1	$fx/2^{10}$ (8.18 kHz)

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

2. Set bits 3-7 to 0.

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with Fx = 8.38 MHz

Figure 9-4. Timer Clock Select Register 51 (TCL51) Format

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	fx/2 (4.19 MHz)
0	1	1	$fx/2^3$ (1.04 MHz)
1	0	0	$fx/2^5$ (261 kHz)
1	0	1	$fx/2^7$ (65.4 kHz)
1	1	0	$fx/2^9$ (16.3 kHz)
1	1	1	$fx/2^{11}$ (4.09 kHz)

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.
2. Set bit 3-7 to 0.

Remarks 1. fx: Main system clock oscillation frequency
2. Figures in parentheses apply to operation with fx = 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 counter 5n (TM5n) count operation control
<2> 8-bit 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 1-bit or 8-bit memory operating command.
$\overline{\text{RESET}}$ input sets to 00H.
The figure 9-5 shows TMC5n format.

Figure 9-5 8-bit Timer Mode Control Register 5n (TMC5n) Format

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	TM5 Count Operation Control
0	After cleaning to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMC5n6	TM5 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 operating enabled	Active low

TOE5n	Timer Output Control
0	Output disabled (Port mode)
1	Output enabled

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, they will be 0.

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 with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 9-6. Port Mode Register 7 (PM7) Format

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 input/output mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

9.4 Operations

9.4.1 8-bit interval timer operation

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

When the count values of the 8-bit 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.

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

[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.

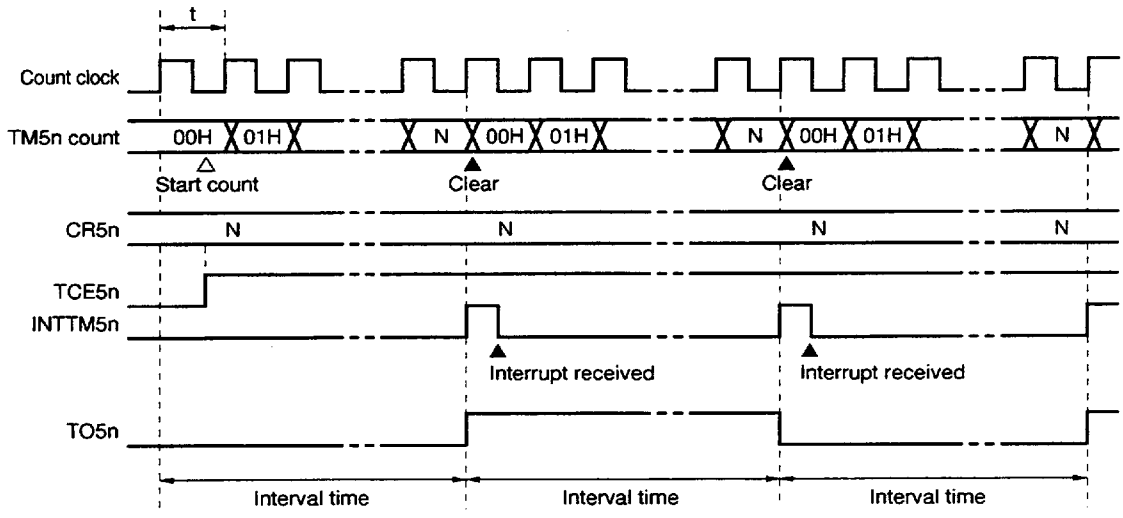
<3> If the values of TM5n and CR5n match, the timer output flip-flop inverts. Also, INTTM5n is generated and TM5n is cleared to 00H.

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

Remark n = 0, 1

Figure 9-7. Interval Timer Operation Timings (1/3)

(a) Basic operation

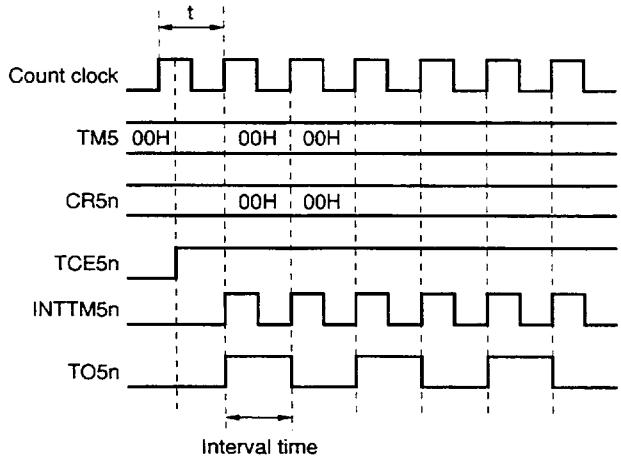


Remarks 1. Interval time = (n+1) × t: N = 00H to FFH

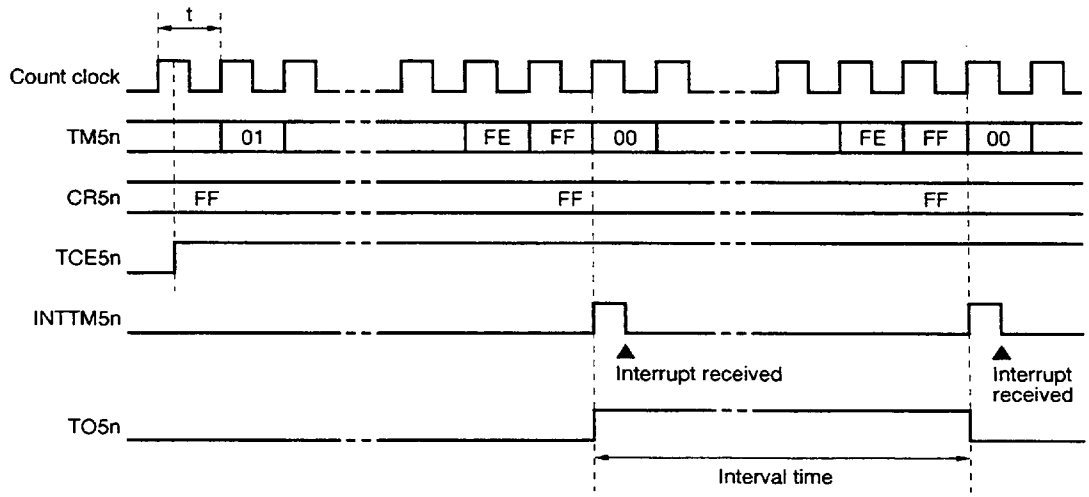
2. n = 0, 1

Figure 9-7. Interval Timer Operation Timings (2/3)

(b) When CR5n = 00H



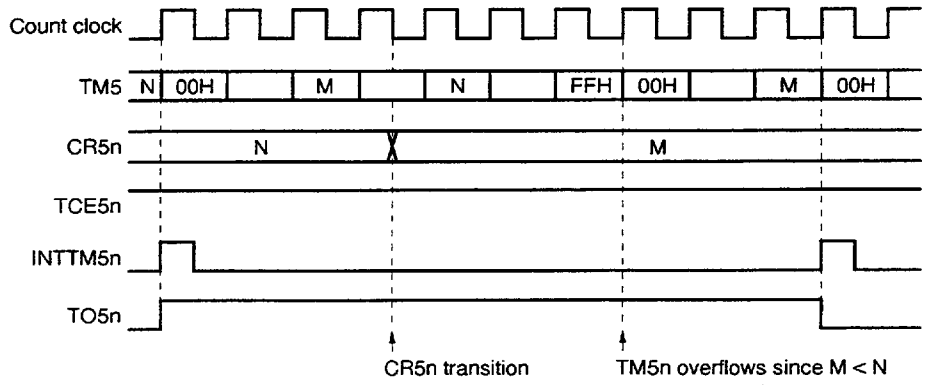
(c) When CR5n = FFH



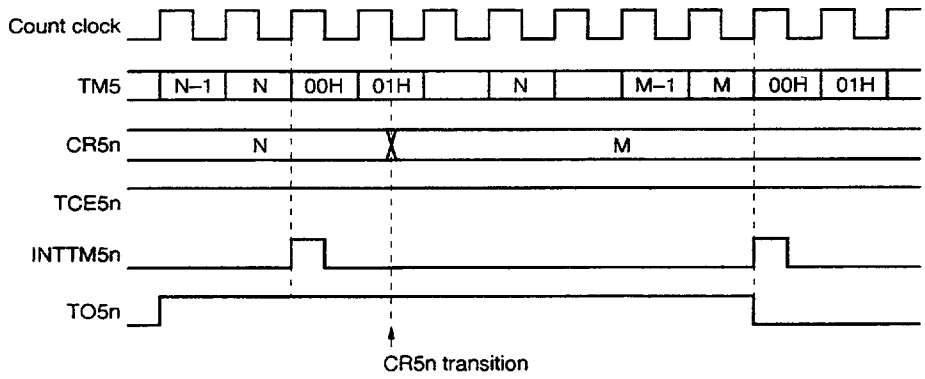
n = 0, 1

Figure 9-7. Interval Timer Operation Timings (3/3)

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



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



n = 0, 1

9.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to theT15n.

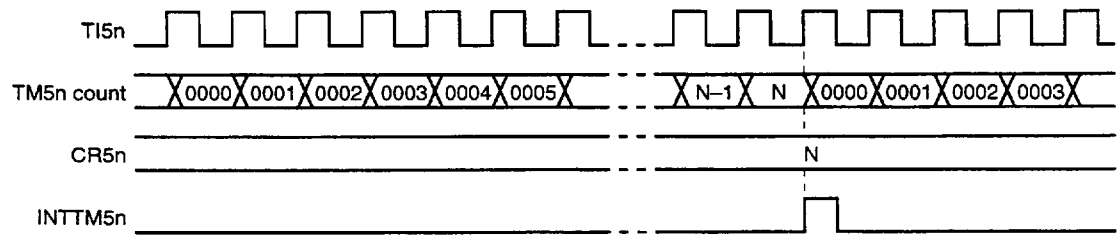
TM5n is incremented each time the valid edge specified with the timer clock select register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

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

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

Remark n = 0, 1

Figure 9-8. External Event Counter Operation Timings (with Rising Edge Specified)



n = 0, 1

9.4.3. Square-wave output operation (8-bit solution)

A square wave with any selected frequency is output at intervals of the value preset to 8-bit compare registers 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 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 permit
Timer output permit → 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

9.4.4 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 compare register 5n (CR5n).

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 bit 0 to bit 2 (TCL5n0 to TCL5n2) of timer clock select register n (TCL5n).

Permit/Prohibit 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, 73) and port mode register 7 (PM72, PM73) to 0.

<2> Set active level width with 8-bit 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 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 counter 5n.

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

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

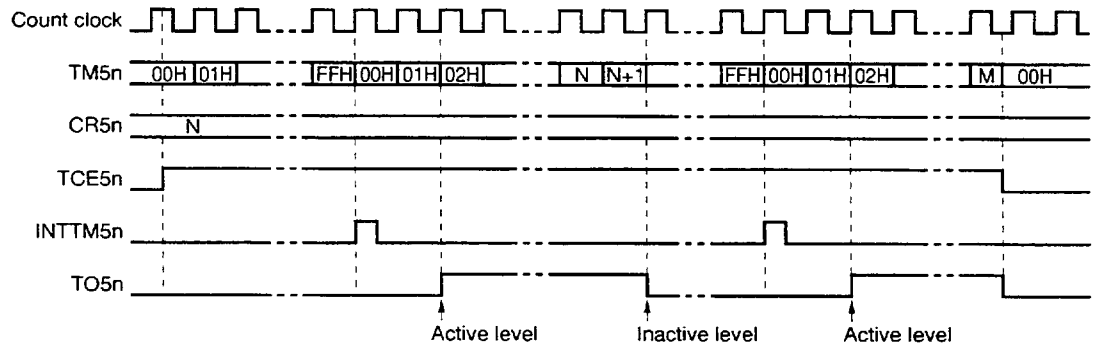
<5> When the count operation is stopped with TCE5n = 0, PWM output comes to inactive level.

Remark n = 0, 1

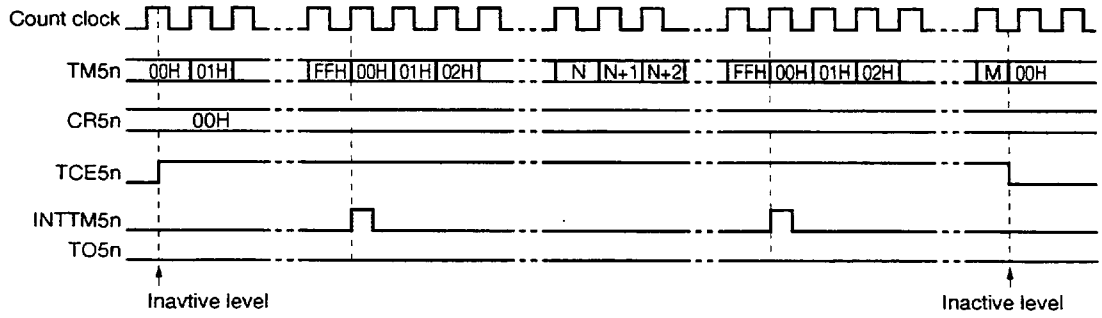
(a) PWM Output Basic Operation

Figure 9-9. PWM Output Operation Timing

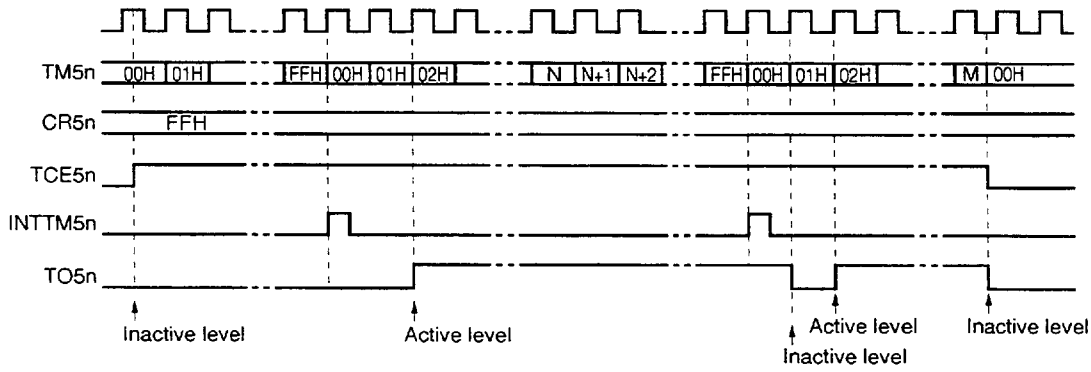
(i) Basic Operation (Active level = H)



(ii) CR5n = 0



(iii) CR5n = FFH

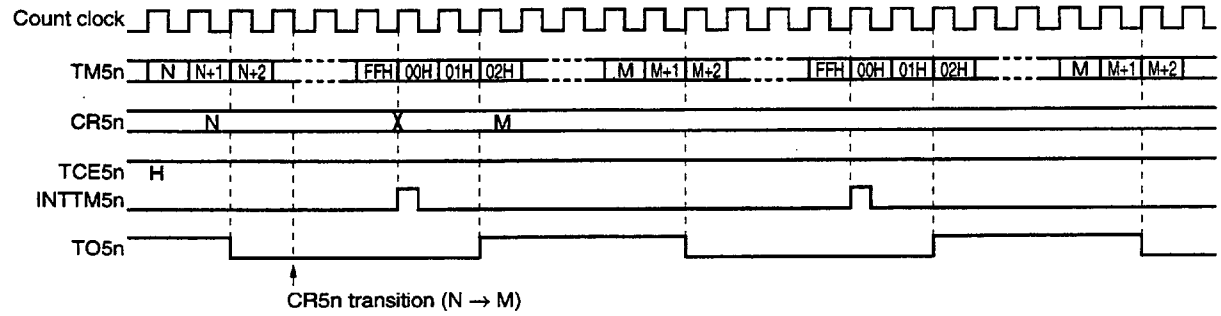


n = 0, 1

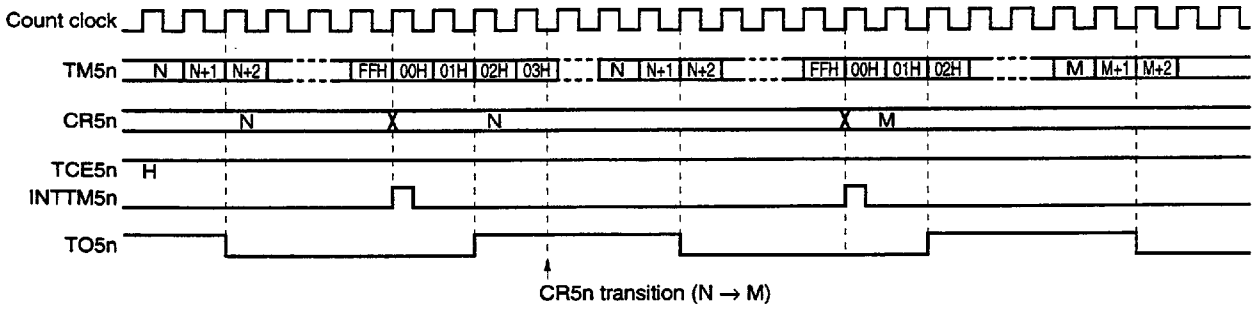
(b) Operation by change of CR5n

Figure 9-10. Timing of Operation by Change of CR5n

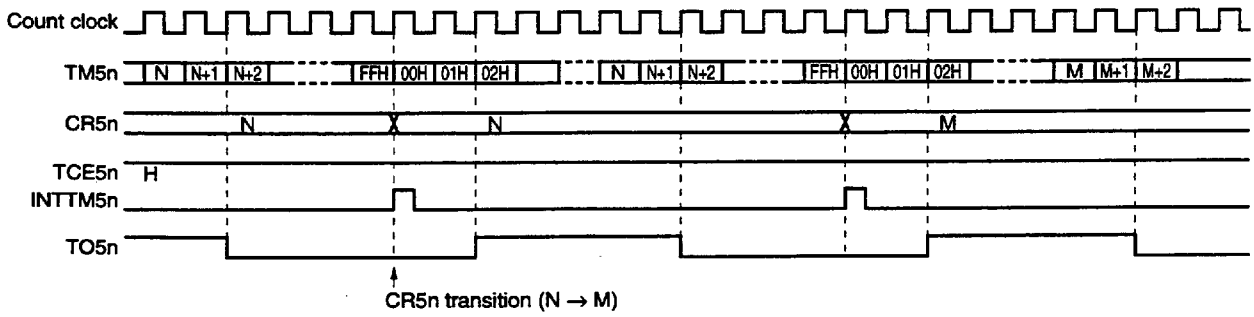
(i) Change of CR5n value to N to M before overflow of TM5n



(ii) Change of CR5n value to N to M after overflow of TM5n



(iii) Change of CR5n value to N to M between two clocks (00H and 01H) after overflow of TM5n



n = 0, 1

(2) Cascade Connection (16-bit timer) Mode

- **Interval timer (16-bit resolution)operation**
  When bit 4 (TMC5n4) of 8-bit timer mode control register 5n (TMC5n) is set to 1, 16-bit resolution timer/counter mode starts.
  The 8-bit timer/event counters operates as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit compare register 5n (CR5n).

[Setting]

<1> Set each register

TCL5n : The lowest timer is count clock selection. The upper timer connected to cascade connection does not need setting.

CR5n : Compare value (Each compare valve can be set to 00H-FFH)

TMC5n : Clear and start mode is selected when TM5n matches CR5n

(The lowest timer → TMC5n = 0000×××0B ×: don't care)
(Others → TMC5n = 0001×××0B ×: don't care)

<2> When TCE5n = 1 is set from the upper timer and TCE5n = 1 is set from the lowest timer, the count operation starts.

<3> When the value of TM5n of all timer (connected in cascade) matches the value of CR5n, INTTM5n of the lowest timer generates (all TM5ns are cleared to 00H).

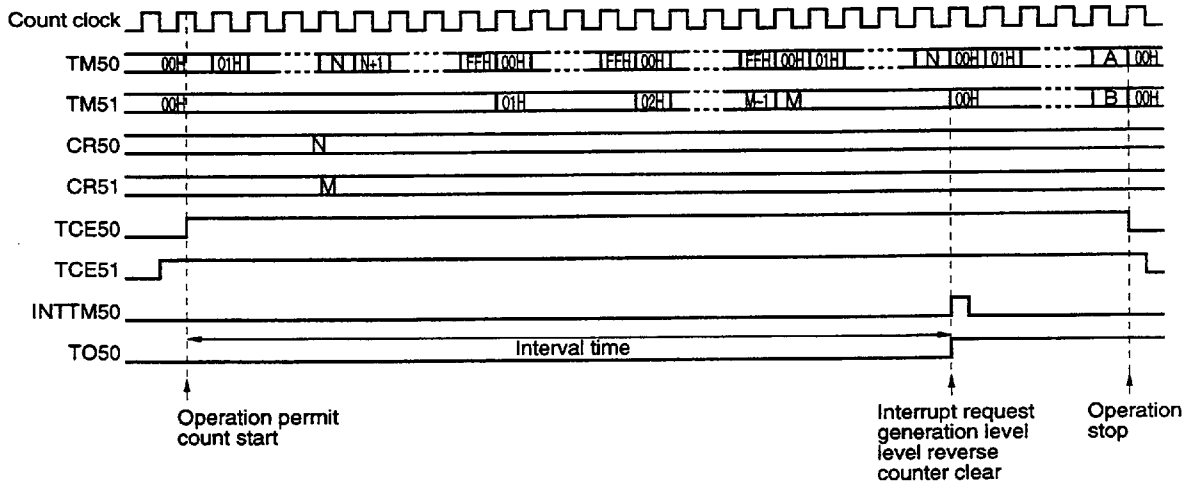
<4> INTTM5n generates repeatedly at the same interval.

Cautions
1. **When compare register 5n (CR5n) is set, timer operation must be stopped.**
2. **If the count value of the upper timer matches CR5n, INTTM5n of the upper timer generates even in use of cascade connection.**
3. **Setting of TCE5n should be done from the upper time to the lower time.**
4. **Count Restart/Stop can operate/Stop by setting TCE5n of the lowest timer to 1/0.**

Remark n = 0, 1

Figure 9-11 indicates an example of timing of 16-bit solution cascade connection mode.

Figure 9-11. 16-bit Solution Cascade Connection Mode

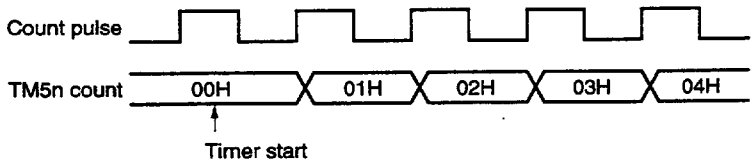


9.5 Cautions

(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 8-bit counter 5n (TM5n) is started asynchronously with the count pulse.

Figure 9-12. Timer 5 Start Timing

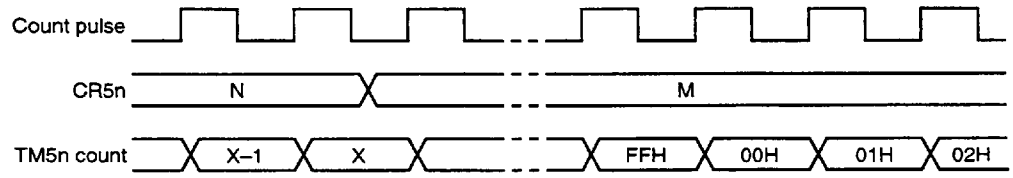


n = 0, 1

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

If the values after the 8-bit compare register 5n (CR5n) are changed is smaller than the value of 8-bit timer register5n(TM5n), TM5n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR5n is smaller than value (N) before the change, it is necessary to restart the timer after changing CR5n.

Figure 9-13. Timing after Compare Register Change during Timer Count Operation



Remarks 1. N > X > M

2. n = 0, 1

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

Remark n = 0, 1

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

When TM5n is read during operation, choose select clock which has longer high/low level wave because select clock is stopped temporary.

PAGE(S) INTENTIONALLY BLANK

CHAPTER 10 WATCH TIMER

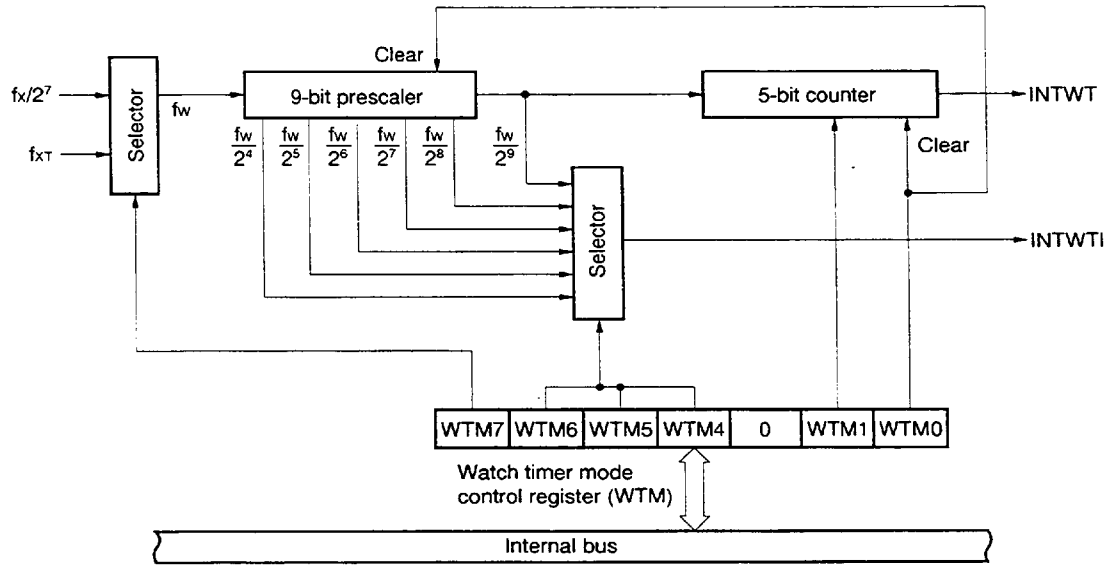
10.1 Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
The figure 10-1 shows Watch Timer Block Doagram

Figure 10-1. Block Diagram of Watch Timer



(1) Watch timer

When the main system clock or subsystem clock is used, interrupt requests (INTWT) are generated at 0.5 second intervals.

(2) Interval timer

Interrupt requests (INTWT1) are generated at the preset time interval.

Table 10-1. Interval Timer Interval Time

Interval Time	When operated at fx = 8.38 MHz	When operated at fx = 4.19 MHz	When operated at fxT = 32.768 kHz
$2^{11} \times 1/fx$	244 μs	489 μs	488 μs
$2^{12} \times 1/fx$	489 μs	978 μs	977 μs
$2^{13} \times 1/fx$	978 μs	1.96 ms	1.95 ms
$2^{14} \times 1/fx$	1.96 ms	3.91 ms	3.91 ms
$2^{15} \times 1/fx$	3.91 ms	7.82 ms	7.81 ms
$2^{16} \times 1/fx$	7.82 ms	15.6 ms	15.6 ms

Remark fx : Main system clock oscillation frequency
fxT : Subsystem clock oscillation frequency

10.2 Configuration

The watch timer consists of the following hardware.

Table 10-2. Watch Timer Configuration

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

10.3 Watch Timer Operating Register

- **Watch timer mode control register (WTM)**

  This register sets the watch timer operating mode, watch flag set time and prescaler interval time and enables/disables prescaler and 5-bit counter operations. WTM is set with a 1-bit or 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets 00H.

Figure 10-2. Watch Timer Mode Control Register (WTM) Format

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 operating mode selection
0	f_{X2^7} (65.4 kHz)
1	f_{XT} (32.768 kHz)

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

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

WTM0	Watch flag set time selection
0	Operation stop (clear both prescaler and timer)
1	Operation enable

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_W = 32.768 KHz.

10.4 Watch Timer Operations

10.4.1 Watch timer operation

When the 32.768-kHz subsystem clock or 8.38-MHz main system clock is used, the timer operates as a watch timer with a 0.5-second interval.

The watch timer is generated interrupt request at the constant time interval.

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

For simultaneous operation of the interval timer, zero-second start can be achieved by setting WTM1 to 0.

10.4.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt request repeatedly at an interval of the preset count value.

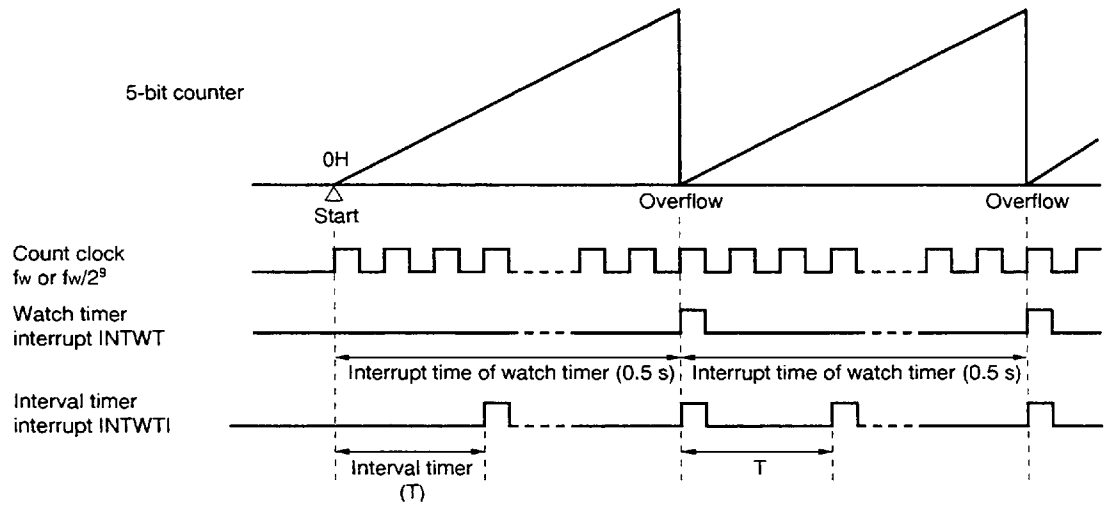
The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer mode control register (WTM).

Table 10-3. 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

Figure 10-3. Operation Timing of Watch Timer/Interval Timer



Remark fw : Watch timer clock frequency
( ) : fw = 32.768 kHz

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 11 WATCHDOG TIMER

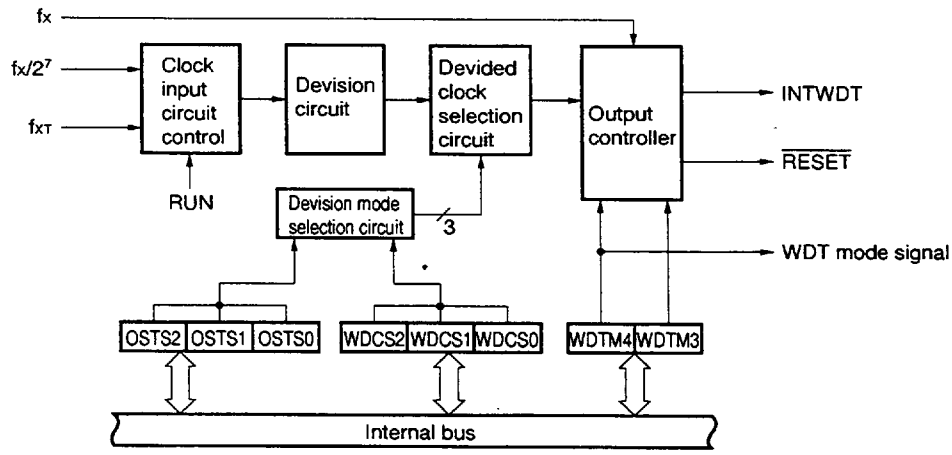
11.1 Functions

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).

Figure 11-1. Block Diagram of Watchdog Timer



(1) Watchdog timer mode

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

Table 11-1. Watchdog Timer Inadvertent Program Overrun Detection Times

Overrun detection times
$2^{12} \times 1/fx$ (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 apply to operation with fx = 8.38 MHz.

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 11-2. Interval Times

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 apply to operation with fx = 8.38 MHz.

11.2 Configuration

The watchdog timer consists of the following hardware.

Table 11-3. Watchdog Timer Configuration

Item	Configuration
Control register	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM)

11.3 Watchdog Timer Control Registers

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

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Watchdog timer clock select register (WDCS)

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

WDCS is set with 8-bit memory manipulation instruction.

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

Figure 11-2. Watchdog Timer Clock Select Register (WDCS) Format

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}/fx (489 μs)
0	0	1	2^{13}/fx (978 μs)
0	1	0	2^{14}/fx (1.96 ms)
0	1	1	2^{15}/fx (3.91 ms)
1	0	0	2^{16}/fx (7.82 ms)
1	0	1	2^{17}/fx (15.6 ms)
1	1	0	2^{18}/fx (31.3 ms)
1	1	1	2^{20}/fx (125 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parenthesises apply to operation with fx = 8.38 MHz

(2) Watchdog timer mode register (WDTM)

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

WDTM is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 11-3. Watchdog Timer Mode Register (WDTM) Format

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 (Maskable interrupt request occurs upon generation of an overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs up 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.

Caution When 1 is set in RUN so that the watchdog timer is cleared, the actual overflow time is up to 0.5 % shorter than the time set by Watchdog Timer Clock Select Register.

Remark ×: don't care

11.4 Operations

11.4.1 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 inadvertent program loop.

The watchdog timer count clock (inadvertent program loop detection time interval) can be selected with bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2).

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 overrun 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 inadvertent program loop detection time is past, system reset or a non-maskable interrupt request is generated according to the 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 overrun detection time may be shorter than the set time by a maximum of 0.5 %.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 11-4. Watchdog Timer Overrun Detection Time

Runover 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 apply to operation with fx = 8.38 MHz.

11.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt request repeatedly at an interval of the preset count value when bit 3 (WDTM3) and bit 4 (WDTM4) of the watchdog timer mode register (WDTM) are set to 1 and 0, respectively.

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specify flag (TMPR4) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, the INTWDT default has the highest priority.

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 (with the watchdog timer mode selected), the interval timer mode is not set unless $\overline{\text{RESET}}$ input is applied.

2. The interval time just after setting with WDTM may be shorter than the set time by a maximum of 0.5 %.

3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 11-5. 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 apply to operation with fx = 8.38 MHz.

CHAPTER 12 CLOCK OUTPUT/BUZZER OUTPUT CONTROL CIRCUITS

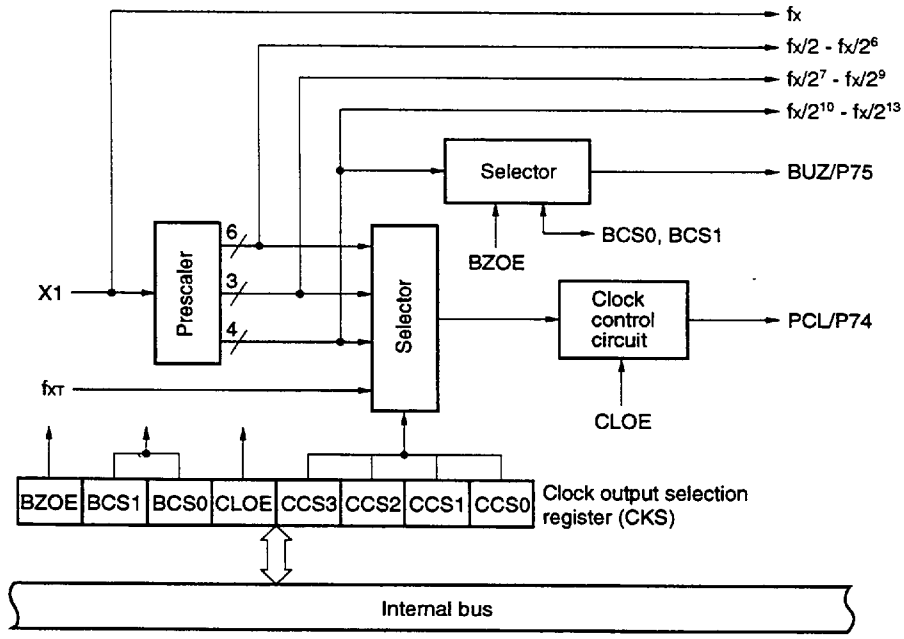
12.1 Functions

The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

In addition, the buzzer output is intended for square wave output of buzzer frequency selected with CKS.

Figure 12-1 shows the CKU block diagram.

Figure 12-1. CKU Block Diagram



12.2 Configuration

The CKU consists of the following hardware.

Table 12-1. CKU Configuration

Item	Configuration
Control register	Clock output selection register (CKS) Port mode register (PM7)

12.3 CKU Control Register

The following two types of registers are used to control the CKU.

- Clock output selection register (CKS)
- Port mode register (PM7)

(1) Clock output selection register (CSK)

This register sets output enable/disable for clock output (CKUPCL) and output enable/disable for the buzzer frequency output (CKUBUZ).

CKS is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 12-2. Clock Output Selection Register (CKS) Format

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	CKUBUZ Output Enable/Disable Specification
0	Stop clock division circuit operation. CKUBUZ fixed to low level.
1	Enable clock division circuit operation. CKUBUZ output enabled.

BCS1	BCS0	CKUBUZ 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	CKUPCL Output Enable/Disable Setting
0	Stop clock division circuit operation. CKUPCL fixed to low level
1	Enable clock division circuit operation. CKUPCL output enabled.

CCS3	CCS2	CCS1	CCS0	CKUPCL Output Clock Selection
0	0	0	0	f_x (8.38 kHz)
0	0	0	1	$f_x/2$ (4.19 kHz)
0	0	1	0	$f_x/2^2$ (2.09 kHz)
0	0	1	1	$f_x/2^3$ (1.04 kHz)
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 apply to operation with f_x = 8.38 MHz or f_{XT} = 32.768.

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

$\overline{\text{RESET}}$ input sets PM7 to 00H.

Figure 12-3. Port Mode Register 7 (PM7) Format

Address: FF27H After Reset: FFH R/W

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

PM7n	P7n Pin Input Output Mode Selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

12.4 Operation

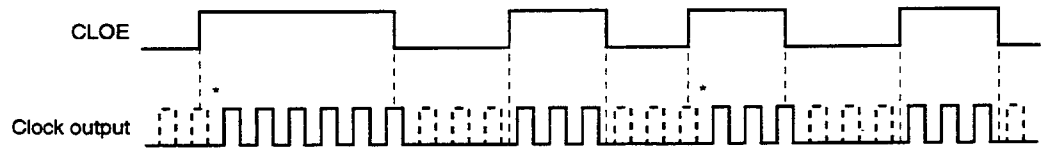
12.4.1 Operation as clock output

To output the clock pulse, follow the procedure described below.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).
<2> Set bit 4 (CLOE) of CKS to 1, and enable clock output.

Remark The clock output control circuit is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 12-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after securing high level of the clock.

Figure 12-4. Remote Control Output Application Example



12.4.2 Operation as buzzer output

To output the buzzer frequency, follow the procedure described below.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output selection register (CKS) (buzzer output in disabled status).
<2> Set bit 7 (BZ0E) of CKS to 1 to enable buzzer output.

13.1 Functions

AD1 is an 8-bit resolution converter that converts analog inputs into digital values. It can control up to 8 analog input channels (ANI0 to ANI7).

(1) Hardware start

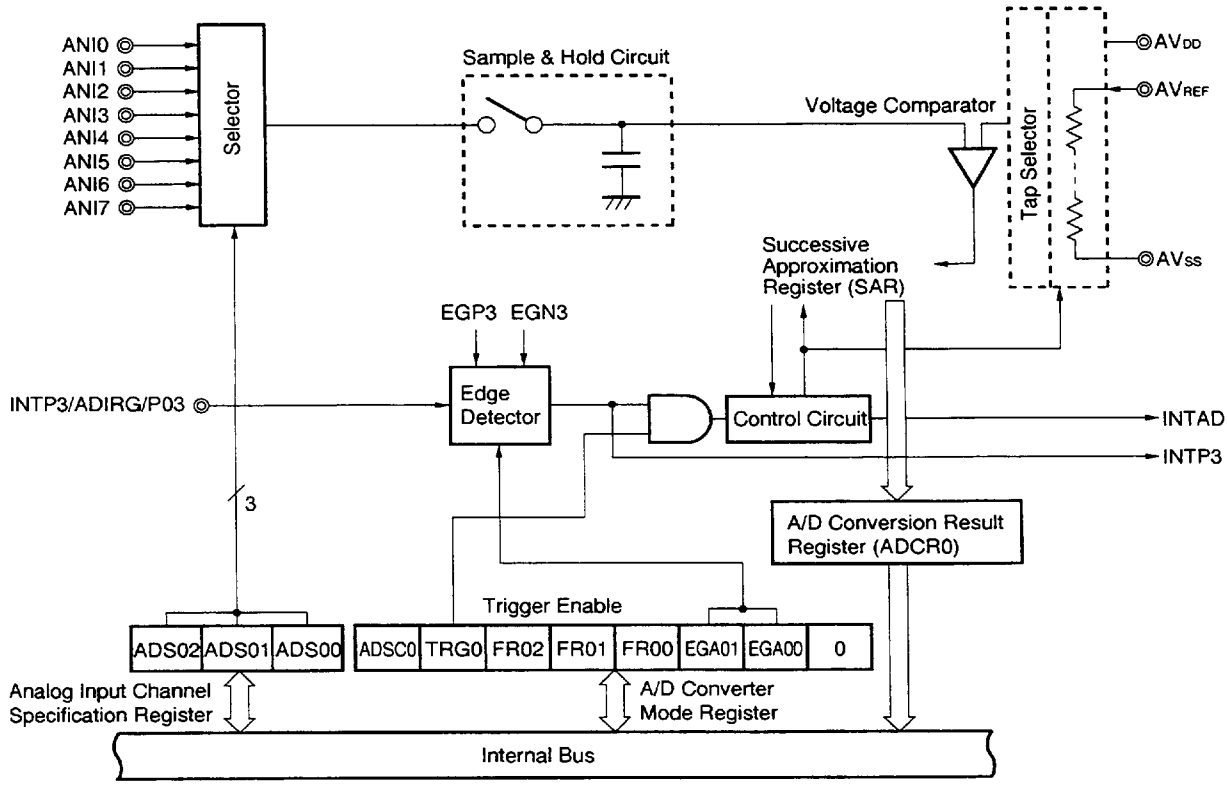
Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

Conversion is started by setting the A/D converter mode register (ADM0).

Select one channel for analog input from ANI0 to ANI7 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 (INTAD) is generated. In the case of software start, A/D conversion is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 13-1. AD1 Block Diagram



13.2 Configuration

AD1 consists of the following hardware.

Table 13-1. AD1 Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR0)
Control register	A/D converter mode register (ADM0) Analog input channel specification register (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 value) 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 set (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

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

This register holds the A/C conversion result. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

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

$\overline{\text{RESET}}$ input sets ADCR0 to 00H.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input 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 in AV_{REF} to AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are eight analog input pins to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 are dual-function pins that can also be used for digital input.

Caution Use ANI0 to ANI7 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.

(7) AV_{REF} pin

This pin inputs the A/D covnerter reference voltage.

It converts signals input to AN10 to AN17 into digital signals according to the voltage applied between AV_{REF} and AVss.

The current flowing in the series resistor string can be reduced by setting the voltage to be input to the AV_{REF} pin to AVss level in the standby mode.

(8) AVss pin

This is the GND potential pin of the A/D converter. Always keep it at the same potential as the Vss 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} pin even when not using the A/D converter.

13.3 A/D1 Control Registers

The following 4 types of registers are used to control AD1.

- A/D converter mode register (ADM0)
- Analog input channel specification register (ADS0)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)

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

This register sets the conversion time for analog input to be A/D converted, conversion start/stop and external trigger. ADM0 is set with an 8-bit memory manipulation instruction.

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

Figure 13-2. A/D Converter Mode Register (ADM0) Format

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.

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 8.38 MHz.

(2) Analog input channel specification register (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set with an 8-bit memory manipulation.

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

Figure 13-3. Analog Input Channel Specification Register (ADS0) Format

Address: FF81H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00

ADS02	ADS01	ADS00	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

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

$\overline{\text{RESET}}$ input sets EGP and EGN to 00H.

Figure 13-4. External Interrupt Rising Edge Enable Register (EGP), Internal Interrupt Falling Edge Enable Register (EGN) Format

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	ENG1	ENG0

EGPn	EGNn	INTPn Pin Valid Edge Selection (n = 0 to 3)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

13.4 A/D Converter Operations

13.4.1 Basic operations of AD1

<1> Select one channel for A/D conversion with the analog input channel specification register (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> Set bit 7 of the successive approximation register (SAR) so that the tap selector sets the series resistor string voltage tap to (1/2) AV_{REF}.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared with 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) AF_{REF}

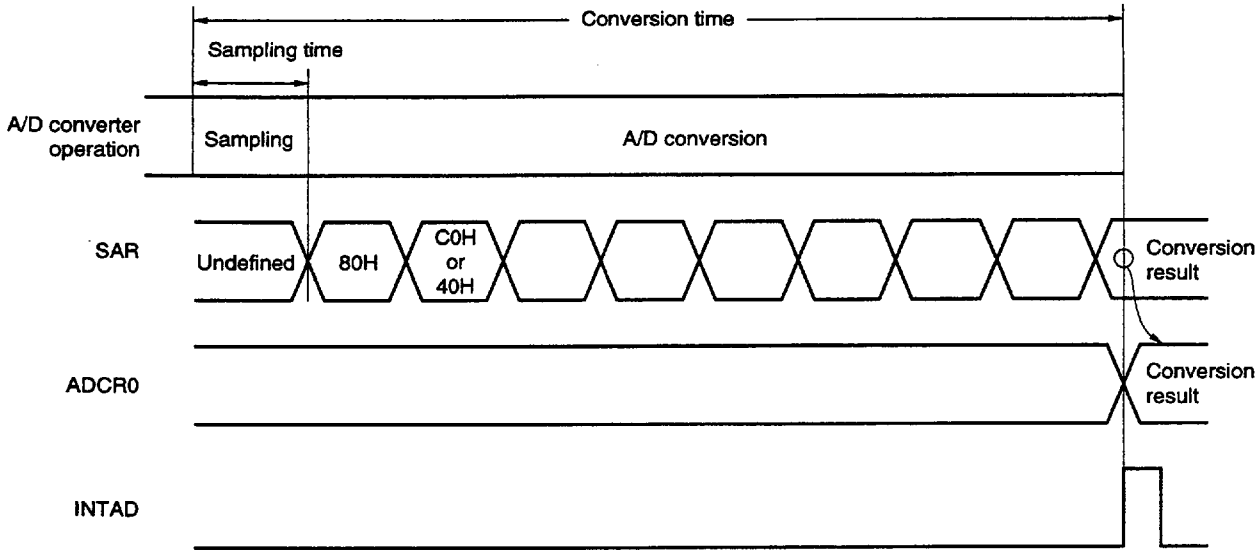
The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated as follows.

- 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 (ADCR0).
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

Figure 13-5. 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 (ADM0) is reset (0) by software.

If a write operation to the ADM0 analog input channel specification register (ADS0) is performed 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.

After $\overline{\text{RESET}}$ input, the value of the A/D conversion result register (ADCR0) becomes $\overline{\text{RESET}}$ input sets the A/D conversion result register (ADCR0) to 00H.

13.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (AN10 to AN17) and the A/D conversion result (stored in the A/D conversion result register (ADCR0)) is shown by the following expression.

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

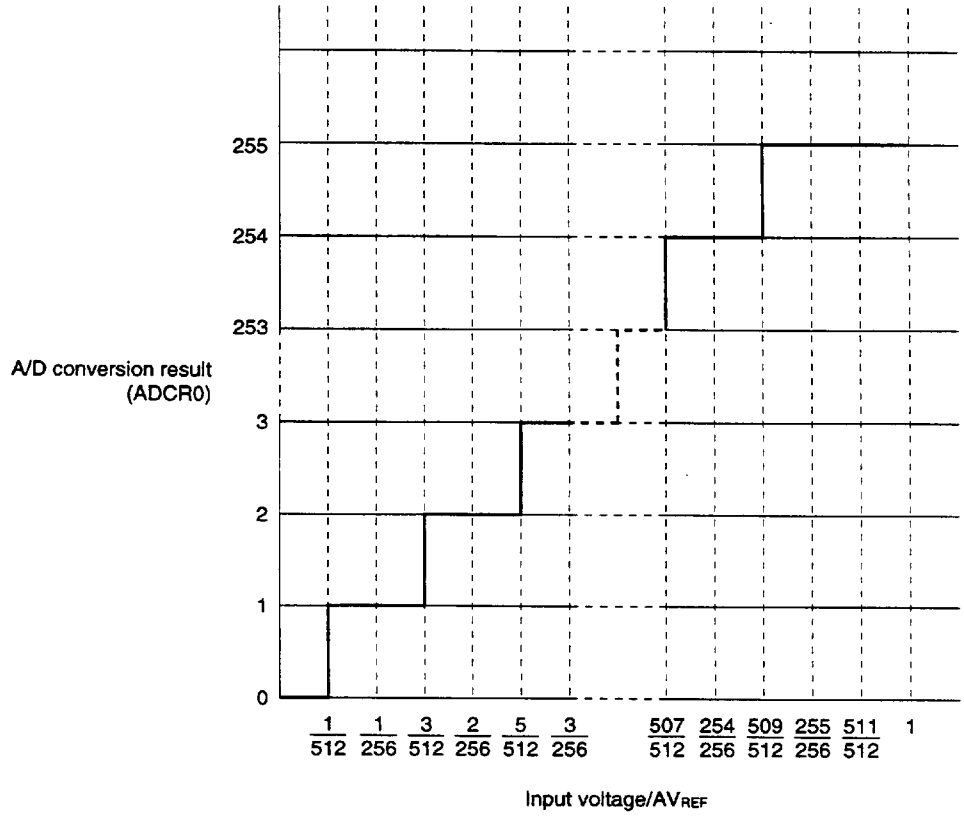
or

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

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 (ADCR0) value

Figure 13-6 shows the relation between the analog input voltage and the A/D conversion result.

Figure 13-6. Relation between Analog Input Voltage and A/D Conversion Result



13.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One analog input channel is selected from among ANI0 to ANI7 with the analog input channel specification register (ADS0) and A/D conversion is performed.

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 enabled).
- Software start : Conversion is started by specifying the A/D converter mode register (ADM0).

The A/D conversion result is stored in the A/D conversion result register (ADCR0), and the interrupt request signal (INTAD) is simultaneously generated.

(1) A/D conversion by hardware start

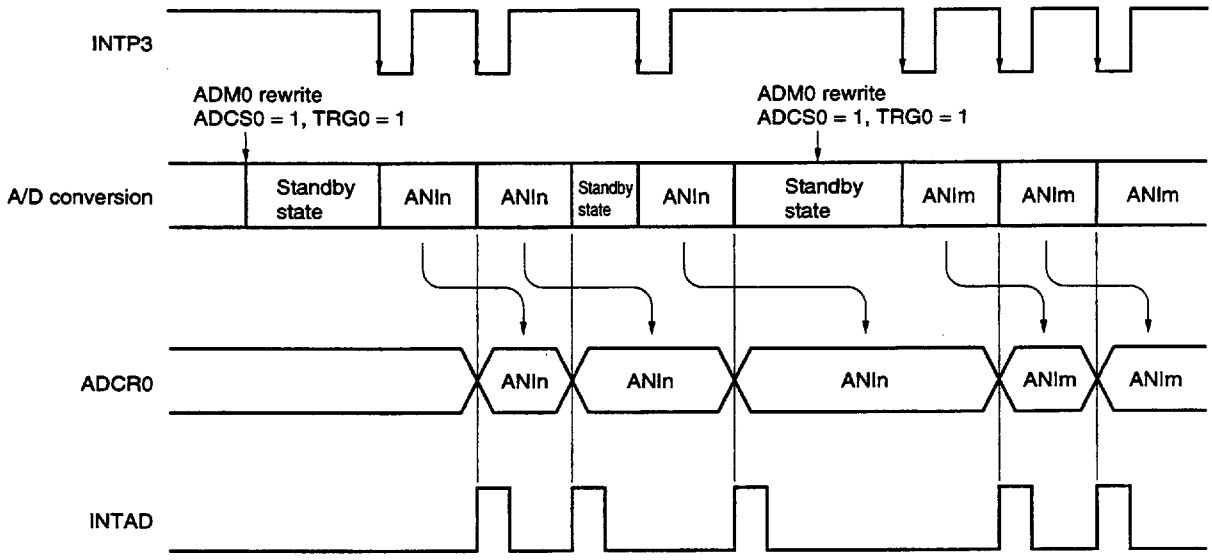
When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register (ADM0) are set to 1, the A/D conversion stanby state is set. When the external trigger signal (INTP3) is intput, A/D conversion of the voltage applied to the analog input pins specified with the analog input channel specification register (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR0), and the interrupt request signal (INTAD) 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 data with ADCS0 set to 1 is written to ADM0 again during A/D conversion, the converter suspends its A/D conversion operation 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 data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 13-7. 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

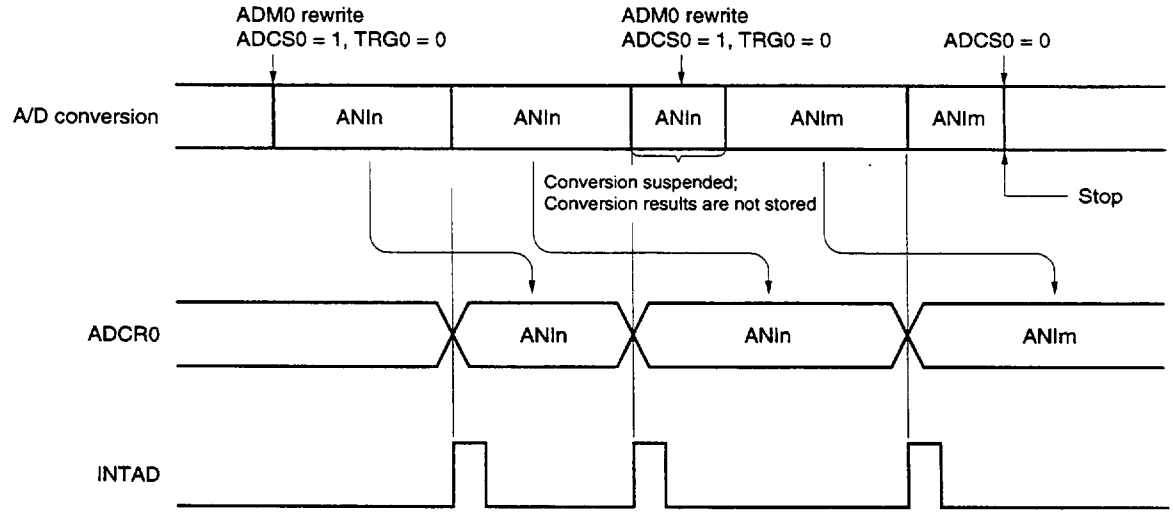
When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register (ADM0) are set to 0 and 1, respectively, A/D conversion of the voltage applied to the analog input pin specified with the analog input channel specification register (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR0), and the interrupt request signal (INTAD) 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 ADM0.

If data with ADCS0 set to 1 is written to ADM0 again during A/D conversion, the converter suspends its A/D conversion operation and A/D conversion of the newly written data 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 13-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

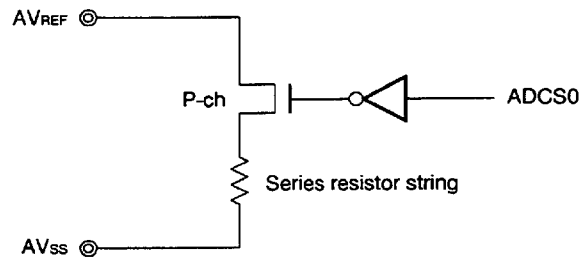
13.5 Cautions

(1) Power consumption in standby mode

AD1 stops operating in the standby mode. At this time, power consumption can be reduced by setting bit 7 (ADCS0) of the A/D converter mode register (ADM0) to stop conversion.

Figure 13-9 shows how to reduce the power consumption in the standby mode.

Figure 13-9. Example of Method of Reducing Power Dissipation in Standby Mode



(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage higher than AVREF or lower than AVss 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 (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.

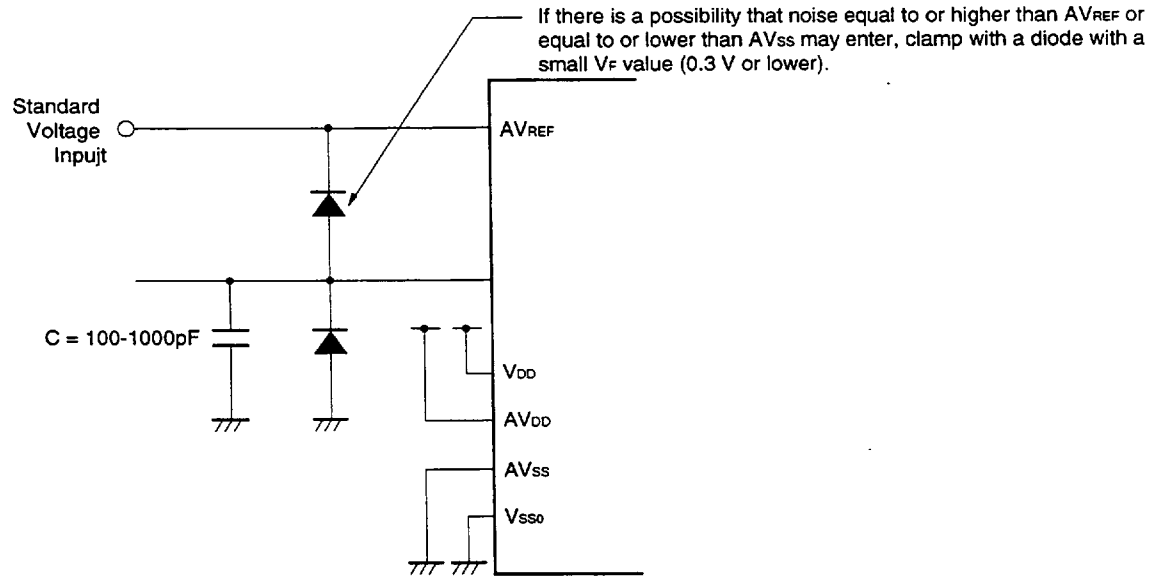
<3> Contention between ADCR0 write and A/D converter mode register (ADM0) write or analog input channel specification register (ADS0) write

ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD) generated.

<4> Noise countermeasures

To maintain 8-bit resolution, attention must be paid to noise input to pin AVREF and pins ANI0 to ANI7. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 13-10 to reduce noise.

Figure 13-10. Analog Input Pin Handling



(5) ANI0 to ANI7

The analog input pins (ANI0 to ANI7) also function as input/output port pins (P10 to P17).

When A/D conversion is performed with any of pins ANI0 to ANI7 selected, do not execute a port input instruction 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) AVREF pin input impedance

A series resistor string of approximately 21 kΩ is connected between the AVREF pin and the AVss pin.

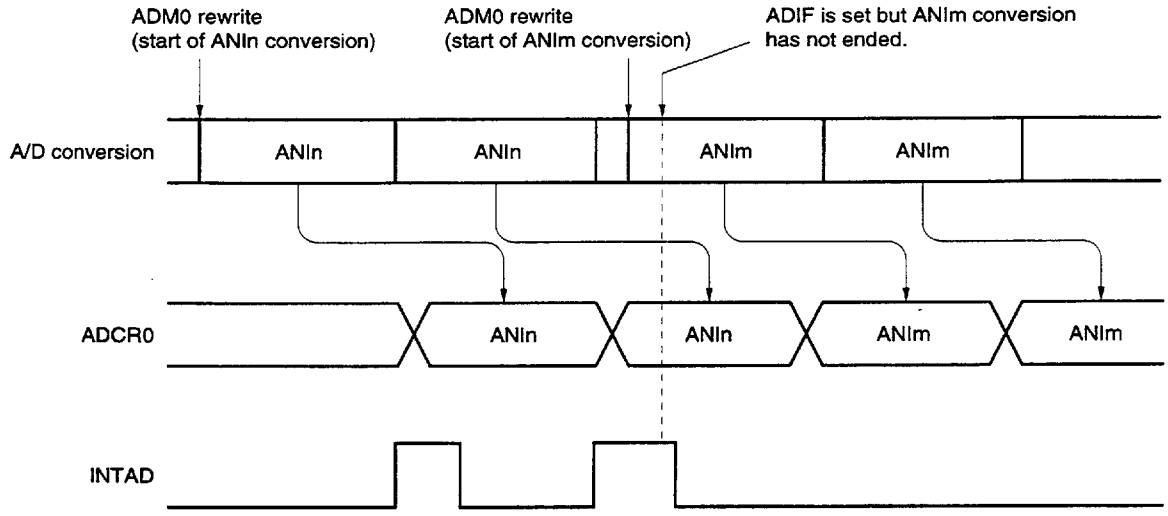
Therefore, if the output impedance of the reference voltage is high, this will result in parallel connection to the series resistor string between the AVREF pin and the AVss pin, and there will be a large reference voltage error.

(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM0) is changed. Caution is therefore required since, if a change of analog input pin is performed 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 ADM rewrite, and when ADIF is read immediately after the ADM rewrite, ADIF may be set despite the fact that the A/D conversion for the post-change analog input has not ended.

When the A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 13-11. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(8) 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 ANI7 pins.

Therefore, be sure to apply the same voltage as V_{DD} to this pin even when the application circuit is designed so as to switch to a backup battery.

14.1 Functions

AD0 is a 10-bit resolution converter that converts analog inputs into digital values. It can control up to 8 analog input channels (ANI0 to ANI7).

(1) Hardware start

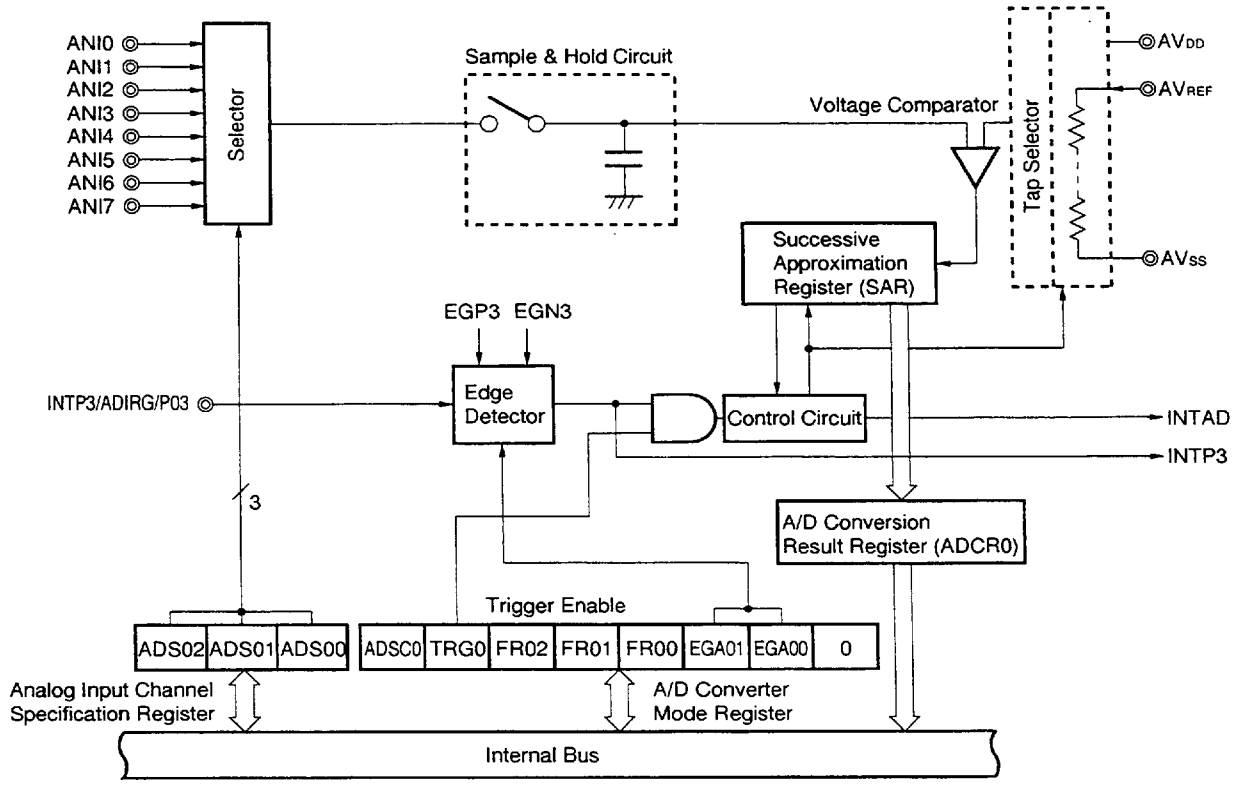
Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

Conversion is started by setting the A/D converter mode register (ADM0).

Select one channel for analog input from ANI0 to ANI7 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 (INTAD) is generated. In the case of software start, A/D conversion is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 14-1. AD0 Block Diagram



14.2 Configuration

AD0 consists of the following hardware.

Table 14-1. AD0 Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR0)
Control register	A/D converter mode register (ADM0) Analog input channel specification register (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 value) 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 set (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

(2) A/D conversion result registers (ADCR0)

Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and held by this register. ADCR0 is read with 1-bit memory manipulation instructions.

$\overline{\text{RESET}}$ input sets ADCR0 to 00H.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input 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 in AV_{REF} to AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are eight analog input pins to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 are dual-function pins that can also be used for digital input.

Caution Use ANI0 to ANI7 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.

(7) AV_{REF} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.

The current flowing in the series resistor string can be reduced by setting the voltage to be input to the AV_{REF} pin to AVss level in the standby mode.

(8) AV_{SS} pin

This is the GND potential pin of the A/D converter. Always keep it at the same potential as the Vss 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 Vss pin even when not using the A/D converter.

14.3 AD0 Control Registers

The following 4 types of registers are used to control AD0.

- A/D converter mode register (ADM0)
- Analog input channel specification register (ADS0)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)

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

This register sets the conversion time for analog input to be A/D converted, conversion start/stop and external trigger. ADM0 is set with an 8-bit memory manipulation instruction.

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

Figure 14-2. A/D Converter Mode Register (ADM0) Format

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)
1	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.

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 8.38 MHz.

(2) Analog input channel specification register (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set with an 8-bit memory manipulation.

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

Figure 14-3. Analog Input Channel Specification Register (ADS0) Format

Address: FF81H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00

ADS02	ADS01	ADS00	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

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

$\overline{\text{RESET}}$ input sets EGP and EGN to 00H.

Figure 14-4. External Interrupt Rising Edge Enable Register (EGP), Internal Interrupt Falling Edge Enable Register (EGN) Format

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	ENG1	ENG0

EGPn	EGNn	INTPn Pin Valid Edge Selection (n = 0 to 3)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

14.4 Operation

14.4.1 Basic operations of AD0

<1> Select one channel for A/D conversion with the analog input channel specification register (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> Set bit 9 of the successive approximation register (SAR) so that the tap selector sets the series resistor string voltage tap to (1/2) AV_{REF}.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared with 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 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AF_{REF}

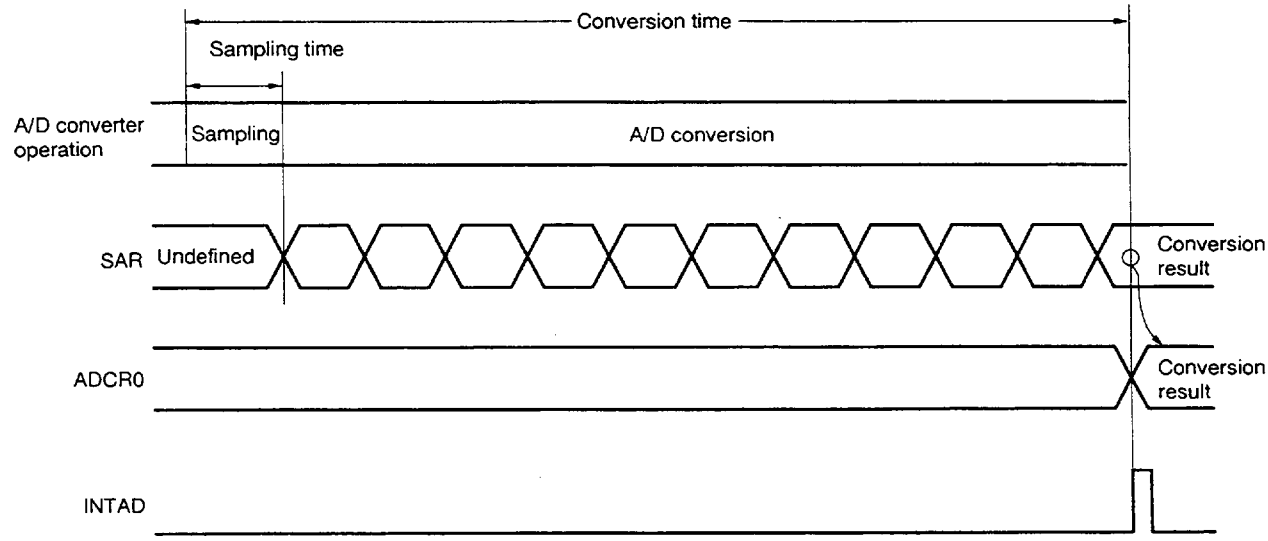
The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

- 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 (ADCR0).
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

Figure 14-5. 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 (ADM0) is reset (0) by software.

If a write operation to the ADM0 analog input channel specification register (ADS0) is performed 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.

After $\overline{\text{RESET}}$ input, the value of the A/D conversion result register (ADCR0H) becomes $\overline{\text{RESET}}$ input sets the A/D conversion result register (ADCR0H) to 00H.

14.4.2 Input voltage and conversion results

The relation 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 (ADCR0H)) is shown by the following expression.

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

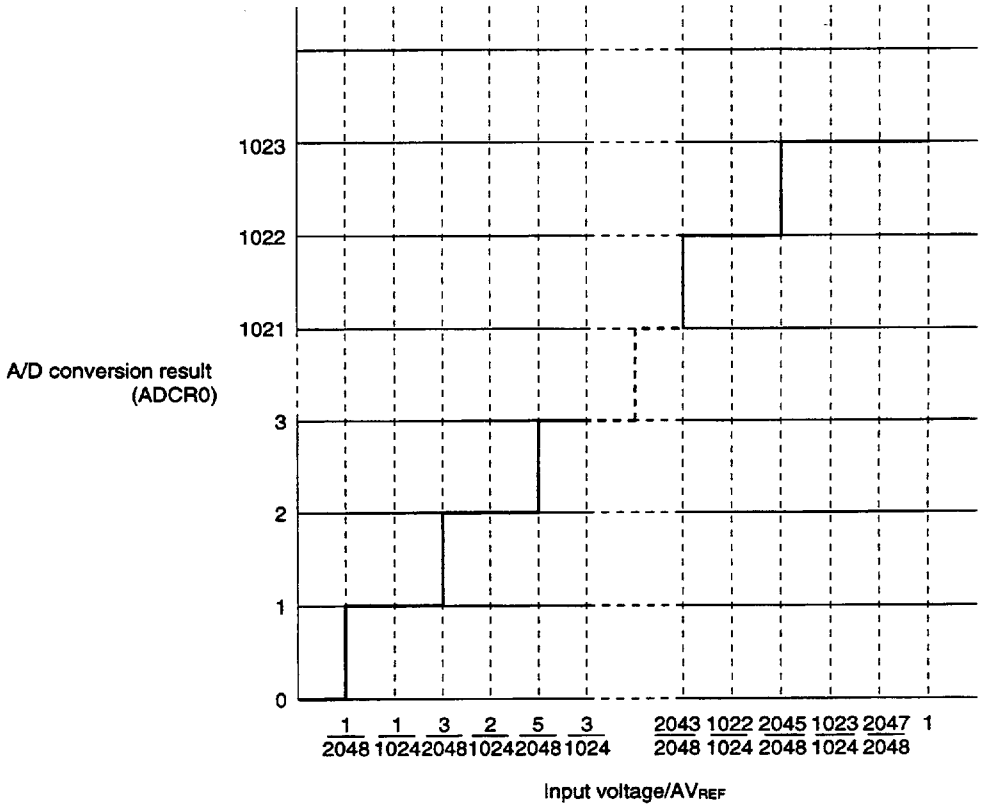
or

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

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 (ADCR0) value

Figure 14-6 shows the relation between the analog input voltage and the A/D conversion result.

Figure 14-6. Relation between Analog Input Voltage and A/D Conversion Result



14.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One analog input channel is selected from among ANI0 to ANI7 with the analog input channel specification register (ADS0) and A/D conversion is performed.

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 enabled).
- Software start : Conversion is started by specifying the A/D converter mode register (ADM0).

The A/D conversion result is stored in the A/D conversion result register (ADCR0), and the interrupt request signal (INTAD) is simultaneously generated.

(1) A/D conversion by hardware start

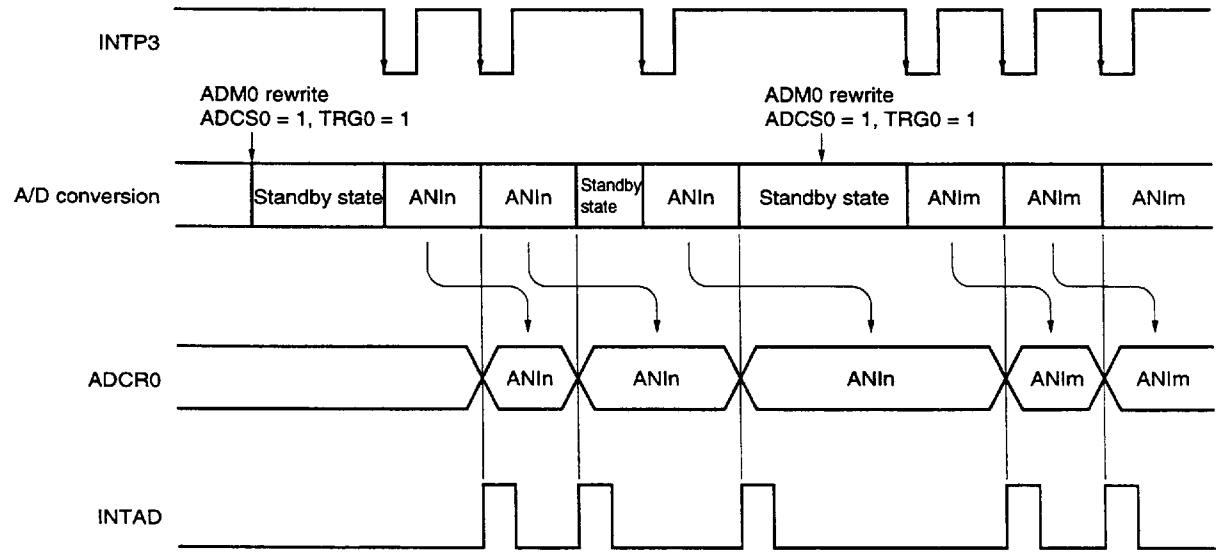
When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register (ADM0) are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is intput, A/D conversion of the voltage applied to the analog input pin specified with the analog input channel specification register (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR0), and the interrupt request signal (INTAD) 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 data with ADCS0 set to 1 is written to ADM0 again during A/D conversion, the converter suspends its A/D conversion operation 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 data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 14-7. 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

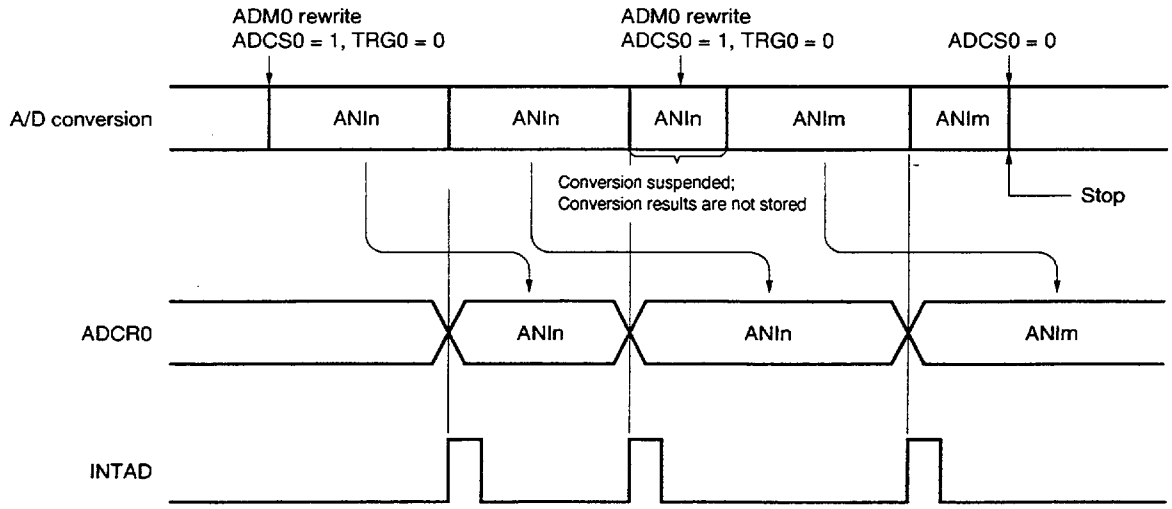
When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register (ADM0) are set to 0 and 1, respectively, A/D conversion of the voltage applied to the analog input pin specified with the analog input channel specification register (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR0H), and the interrupt request signal (INTAD) 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 ADM0.

If data with ADCS0 set to 1 is written to ADM0 again during A/D conversion, the converter suspends its A/D conversion operation and A/D conversion of the newly written data 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 14-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

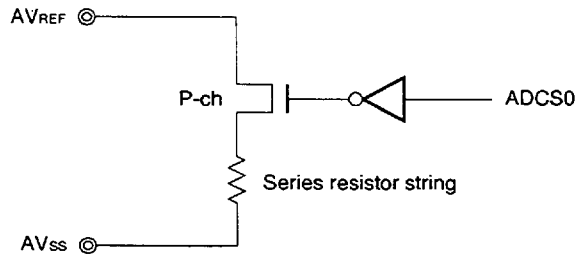
14.5 Cautions

(1) Power consumption in standby mode

AD0 stops operating in the standby mode. At this time, power consumption can be reduced by setting bit 7 (ADCS0) of the A/D converter mode register (ADM0) to stop conversion.

Figure 14-9 shows how to reduce the power consumption in the standby mode.

Figure 14-9. Example of Method of Reducing Power Dissipation in Standby Mode



(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage higher than AVREF or lower than AVss 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 (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.

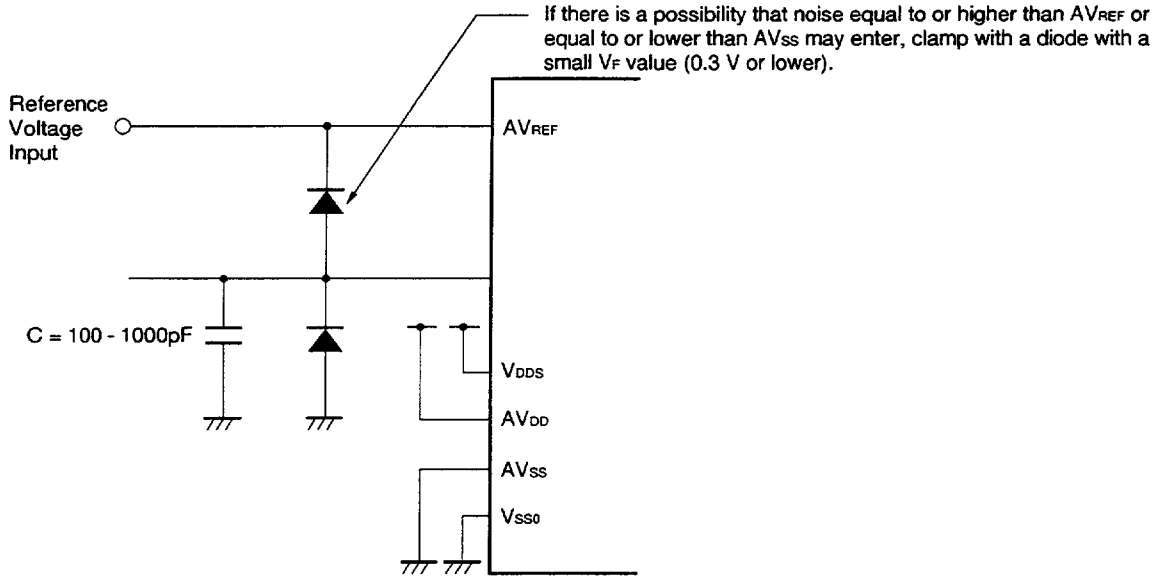
<3> **Contention between ADCR0 write and A/D converter mode register (ADM0) write or analog input channel specification register (ADS0) write**

ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD) generated.

<4> **Noise countermeasures**

To maintain 10-bit resolution, attention must be paid to noise input to pin AVREF and pins ANI0 to ANI7. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 14-10 to reduce noise.

Figure 14-10. Analog Input Pin Handling



(5) ANI0 to ANI7

The analog input pins (ANI0 to ANI7) also function as input/output port pins (P10 to P17).

When A/D conversion is performed with any of pins ANI0 to ANI7 selected, do not execute a port input instruction 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) AVREF pin input impedance

A series resistor string of approximately 47 kΩ is connected between the AVREF pin and the AVSS pin.

Therefore, if the output impedance of the reference voltage is high, this will result in parallel connection to the series resistor string between the AVREF pin and the AVSS pin, and there will be a large reference voltage error.

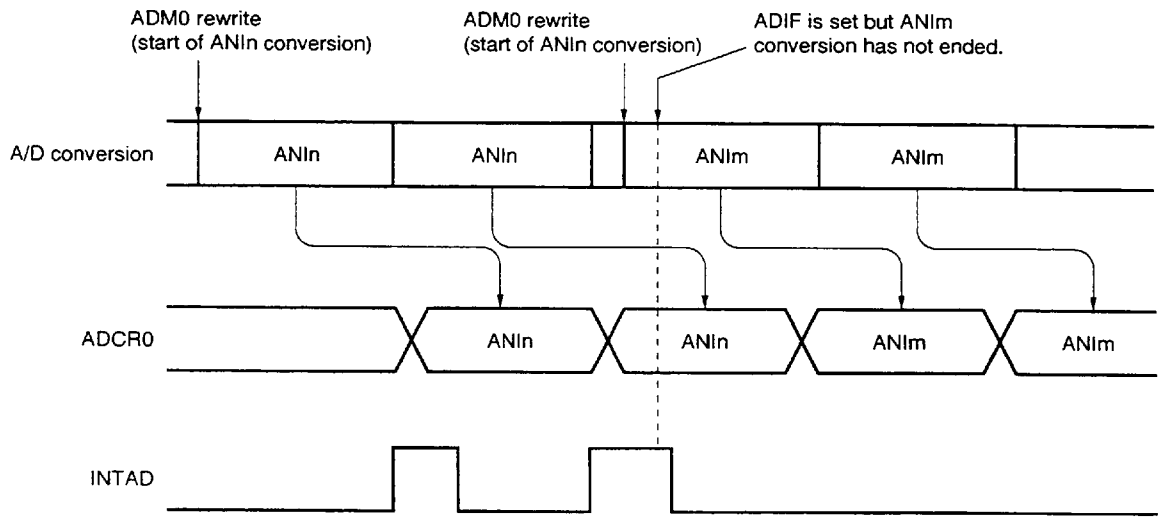
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM) is changed.

Caution is therefore required since, if a change of analog input pin is performed 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 ADM0 rewrite, and when ADIF is read immediately after the ADM0 rewrite, ADIF may be set despite the fact that the A/D conversion for the post-change analog input has not ended.

When the A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 14-11. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 7
2. m = 0, 1, ..., 7

(8) AV_{DD} pin

The AV_{DD} pin is the analog circuit power supply pin. It supplies power to the input circuits of the AN10 to AN17 pins.

Therefore, be sure to apply the same voltage as V_{DD} to this pin even when the application circuit is designed so as to switch to a backup battery.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 15 SERIAL INTERFACE OUTLINE

The μPD780024, 780034 Subseries and the μPD780024Y, 780034Y Subseries have differences in their interfaces. These differences are listed in Table 15-1.

Table 15-1. Differences between μPD780024, 780034 Subseries and the μPD780024Y, 780034Y Subseries

Item		μPD780024, 780034	μPD780024Y, 780034Y	Relevant Section
UART0		O	O	CHAPTER 16
SIO3	SIO30	O	O	CHAPTER 17
	SIO31	O	—	
IIC0		—	O	CHAPTER 18

Remark O: Provided; —: Not provided

[MEMO]

PAGE(S) INTENTIONALLY BLANK

16.1 FUNCTIONS

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 **16.5.1 Operation Stop Mode**.

(2) Asynchronous serial interface (UART0) mode

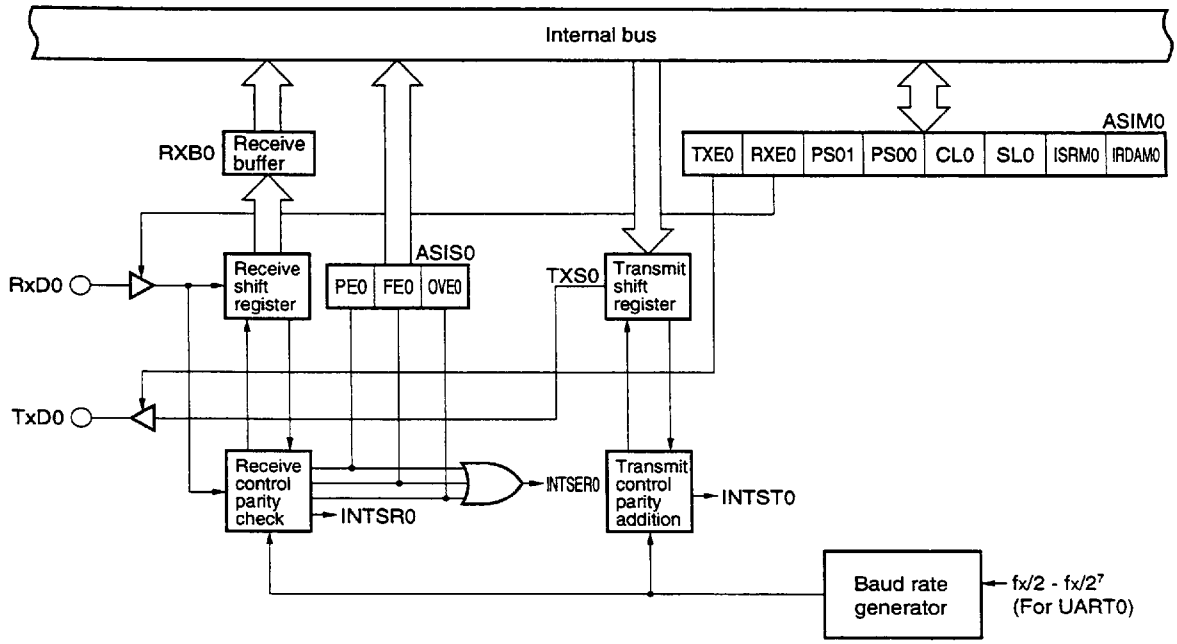
This mode enables full-duplex operation wherein one byte of data is transmitted and received after the start bit. The on-chip dedicated UART0 baud rate generator enables communications using a wide range of selectable baud rates. In addition, a baud rate based on divided clock input to the ASCK pin can also be defined. The UART0 baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).
For details, see **16.5.2 A synchronous Serial Interface (UART0) Mode**.

(3) Infrared data transfer (IrDA0) mode

For details, see **16.5.3 Infrared Data Transfer (IrDA0) Mode**.

Figure 16-1 shows a block diagram of the UART0 macro.

Figure 16-1. Block Diagram of UART0



16.2 CONFIGURATION

The UART0 includes the following hardware.

Table 16-1. Configuration of UART0

Item	Configuration
Registers	Transmit shift register 1 (TXS0) Receive shift register 1 (RX0) Receive buffer register (RXB0)
Control registers	Asynchronous serial interface mode register (ASIM0) Asynchronous serial interface status register (ASIS0) Baud rate generator control register (BRGC0)

(1) Transmit shift register 1 (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 transmitted as serial data. Writing data to TXS0 starts the transmit operation.

TXS0 can be written to via 8-bit memory manipulation instructions. It cannot be read.

When $\overline{\text{RESET}}$ is input, its value is FFH.

Caution Do not write to TXS0 during a transmit operation.
The same address is assigned to TXS0 and the receive buffer register (RXB0). A read operation reads values from RXB0.

(2) Receive shift register 1 (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 (RXB0).

RXS0 cannot be manipulated directly by a program.

(3) Receive buffer register (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 RXB0, the MSB must be set to "0".

RXB0 can be read to via 8-bit memory manipulation instructions. It cannot be written to.

When $\overline{\text{RESET}}$ is input, its value is FFH.

Caution The same address is assigned to RXB0 and the transmit shift register (TXS0). During a write operation, values are written to TXS0.

(4) Transmission control circuit

The transmission control circuit controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to the transmit shift register (TXS0), based on the values set to the asynchronous serial interface mode register (ASIM0).

(5) Reception control circuit

The reception control circuit controls receive operations based on the values set to the asynchronous serial interface mode register (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 (ASIS0) according to the type of error that is detected.

16.3 LIST OF SFRS (SPECIAL FUNCTION REGISTERS)

Table 16-2. List of SFRs (Special Function Registers)

SFR name	Symbol	R/W	Units available for bit manipulation			Value when reset
			1 bit	8 bits	16 bits	
Transmit shift register	TXS0	W	—	○	—	FFH
Receive buffer register	RXB0	R				
Asynchronous serial interface mode register	ASIM0	R/W	○	○	—	00H
Asynchronous serial interface status register	ASIS0	W	—	○	—	
Baud rate generator control register	BRGC0	R/W	—	○	—	

16.4 SERIAL INTERFACE CONTROL REGISTERS

The UART0 uses the following three types of registers for control functions.

- Asynchronous serial interface mode register (ASIM0)
- Asynchronous serial interface status register (ASIS0)
- Baud rate generator control register (BRGC0)

(1) Asynchronous serial interface mode register (ASIM0)

This is an 8-bit register that controls UART0's serial transfer operations.

ASIM0 can be set by 1-bit or 8-bit memory manipulation instructions.

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

Figure 16-2 shows the format of ASIM0.

Figure 16-2. Format of Asynchronous Serial Interface Mode Register (ASIM0)

Address: FFA0H When 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/PXX pin function	TxD0/PXX pin function
0	0	Operation stop	Port function	Port function
0	1	UART0 mode (receive only)	Serial function	Port function
1	0	UART0 mode (transmit only)	Port function	Serial function
1	1	UART0 mode (transmit and receive)	Serial function	Serial function

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmittion No parity detection during reception (parity errors do not occur)
1	0	Odd aprity
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 is issued when an error occurs
1	Receive completion interrupt is not issued when an error occurs

IRDAM0	Operation specified for infrared data transfer (IrDA0) mode[Note 1]
0	UART0 (transmit/receive) mode
1	IrDA0 (transmit/receive) mode[Note 2]

Notes 1. The UART0/IrDA0 mode specification is controlled by TXE0 and RXE0.

2. When using IrDA0 mode, be sure to set "00H" to the baud rate generator control register (BRGC0).

Caution Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

(2) Asynchronous serial interface status register (ASIS0)

When a receive error occurs during UART0 mode, this register indicates the type of error.

ASIS0 can be read using an 8-bit memory manipulation instruction.

When $\overline{\text{RESET}}$ is input, its value is 00H.

Figure 16-3. Format of Asynchronous Serial Interface Status Register (ASIS0)

Address: FFA1H When 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
0	Parity error (Incorrect parity bit detected)

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)

Notes 1. Even if a stop bit length of two bits has been set to bit 2 (SL0) in the asynchronous serial interface mode register (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 (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 (BRGC0)

This register sets the serial clock for UART0.

BRGC0 can be set via an 8-bit memory manipulation instruction.

When $\overline{\text{RESET}}$ is input, its value is 00H.

Figure 16-4 shows the format of BRGC0.

Figure 16-4. Format of Baud Rate Generator Control Register (BRGC0)

Address: FFA2H When reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(fx = 8.38 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P25/ASCK0	1
0	0	1	fx/2	2
0	1	0	$fx/2^2$	3
0	1	1	$fx/2^3$	4
1	0	0	$fx/2^4$	5
1	0	1	$fx/2^5$	6
1	1	0	$fx/2^6$	7
1	1	1	$fx/2^7$	8

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	fsck/16	0
0	0	0	1	fsck/17	1
0	0	1	0	fsck/18	2
0	0	1	1	fsck/19	3
0	1	0	0	fsck/20	4
0	1	0	1	fsck/21	5
0	1	1	0	fsck/22	6
0	1	1	1	fsck/23	7
1	0	0	0	fsck/24	8
1	0	0	1	fsck/25	9
1	0	1	0	fsck/26	10
1	0	1	1	fsck/27	11
1	1	0	0	fsck/28	12
1	1	0	1	fsck/29	13
1	1	1	0	fsck/30	14
1	1	1	1	Setting prohibit	—

Caution 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.

Remarks 1. fsck : Source clock for 5-bit counter

2. n : Value set via TPS00 to TPS02 ($1 \leq n \leq 8$)

3. k : Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

16.5 OPERATIONS

This section explains the three modes of the UART0.

16.5.1 Operation Stop Mode

This mode is used when serial transfers are not performed to reduce power consumption.

In the operation stop mode, pins can be used as ordinary ports.

(1) Register settings

Operation stop mode settings are made via the asynchronous serial interface mode register (ASIM0).

ASIM0 can be set via 1-bit or 8-bit memory manipulation instructions.

When $\overline{\text{RESET}}$ is input, its value is 00H.

Address: FFA0H When 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/PXX pin function	TxD0/PXX pin function
0	0	Operation stop	Port function	Port function
0	1	UART0 mode (receive only)	Serial function	Port function
1	0	UART0 mode (transmit only)	Port function	Serial function
1	1	UART0 mode (transmit and receive)	Serial function	Serial function

Caution Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

16.5.2 Asynchronous Serial Interface (UART0) Mode

This mode enables full-duplex operation wherein one byte of data is transmitted or received after the start bit.

The on-chip dedicated UART0 baud rate generator enables communications using a wide range of selectable baud rates.

The UART0 baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(1) Register settings

UART0 mode settings are made via the asynchronous serial interface mode register (ASIM0), asynchronous serial interface status register (ASIS0), and the baud rate generator control register (BRGC0n).

(a) Asynchronous serial interface mode register (ASIM0)

ASIM0 can be set by 1-bit or 8-bit memory manipulation instructions.

When $\overline{\text{RESET}}$ is input, its value is 00H.

Address: FFA0H When reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	PEX0	Operation mode	RxD0/PXX pin function	TxD0/PXX pin function
0	0	Operation stop	Port function	Port function
0	1	UART0 mode (receive only)	Serial function	Port function
1	0	UART0 mode (transmit only)	Port function	Serial function
1	1	UART0 mode (transmit and receive)	Serial function	Serial function

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmittion No parity detection during reception (parity errors do not occur)
1	0	Odd aprity
1	1	Even parity

CL0	Character length specification
0	7 bits
0	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 is issued when an error occurs
1	Receive completion interrupt is not issued when an error occurs

IRDAM0	Operation specified for infrared data transfer (IrDA0) mode[Note 1]
0	UART0 (transmit/receive) mode
1	IrDA0 (transmit/receive) mode[Note 2]

Notes 1. The UART0/IrDA0 mode specification is controlled by TXE0 and RXE0.

2. When using IrDA0 mode, be sure to set "00H" to the baud rate generator control register (BRGC0).

Caution Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

(b) Asynchronous serial interface status register (ASIS0)

ASIS0 can be read using an 8-bit memory manipulation instruction.

When $\overline{\text{RESET}}$ is input, its value is 00H.

Address: FFA1H When 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 (Incorrect parity bit detected)

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)

Notes 1. Even if a stop bit length of two bits has been set to bit 2 (SL0) in the asynchronous serial interface mode register (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 (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 (BRGC0)

BRGC0 can be set by an 8-bit memory manipulation instruction.

When $\overline{\text{RESET}}$ is input, its value is 00H.

Address: FFA2H When reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(fx = 8.38 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P25/ASCK0	1
0	0	1	fx/2	2
0	1	0	$fx/2^2$	3
0	1	1	$fx/2^3$	4
1	0	0	$fx/2^4$	5
1	0	1	$fx/2^5$	6
1	1	0	$fx/2^6$	7
1	1	1	$fx/2^7$	8

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	fsck/16	0
0	0	0	1	fsck/17	1
0	0	1	0	fsck/18	2
0	0	1	1	fsck/19	3
0	1	0	0	fsck/20	4
0	1	0	1	fsck/21	5
0	1	1	0	fsck/22	6
0	1	1	1	fsck/23	7
1	0	0	0	fsck/24	8
1	0	0	1	fsck/25	9
1	0	1	0	fsck/26	10
1	0	1	1	fsck/27	11
1	1	0	0	fsck/28	12
1	1	0	1	fsck/29	13
1	1	1	0	fsck/30	14
1	1	1	1	Setting prohibit	—

Caution 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.

Remarks 1. fsck : Source clock for 5-bit counter

2. n : Value set via TPS00 to TPS02 ($1 \leq n \leq 8$)

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.

- Use of main system clock to generate a transmit/receive clock for baud rate
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated by the main system clock is determined according to the following formula.

$$[\text{Baud rate}] = \frac{fx}{2^{n+1}(k + 16)} \text{ [Hz]}$$

fx : Oscillation frequency of main system clock
n : Value set via TPS00 to TPS02 ($1 \leq n \leq 8$)
For details, see Table 16-3.
k : Value set via MDL00 to MDL02 ($0 \leq k \leq 14$)

Table 16-3 shows the relation 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 16-3. Relation 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	1
0	0	1	fx/2	2
0	1	0	$fx/2^2$	3
0	1	1	$fx/2^3$	4
1	0	0	$fx/2^4$	5
1	0	1	$fx/2^5$	6
1	1	0	$fx/2^6$	7
1	1	1	$fx/2^7$	8

Remark fx: Oscillation frequency of main system clock.

- **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)].

  Table 16-4 describes the relation between the main system clock and the baud rate and Figure 16-5 shows an example of a baud rate error tolerance range.

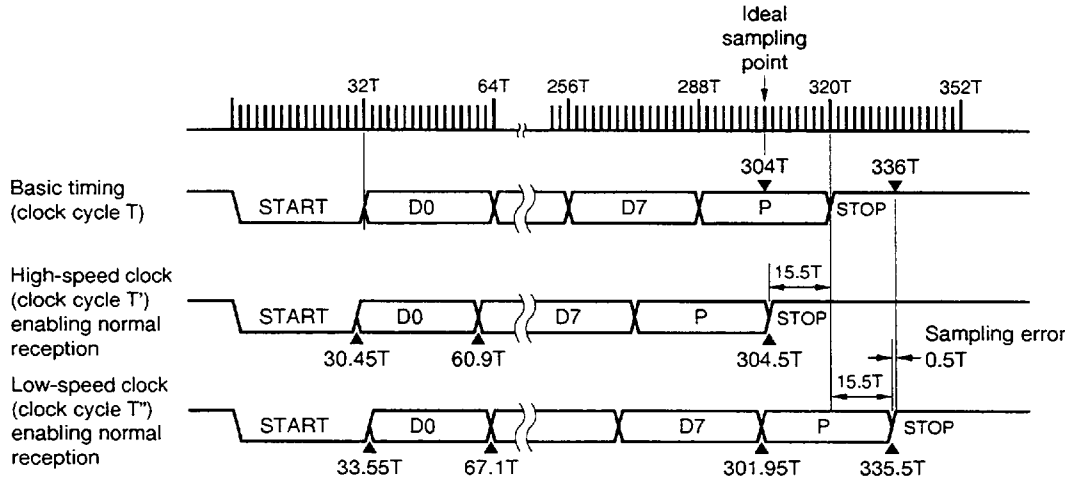
Table 16-4. Relation between Main System Clock and Baud Rate

Baud rate (bps)	fx = 8.386 MHz		fx = 8.000 MHz		fx = 7.3728 MHz		fx = 5.000 MHz		fx = 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	–	–	–	–
IrDA0 mode[Note]	65536 bps		62500 bps		115200 bps		39063 bps		32768 bps	

Note The UART0/IrDA0 mode specification is controlled by TXE0 and RXE0.
When using the IrDA0 mode, be sure to set the baud rate generator register (BRGL0) as follows.
- k = 0 (MDL0 to MDL3 = 0000)
- n = 1 (TPS00 to TPS02 = 000)

Remarks fx : Oscillation frequency of main system clock
n : Value set via TPS00 to TPS02 ($1 \leq n \leq 8$)
k : Value set via MLD00 to MLD03 ($0 \leq k \leq 14$)

Figure 16-5. Error Tolerance (when k = 0), including Sampling Errors



Remark T: 5-bit counter's source clock cycle

Baud rate error tolerance (when k = 0) = $\frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$

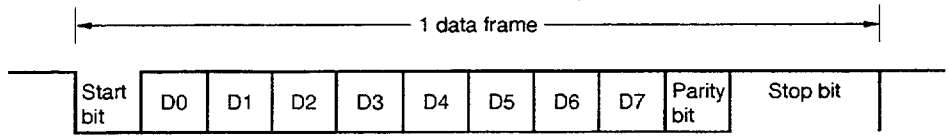
(2) Communication operations

(a) Data format

As shown in Figure 16-6, the format of the transmit/receive data consists of a start bit, character bits, a parity bit, and one or more stop bits.

The asynchronous serial interface mode register (ASIM0) is used to set the character bit length, parity selection, and stop bit length within each data frame.

Figure 16-6. Format of Transmit/Receive Data in Asynchronous Serial Interface



- 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

When "7 bits" is selected as the number of character bits, only the low-order 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 asynchronous serial interface mode register (ASIM0) and the baud rate generator control register (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 (ASIS0).

(b) Parity types and operations

The parity bit is used to detect bit errors in transfer 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 "1" bits. The value of the parity bit is as follows.

  If the transmit data contains an odd number of "1" bits : the parity bit value is "1"
  If the transmit data contains an even number of "1" bits: the parity bit value is "0"

- During reception
  The number of "1" bits is counted among the transfer data that include a parity bit, and a parity error occurs when the 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 "1" bits. The value of the parity bit is as follows.

  If the transmit data contains an odd number of "1" bits : the parity bit value is "0"
  If the transmit data contains an even number of "1" bits: the parity bit value is "1"

- During reception
  The number of "1" bits is counted among the transfer data that include a parity bit, and a parity error occurs when the 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

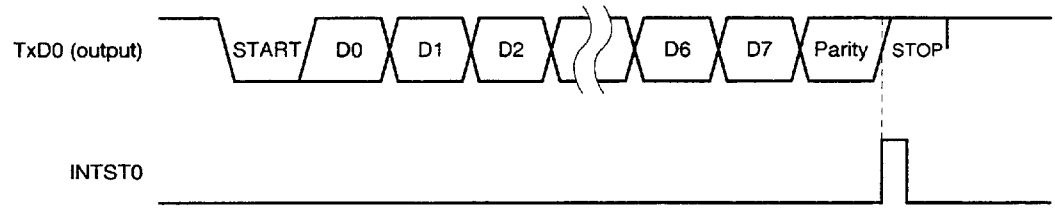
The transmit operation is started when transmit data is written to the transmit shift register (TXS0). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS0, thereby emptying TXS0, after which a transmit completion interrupt (INTST0) is issued.

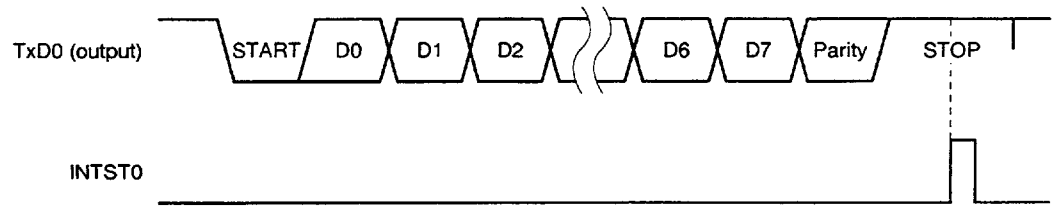
The timing of the transmit completion interrupt is shown in Figure 16-7.

Figure 16-7. Timing of Asynchronous Serial Interface Transmit Completion Interrupt

(i) Stop bit length: 1 bit



(ii) Stop bit length: 2 bits



Caution Do not write to the asynchronous serial interface mode register (ASIM0) during a transmit operation. Writing to ASIM0n during a transmit operation may disable further transmit operations (in such cases, enter 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 (INTST0) or the interrupt request flag (STIF0) that is set by INTST0.

(d) Reception

The receive operation is enabled when "1" is set to bit 6 (RXE0) of the asynchronous serial interface mode register (ASIM0), and input via the RxD0 pin is sampled.

The serial clock specified by ASIM0 is used when sampling the RxD0 pin.

When the RxD0 pin goes low, the 5-bit counter 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.

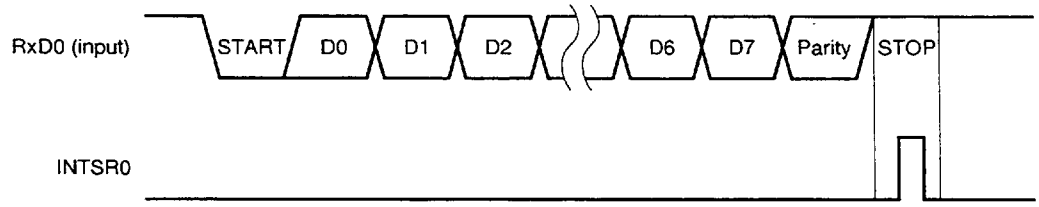
Once reception of one data frame is completed, the receive data in the shift register is transferred to the receive buffer register (RXB0) and a receive completion interrupt (INTSR0) occurs.

Even if an error has occurred, the receive data in which the error occurred is still transferred to RXB0 and INTSR0 occurs (see Figure 16-9).

If the RXE0 bit is reset (to "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 16-8 shows the timing of the asynchronous serial interface receive completion interrupt.

Figure 16-8. Timing of Asynchronous Serial Interface Receive Completion Interrupt



Caution Be sure to read the contents of the receive buffer register (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 (ASIS0), a receive error interrupt (INTSER0) will occur. Receive error interrupts are generated before receive interrupts (INTSR0). Table 16-5 lists the causes behind receive errors.

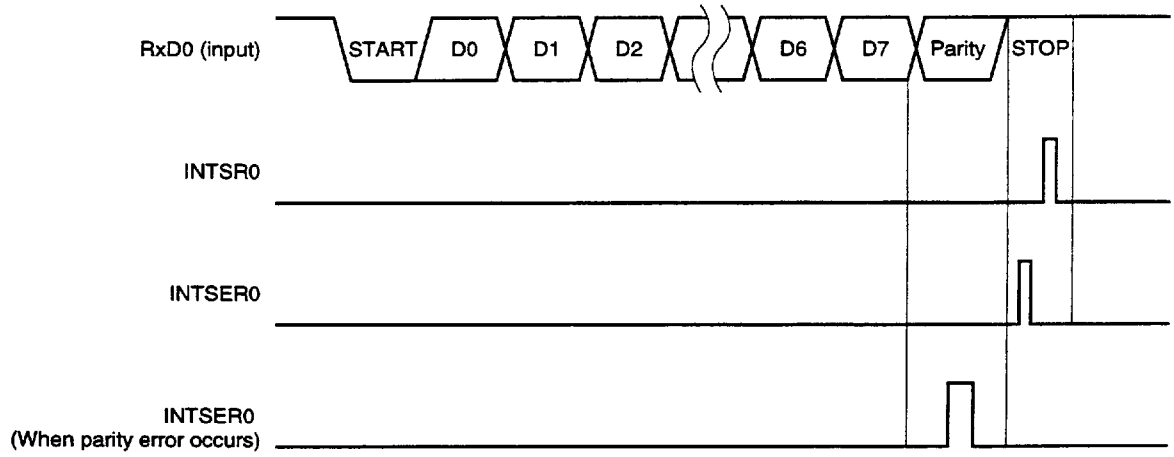
As part of receive error interrupt (INTSER0) servicing, the contents of ASIS0 can be read to determine which type of error occurred during the receive operation (see Table 16-5 and Figure 16-9).

The contents of ASIS0 are reset (to "0") when the receive buffer register (RXB0) is read or when the next data is received (if the next data contains an error, another error flag will be set).

Table 16-5. 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	01H

Figure 16-9. Receive Error Timing



Cautions 1. The contents of ASIS0 are reset (to "0") when the receive buffer register (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 (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.

16.5.3 Infrared Data Transfer (IrDA0) Mode

(1) Data format

Figure 16-10 compares the data format used in UART0 mode with that used in IrDA0 mode.

Each IR (infrared) frame includes a start bit, eight data bits, and a pulse-stopping one-bit stop bit. This corresponds to the bit stream in UART0 frames.

The length of the electrical pulses that are used to transmit and receive IR frames 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).

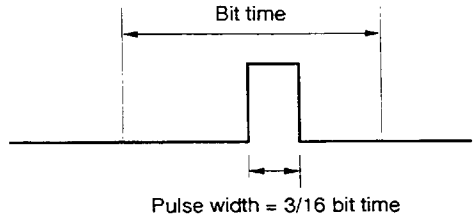
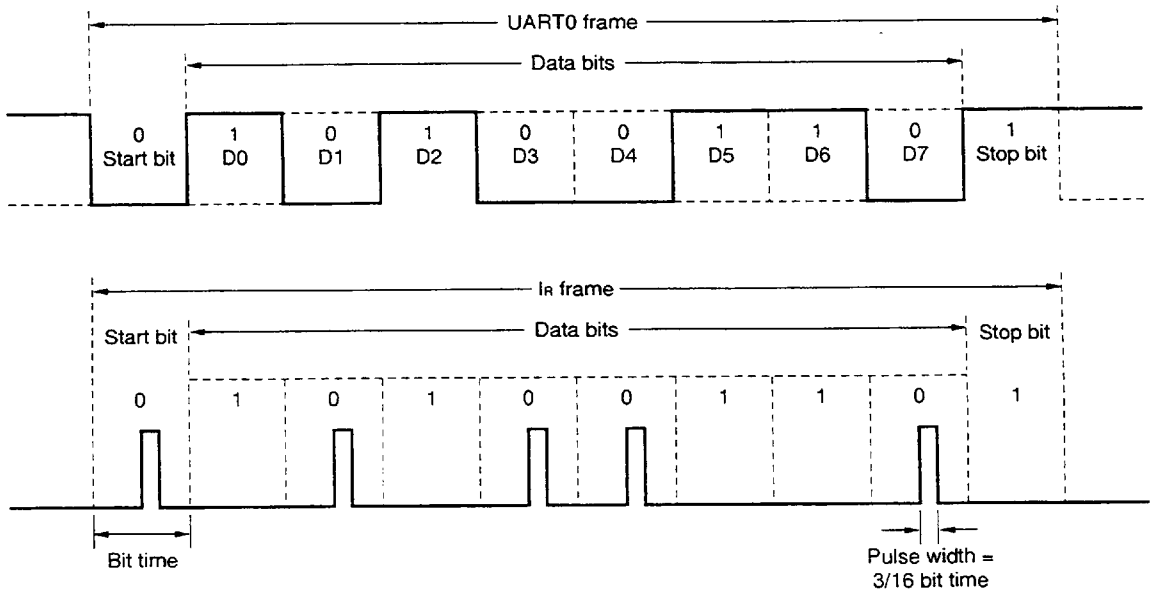


Figure 16-10. Data Format Comparison between IrDA0 Mode and UART0 Mode



(2) Bit rate and pulse width

Table 16-6 lists bit rates, bit rate error tolerances, and pulse width values.

According to the specifications, the minimum pulse width can be either 3/16 the bit time or the minimum pulse width of a 115.2-kbps signal (error tolerance: 1.63 μs to 22 (μs). The value is the same for all bit rates.

The value that is added as the maximum pulse width is the largest value among 3/16 the bit time, 2.5% of the bit time, or 1.08 μs.

Table 16-6. Bit Rate and Pulse Width Values

Bit rate (kbits/s)	Bit rate error tolerance (% of bit rate)	Pulse width minimum value (μs) Note	3/16 pulse width (nominal value)	Maximum pulse width (μs)
2.4	+/– 0.87	1.41	78.13	88.55
9.6	+/– 0.87	1.41	19.53	22.13
19.2	+/– 0.87	1.41	9.77	11.07
38.4	+/– 0.87	1.41	4.88	5.96
57.6	+/– 0.87	1.41	3.26	4.34
115.2	+/– 0.87	1.41	1.63	2.71

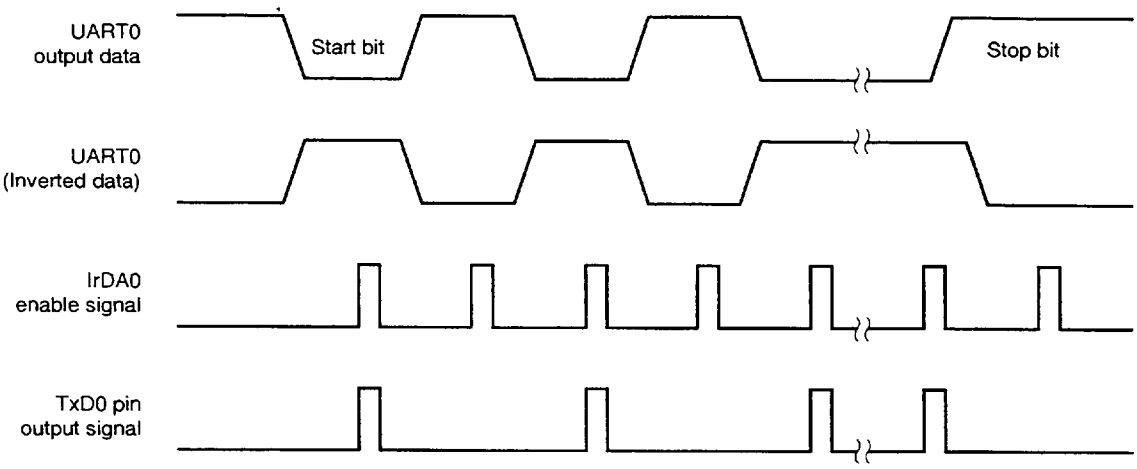
Note When a digital noise elimination circuit is used in a microcontroller operating at 1.41 MHz or above.

The following is an example of the formula for calculating the maximum pulse width when the bit rate is 2.4 kbps.

$$78.13 + (78.13 \times \frac{16}{3} \times 0.025) = 88.55\ [\mu s]$$

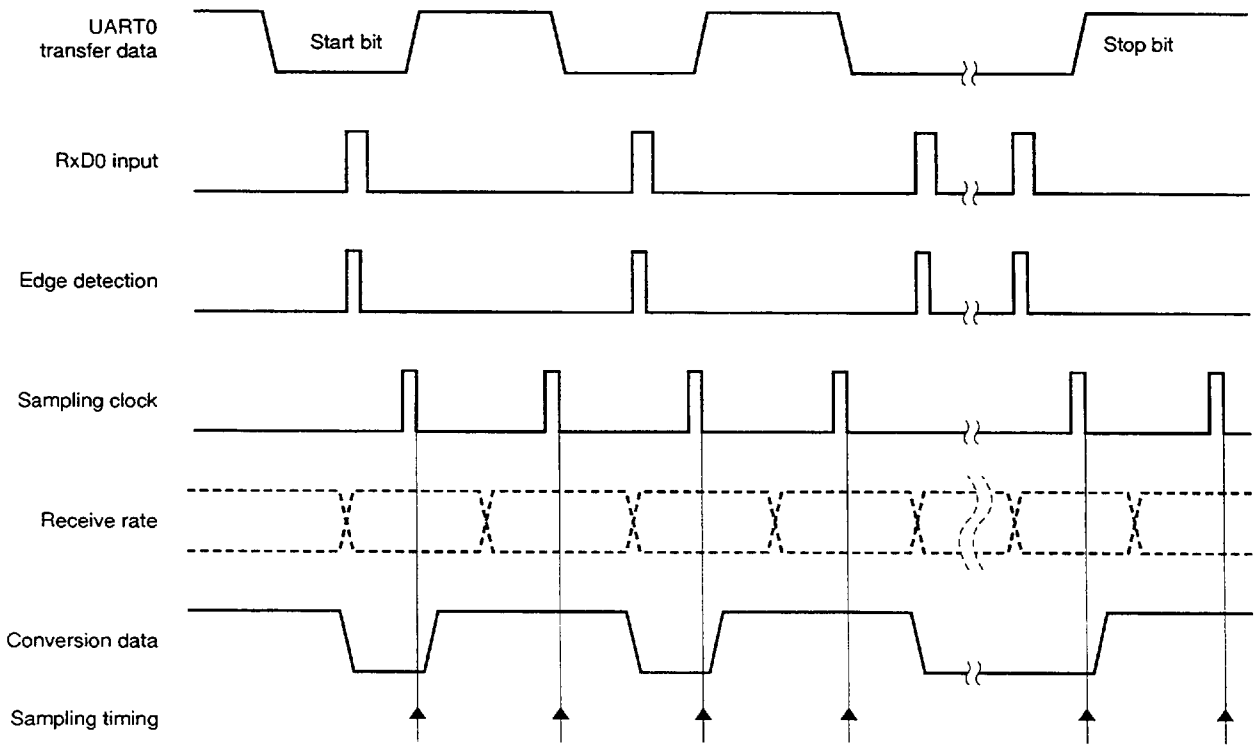
(3) I/O data and internal signals

- **Transmit operation timing**



- **Receive operation timing**

  Data reception is delayed for one-half of the specified baud rate.



16.6 STANDBY FUNCTION

Serial transfer operations can be performed during HALT mode.

During STOP mode, serial transfer operations are stopped and the values in the asynchronous serial interface mode register (ASIM0), transmit shift register (TXS0), receive shift register (RX0), and receive buffer register (RXB0) remain as they were just before the clock was stopped.

Output from the TxD0 pin retains the immediately previous data if the clock is stopped (if the system enters STOP mode) during a transmit operation. If the clock is stopped during a receive operation, the data received before the clock was stopped is retained and all subsequent operations are stopped. The receive operation can be restarted once the clock is restarted.

CHAPTER 17 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.

Therefore, unless otherwise specified, the SIO30 is used throughout this chapter to describe the functions of both the SIO30 and SIO31. If using the SIO31, refer to Table 17-1 for the register, bit, and pin names.

Caution The μPD780024Y and 780034Y Subseries do not have the SIO31.

Table 17-1 SIO30 and SIO31 Naming Differences

Item	SIO30	SIO31
Pins	P20/SI30 P21/SO30 P22/$\overline{SCK30}$	P34/SI31 P35/SO31 P36/$\overline{SCK31}$
Serial operation mode register 3	CSIM30	SCIM31
Serial operation mode register 3 address	FFB0H	FFB8H
Serial operation mode register 3 bits	CSIE30 MODE30 SCL301 SCL300	CSIE31 MODE31 SCL311 SCL310
Serial I/O shift register 3	SIO30	SIO31
Serial I/O shift register 3 address	FF1AH	FF1BH
Interrupt request	INTCSI30	INTCSI31
Interrupt control register adn bits mentioned in this chapter	CSIIF30 CSIMK30 CSIPR30	CSIIF31 CSIMK31 CSIPR31

17.1 FUNCTIONS

The SIO3 serial interface is a CoreMicro macro.
The SIO3 has the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **17.5.1 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 ($\overline{SCK30}$), serial output line (SO30), and serial input line (SI30).

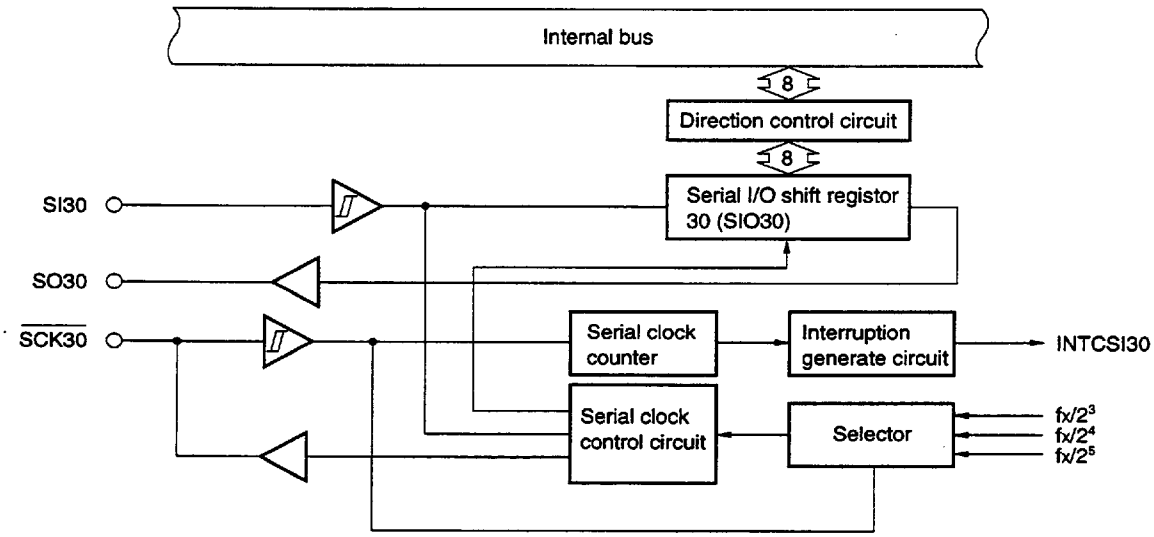
Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit in the 8-bit data in serial transfers is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O device that includes a clock-synchronous serial interface, or a display controller, etc. For details see **17.5.2 Three-Wire Serial I/O Mode.**

Figure 1-1 shows a block diagram of the SIO30 macro.

Figure 17-1. Block Diagram of SIO3 Macro



17.2 CONFIGURATION

The SIO30 includes the following hardware.

Table 17-2. Composition of SIO30

Item	Configuration
Registers	Serial I/O shift register 30 (SIO30)
Control registers	Serial operation mode register 30 (CSIM30)

(1) Serial I/O shift register 30 (SIO30)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO30 is set by an 8-bit memory manipulation instruction.

When "1" is set to bit 7 (CSIE30) of the serial operation mode register 30 (CSIM30), a serial operation can be started by writing data to or reading data from SIO30.

When transmitting, data written to SIO30 is output via the serial output (SO30).

When receiving, data is read from the serial input (SI30) and written to SIO30.

The $\overline{\text{RESET}}$ signal resets the register value to 00H.

Caution Do not access SIO30 durig a transmit operation unless the access is triggered by a transfer start. (Read is disabled when MODE = 0 and write is disabled when MODE = 1.)

17.3 LIST OF SFRS (SPECIAL FUNCTION REGISTERS)

Table 17-3. List of SFRs (Special Function Registers)

SFR name	Symbol	R/W	Units available for bit manipulation			Value when reset
			1 bit	8 bits	16 bits	
Serial operation mode register 30	CSIM30	R/W	○	○	—	00H
Serial I/O shift register 30	SIO30		—	○	—	

17.4 SERIAL INTERFACE CONTROL REGISTERS

The SIO3 uses the following type of register for control functions.

- Serial operation mode register 30 (CSIM30)

(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 can be set via a 1-bit or 8-bit memory manipulation instruction.
The $\overline{\text{RESET}}$ input sets the value to 00H.

Figure 17-2. Format of Serial Operation Mode Register 30 (CSIM30)

Address: FFB0H When 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[Note]
0	Operation stop	Clear	Port function
1	Operation enable	Count operation enable	Serial function + port function

MODE0	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO30 output
0	Transmit/receive mode	Write to SIO30	Normal output
1	Receive-only mode	Read from SIO30	Fixed at low level

SCL301	SCL300	Clock selection (f_X = 8.38 MHz)
0	0	External clock input
0	1	$f_X/2^3$
1	0	$f_X/2^4$
1	1	$f_X/2^5$

Note When CSIE30 = 0 (SIO30 operation stop status), the pins connected to SI30 and SO30 can be used for port functions.

17.5 OPERATIONS

This section explains on two modes of SIO3.

17.5.1 Operation Stop Mode

This mode if used when serial transfers are not performed to reduce power consumption.
When in operation stop mode, the pins can be used as normal I/O ports as well.

(1) Register settings

Operation stop mode are set via serial operation mode register 30 (CSIM30).
CSIM30 can be set via 1-bit or 8-bit memory manipulation instructions.
The $\overline{\text{RESET}}$ input sets the value to 00H.

Figure 17-3. Format of Serial Operation Mode Register 30 (CSIM30)

Address: FFB0H When reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	SIO30 operation enable/disable specification		
	Shift register operation	Serial counter	Port[Note]
0	Operation stop	Clear	Port function
1	Operation enable	Count operation enable	Serial function + port function

Note When CSIE30 = 0 (SIO30 operation stop status), the pins connected to SI30 and SO30 can be used for port functions.

17.5.2 Three-Wire Serial I/O Mode

The three-wire serial I/O mode is useful when connecting a peripheral I/O device that includes a clock-synchronous serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{SCK30}$), serial output line (SO30), and serial input line (SI30).

(1) Register settings

3-wire serial I/O mode is set via serial operation mode register 30 (CSIM30).

CSIM30 can be set via 1-bit or 8-bit memory manipulation instructions.

The $\overline{RESET}$ input set the value to 00H .

Figure 17-4. Format of Serial Operation Mode Register 30 (CSIM30)

Address: FFB0H When 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[Note]
0	Operation stop	Clear	Port function
1	Operation enable	Count operation enable	Serial function + port function

MODE0	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO30 output
0	Transmit/receive mode	Write to SIO30	Normal output
1	Receive-only mode	Read from SIO30	Fixed at low level

SCL301	SCL300	Clock selection (fx = 8.38 MHz)
0	0	External clock input
0	1	$fx/2^3$
1	0	$fx/2^4$
1	1	$fx/2^5$

Note When CSIE30 = 0 (SIO30 operation stop status), the pins connected to SI30 and SO30 can be used for port functions.

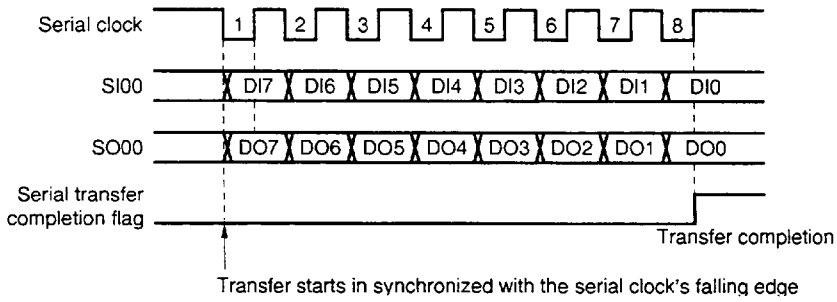
(2) Communication Operations

In the three-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is sent or received in synchronized with the serial clock.

The serial I/O shift register 30 (SIO30) is shifted in synchronized with the falling edge of the serial clock. Transmission data is held in the SO30 latch and is output from the SO30 pin. Data that is received via the SI30 pin in synchronized with the rising edge of the serial clock is latched to SIO30.

Completion of an 8-bit transfer automatically stops operation of SIO30 and sets a serial transfer completion flag.

Figure 17-5. Timing of Three-wire Serial I/O Mode



(3) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set to serial I/O shift register 30 (SIO30).

- The SIO30 operation control bit (CSIE30) = 1
- After an 8-bit serial transfer, the internal serial clock is either stopped or is set to high level.
- Transmit/receive mode
  When CSIE30 = 1 and MODE0 = 0, transfer starts when writing to SIO30.
- Receive-only mode
  When CSIE30 = 1 and MODE0 = 1, transfer starts when reading from SIO30.

Caution After data has been written to SIO30, transfer will not start even if the CSIE30 bit value is set to "1".

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets a serial transfer completion flag.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 18 SERIAL INTERFACE IIC0 (μPD780024Y, 780034Y SUBSERIES ONLY)

18.1 FUNCTIONS

The IIC0 serial interface has the following two modes.

- Operation stop mode
- I^2C (Inter IC) bus (multi-master support)

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multi-master support)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I^2C bus format and can output "start condition", "data", and "stop condition" data segments when transmitting via the serial data bus. These data segments are automatically detected by hardware during reception.

Since SCL0 and SDA0 are open-drain outputs, the IIC0 requires pull-up resistors for the serial clock line (SCL0) and the serial data bus line (SDA0).

Figure 18-1 shows a block diagram of IIC0.

Figure 18-1. Block Diagram of IIC0

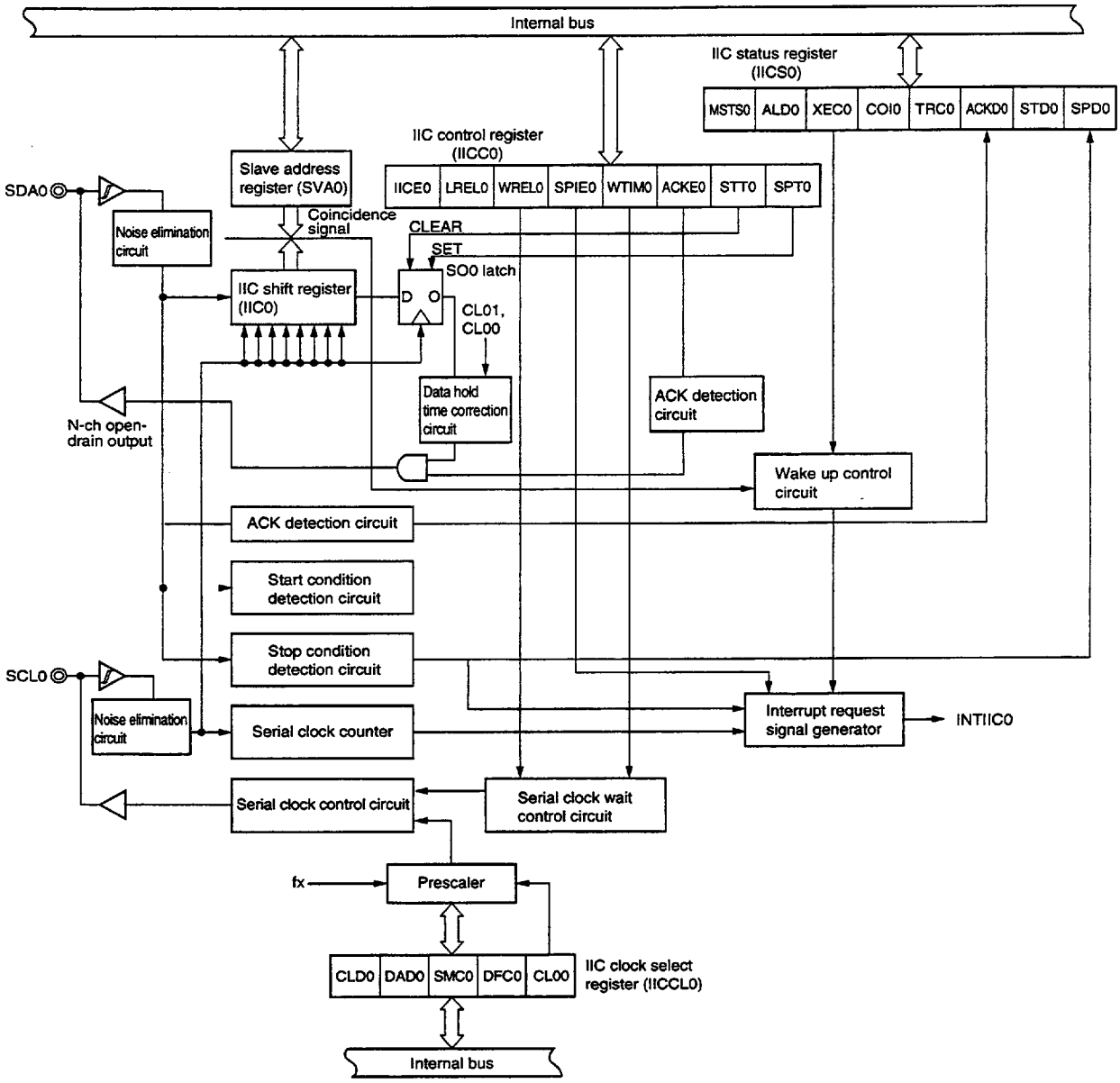
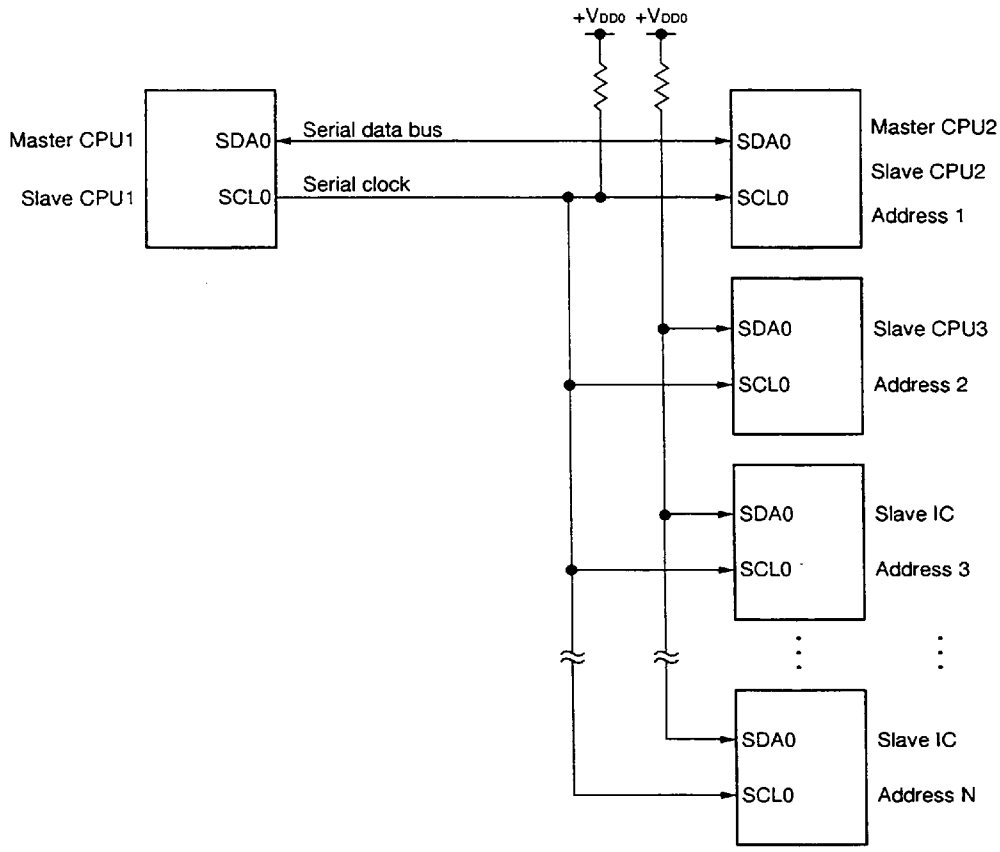


Figure 18-2 shows a serial bus configuration example.

Figure 18-2. Serial Bus Configuration Example Using I²C Bus



18.2 CONFIGURATION

The IIC0 includes the following hardware.

Table 18-1. Configuration of IIC0

Item	Configuration
Registers	IIC shift register (IIC0) Slave address register (SVA0)
Control registers	IIC control register (IICC0) IIC status register (IICS0) IIC clock select register (IICCL0)

(1) IIC shift register (IIC0)

IIC0 is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data.
IIC0 can be used for both transmission and reception.
Write and read operations to IIC0 are used to control the actual transmit and receive operations.
IIC0 is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value to 00H.

(2) Slave address register (SVA0)

This register sets local addresses when in slave mode.
SVA0 is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value to 00H.

(3) SO0 latch

The SO0 latch is used to retain the SDA0 pin's output level.

(4) Wake-up control circuit

This circuit generates an interrupt request when the address received by this register matches the address value set to the slave address register (SVA0) or when an extension code is received.

(5) Clock selector

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent and/or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC0).
An I^2C interrupt request is generated following either of two triggers.

- Eighth or ninth clock of the serial clock (set by WTIM0 bit) **(Note)**
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit) **(Note)**

Note WTIM0 bit : bit 3 of the IIC control register (IICC0)
SPIE0 bit : bit 4 of the IIC control register (IICC0)

(8) Serial clock control circuit

During master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

(9) Serial clock wait control circuit

This circuit controls the wait timing.

(10) ACK output circuit, stop condition detection circuit, start condition detection circuit, and ACK detection circuit

These circuits are used to output and detect various control signals.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

18.3 LIST OF SFRS (SPECIAL FUNCTION REGISTERS)

Table 18-2. List of SFRs (Special Function Registers)

SFR name	Symbol	R/W	Units available for bit manipulation			Value when reset
			1 bit	8 bits	16 bits	
Slave address register	SVA0	R/W	—	O	—	00H
IIC shift register	IIC0		—	O	—	
IIC control register	IICC0		—	O	—	
IIC status register	IICS0	R	O	O	—	
IIC clock select register	IICCL0	R/W	O	O	—	

18.4 SERIAL INTERFACE CONTROL REGISTER

IIC0 is controlled via three types of registers.

- IIC control register (IICC0)
- IIC status register (IICS0)
- IIC clock select register (IICCL0)

The following registers are also used.

- IIC shift register (IIC0)
- Slave address register (SVA0)

(1) IIC control register (IICC0)

This register is used to enable/disable I^2C operations, set wait timing, and set other I^2C operations.

IICC0 can be set by 1-bit or 8-bit memory manipulation instructions.

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

Figure 18-3. Format of IIC Control Register (IICC0) (1/3)

Address: FFA8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stops operation. Presets expansion register (IICS0). Stops internal operation.
1	Enables operation.

Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

LREL0	Exit from communications
0	Normal operation
1	This exits from the current communications operation and sets standby mode. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set for high impedance. The following flags are cleared. • STD0 • ACK0 • TRC0 • COI0 • EXC0 • MSTS0

The standby mode following exit from communications remains in effect until the following communications entry conditions are met.
- After a stop condition is detected, restart is in master mode.
- An address match or extension code receptions occurs after the start condition.

Condition for clearing (LREL0 = 0) (Note)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

WREL0	Cancel wait
0	Does not cancel wait
1	Cancels wait. This setting is automatically cleared after wait is canceled.

Condition for clearing (WREL0 = 0) (Note)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

SPIE0	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable

Condition for clearing (SPIE0 = 0) (Note)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICE0 = 0.

Figure 18-3. Format of IIC Control Register (IICC0) (2/3)

WTIM0	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode : After output of eight clocks, clock output is set to low level and wait is set. Slave mode : After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode : After output of nine clocks, clock output is set to low level and wait is set. Slave mode : After input of nine clocks, the clock is set to low level and wait is set for master device.

This bit's setting is invalid during an address transfer and is valid as the transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an ACK signal is issued. When the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.

Condition for clearing (WTIM0 = 0) (Note)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

ACKE0	Acknowledge control
0	Disable acknowledge.
1	Enable acknowledge. During the ninth clock period, the SDA0 line is-set to low level. However, the ACK is invalid during address transfers and is valid when EXC0 = 1.

Condition for clearing (ACKE0 = 0) (Note)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

STT0	Start condition trigger
0	Does not generate a start condition.ce.
1	When bus is released (during STOP mode): Generates a start condition (for starting as master). The SDA0 line is changed from high level to low level and then the start condition is generated. Next, after the rated amount of time has elapsed, SCL0 is changed to low level. When bus is not used: This trigger functions as a start condition reserve flag. When set, it releases the bus and then automatically generates a start condition.

Cautions concerning set timing
- For master reception : Cannot be set during transfer. Can be set only when ACKE0n has been set to 0 and slave has been notified of final reception.
- For master transmission : Note that a start condition cannot be generated normally during the ACK period.
- Cannot be set at the same time as SPT0

Condition for clearing (STT0 = 0) (Note)	Condition for setting (STT0 = 1)
• Cleared by instruction • Cleared by loss in arbitration • Cleared after start condition is generated by master device • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICE0 = 0.

Figure 18-3. Format of IIC Control Register (IICC0) (3/3)

SPT0	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer). After the SDA0 line goes to low level, either set the SCL0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA0 line is changed from low level to high level and a stop condition is generated.

Cautions concerning set timing

- For master reception : Cannot be set during transfer.
  Can be set only when ACKE0 has been set to 0
  and slave has been notified of final reception.
- For master transmission: Note that a stop condition cannot be generated normally during the ACK0 period.
- Cannot be set at the same time as STT0.
- SPT0 can be set only when in master mode. (Note 1)
- When WTIM0 has been set to 0, if SPT0 is set during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high level period of the ninth clock.
  When a ninth clock must be output, WTIM0 should be changed from 0 to 1 during the wait period following output of eight clocks, and SPT0 should be set during the wait period that follows output of the ninth clock.

Condition for clearing (SPT0 = 0) (Note 2)	Condition for setting (SPT0 = 1)
• Cleared by instruction • Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared when $\overline{\text{RESET}}$ is input	• Set by instruction

Notes 1. Set SPT0 only during master mode. However, you must set SPT0 and generate a stop condition before the first stop condition is detected following the switch to operation enable status. For details, see **"18.6.15 Other cautions"**.

2. This flag's signal is invalid when IICE0 = 0.

Caution When bit 3 (TRC0) of the IIC status register (IICS0) is set to "1", WREL0 is set during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set for high impedance.

Remarks STD0 : Bit 1 of IIC status register (IICS0)
ACKD0 : Bit 2 of IIC status register (IICS0)
TRC0 : Bit 3 of IIC status register (IICS0)
C0I0 : Bit 4 of IIC status register (IICS0)
EXC0 : Bit 5 of IIC status register (IICS0)
MSTS0 : Bit 7 of IIC status register (IICS0)

(2) IIC status register (IICS0)

This register indicates the status of the I^2C bus.

IICS0 can be set by 1-bit or 8-bit memory manipulation instructions. IICS0n is a read-only register.

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

Figure 18-4. Format of IIC Status Register (IICS0) (1/3)

Address: FFA9H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
IICS0	MSTS0	ALD0	EXC0	C0I0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a "win".
1	This status indicates the arbitration result was a "loss". MSTS0 is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after IICS0 is read (Note) • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the arbitration result is a "loss".

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the high-order four bits of the received address data is either "0000" or "1111" (set at the rising edge of the eighth clock).

Note This register is also cleared when a bit manipulation instruction is executed for bits other than IICS0.

Figure 18-4. Format of IIC Status Register (IICS0) (2/3)

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the received address matches the local address (SVA0) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the rising edge of the first byte's ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
• When a start condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • Cleared by WREL0 = 1 • When ALD0 changes from 0 to 1 • When $\overline{\text{RESET}}$ is input Master • When "1" is output to the first byte's LSB (transfer direction specification bit) Slave • When a start condition is detected • When not used for communication	Master • When a start condition is generated Slave • When "1" is input by the first byte's LSB (transfer direction specification bit)

ACKD0	Detection of ACK
0	ACK was not detected.
1	ACK was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• After the SDA0 line is set to low level at the rising edge of the SCL0's ninth clock

Figure 18-4. Format of IIC Status Register (IICS0) (3/3)

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect

Condition for clearing (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD0 = 0)	Condition for setting (SPD0 = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a stop condition is detected

Remarks LREL0: Bit 6 of IIC control register (IICC0)
IICE0 : Bit 7 of IIC control register (IICC0)

(3) IIC clock select register (IICCL0)

This register is used to set the transfer clock for the I^2C bus.

IICCL0 can be set by 1-bit or 8-bit memory manipulation instructions.

RESET input sets the value to 00H.

Figure 18-5. Format of IIC Clock Select Register (IICCL0) (1/2)

Address: FFAAH After reset: 00H R/W (Note)

Symbol	7	6	5	4	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	0	CL00

CLD0	Detection of SCL0 line level (valid only when IICE0 = 1)
0	SCL0 line was detected at low level.
1	SCL0 line was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCL0 line is at low level • When IICE0 = 0 • When $\overline{\text{RESET}}$ is input	• When the SCL0 line is at high level

DAD0	Detection of SDA0 line level (valid only when IICE0 = 1)
0	SDA0 line was detected at low level.
1	SDA0 line was detected at high level.

Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDA0 line is at low level • When IICE0 = 0 • When $\overline{\text{RESET}}$ is input	• When the SDA0 line is at high level

SMC0	Operation mode switching
0	Operation in standard mode
1	Operation in high-speed mode

Condition for clearing (SMC0 = 0)	Condition for setting (SMC0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note Bits 4 and 5 are read-only bits.

Figure 18-5. Format of IIC Clock Select Register (IICCL0) (2/2)

DFC0	Control of digital filter operation (Note)
0	Digital filter ON
1	Digital filter OFF

CL00	Selected clock	Selected clock speed	
		Standard mode	High-speed mode
0	fx	2 MHz ≤ fx ≤ 4.19 MHz	4 MHz ≤ fx ≤ 8.38 MHz
1	fx	4.19 MHz ≤ fx ≤ 8.38 MHz	4 MHz ≤ fx ≤ 8.38 MHz

Note The digital filter can be used when in high-speed mode. Response time is slower when the digital filter is used.

Remark IICE0: Bit 7 of IIC control register (IICC0)

(4) I^2C shift register (IIC0)

This register is used for serial transmission/reception (shift operations) that are synchronized with the serial clock. It can be read from or written to in 8-bit units, but data should not be written to IIC0 during a data transfer.

Address: FF1FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICC0								

(5) Slave address register (SVA0)

This register holds the I^2C's slave addresses.
It can be read from or written to in 8-bit units, but bit 0 should be fixed as "0".

Address: FFABH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0								0

18.5 I²C BUS MODE FUNCTIONS

18.5.1 Pin Configuration

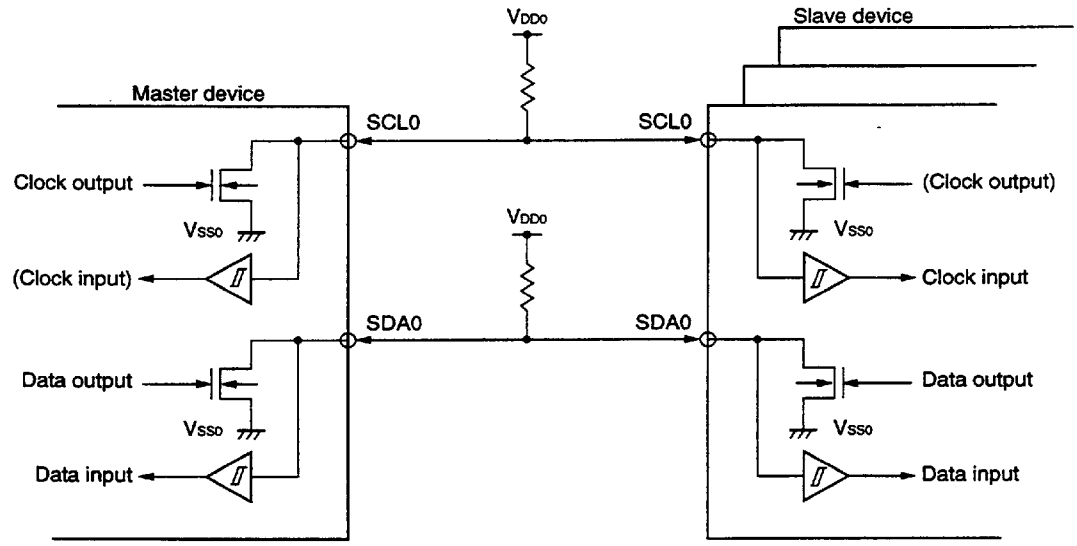
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0 ..... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDA0 ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open drain outputs, an external pull-up resistor is required.

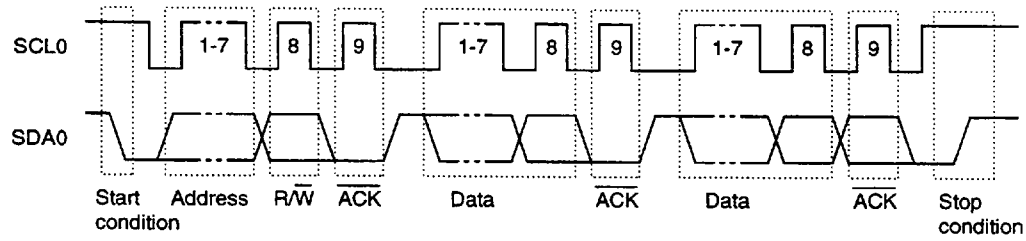
Figure 18-6. Pin Configuration Diagram



18.6 I²C BUS DEFINITIONS AND CONTROL METHODS

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. Figure 18-7 shows the transfer timing for the "start condition", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 18-7. I²C Bus's Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

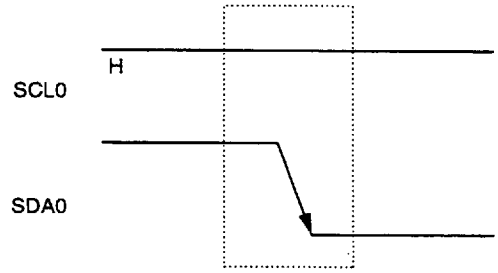
The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's low level period can be extended and a wait can be inserted.

18.6.1 Start Conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device outputs to the slave device when starting a serial transfer. The slave device includes hardware for detecting start conditions.

Figure 18-8. Start Conditions



A start condition is output when the IIC control register (IICC0)'s bit 1 (STT0) is set (to "1") after a stop condition has been detected (SPD0: Bit 0 = 1 in the IIC status register (IICS0)). When a start condition is detected, IICS0's bit 1 (STD0) is set (to "1").

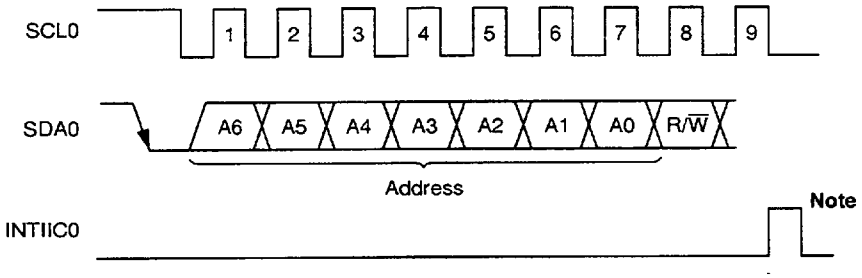
18.6.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition.

Figure 18-9. Address



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

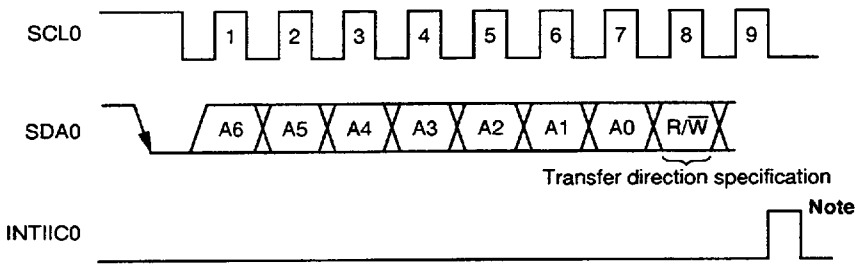
The slave address and the eighth bit, which specifies the transfer direction as described in **"18.6.3 Transfer Direction Specification"** below, are together written to the IIC shift register (IIC0) and are then output. Received addresses are written to IIC0.

The slave address is assigned to the high-order 7 bits of IIC0.

18.6.3 Transfer Direction Specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of "0", it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of "1", it indicates that the master device is receiving data from a slave device.

Figure 18-10. Transfer Direction Specification



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

18.6.6 Wait Signal ($\overline{\text{WAIT}}$)

The wait signal ($\overline{\text{WAIT}}$) is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 18-13. Wait Signal (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)

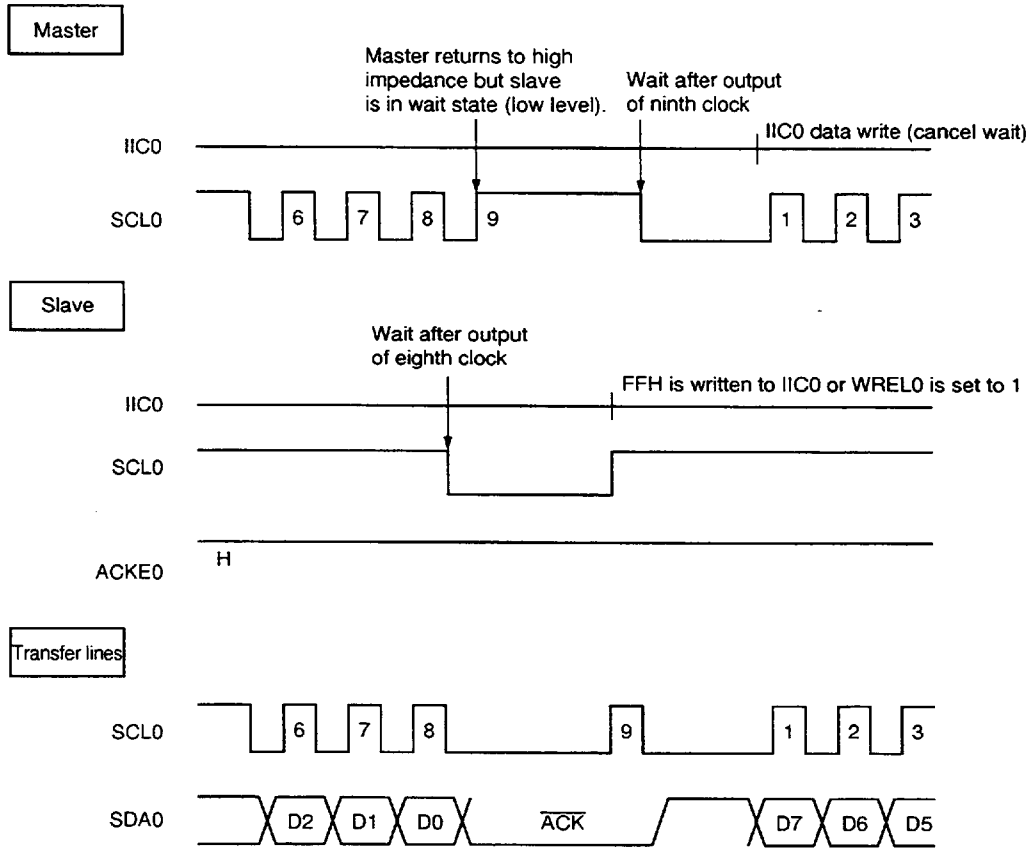
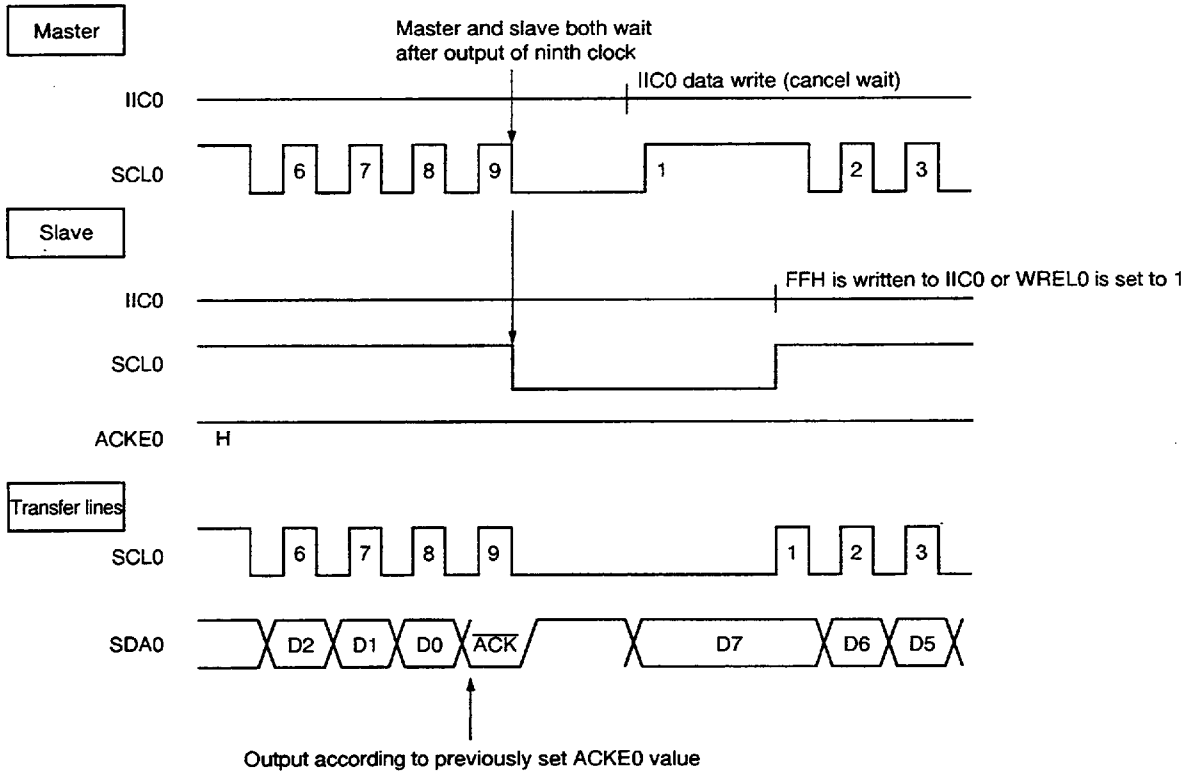


Figure 18-13. Wait Signal (2/2)

(2) When master and slave devices both have a nine-clock wait (master device transmits, slave receives, and ACKE0 = 1)



Remarks ACKE0 : Bit 2 of IIC control register (IICC0)
WREL0 : Bit 5 of IIC control register (IICC0)

A wait may be automatically generated depending on the setting for bit 3 (WTIM0) of the IIC control register (IICC0).

Normally, when bit 5 (WREL0) of IICC0n is set to "1" or when FFH is written to the IIC shift register (IIC0), the wait status is canceled and the transmitting side write data to IIC0 to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting bit 1 (STT0) of IICC0 to "1"
- By setting bit 0 (SPT0) of IICC0 to "1"

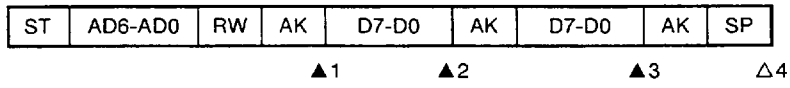
18.6.7 I²C Interrupt Requests (INTIIC0)

The INTIIC0 interrupt request timing and the IIC status register (IICS0) settings corresponding to that timing are described below.

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

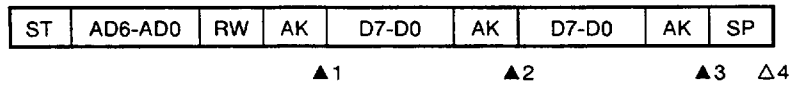
(i) When WTIM0 = 0



▲1 : IICS0 = 10×××110B
▲2 : IICS0 = 10×××000B
▲3 : IICS0 = 10×××000B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1

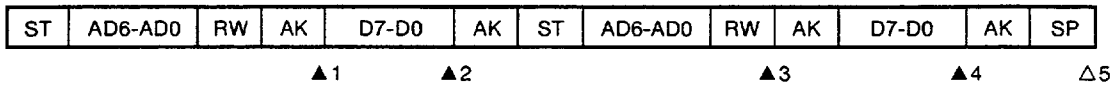


▲1 : IICS0 = 10×××110B
▲2 : IICS0 = 10×××100B
▲3 : IICS0 = 10××××00B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

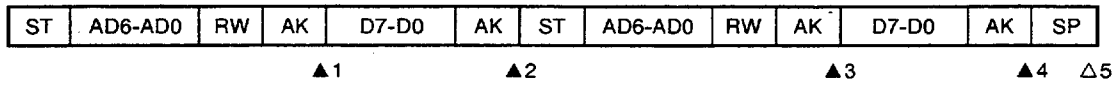
(i) When WTIM0 = 0



▲1 : IICS0 = 10×××110B
▲2 : IICS0 = 10×××000B
▲3 : IICS0 = 10×××110B
▲4 : IICS0 = 10×××000B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care

(ii) When WTIM0 = 1

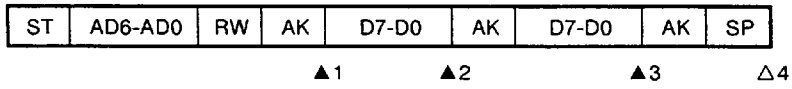


▲1 : IICS0 = 10×××110B
▲2 : IICS0 = 10×××000B
▲3 : IICS0 = 10×××110B
▲4 : IICS0 = 10××××00B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

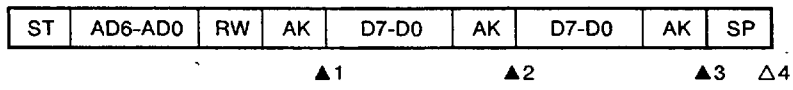
(i) When WTIM0n = 0



▲1 : IICS0 = 1010×110B
▲2 : IICS0 = 1010×000B
▲3 : IICS0 = 1010×000B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care

(ii) When WTIM0n = 1



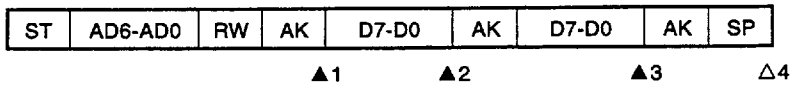
▲1 : IICS0 = 1010×110B
▲2 : IICS0 = 1010×100B
▲3 : IICS0 = 1010××00B
△4 : IICS0 = 00001001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care

(2) Slave device operation (Slave address data reception time (matches with SVA0))

(a) Start ~ Address ~ Data ~ Data ~ Stop

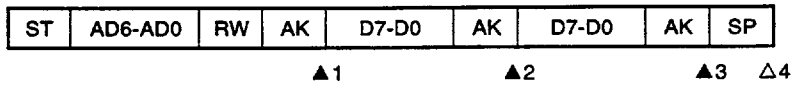
(i) When WTIM0 = 0



▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001×000B
▲3 : IICS0 = 0001×000B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1

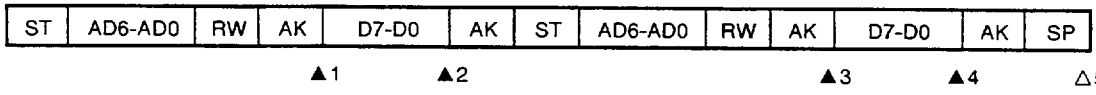


▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001×100B
▲3 : IICS0 = 0001××00B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

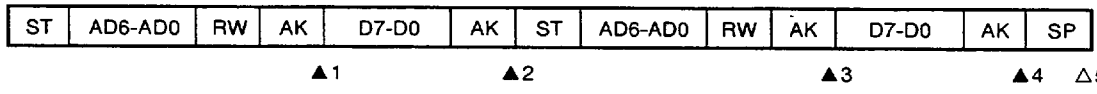
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001×000B
▲3 : IICS0 = 0001×110B
▲4 : IICS0 = 0001×000B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

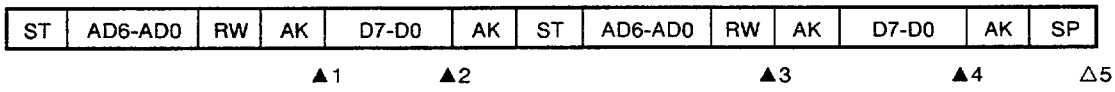


▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001××00B
▲3 : IICS0 = 0001×110B
▲4 : IICS0 = 0001××00B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

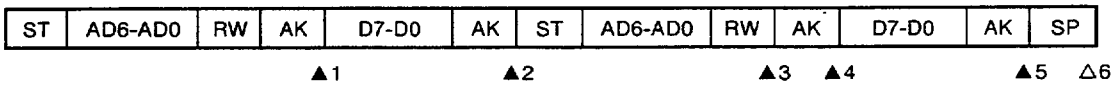
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001×000B
▲3 : IICS0 = 0010×010B
▲4 : IICS0 = 0010×000B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

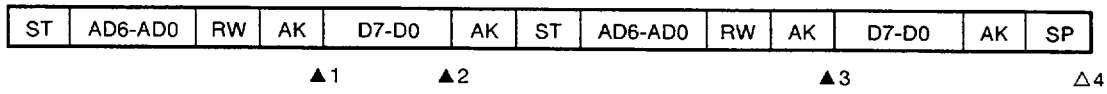


▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001××00B
▲3 : IICS0 = 0010×010B
▲4 : IICS0 = 0010×110B
▲5 : IICS0 = 0010××00B
△6 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

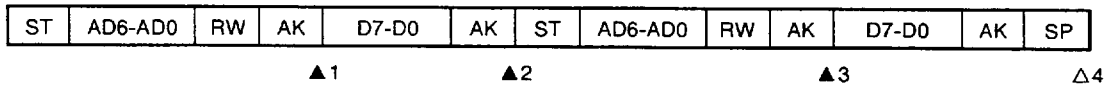
(i) When WTIM0 = 0 (after restart, does not match with SVA0n (= not extension code))



▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001×000B
▲3 : IICS0 = 0000××10B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1 (after restart, does not match with SVA0n (= not extension code))



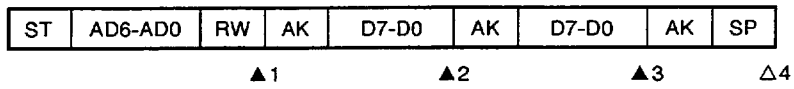
▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001××00B
▲3 : IICS0 = 0000××10B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(3) Slave device operation (when receiving extension code)

(a) Start ~ Code ~ Data ~ Data ~ Stop

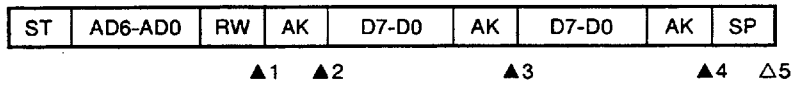
(i) When WTIM0 = 0



▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×000B
▲3 : IICS0 = 0010×000B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1

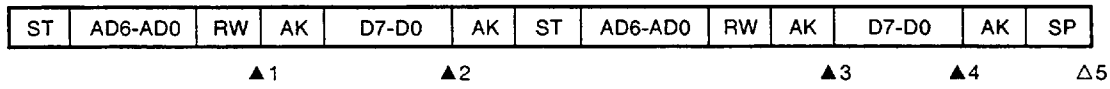


▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×110B
▲3 : IICS0 = 0010×100B
▲4 : IICS0 = 0010××00B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

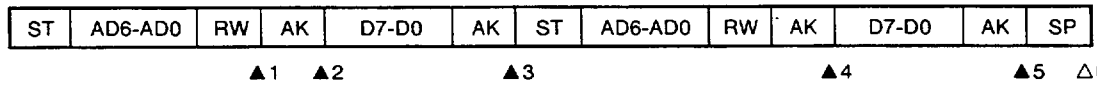
(i) When WTIM0 = 0 (after restart, matches with SVA0n)



▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×000B
▲3 : IICS0 = 0001×110B
▲4 : IICS0 = 0001×000B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

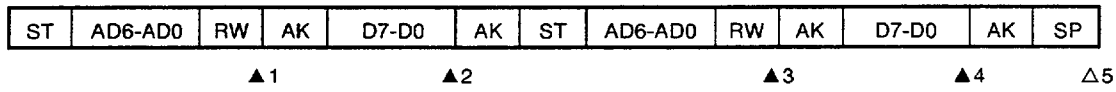


▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×110B
▲3 : IICS0 = 0010××00B
▲4 : IICS0 = 0001×110B
▲5 : IICS0 = 0001××00B
△6 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

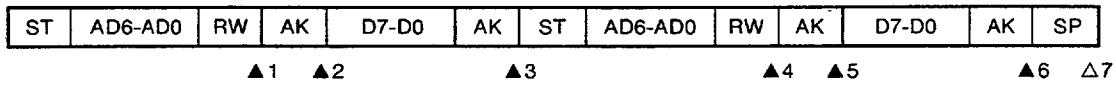
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×000B
▲3 : IICS0 = 0010×010B
▲4 : IICS0 = 0010×000B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

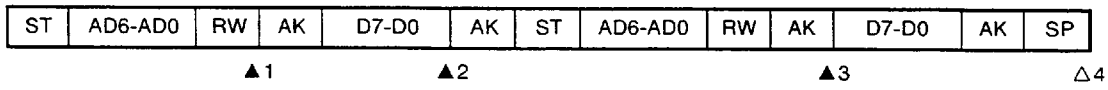


▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×110B
▲3 : IICS0 = 0010××00B
▲4 : IICS0 = 0010×010B
▲5 : IICS0 = 0010×110B
▲6 : IICS0 = 0010××00B
△7 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

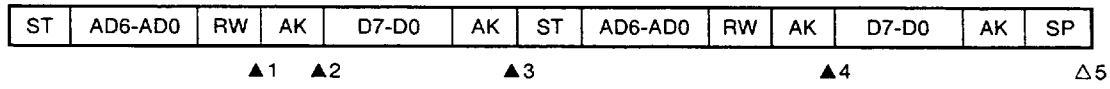
(i) When WTIM0 = 0 (after restart, does not match with address (= not extension code))



▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×000B
▲3 : IICS0 = 00000×10B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1 (after restart, does not match with address (= not extension code))

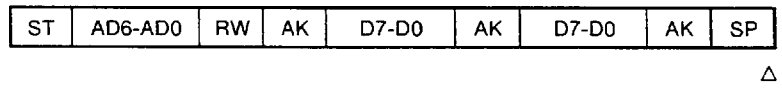


▲1 : IICS0 = 0010×010B
▲2 : IICS0 = 0010×110B
▲3 : IICS0 = 0010××00B
▲4 : IICS0 = 00000×10B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



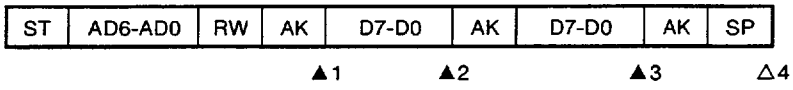
△1 : IICS0 = 00000001B

Remarks △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

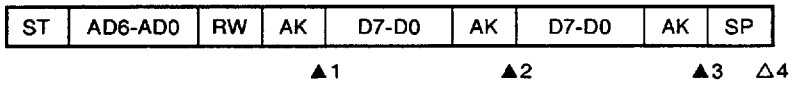
(i) When WTIM0 = 0



▲1 : IICS0 = 0101×110B (Example: when ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0001×000B
▲3 : IICS0 = 0001×000B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1

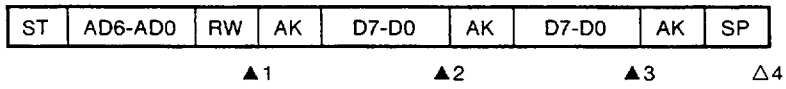


▲1 : IICS0 = 0101×110B (Example: when ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0001×100B
▲3 : IICS0 = 0001××00B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(b) When arbitration loss occurs during transmission of extension code

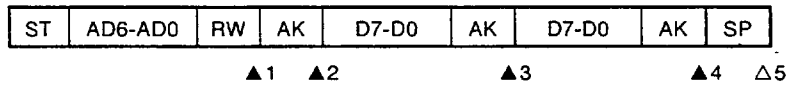
(i) When WTIM0n = 0



▲1 : IICS0 = 0110×010B (Example: when ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0010×000B
▲3 : IICS0 = 0010×000B
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1

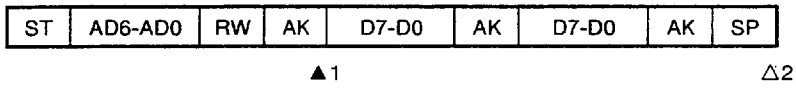


▲1 : IICS0 = 0110×010B (Example: when ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0010×110B
▲3 : IICS0 = 0010×100B
▲4 : IICS0 = 0010××00B
△5 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

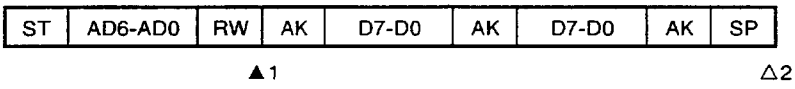
(a) When arbitration loss occurs during transmission of slave address data



▲1 : IICS0 = 01000110B (Example: when ALD0 is read during interrupt servicing)
△2 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension data

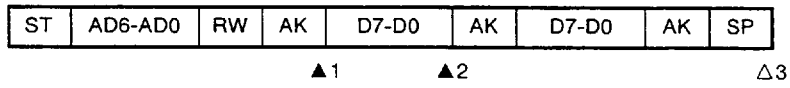


▲1 : IICS0 = 0110×010B (Example: when ALD0 is read during interrupt servicing)
LREL0 is set to "1" by software
△2 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(c) When arbitration loss occurs during transmission of data

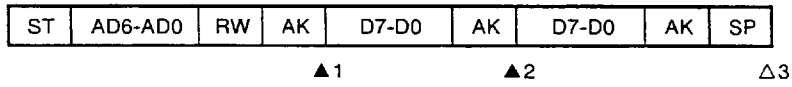
(i) When WTIM0 = 0



▲1 : IICS0 = 10001110B
▲2 : IICS0 = 01000000B (Example: when ALD0 is read during interrupt servicing)
△3 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

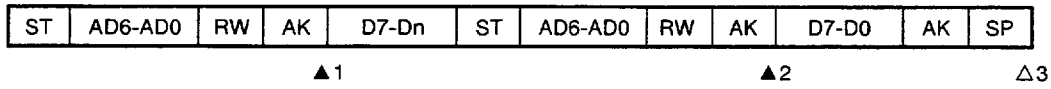


▲1 : IICS0 = 10001110B
▲2 : IICS0 = 01000100B (Example: when ALD0 is read during interrupt servicing)
△3 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

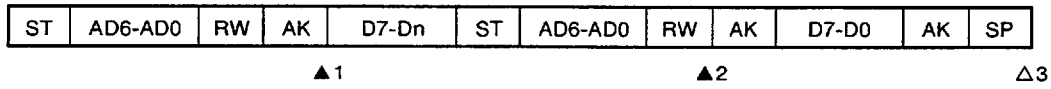
(i) Not extension code (Example: matches with SVA0)



▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 01000110B (Example: when ALD0 is read during interrupt servicing)
△3 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care
n = 6 - 0

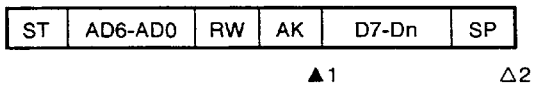
(ii) Extension code



▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 0110×010B (Example: when ALD0 is read during interrupt servicing)
IICC0's LREL0n is set to "1" by software
△3 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care
n = 6 - 0

(e) When loss occurs due to stop condition during data transfer

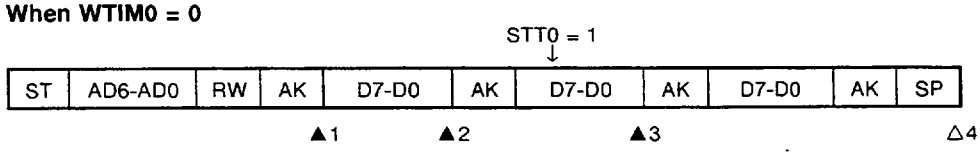


▲1 : IICS0 = 1000×110B
△2 : IICS0 = 01000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care
n = 6 - 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

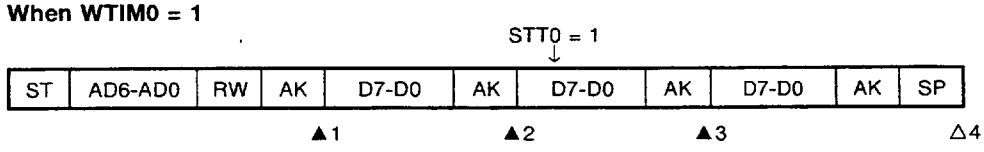
(i) When WTIM0 = 0



▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000×000B
▲3 : IICS0 = 01000000B (Example: when ALD0 is read during interrupt servicing)
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care

(ii) When WTIM0 = 1

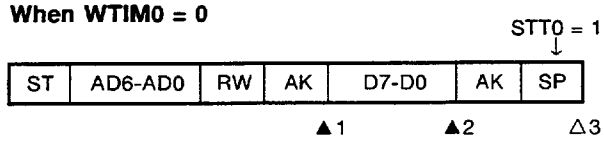


▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000××00B
▲3 : IICS0 = 01000100B (Example: when ALD0 is read during interrupt servicing)
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
× : don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

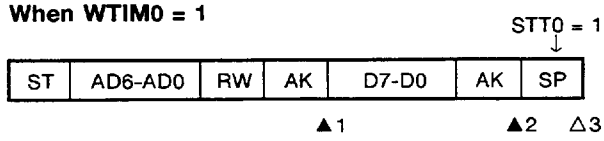
(i) When WTIM0 = 0



▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000×000B
△3 : IICS0 = 01000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1

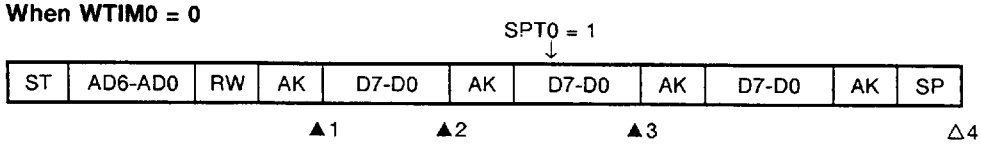


▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000××00B
△3 : IICS0 = 01000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

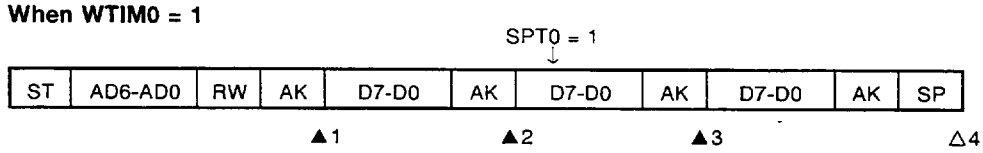
(i) When WTIM0 = 0



▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000×000B
▲3 : IICS0 = 01000000B (Example: when ALD0 is read during interrupt servicing)
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

(ii) When WTIM0 = 1



▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000××00B
▲3 : IICS0 = 01000000B (Example: when ALD0 is read during interrupt servicing)
△4 : IICS0 = 00000001B

Remarks ▲: Always generated
△: Generated only when SPIE0 = 1
×: don't care

18.6.8 Interrupt Request (INTIIC0) Generation Timing and Wait Control

The setting of bit 3 (WTIM0) in the IIC control register (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown in Table 18-3.

Table 18-3. INTIIC0 Timing and Wait Control

WTIM0	During slave device operation			During master device operation		
	Address	Data reception	Data transmission	Address	Data reception	Data transmission
0	9 (Notes 1, 2)	8 (Notes 2)	8 (Notes 2)	9	8	8
1	9 (Notes 1, 2)	9 (Notes 2)	9 (Notes 2)	9	9	9

Notes 1. The slave device's INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register (SVA0).
At this point, $\overline{ACK}$ is output regardless of the value set to IICC0's bit 2 (ACKE0). For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of the slave address register (SVA0), neither INTIIC0 nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation : Interrupt and wait timing are determined regardless of the WTIM0 bit.
- Master device operation : Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WREL0) of IIC control register (IICC0) to "1"
- By writing to the IIC shift register (IIC0)
- By setting a start condition (setting bit 1 (STT0) of IIC control register (IICC0) to "1")
- By setting a stop condition (setting IIC0's bit 0 (SPT0) to "1")

When 8-clock wait has been selected (WTIM0 = 0), the output level of $\overline{ACK}$ must be determined prior to wait cancellation.

(5) Stop condition detection

INTIIC0 is generated when a stop condition is detected.

18.6.9 Address Match Detection Method

When in I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An interrupt frequency (INTIIC0) occurs when a local address has been set to the slave address register (SVA0) and when the address set to SVA0 matches the slave address sent by the master device, or when an extension code has been received.

18.6.10 Error Detection

During I^2C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by the IIC shift register (IIC0) of the transmitting device, so the IIC0 data prior to transmission can be compared with the transmitted IIC0 data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

18.6.11 Extension Code

(1) When the high-order 4 bits of the receive address are either "0000" or "1111", the extension code flag (EXC0) is set for extension code reception and an interrupt request (INTIIC0) is issued at the falling edge of the eighth clock. The local address stored in the slave address register (SVA0) is not affected.

(2) If "111110××" is set to SVA0 by a 10-bit address transfer and "111110××0" is transferred from the master device, the results are as follows. Note that INTIIC0 occurs at the falling edge of the eighth clock.

- High-order four bits of data match: EXC0 = 1 (Note)
- Seven bits of data match: COI0 = 1 (Note)

Note EXC0: Bit 5 of IIC status register (IICS0)
COI0 : Bit 4 of IIC status register (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, you can set bit 6 (LREL0) of the IIC control register (IIC0) to "1" to set standby mode for the next communication operation.

Table 18-4. Extension Code Bit Definitions

Slave address	R/W bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	×	CBUS address
0000 010	×	Address that is reserved for different bus format
1111 0××	×	10-bit slave address specification

18.6.12 Arbitration

When several master devices simultaneously output a start condition (when STT0 is set to 1 before STD0 is set to 1, (Note)), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in the IIC status register (IICS0) is set via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set for high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **"18.6.7 I^2C Interrupt Requests (INTIIC0)"**.

Note STD0 : Bit 1 of IIC status register (IICS0)
STT0 : Bit 1 of IIC control register (IICC0)

Figure 18-14. Arbitration Timing Example

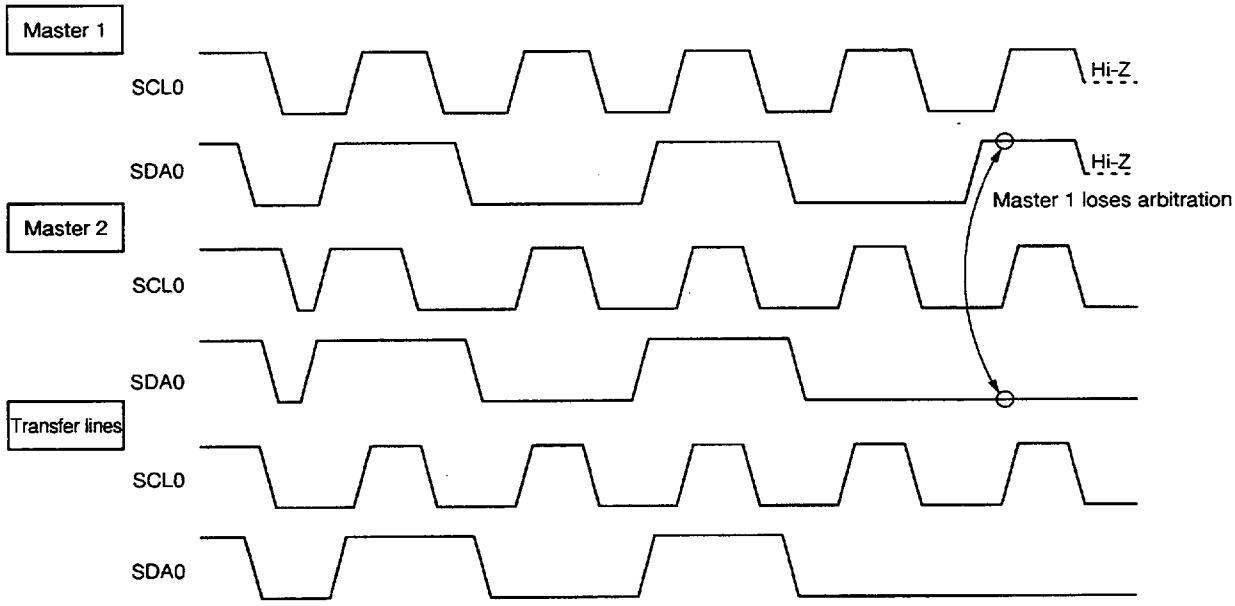


Table 18-5. Status during Arbitration and Interrupt Request Generation Timing

Status during arbitration	Interrupt request generation timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer (Note 1)
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{ACK}$ signal transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when SPIE0 = 1) (Note 2)
When data is at low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer (Note 1)
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when SPIE0 = 1) (Note 2)
When data is at low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer (Note 1)
When SCL0 is at low level while attempting to output a restart condition	

Notes 1. When WTIM0 (bit 3 of the IIC control register IICC0) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0 : Bit 5 of the IIC control register (IICC0)

18.6.13 Wake Up Function

The I^2C bus slave function is a function that generates an interrupt request (INTIIC0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wake-up standby mode is set. This wake-up standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has output a start condition) to a slave device.

However, when a stop condition is detected, bit 5 (SPIE0) of the IIC control register (IICC0) is set regardless of the wake up function, and this determines whether interrupt requests are enabled or prohibited.

18.6.14 Communication Reservation

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when bit 6 (LREL0) of the IIC control register (IICC0) was set to "1").

If bit 1 (STT0) of IICC0 is set while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait status is set.

When the bus release is detected (when a stop condition is detected), writing to the IIC shift register (IIC0) causes the master's address transfer to start. At this point, IICC0's bit 4 (SPIE0) should be set.

When STT0 has been set, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode) ..... communication reservation

To detect which operation mode has been determined for STT0, set STT0, wait for the wait period, then check the STT0 bit again.

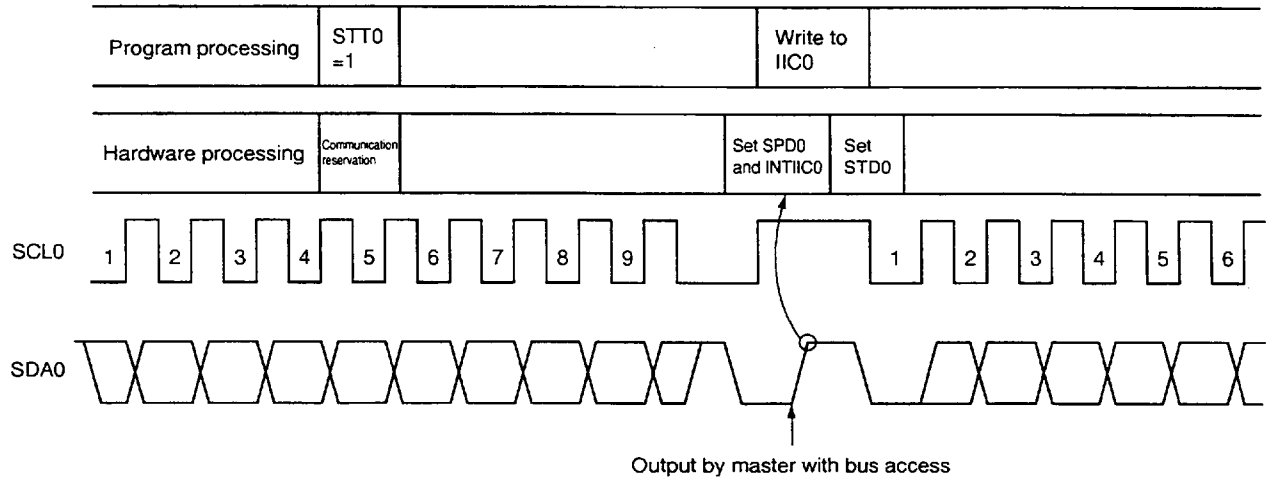
Wait periods, which should be set via software, are listed in Table 18-6. These wait periods can be set via the settings for bits 3 and 0 (SMC0 and CL00) in the IIC clock select register (IICCL0).

Table 18-6. Wait Periods

SMC0	CL00	Wait period
0	0	26 clocks
0	1	46 clocks
1	0	16 clocks
1	1	

Figure 18-5 shows communication reservation timing.

Figure 18-15. Communication Reservation Timing



IIC0 : IIC shift register
STT0: Bit 1 of IIC control register (IICC0)
STD0: Bit 1 of IIC status register (IICS0)
SPD0: Bit 0 of IIC status register (IICS0)

Communication reservations are accepted via the following timing. After bit 1 (STD0) of the IIC status register (IICS0) is set to "1", a communication reservation can be made by setting bit 1 (STT0) of the IIC control register (IICC0) to "1" before a stop condition is detected.

Figure 18-16. Timing for Accepting Communication Reservations

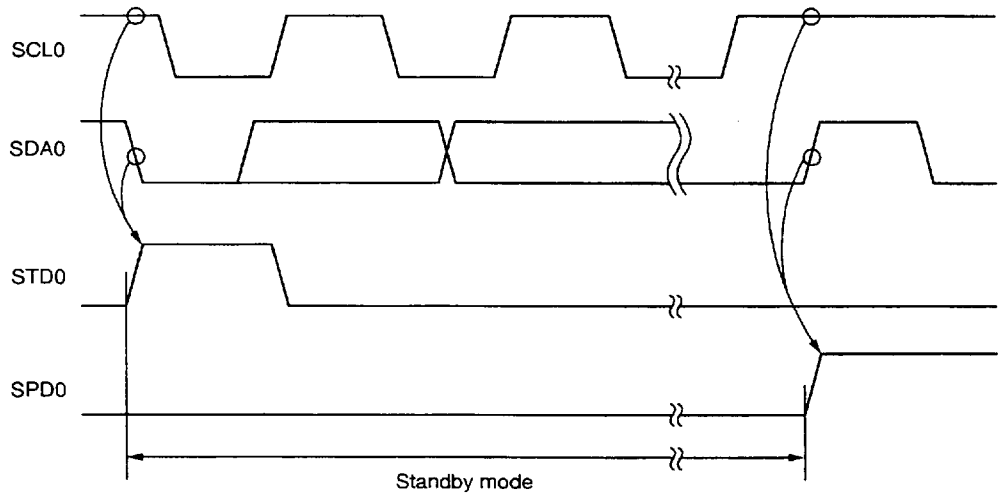
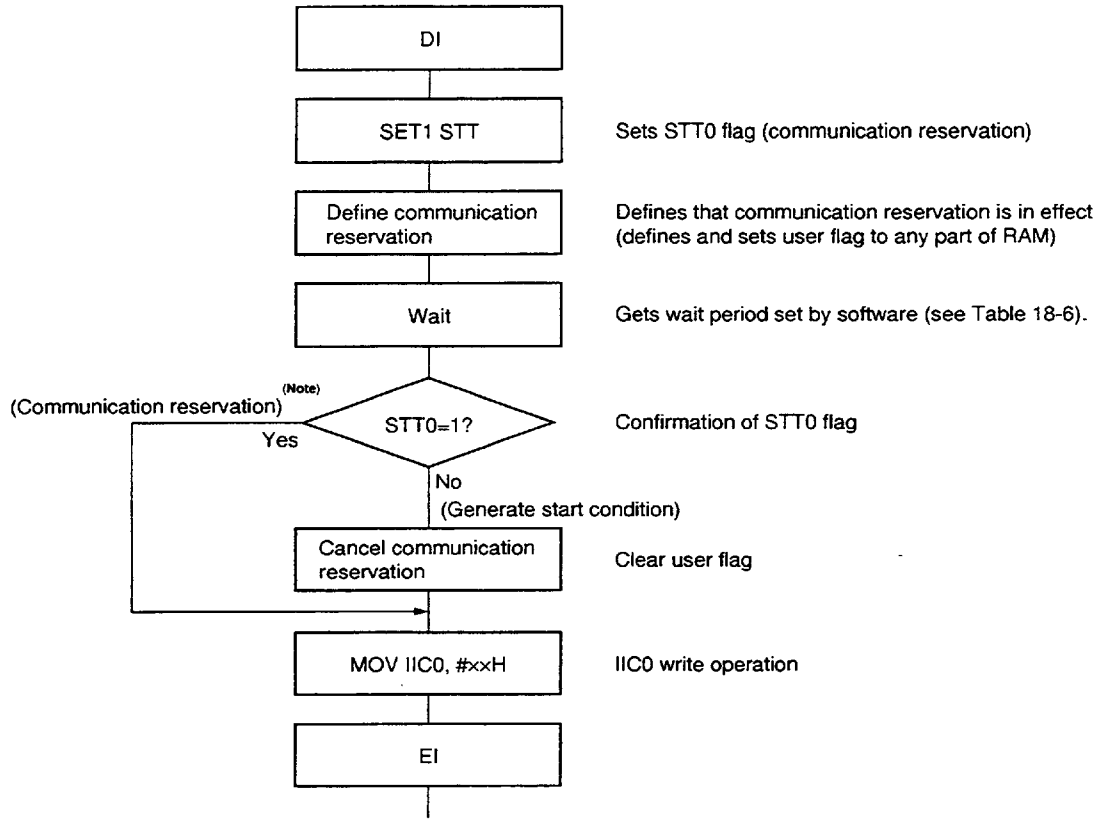


Figure 18-17 shows the communication reservation protocol.

Figure 18-17. Communication Reservation Protocol



Note The communication reservation operation executes a write to the IIC shift register (IIC0) when a stop condition interrupt request occurs.

Remark STT0 : Bit 1 of IIC control register (IICC0)
IIC0 : IIC shift register

18.6.15 Other Cautions

After a reset, when changing from a mode in which no stop condition has been detected (the bus has not been released) to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

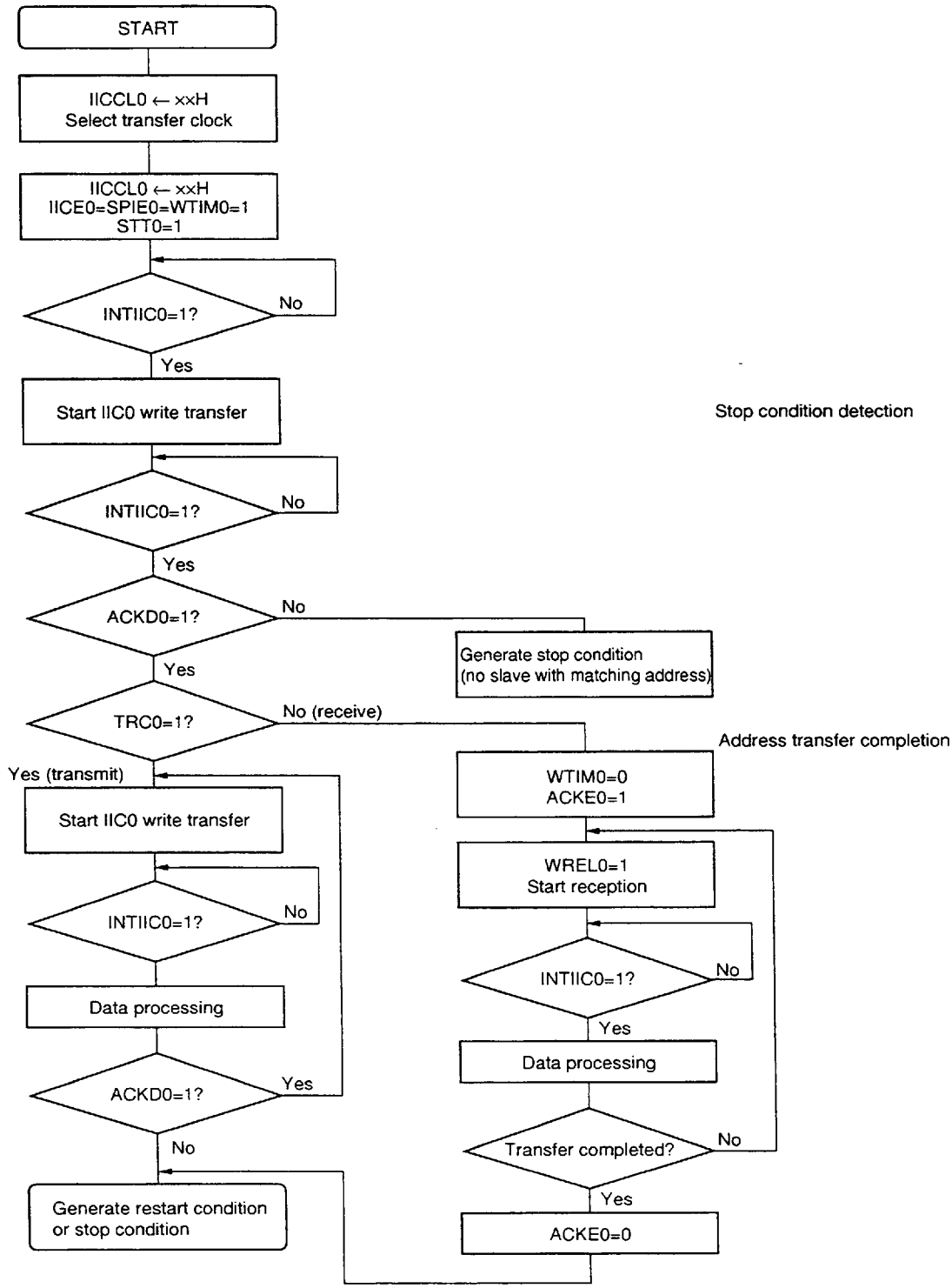
(a) Set IIC clock select register (IICCL0).
(b) Set bit 7 (IICE0) of the IIC control register (IICC0).
(c) Set bit 0 of IICC0.

18.6.16 Communication Operations

(1) Master operations

The following is a flow chart of the master operations.

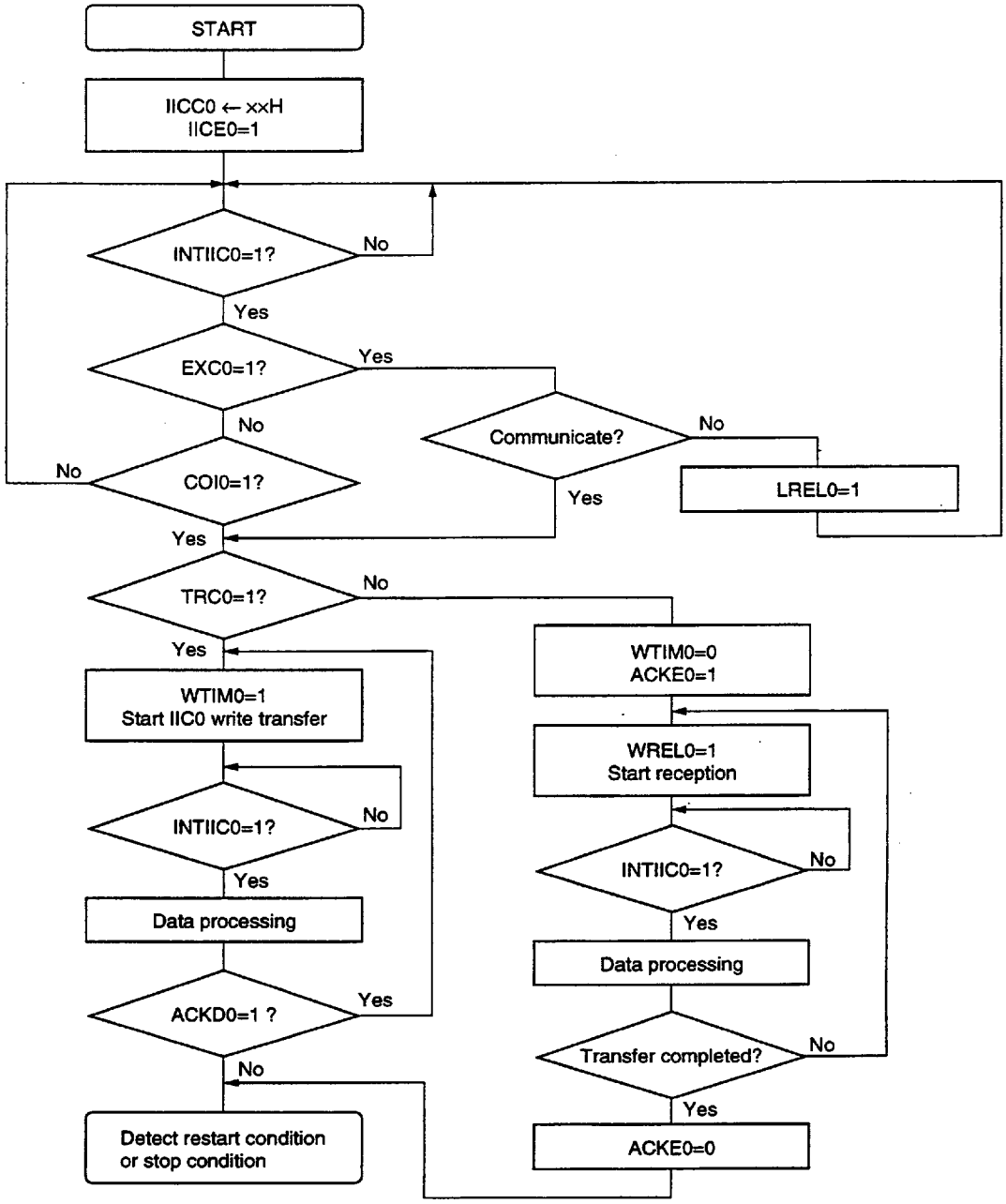
Figure 18-18. Master Operation Flow Chart



(2) Slave operation

An example of slave operation is shown below.

Figure 18-19. Slave Operation Flow Chart



18.7 TIMING CHARTS

When using I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of the IIC status register (IICS0)) that specifies the data transfer direction and then starts serial communication with the slave device.

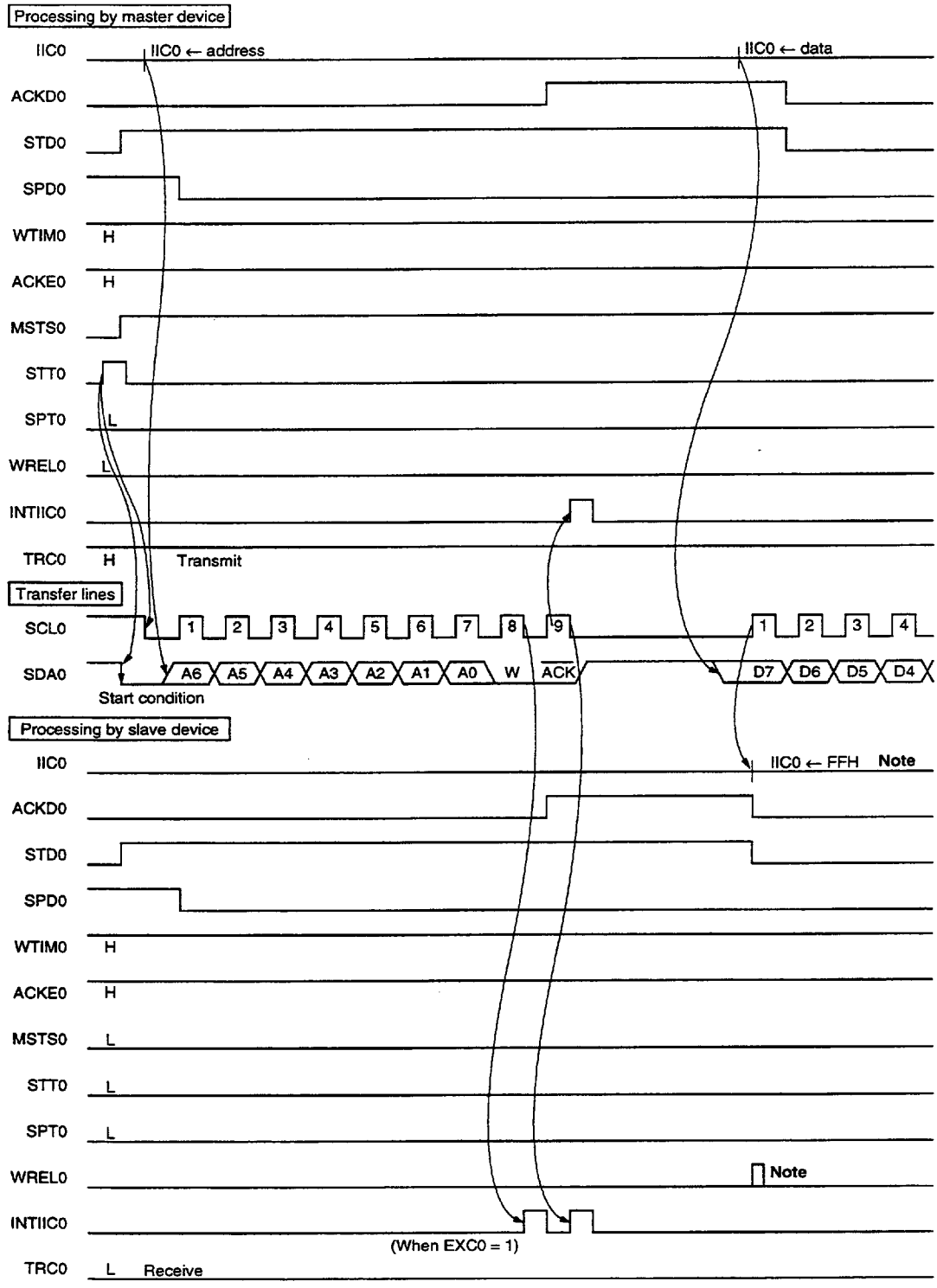
Figures 18-20 and 18-21 show timing charts of the data communication.

The IIC bus shift register (IIC0)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO0 latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured by IIC0 at the rising edge of SCL0.

Figure 18-20. Example of Master to Slave Communication (when 9-clock Wait Is Selected for Both Master and Slave) (1/3)

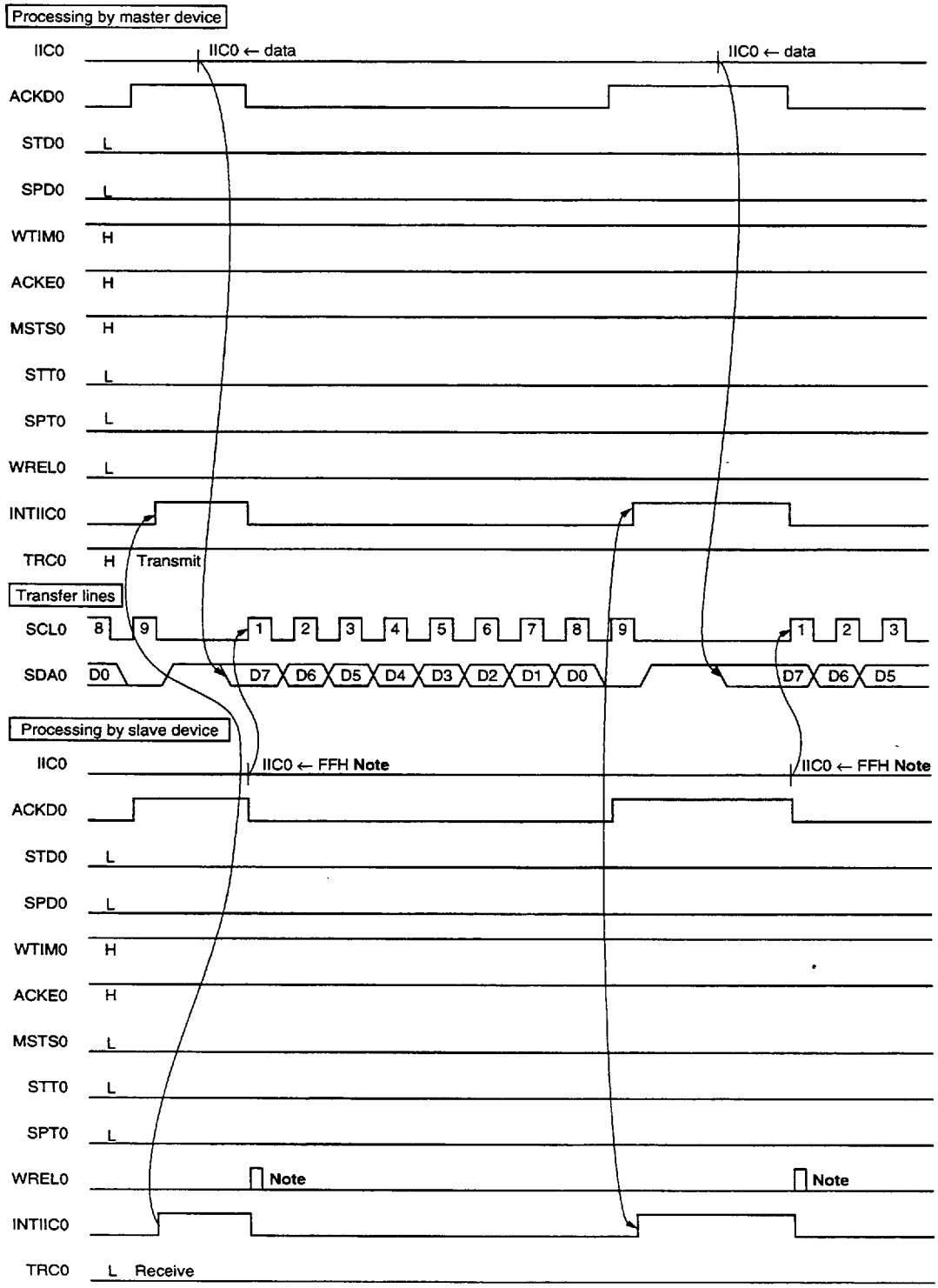
(1) Start condition ~ address



Note To cancel slave wait, write "FFH" to IICC0 or set WREL0.

Figure 18-20. Example of Master to Slave Communication (when 9-clock Wait Is Selected for Both Master and Slave) (2/3)

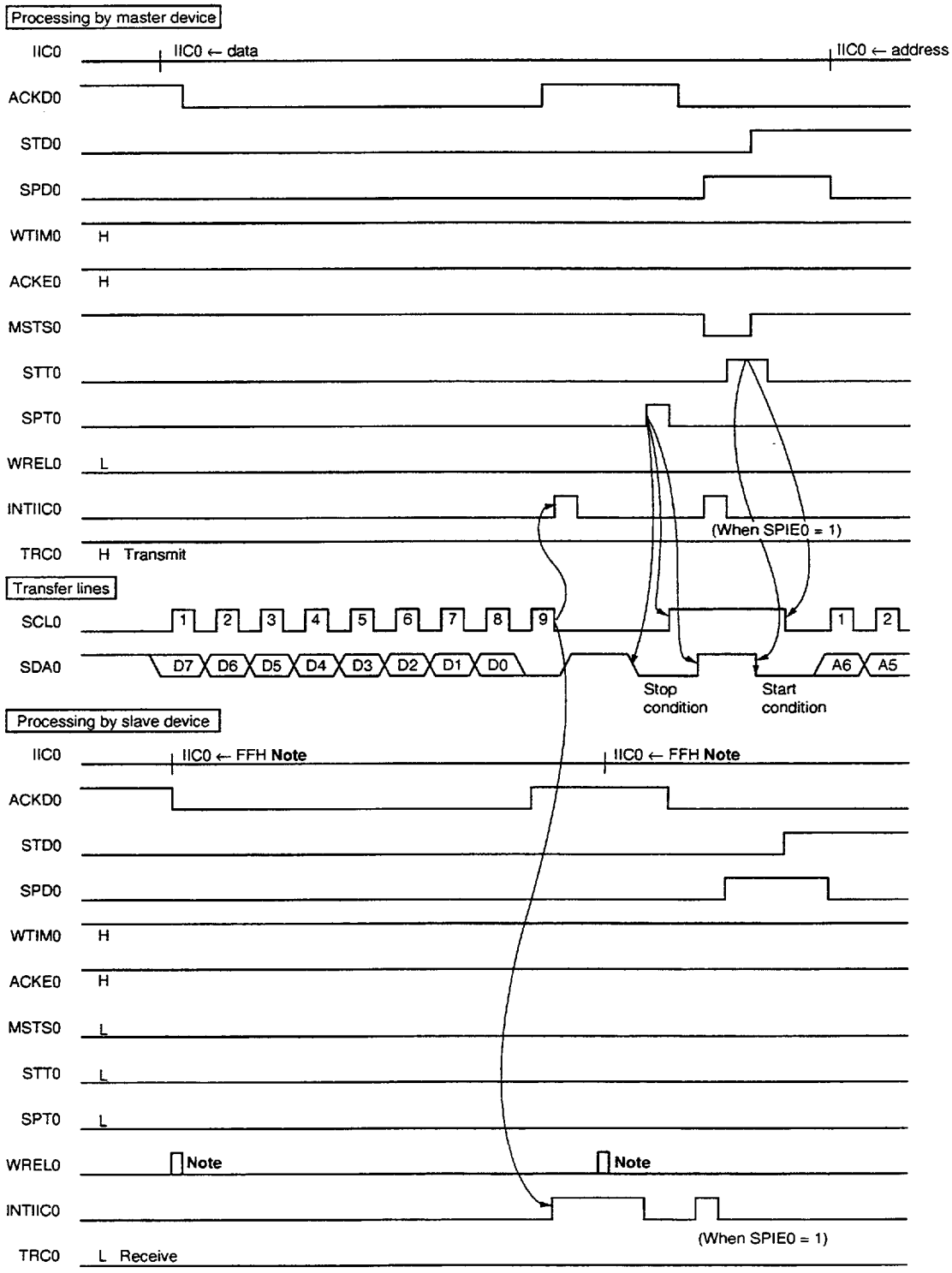
(2) Data



Note To cancel slave wait, write "FFH" to IICC0 or set WREL0.

Figure 18-20. Example of Master to Slave Communication (when 9-clock Wait Is Selected for Both Master and Slave) (3/3)

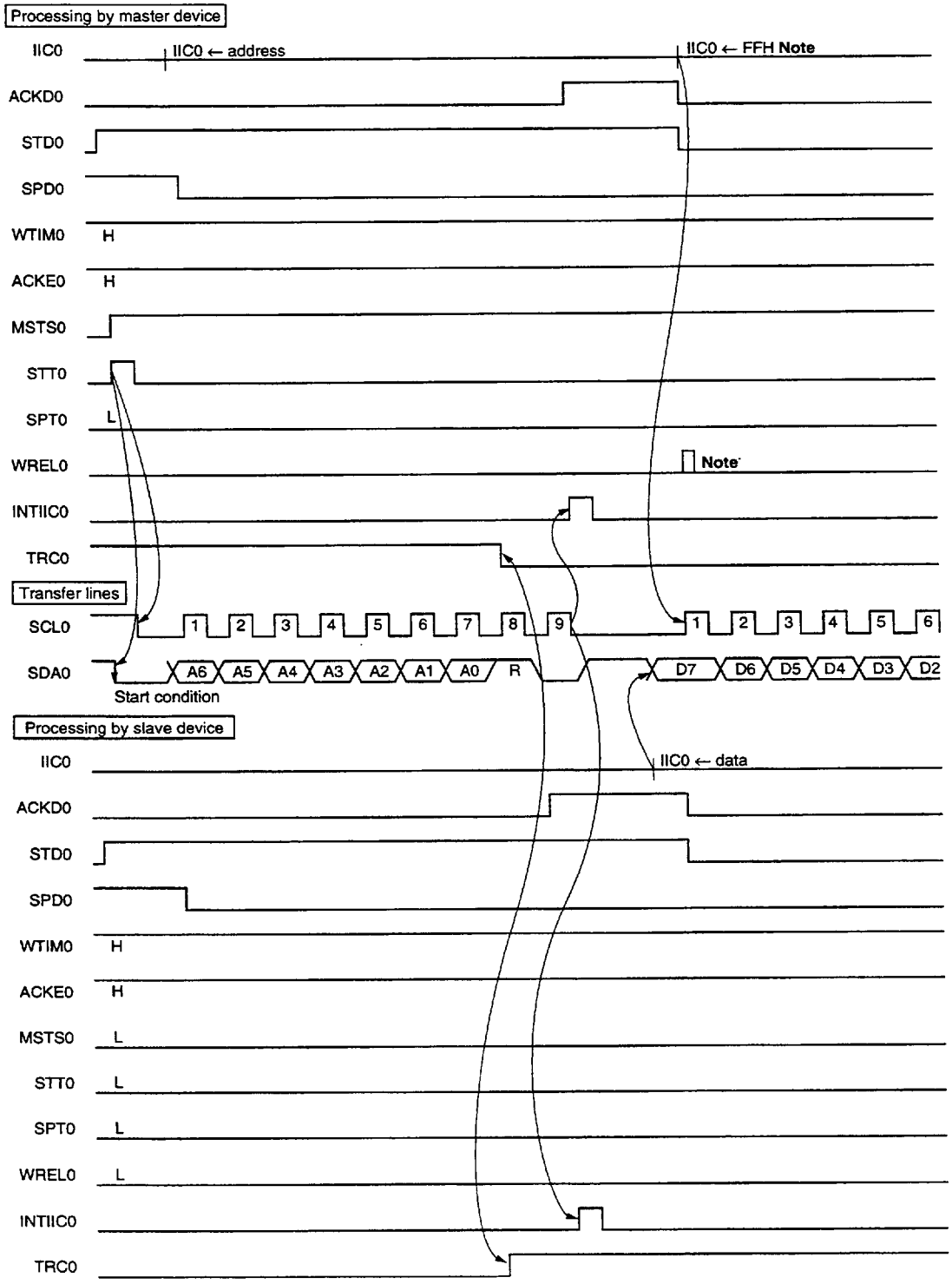
(3) Stop condition



Note To cancel slave wait, write "FFH" to IICC0 or set WREL0.

Figure 18-21. Example of Slave to Master Communication (when 9-clock Wait Is Selected for Both Master and Slave) (1/3)

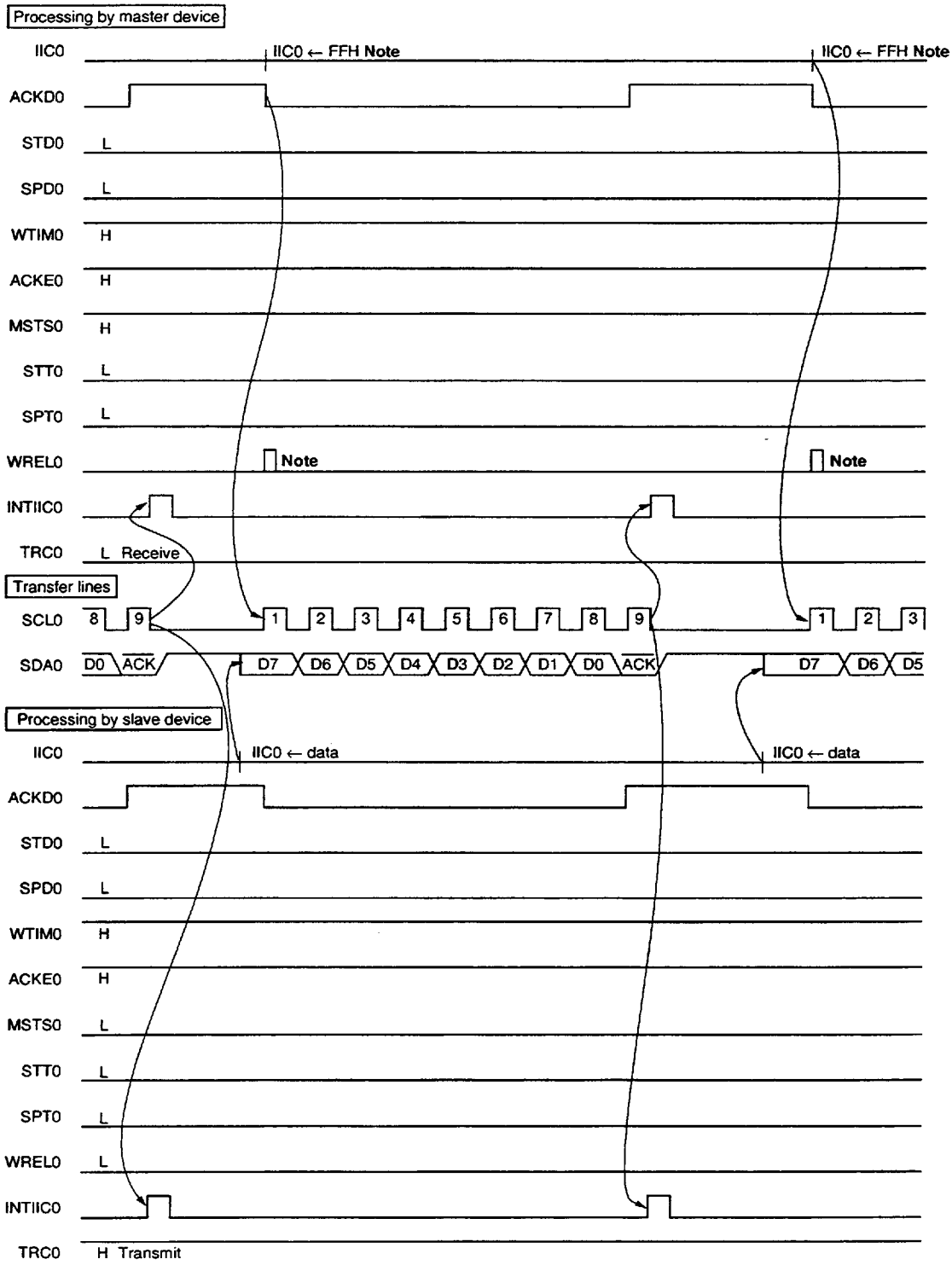
(1) Start condition ~ address



Note To cancel slave wait, write "FFH" to IICC0 or set WREL0.

Figure 18-21. Example of Slave to Master Communication (when 9-clock Wait Is Selected for Both Master and Slave) (2/3)

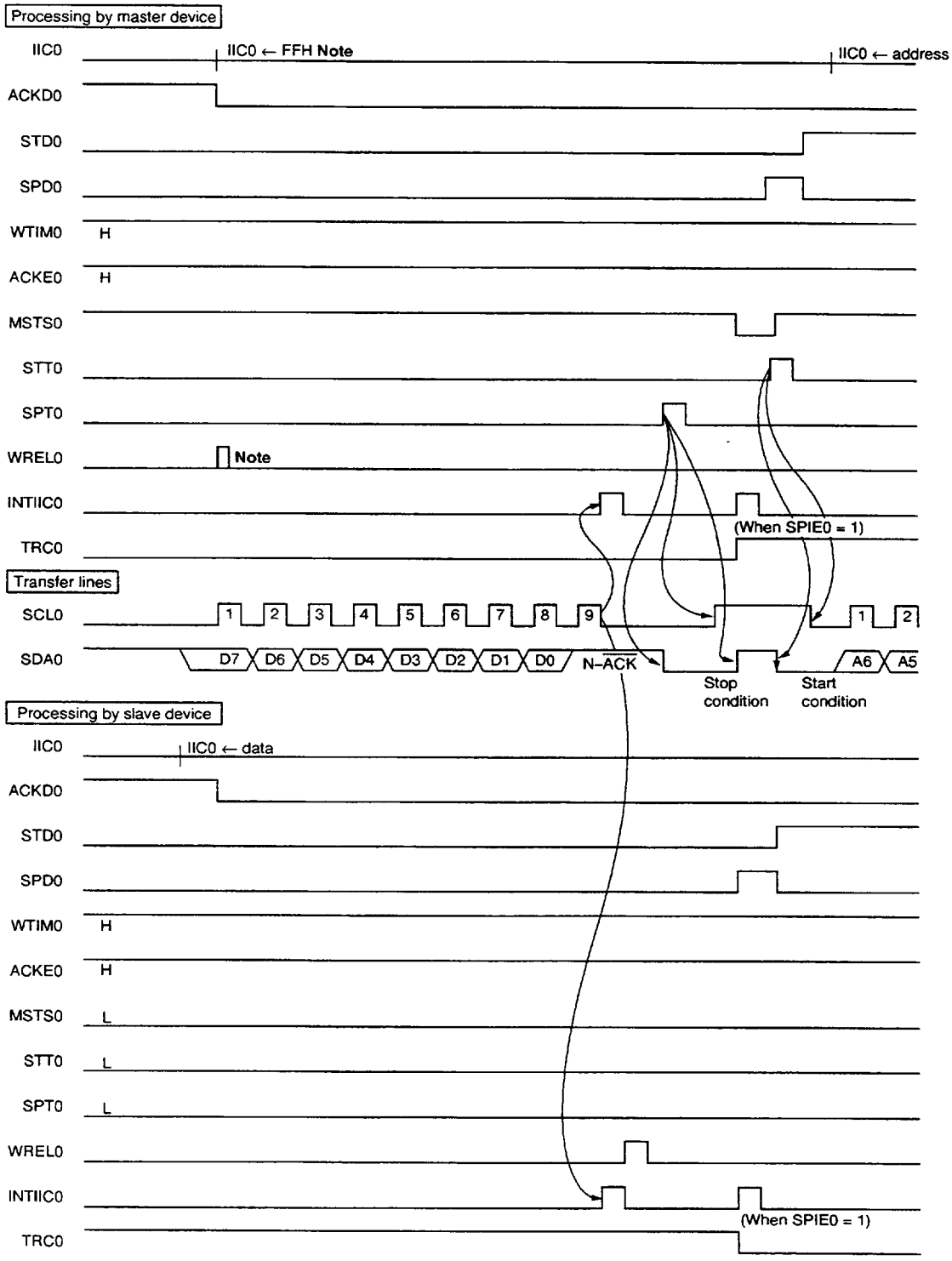
(2) Data



Note To cancel slave wait, write "FFH" to IICC0 or set WREL0.

Figure 18-21. Example of Slave to Master Communication (when 9-clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition



Note To cancel slave wait, write "FFH" to IICC0 or set WREL0.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 19 INTERRUPT FUNCTIONS

19.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo priority control and is given top priority over all other interrupt requests.

It generates a standby release signal.

One interrupt 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 specify 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 19-1**).

A standby release signal is generated.

Seven external interrupts and thirteen internal interrupts 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 a disabled stated. The software interrupt does not undergo interrupt priority control.

19.2 Interrupt Sources and Configuration

A total of 22 interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 19-1**).

Table 19-1. Interrupt Source List

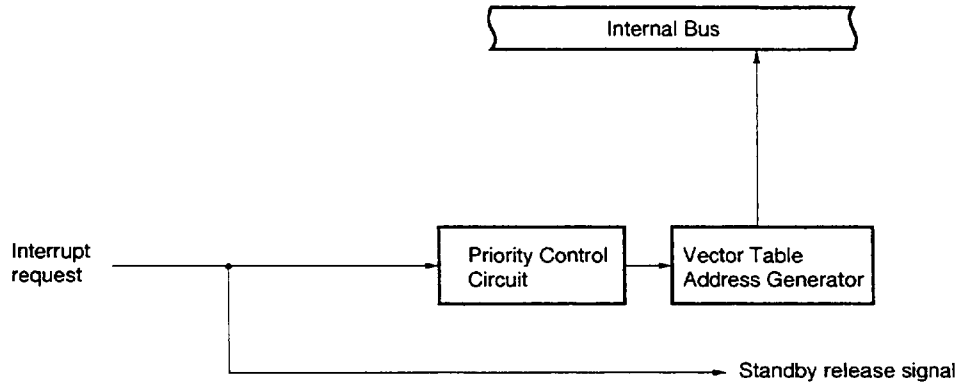
Interrupt Type	Default Priority	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval 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 [Only for μPD780024, 780034 Subseries]		0016H	
	10	INTIIC0	End of serial interface IIC0 transfer [Only for μPD780024Y, 780034Y Subseries]		0018H	
	11	INTWTI	Reference time interval signal from watch timer		001AH	
	12	INTTM00	Generation of 16-bit timer register and capture/compare register 00 (CR00) match signal (with CR00 specified to compare register)		001CH	(B)
			Detection of TI00/P70/TO0 pin input edge (with CR00 specified to capture register)	External		(D)
	13	INTTM01	Generation of 16-bit timer register and capture/compare register 01 (CR01) match signal (with CR01 specified to compare register)	Internal	001EH	(B)
			TI01/P71 pin input edge detection (with CR01 specified to capture register)	External		(D)
	14	INTTM50	Generation of 8-bit timer/event counter 50 match signal	Internal	0020H	(B)
	15	INTTM51	Generation of 8-bit timer/event counter 51 coincidence signal		0022H	
	16	INTAD0	End of A/D converter conversion		0024H	
	17	INTWT	Watch timer overflow		0026H	
	18	INTKR	Port 4 falling edge detection	External	0028H	(E)
Software	—	BRK	BRK Instruction execution	—	003EH	(F)

Notes 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 18 is the lowest.

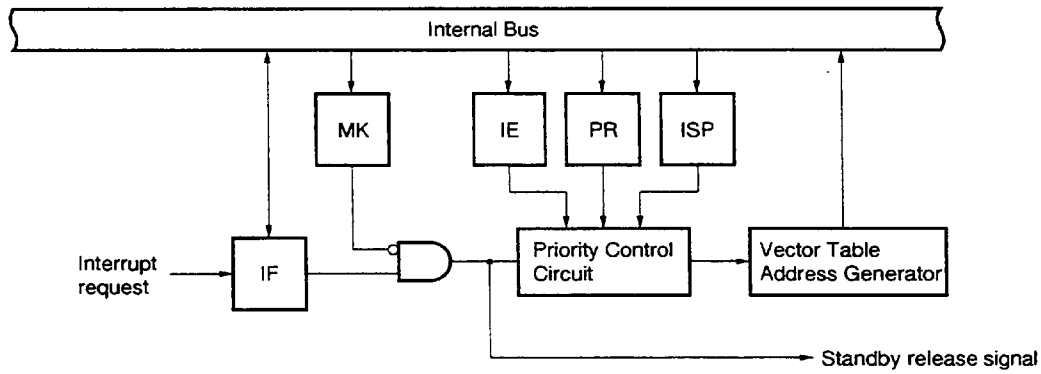
2. Basic configuration types (A) to (F) correspond to (A) to (F) in Figure 19-1.

Figure 19-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrup (INTP0 to INTP3)

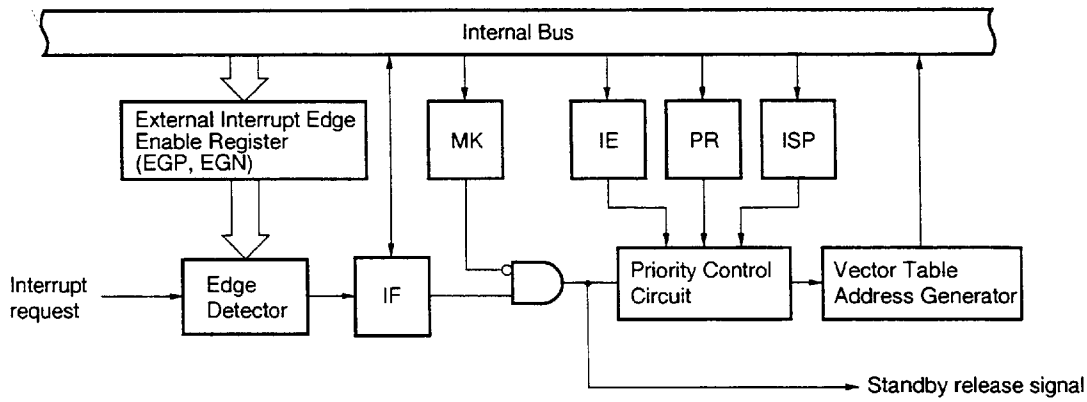
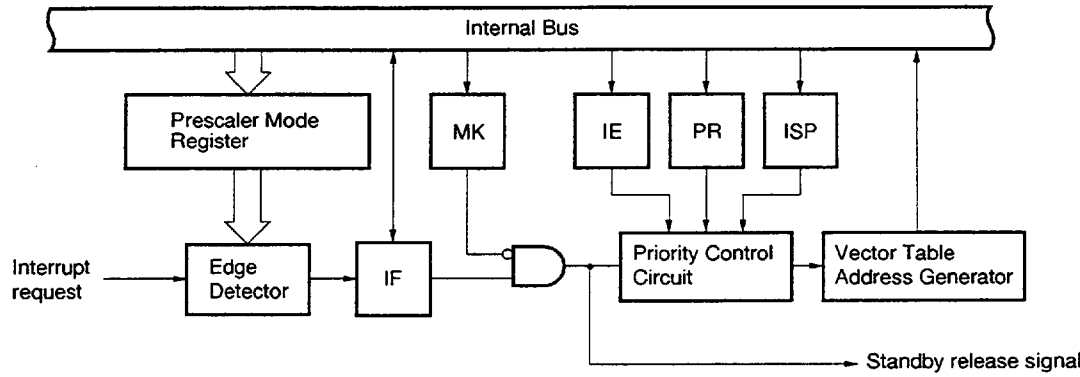
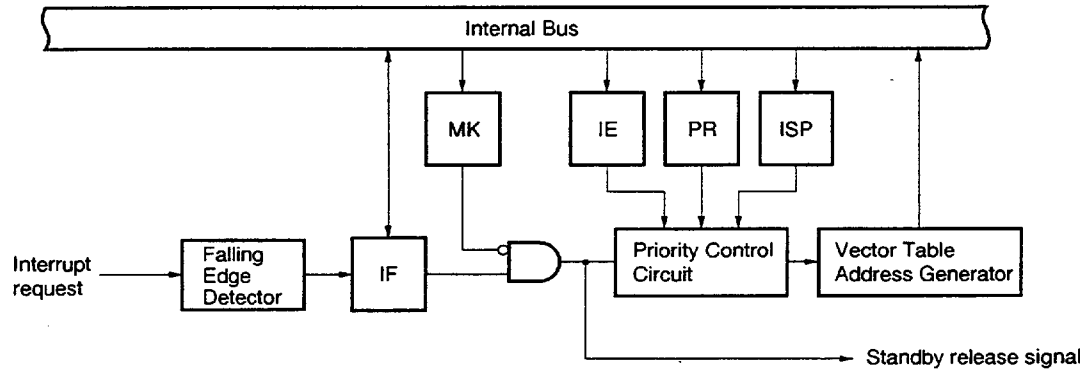


Figure 19-1. Basic Configuration of Interrupt Function (2/2)

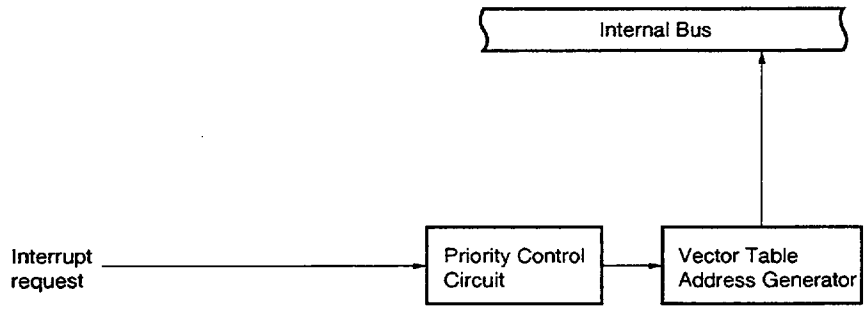
(D) External maskable interruption (INTTM00, INTTM01)



(E) External maskable interruption (INTKR)



(F) Software interrupt



IF : Interrupt request flag
IE : Interrupt enable flag
ISP : In-service priority flag
MK : Interrupt mask flag
PR : Priority specify flag

19.3 Interrupt Function Control Registers

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 specify flag register (PR0L, PR0H, PR1)
- External interrupt rising edge enable flag (EGP)
- External interrupt falling edge enable flag (EGN)
- Prescaler mode register (PRM0)
- Program status word (PSW)

Table 19-2 gives a list of interrupt request flags, interrupt mask flags, and priority specify flags corresponding to interrupt request sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources

Interrupt Request	Interrupt Request Flag		Interupt Mask Flag		Priority Specify Flag	
		Register		Register		Register
INTWDT	WDTIF	IF0L	WDTMK	MK0L	WDTPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTSER0	SERIF0		SERMK0		SERPR	
INTSR0	SRIF0		SRMK0		SRPR	
INTST0	STIF0		STMK0		STPR	
INTCSI30	CSIIF30	IF0H	CSIMK30	MK0H	CSIPR30	PR0H
INTCSI31[Note 1]	CSIIF31[Note 1]		CSIMK31[Note 1]		CSIPR31[Note 1]	
INTIIC0[Note 2]	IICIF0[Note 2]		IICMK0[Note 2]		IICPR[Note 2]	
INTWTI	WTIIF		WTIMK		TMPR3	
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTAD0	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
INTWT	WTIF		WTMK		WTPR	
INTKR	KRIF		KRMK		PT4PR	

Notes 1. μPD780024, 780034 Subseries only
2. μPD780024Y, 780034Y Subseries only

(1) Interrup 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 with a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are read with a 16-bit memory manipulation instruction.

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

Figure 19-2. Interrupt Request Flag Register (IF0L, IF0H, IF1L) Format

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	IICIF0Note 1	CSIIF31Note 2	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	ADIF

XXIFX	Interrupt Request Flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Notes 1. Incorporated only in the μPD780024 and 780034 Subseries. Be sure to set 0 for the μPD780024Y, 780034Y Subseries.

2. Incorporated only in the μPD780024Y, 780034Y Subseries. Be sure to set 0 for the μPD780024, 780034 Subseries.

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 0 to IF1L bits 3 to 7.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L, MK0H, and MK1L are set with a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form a 16-bit register, they are set with a 16-bit memory manipulation instruction.

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

Figure 19-3. Interrupt Mask Flag Register (MK0L, MK0H, MK1L) Format

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	IICMK0Note 1	CSIMK31Note 2	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	ADMK

XXMKX	Interrupt Servicing Control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Notes 1. Incorporated only in the μPD780024 and 780034 Subseries. Be sure to set 1 for the μPD780024Y, 780034Y Subseries.

2. Incorporated only in the μPD780024Y, 780034Y Subseries. Be sure to set 1 for the μPD780024, 780034 Subseries.

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. Always set 1 in MK1L bits 3 to 7.

(3) Priority specify flag registers (PR0L, PR0H, PR1L)

The priority specify flag registers are used to set the corresponding maskable interrupt priority orders. PR0L, PR0H, and PR1L are set with a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set with a 16-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets these registers to 00H.

Figure 19-4. Priority Specify Flag Register (PR0L, PR0H, PR1L) Format

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	IICPR0Note 1	CSIPR31Note 2	CSIPR30

Address: FFEAH After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR1L	1	1	1	1	1	PT4PR	WTPR	ADPR

XXPRX	Priority Level Selection
0	High priority level
1	Low priority level

Notes 1. Incorporated only in the μPD780024 and 780034 Subseries. Be sure to set 1 for the μPD780024Y, 780034Y Subseries.

2. Incorporated only in the μPD780024Y, 780034Y Subseries. Be sure to set 1 for the μPD780024, 780034 Subseries.

Cautions 1. When the watchdog timer is used in the watchdog timer 1 mode, set 1 in the WDTPR flag.

2. Always set 1 in PR1L bits 3 to 7.

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

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

Figure 19-5. External interrupt rising edge enable register (EGP), External interrupt falling edge enable register (EGN) Format

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 disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

(5) Prescaler mode register 0 (PRM0)

This register specifies the valid edge for TI00/P70/TO0/P71 pin inputs.

FRM0 is set with an 8-bit memory manipulation instruction.

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

Figure 19-6. Prescaler Mode Register 0 (PRM0) Format

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 rising and falling edges

ES01	ES00	TI00 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Caution Set the valid edge of the TI00/P70/TO0 and TI01/P71 pins after setting bits 1 to 3 of 16-bit timer mode control register 0 (TMC0) to 0 to stop the timer operation.

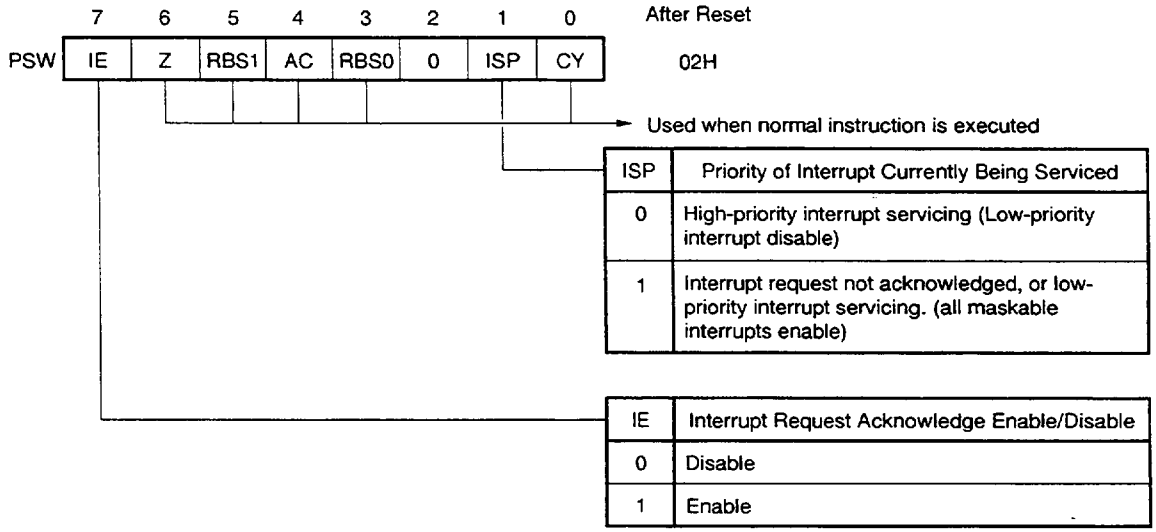
(6) Program status word (PSW)

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 multiple 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 specify 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 19-7. Program Status Word Format



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

19.4 Interrupt Servicing Operations

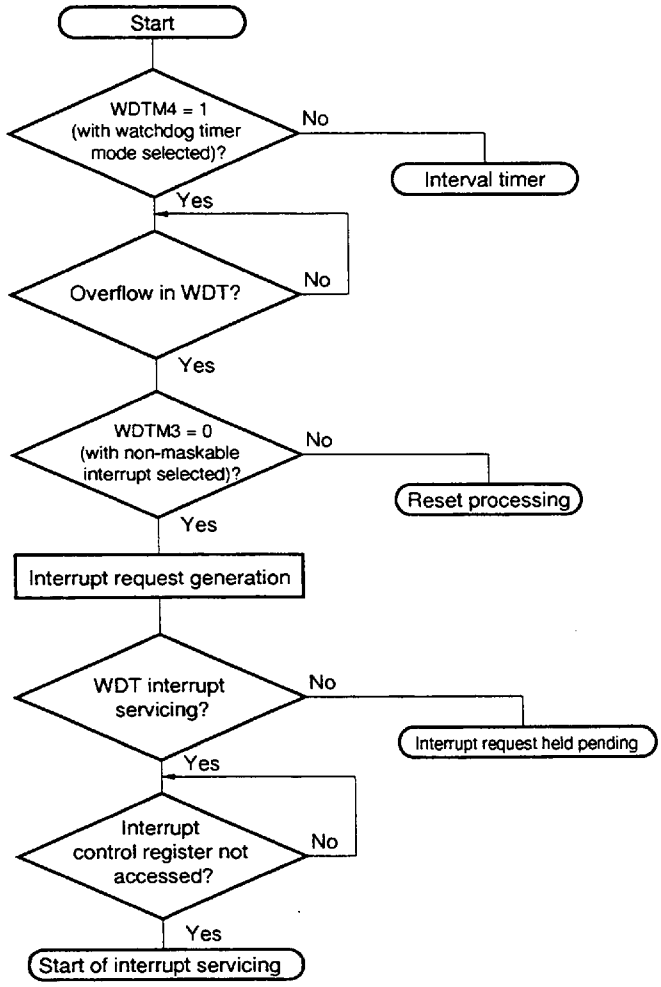
19.4.1 Non-maskable interrupt request acknowledge operation

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.

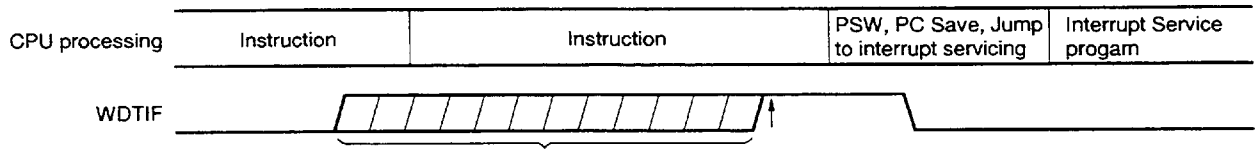
A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. 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. Figure 19-8, 19-9, and 19-10 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 19-8. Non-Maskable Interrupt Request Generation to Acknowledge Flowchart



WDTM: Watchdog timer mode register
WDT : Watchdog timer

Figure 19-9. Non-Maskable Interrupt Request Acknowledge Timing

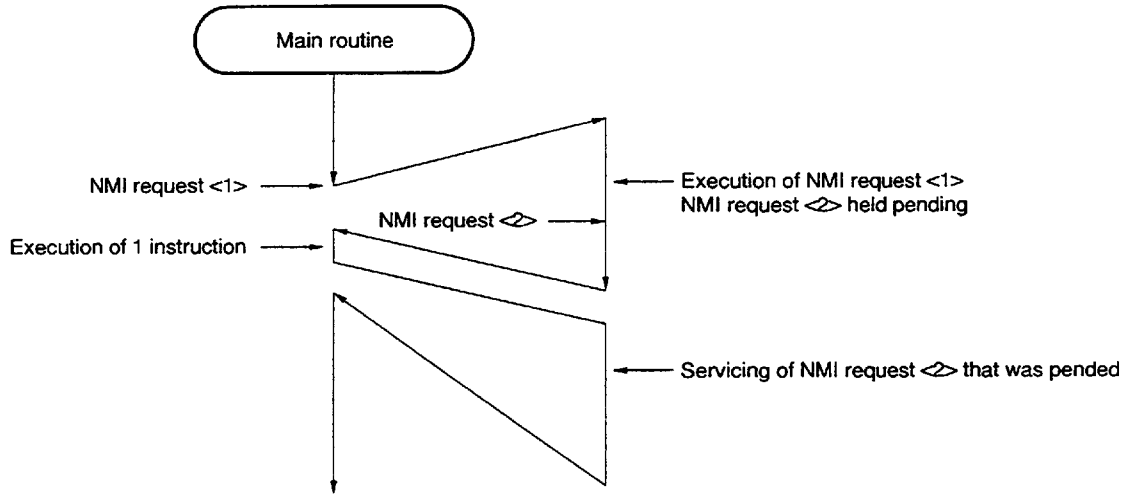


Interrupt request generated during this interval is acknowledged at ↑.

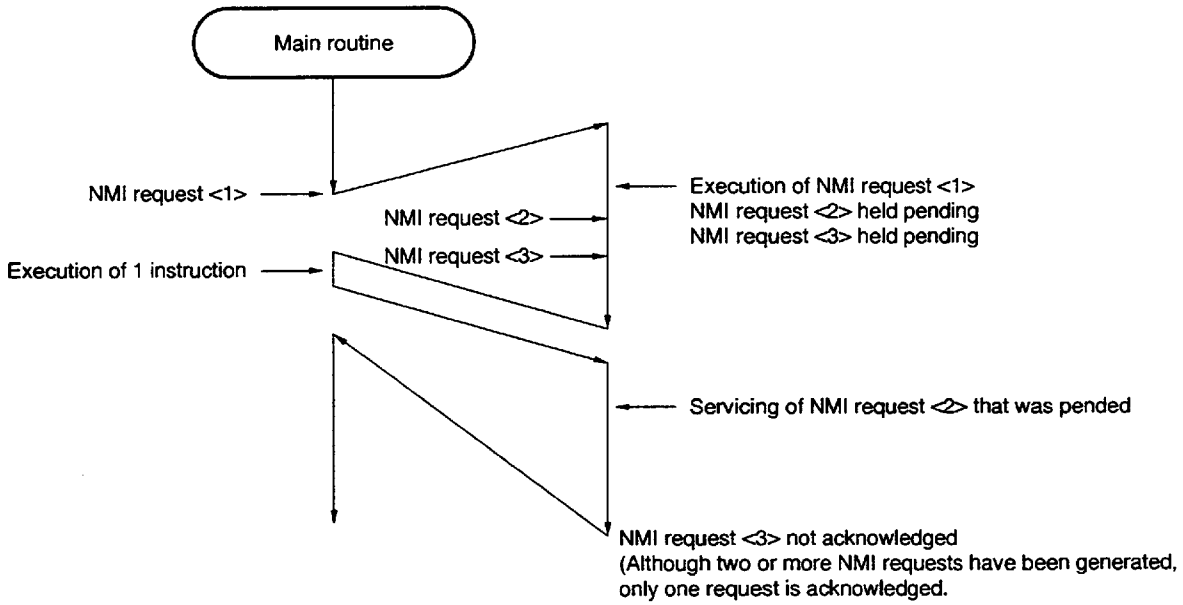
WDTIF: Watchdog timer interrupt request flag

Figure 19-10. 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



19.4.2 Maskable interrupt acknowledge operation

A maskable interrupt becomes acknowledgeable when an interrupt request flag is set to 1 and the mask flag corresponding to that interrupt 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 19-3 below.

For the interrupt request aknowledge timing, see the figures 19-12 and 19-13.

Table 19-3. Times from Generation of Maskable Interrupt until Servicing

	Minimum Time	Maximum Time[Note]
When xxPR = 0	7 clocks	32 clocks
When xxPR = 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/fCPU (fCPU: CPU clock)

If two or more interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specify flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

An interrupt request that is held pending is acknowledged when it becomes aknowledgeable.

Figure 19-11 shows the interrupt request acknowledge algorithm.

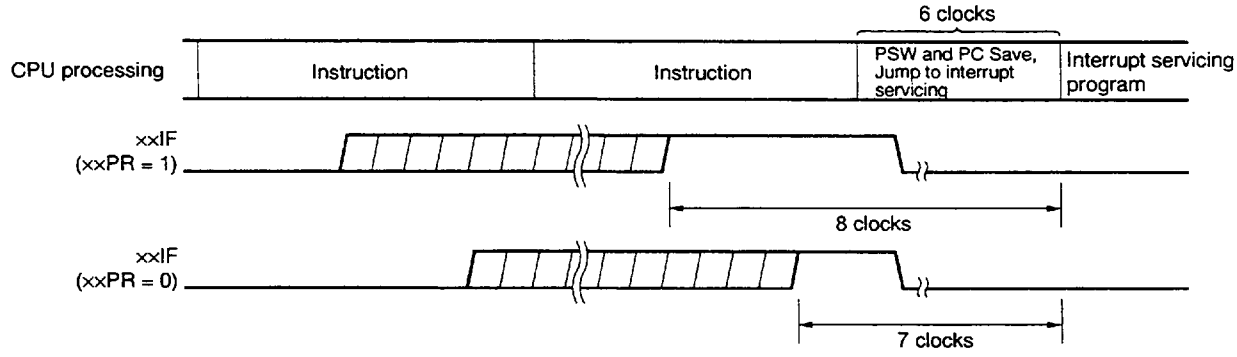
If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specify flag corresponding to the acknowledged interrupt are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

Return from an interrupt is possible with the RETI instruction.

Figure 19-11. Interrupt Request Acknowledge Processing Algorithm

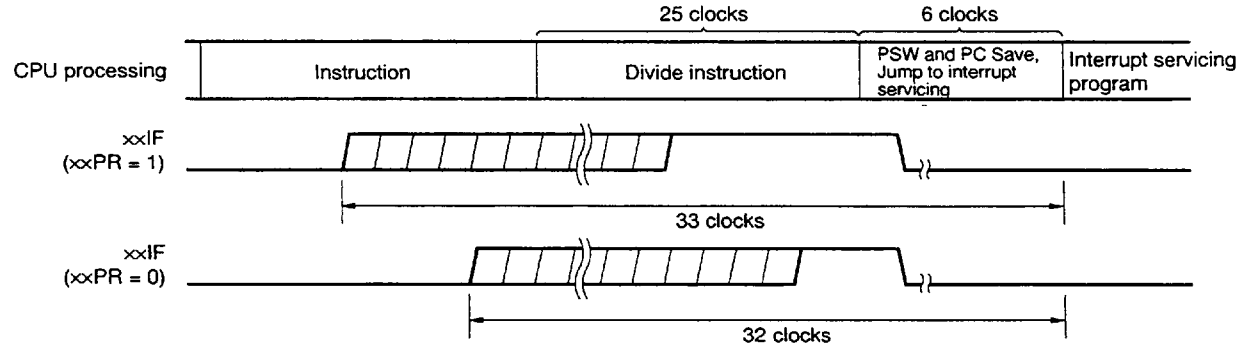
××IF : Interrupt request flag
××MK : Interrupt mask flag
××PR : Priority specify flag
IE : Flag that controls acknowledge of maskable interrupt request (1 = Enable, 0 = Disable)
ISP : Flat that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request received, or low-priority interrupt servicing)

Figure 19-12. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

Figure 19-13. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

19.4.3 Software interrupt request acknowledge operation

A software interrupt acknowledge 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.

19.4.4 Multiple interrupt servicing

Multiple interrupts occur when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupts do not occur unless the interrupt request acknowledge enable state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is received, interrupt requests acknowledge becomes disabled (IE = 0). Therefore, to enable multiple interrupts, 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, multiple interrupts may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupts.

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 multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because 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.

Multiple interrupt servicing is not possible during non-maskable interrupt servicing.

Table 19-4 shows interrupt requests enabled for multiple interrupt servicing, and Figure 19-14 shows multiple interrupt examples.

Table 19-4. Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing

Multiple Interrupt 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. ○ : Multiple interrupt enable

2. × : Multiple interrupt disable

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.

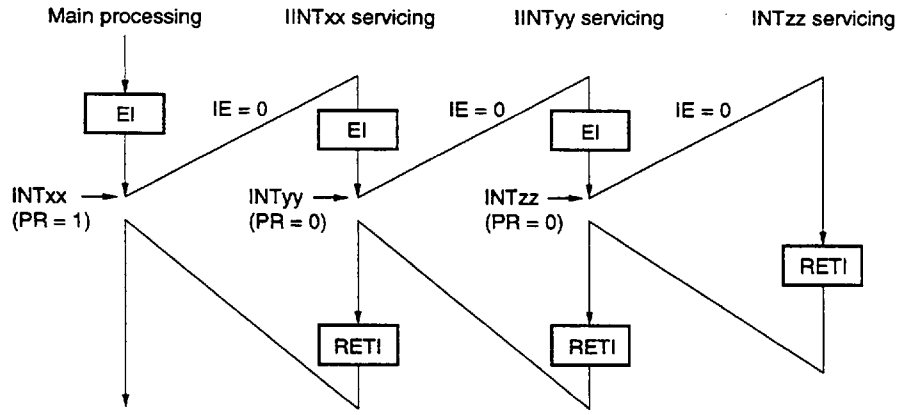
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0 : Higher priority level

PR = 1 : Lower priority level

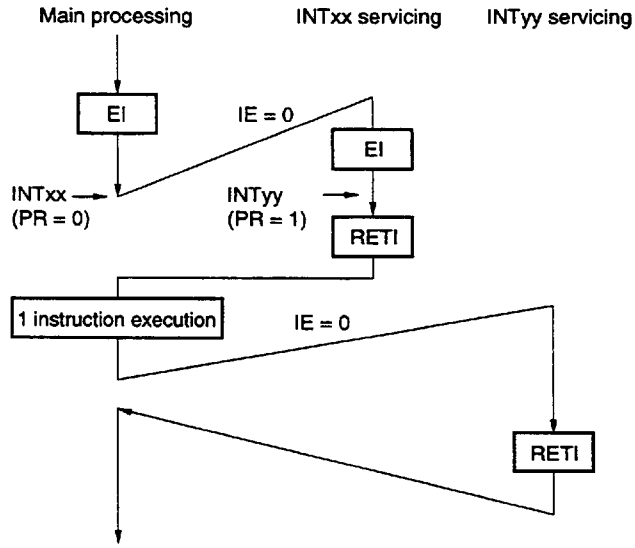
Figure 19-14. Multiple Interrupt Examples (1/2)

Example 1 Multiple interrupts occur twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledge.

Example 2 Multiple interrupt servicing does not occur due to priority control

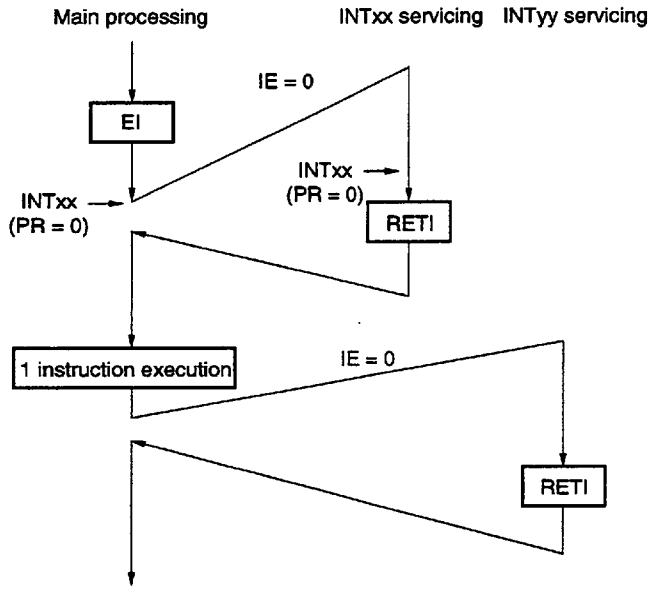


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0 : Higher priority level
PR = 1 : Lower priority level
IE = 0 : Interrupt request acknowledge disable

Figure 19-14. Multiple Interrupt Examples (2/2)

Example 3 Multiple interrupt servicing 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 multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0 : Higher priority level
IE = 0 : Interrupt request acknowledge disabled

19.4.5 Interrupt request hold

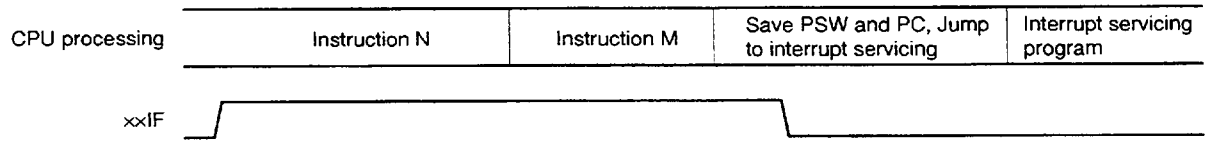
There are instructions where, even if an interrupt request is issued for them while another instruction is executed, request acknowledge is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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, PR1L, EGP, and EGN registers.

Caution The BRK instruction is not one of the above-listed interrupt request hold instruction. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt requests is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

The timing with which interrupt requests are held pending is shown in Table 19-15.

Figure 19-15. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The xxRP (priority level) values do not affect the operation of xxIF (instruction request)

PAGE(S) INTENTIONALLY BLANK

CHAPTER 20 EXTERNAL DEVICE EXPANSION FUNCTION

20.1 External Device Expansion Function

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

Table 20-1. Pin Functions in External Memory Expansion Mode

Pin Function at External Device Connection		Alternate Function
Name	Function	
AD0 - AD7	Muliplexed address/data bus	P40 - P47
A8 - A15	Address bus	P50 - P57
$\overline{RD}$	Read strobe signal	P64
$\overline{WR}$	Write strobe signal	P65
$\overline{WAIT}$	Wait signal	P66
ASTB	Address strobe signal	P67

Table 20-2. State of Port 4 to 6 Pins in External Memory Expansion Mode

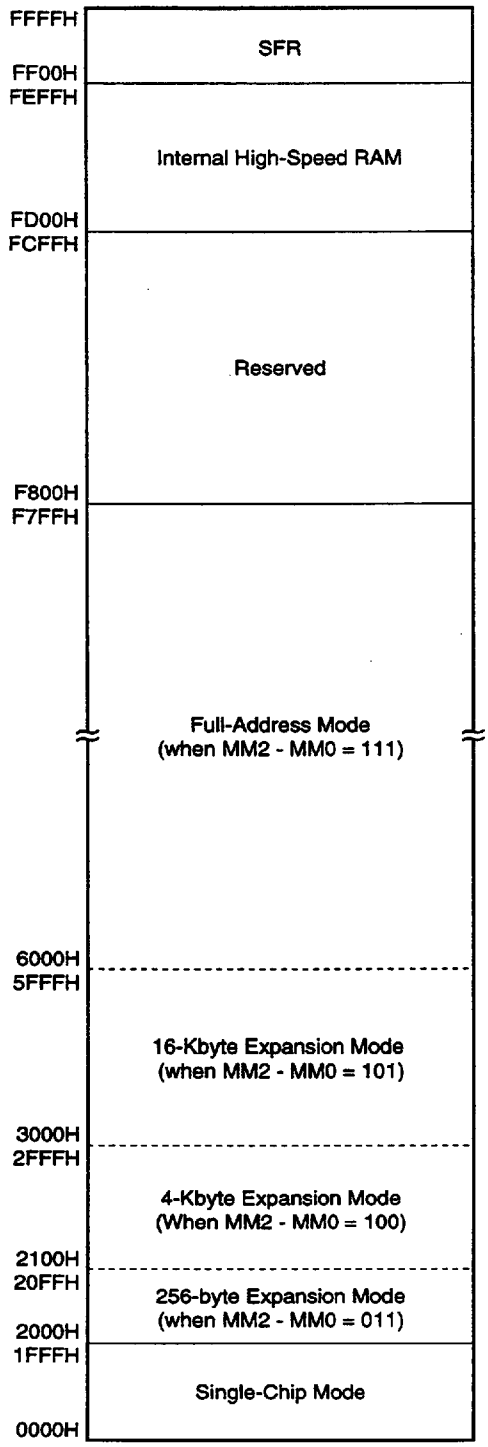
Port / External expansion mode	Port 4	Port 5								Port 6							
	0 - 7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Single-chip mode	Port	Port								Port							
256-byte expansion mode	Address/data	Port								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
4-kbyte expansion mode	Address/data	Address				Port				Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
16-kbyte expansion mode	Address/data	Address						Port		Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
Address mode	Address/data	Address								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			

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

The memory maps when the external device expansion function is used are as follows.

Figure 20-1. External Map when Using External Device Function (1/2)

(a) Memory map of μPD780021, 780031, 780021Y, 780031Y and of μPD78F0034, 78F0034Y when internal ROM (flash memory) size is 8 Kbytes



(b) Memory map of μPD780022, 780032, 780022Y, 780032Y and of μPD78F0034, 78F0034Y when internal ROM (flash memory) size is 16 Kbytes

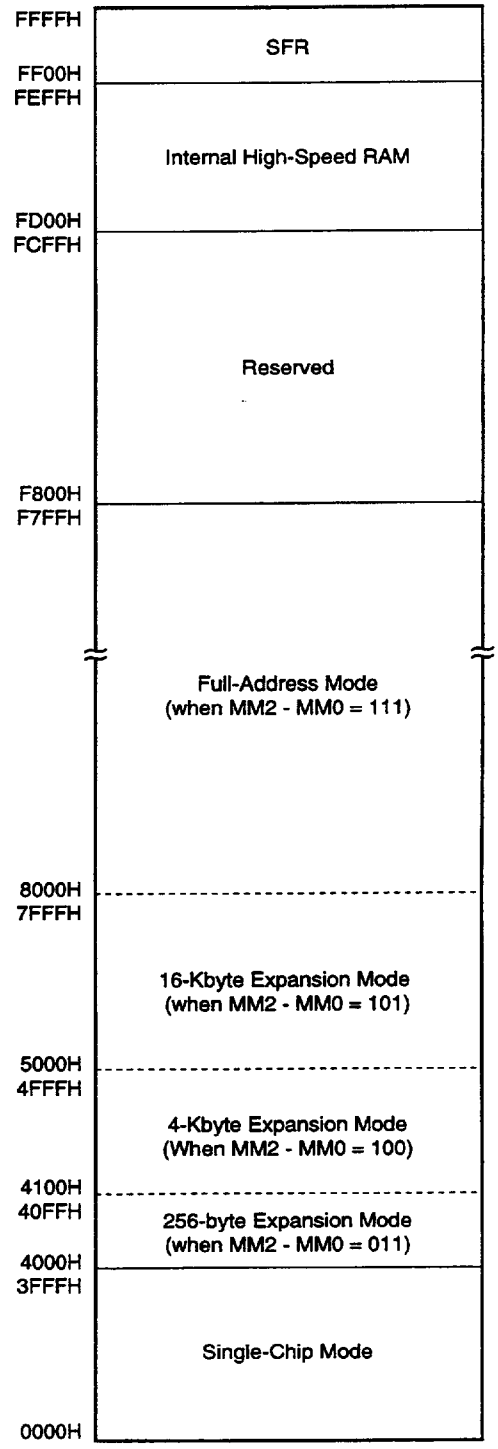
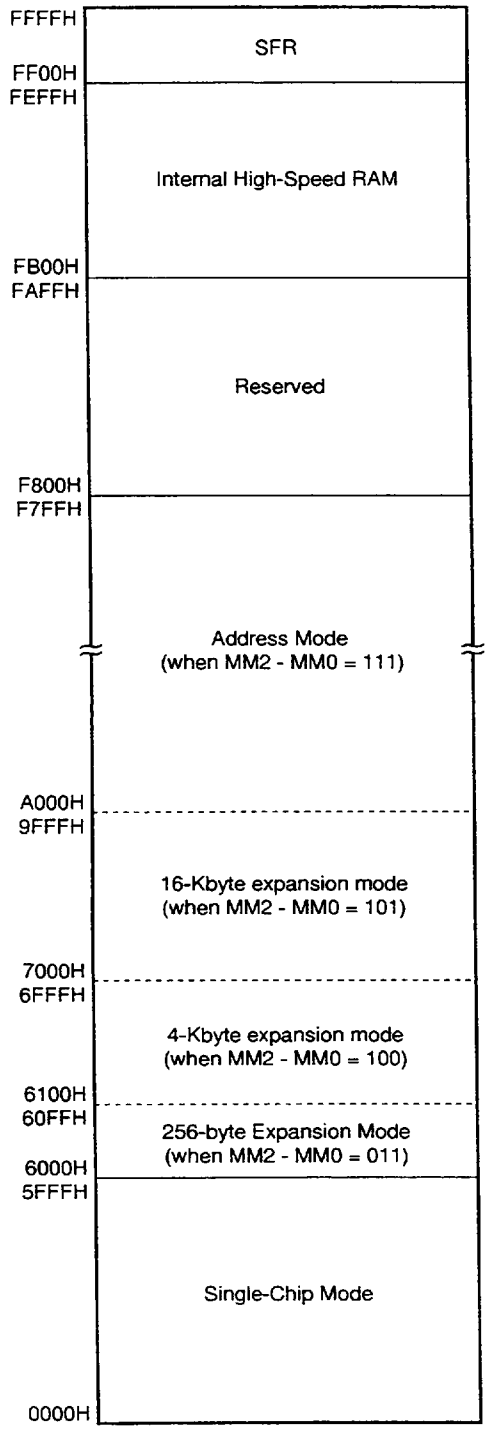
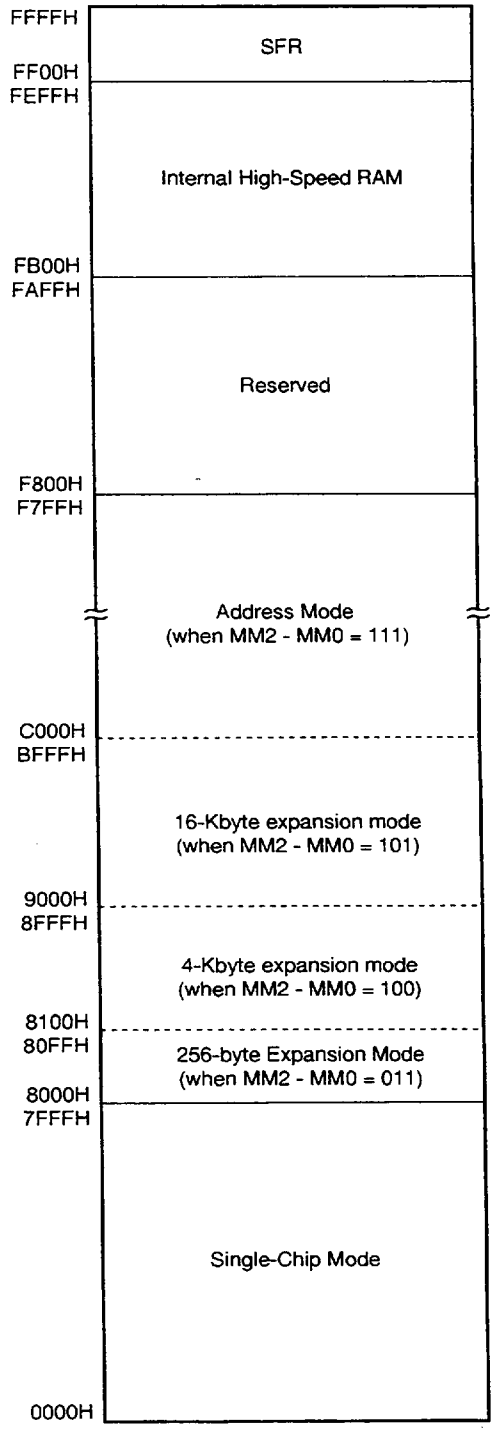


Figure 20-1. External Map when Using External Device Function (2/2)

(c) Memory map of μPD780023, 780033, 780023Y, 780033Y and of μPD78F0034, 78F0034Y when internal ROM (flash memory) size is 24 Kbytes



(d) Memory map of μPD780024, 780034, 780024Y, 780034Y and of μPD78F0034, 780034Y when ROM (flash memory) size is 32 kbytes



20.2 External Device Expansion Function Control Register

The external device expansion function is controlled by the following two types of registers.

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

(1) Memory expansion mode register (MEM)

MEM sets the external expansion area.

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

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

Figure 20-2. Memory Expansion Mode Register (MEM) Format

Address: FF47H After Reset: 00H R/W

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

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

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

(2) Memory expansion wait setting register (MM)

MM sets the number of waits.

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

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

Figure 20-3. Memory Expansion Wait Setting Register (MM) Format

Address: FFF8H After Reset: 10H R/W

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

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

20.3 External Device Expansion Function Timing

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

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

Read strobe output pin. The read strobe output pin is output in data accesses and instruction fetches from external memory.

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

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

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

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

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

External wait signal input pin.

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

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

(4) ASTB pin (Alternate function: P67)

Address strobe signal output pin. The address strobe signal is output regardless of data access and instruction fetch from external memory. During internal memory access, the address strobe signal is not output.

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

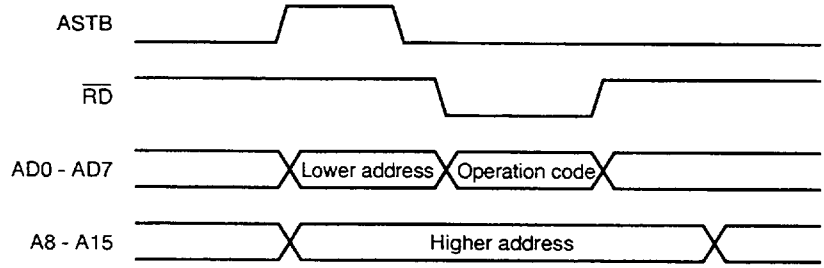
Address/data signal output pins. Valid signal is output or input during data accesses and instruction fetches from external memory.

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

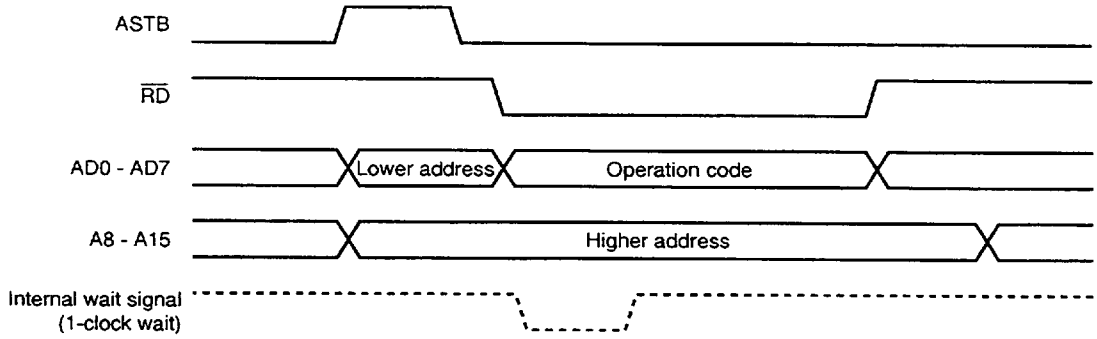
The timing charts are shown in Figures 20-4 to 20-7.

Figure 20-4. Instruction Fetch from External Memory

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



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



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

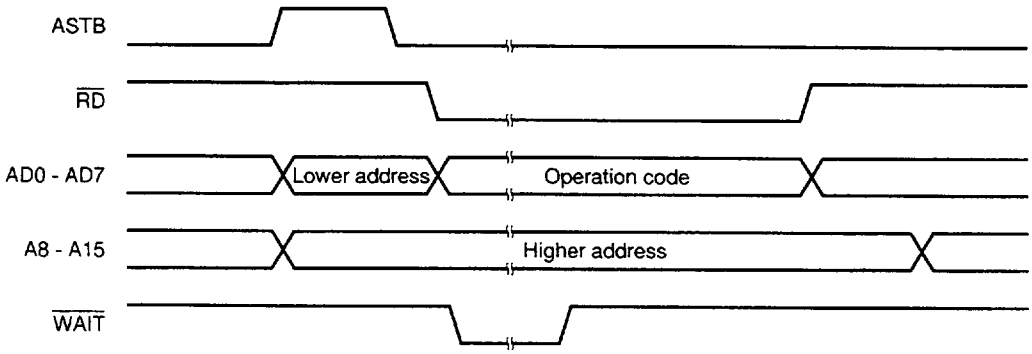
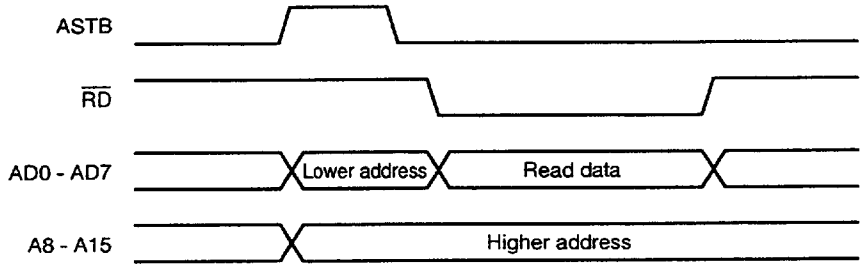
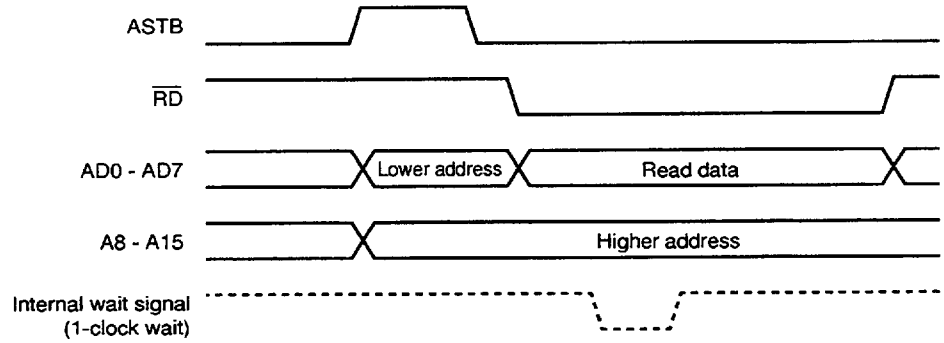


Figure 20-5. External Memory Read Timing

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



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



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

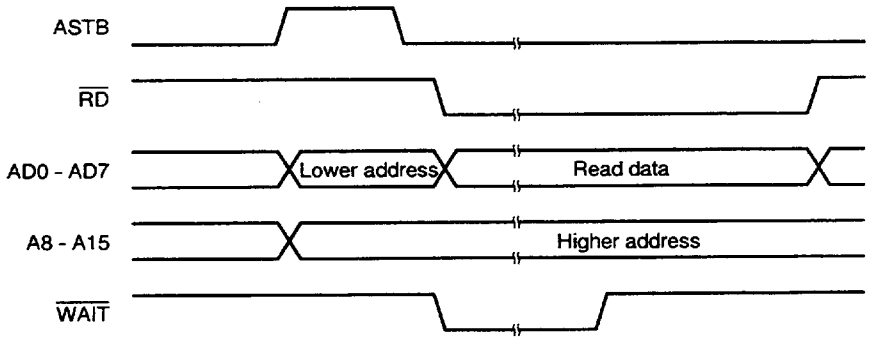
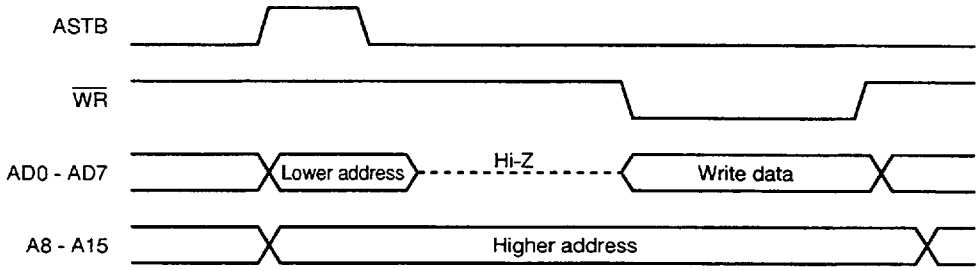
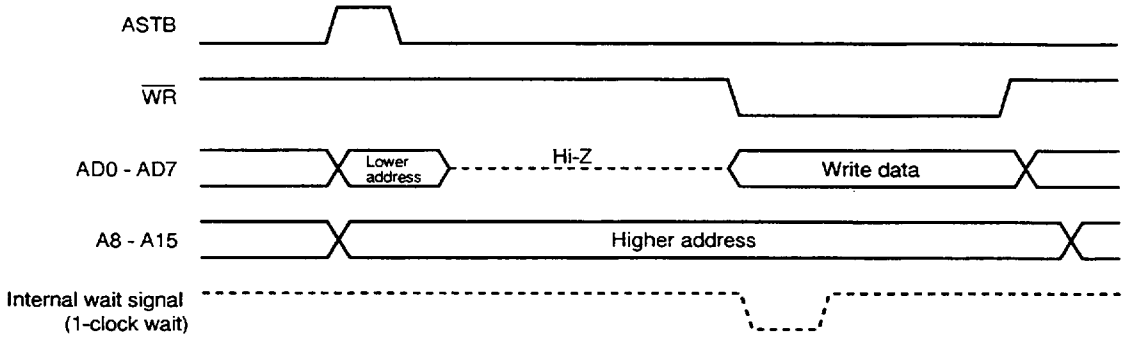


Figure 20-6. External Memory Write Timing

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



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



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

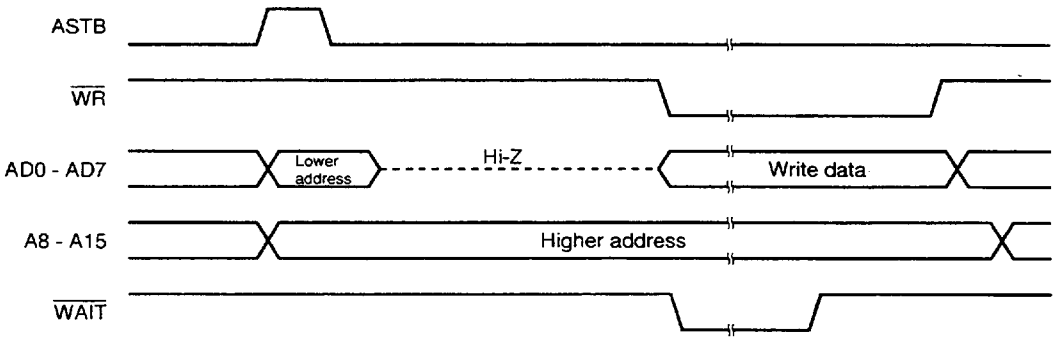
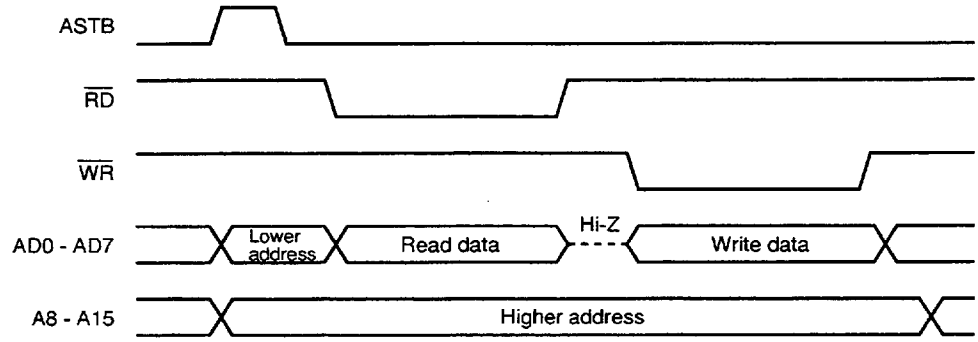
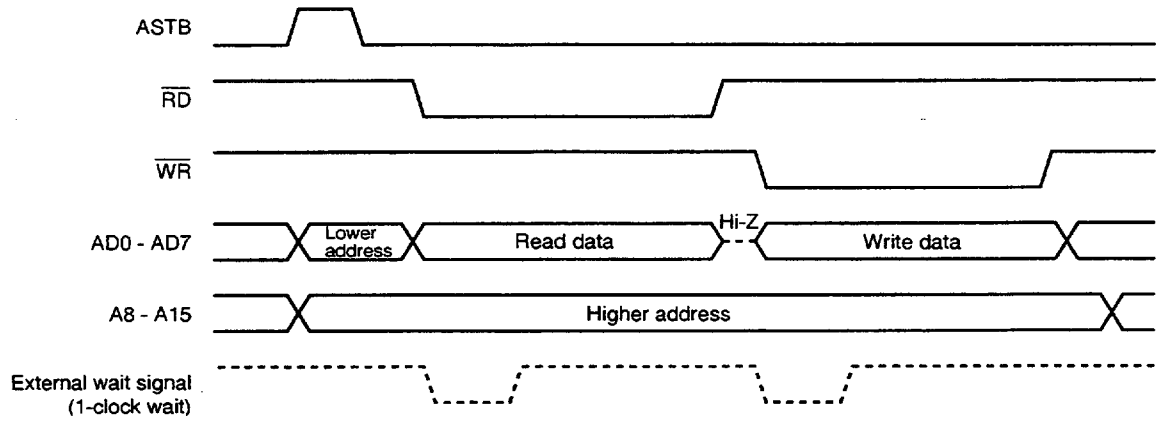


Figure 20-7. External Memory Read Modify Write Timing

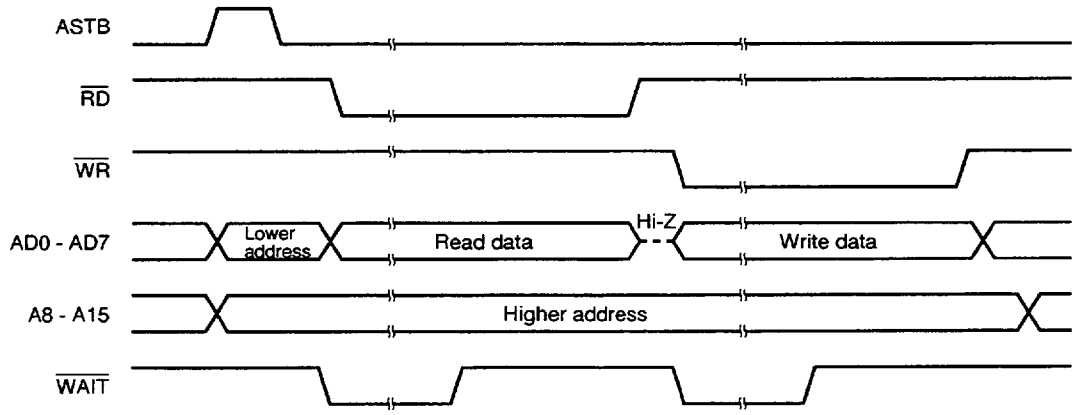
(a) No wait (PW1, PW0 = 0, 0) setting



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



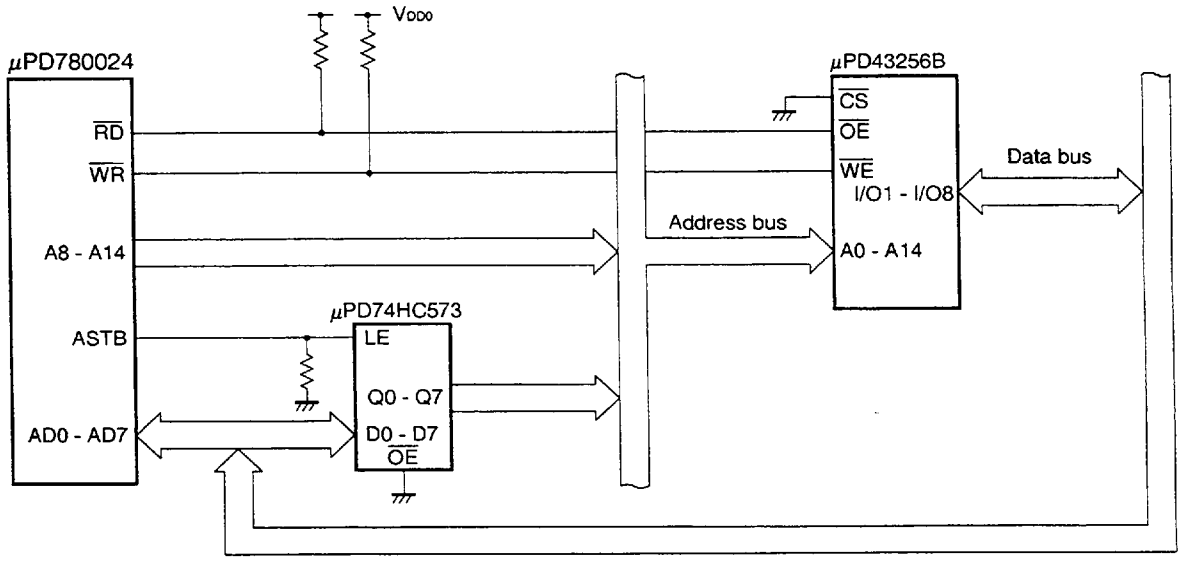
(c) External wait (PW1, PW0 = 1, 1) setting



20.4 Example of Connection with Memory

This section provide an example of connecting the μPD780024 with external memory in Figure 20-8. In addition, the external device expansion function is used in the full-address mode, and the addresses from 0000H to 7FFFH (32 Kbytes) are allocated for internal ROM, and the addresses after 8000H from SRAM.

Figure 20-8. Connection Example of μPD780024 and Memory



PAGE(S) INTENTIONALLY BLANK

CHAPTER 21 STANDBY FUNCTION

21.1 Standby Function and Configuration

21.1.1 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} = 1.8 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 input/output 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 (ADM0) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

21.1.2 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 with an 8-bit memory manipulation instruction.

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

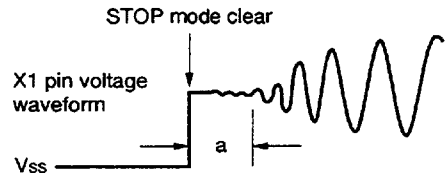
Figure 21-1. Oscillation Stabilization Time Select Register Format

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, regardless of clearance by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses apply to operation with fx = 5.0 MHz.

21.2 Standby Function Operations

21.2.1 HALT mode

(1) HALT mode setting and operating statuses

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 statuses in the HALT mode are described below.

Table 21-1. HALT Mode Operating Statuses

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 TM0		Operable			Operable when TI00 is selected.
8-bit timer/event counter TM5		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		Operable			Operation stops.
Serial interface		Operable			Operable during external SCK.
External interrupt		Operable			
Bus line during external expansion	AD0 - AD7	High impedance			
	A0 - A15	Status before HALT mode setting is held.			
	ASTB	Low level			
	$\overline{WR}$, $\overline{RD}$	High level			
	$\overline{WAIT}$	High impedance			

Notes 1. Including case when external clock is not supplied.

2. Including case when external clock is supplied.

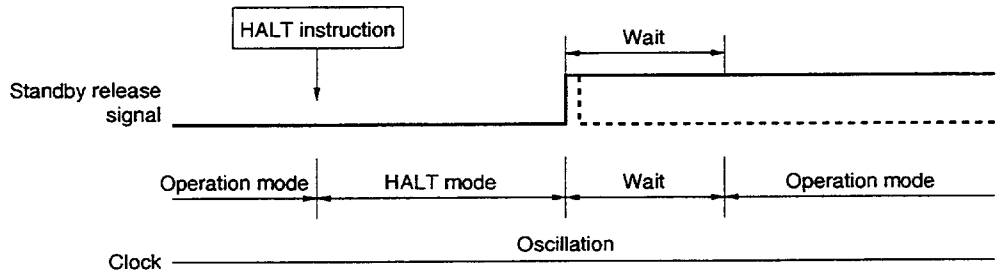
(2) HALT mode clear

The HALT mode can be cleared with the following three types of sources.

(a) Clear upon unmasked interrupt request

An unmasked interrupt request is used to clear the HALT mode. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 21-2. HALT Mode Clear Upon Interrupt Generation



Remarks 1. The broken line indicates the case when the interrupt request which has cleared the standby mode is acknowledged.

2. Wait times are as follows:

- When vectored interrupt service is carried out : 8 to 9 clocks
- When vectored interrupt service is not carried out : 2 to 3 clocks

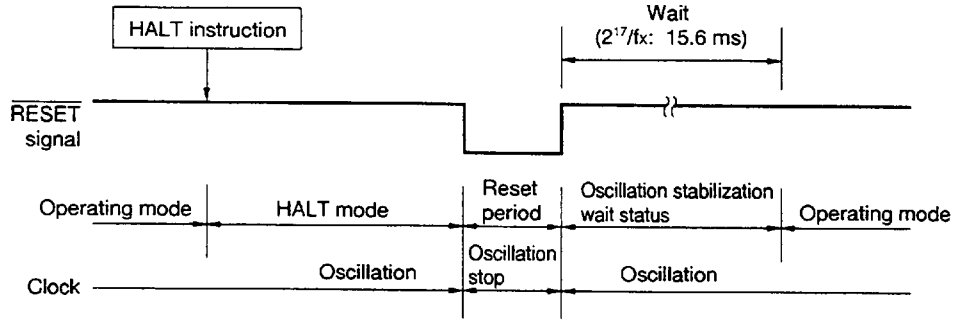
(b) Clear upon non-maskable interrupt request

The HALT mode is cleared and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

(c) Clear upon $\overline{\text{RESET}}$ input

As in the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 21-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses apply to operation with fx = 8.38 MHz.

Table 21-2. Operation after HALT Mode Release

Release Source	MKxx	PRxx	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

21.2.2 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 in Table 21-3 below.

Table 21-3. STOP Mode Operating Status

Item \ STOP Mode Setting		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 TM0		Operation stops.	
8-bit timer/event counter TM5		Operable only when TI50, TI51 are selected as count clock.	
Watch timer		Operable when f_{XT} is selected as counter clock.	Operation stops.
Watchdog timer		Operation stops.	
A/D converter			
Serial interface	Other than UART	Operable only when externally supplied clock is specified as the serial clock.	
	UART	Operation stops.	
External interrupt		Operatable	
Bus line during external expansion	AD0 - AD7	High impedance	
	A0 - A15	Status before STOP mode setting is held.	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

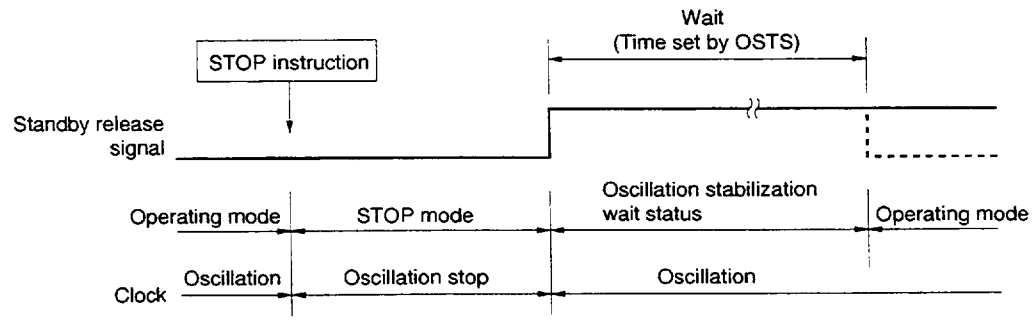
(2) STOP mode release

The STOP mode can be cleared with the following two types of sources.

(a) Release by unmasked interrupt request

An unmasked interrupt reqeust is used to release the STOP mode. In 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 21-4. STOP Mode Release by Interrupt Generation

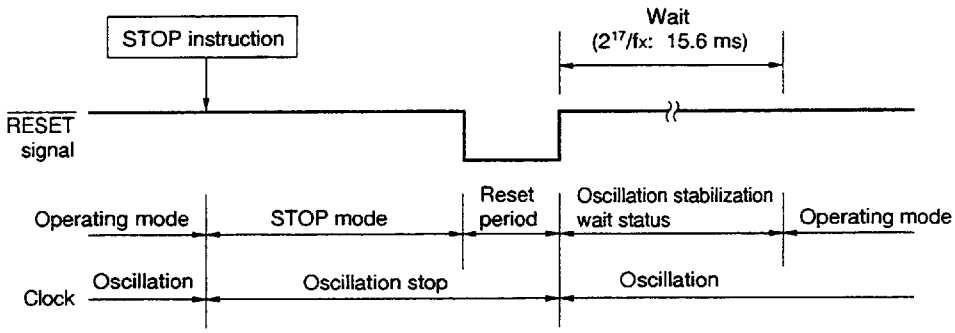


Remark The broken line indicates the case when the interrupt request which has cleared the standby status is aknowledged.

(b) Release by $\overline{\text{RESET}}$ input

The STOP mode is cleared and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 21-5. Release of STOP Mode $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses apply to operation with fx = 8.38 MHz.

Table 21-4. Operation after STOP Mode Release

Release Source	MKxx	PRxx	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 22 RESET FUNCTION

22.1 Reset Function

The following two operations are available to generate the reset function.

(1) External reset input with $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer overrun 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 22-1. 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 22-2 to 22-4).

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 22-1. Block Diagram of Reset Function

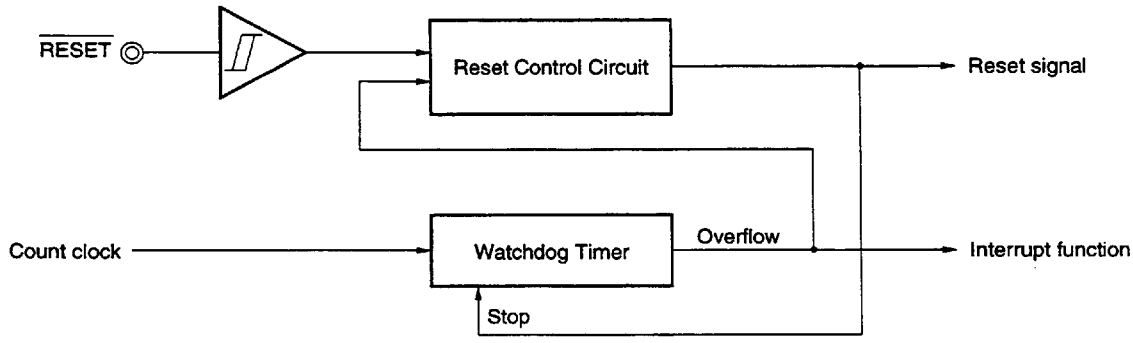


Figure 22-2. Timing of Reset by $\overline{\text{RESET}}$ Input

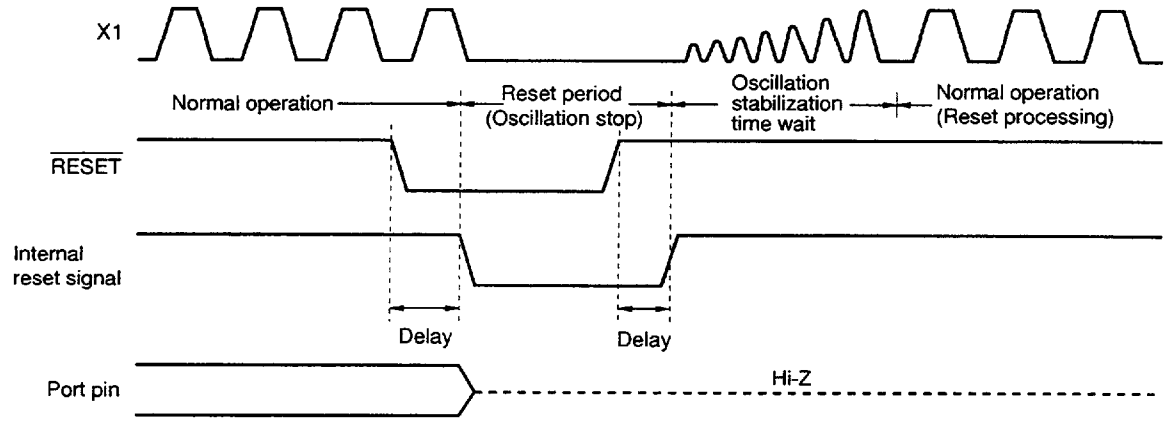


Figure 22-3. Timing of Reset due to Watchdog Timer Overflow

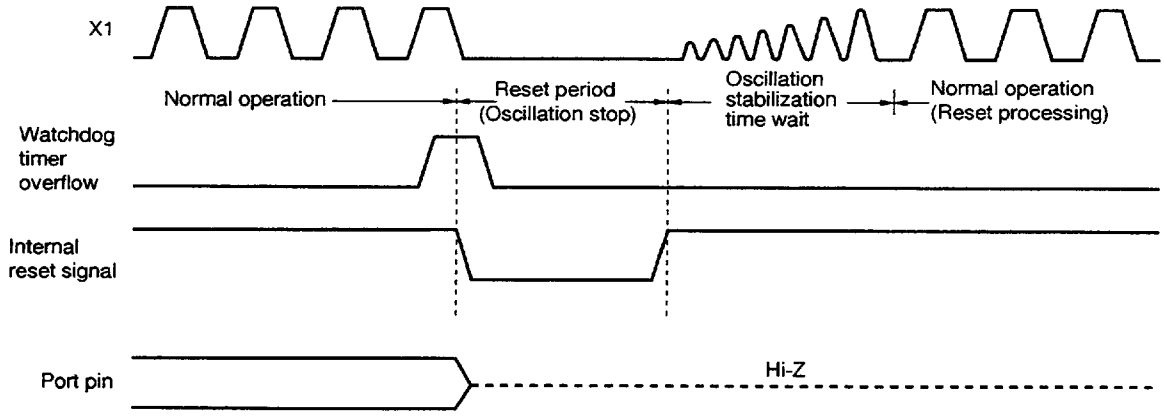


Figure 22-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

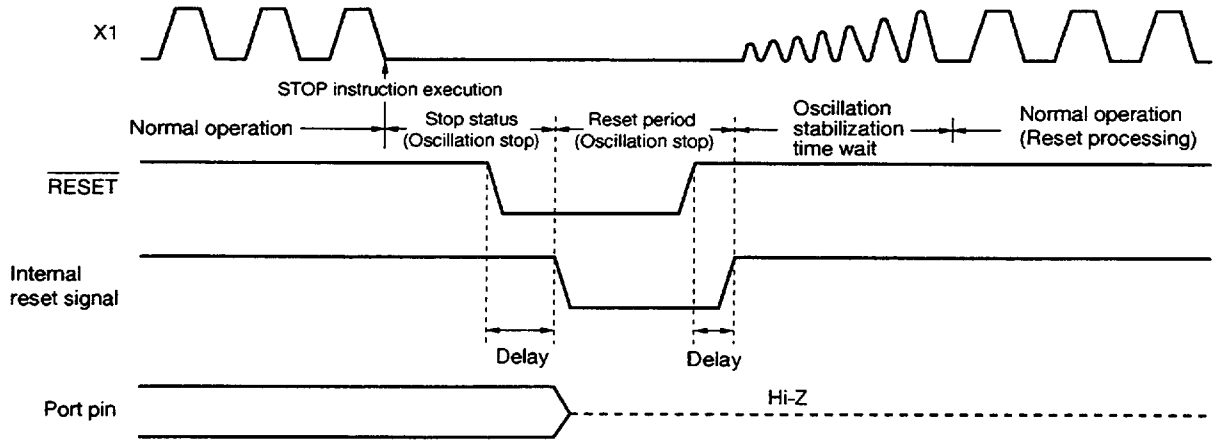


Table 22-1. Hardware Statuses after Reset (1/2)

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 register	Undefined[Note 2]
Port (Output latch)		00H
Port mode registers (PM0, PM2 to PM7)		FFH
Pull-up resistor option registers (PU0, PU2 to PU7)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		CFH[Note 3]
Memory expansion mode register (MEM)		00H
Memory expansion wait setting register (MM)		10H
Oscillation stabilization selection register (OSTS)		04H
16-bit timer/event counter TM0	Timer register (TM0)	00H
	Capture/compare register (CR00, CR01)	Undefined
	Prescaler mode register (PRM0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	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. The post-reset status is held in the standby mode.

3. Although the itntial value is CFH, use the following value to be set for each version.

μPD780021, 780021Y, 780031, 780031Y : 42H
μPD780022, 780022Y, 780032, 780032Y : 44H
μPD780023, 780023Y, 780033, 780033Y : C6H
μPD780024, 780024Y, 780034, 780034Y : C8H
μPD78F0034, 78F0034Y : Value for mask ROM versions

Table 22-1. Hardware Statuses after Reset (2/2)

Hardware		Status After Reset
8-bit timer/event counter TM5	Timer counters (TM50, TM51)	00H
	Compare registers (CR50, CR51)	Undefined
	Clock selection registers (TCL50, TCL51)	00H
	Mode control register (TMC50, TMC51)	04H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Clock selection register (WDCS)	00H
	Mode register (WDTM)	00H
Clock output/buzzer output controller	Clock output selection register (CKS)	00H
A/D converter	Conversion result registers (ADCR0)	Undefined
	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[Note 1])	Undefined
	Operating mode registers (CSIM30, CSIM31[Note 1])	00H
Serial interface IIC0[Note 2]	Clock selection register (IICCL0)	00H
	Shift register (IIC0)	00H
	Control register (IICC0)	00H
	Status register (IICS0)	00H
	Slave address register (SVA0)	00H
Interrupt	Request flag registers (IF0L, IF0H, IF1L)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Priority specify flag registers (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable flag (EGP)	00H
	External interrupt falling edge enable flag (EGN)	00H

Notes 1. Provided only in the μPD780024, 780034 Subseries.

2. Provided only in the μPD780024Y, 780034Y Subseries.

CHAPTER 23 μPD78F0034, 78F0034Y

The flash memory versions of the μPD780024, 780024Y, 780034, 780034Y Subseries include the μPD78F0034 and 78F0034Y.

For purposes of simplification, throughout this chapter, the μPD78F0034 is used to refer to both the μPD78F0034 and 78F0034Y. In the same way, with regard to mask ROM versions, the μPD78F0034 is used to refer to the μPD780021, 780022, 780023, 780024 and the μPD780031, 780032, 780033, 780034, respectively.

The μPD78F0034 replace the internal mask ROM of the μPD780034 with flash memory to which a program can be written, deleted and overwritten while mounted on the substrate. Table 23-1 lists the differences among the μPD78F0034 and the mask ROM versions.

Table 23-1. Differences among μPD78F0034 and Mask ROM Versions

Item	μPD78F0034	Mask ROM Versions
Internal ROM configuration	Flash memory	Mask ROM
Internal ROM capacity	32 Kbytes	μPD780021, 780031: 8 Kbytes μPD780022, 780032: 16 Kbytes μPD780023, 780033: 24 Kbytes μPD780024, 780034: 32 Kbytes
Internal high-speed RAM capacity	1024 bytes	μPD780021, 780031: 512 bytes μPD780022, 780032: 512 bytes μPD780023, 780033: 1024 bytes μPD780024, 780034: 1024 bytes
Change of internal ROM and internal high-speed RAM capacity using memory size switching register IROM and internal high-speed RAM	Possible[Note]	Not provided
IC pin	None	Available
V_{PP} pin	Aailable	None
Electrical characteristics	See data sheet of each product.	

Note $\overline{\text{RESET}}$ input causes the flash memory and internal high-speed RAM capacities to become 32 Kbytes and 1024 bytes, respectively.

Caution Flash memory versions and mask ROM versions differ in their noise tolerance and noise emission. If replacing flash memory versions with mask ROM versions when changing from test production to mass production, be sure to perform sufficient evaluation with CS versions (not ES versions) of mask ROM versions.

23.1 Memory Size Switching Register

The μPD78F0034 allow users to select the internal memory capacity using the memory size switching register (IMS) so that the same memory mapping as that of the μPD780021, 780022, 780023, 780024 and μPD780031, 780032, 780033, 780034 with a different size of internal memory capacity can be achieved.

IMS is set by using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to CFH.

Figure 23-1. Memory Size Switching Register (IMS) Format

Address: FFF0H After Reset: CFH 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 Kbytes
0	1	0	0	16 Kbytes
0	1	1	0	24 Kbytes
1	0	0	0	32 Kbytes
Other than above				Setting prohibited

The IMS settings to obtain the same memory map as mask ROM versions are shown in Table 23-2.

Table 23-2. Memory Size Switching Register Settings

Product	IMS Setting
μPD780021, 780031	42H
μPD780022, 780032	44H
μPD780023, 780033	C6H
μPD780024, 780034	C8H

23.2 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 writer (Flashpro) 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.

Remark Flashpro is a product of Naitoudensei Machida Seisakusho, Co., Ltd.

23.2.1 Selection of transmission method

Writing to flash memory is performed using Flashpro and serial communication. Select the transmission method for writing from Table 23-3. For the selection of the transmission method, a format like the one shown in Figure 23-2 is used. The transmission methods are selected with the V_{PP} pulse numbers shown in Table 23-3.

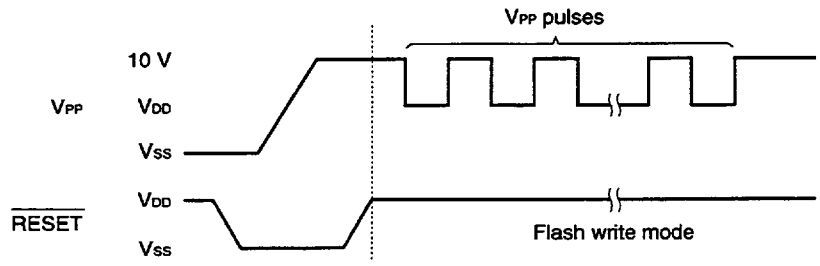
Table 23-3. Transmission Method List

Transmission Method	Number of Channels	Pin Used	Number of V_{PP} Pulses
3-wire serial I/O	1	SI30/P20 SO30/P21 $\overline{SCK30}$/P22	0
3-wire serial I/O (μPD78F0034 only)	1	SI31/P34 SO31/P35 $\overline{SCK31}$/P36	1
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
UART	1	RxD0/P23 TxD0/P24 ASCK0/P25	8
I^2C bus (μPD78F0034Y only)	1	SDA0/P32 SCL0/P33	4

Cautions 1. Be sure to select the number of VPP pulses shown in Table 23-3 for the transmission method.

2. If performing write operations to flash memory with the UART transmission method, set the main system clock oscillation frequency to 3 MHz or higher.

Figure 23-2. Transmission Method Selection Format



23.2.2 Flash memory programming function

Flash memory writing is performed through command and data transmit/receive operations using the selected transmission method. The main functions are listed in Table 23-4.

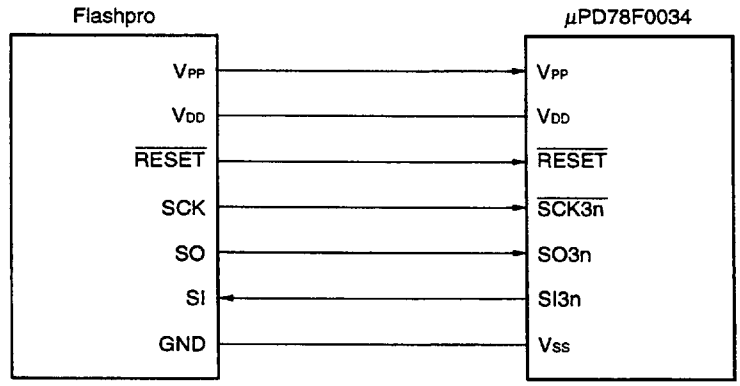
Table 23-4. Main Functions of Flash Memory Programming

Function	Description
Reset	Detects write stop and transmission synchronization.
Batch verify	Compares entire memory contents and input data.
Batch delete	Deletes 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 successive write operations using the data input with high-speed write operation.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Delete time setting	Inputs the memory delete time.
Baud rate setting	Sets the transmission rate when the UART method is used.
I^2C mode setting	Sets the standard/high-speed mode when the I^2C bus method is used.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

23.2.3 Flashpro connection

Connection of Flashpro and μPD78F0034 differs depending on communication method (3-wire serial I/O, UART, and I^2C bus). Each case of connection shows in Figures 23-3, 23-4, and 23-5.

Figure 23-3. Connection of Flashpro Using 3-Wire Serial I/O Method



n = 0, 1

Figure 23-4. Flashpro Connection Using UART Method

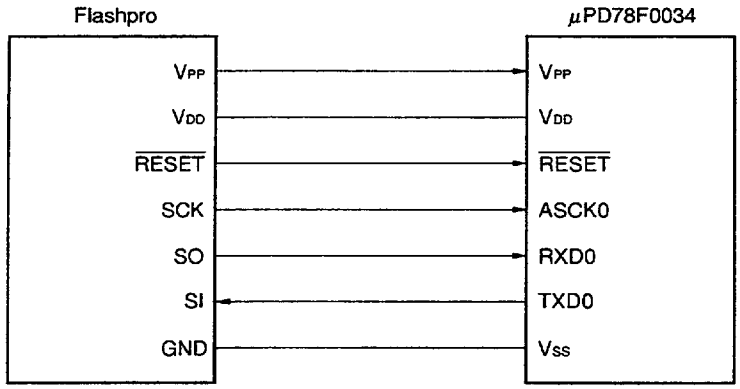


Figure 23-5. Flashpro Connection Using I²C Bus Method

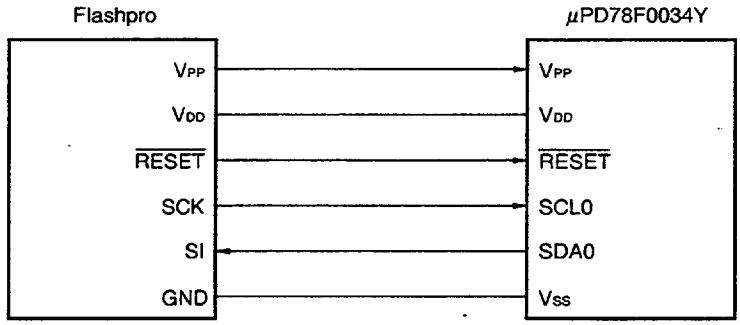
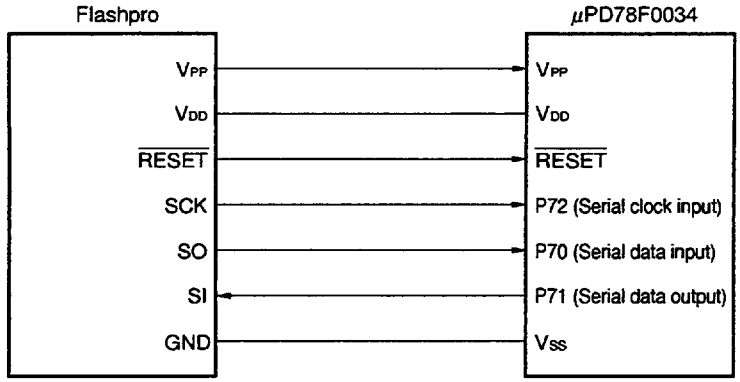


Figure 23-6. Flashpro Connection Using Pseudo 3-wire Serial I/O



CHAPTER 24 INSTRUCTION SET

This chapter describes each instruction set of the μPD780024, 780034, 780024Y, 780034Y subseries as list table. For details of its operation and operation code, refer to the separate document **"78K/0 series USER'S MANUAL—Instruction (IEU-1372)."**

24.1 Legends Used in Operation List

24.1.1 Operand identifiers and description methods

Operands are described in "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and must be described as they are. Each symbol has the following meaning.

- # : Immediate data specification
- ! : Absolute address specification
- $: Relative address specification
- [ ] : Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 24-1. Operand Identifiers and Description Methods

Identifier	Description 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-FF1FH Immediate data or labels
saddrp	FE20H-FF1FH Immediate data or labels (even address only)
addr16	0000H-FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H-0FFFH Immediate data or labels
addr5	0040H-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 5-5 Special-Function Register List.**

24.1.2 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)

24.1.3 Description of "flag operation" column

(Blank) : Nt affected
0 : Cleared to 0
1 : Set to 1
× : Set/cleared according to the result
R : Previously saved value is restored

24.2 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 PCC register.
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 PCC register.
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 PCC register.
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 ← AVbyte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)Vbyte	×
		A, r Note 3	2	4	–	A ← AVr	×
		r, A	2	4	–	r ← rVA	×
		A, saddr	2	4	5	A ← AV(saddr)	×
		A, !addr16	3	8	9 + n	A ← AV(addr16)	×
		A, [HL]	1	4	5 + n	A ← AV(HL)	×
		A, [HL + byte]	2	8	9 + n	A ← AV(HL + byte)	×
		A, [HL + B]	2	8	9 + n	A ← AV(HL + B)	×
		A, [HL + C]	2	8	9 + n	A ← AV(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 PCC register.
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 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7 + n	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8 + n + m	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or 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 PCC register.
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 PCC register.
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) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15 - 11 ← 00001, PC10 - 0 ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (00000000, addr5 + 1), PCL ← (00000000, addr5), SP ← SP – 2	
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)H, (SP – 3) ← (PC + 1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP – 3, IE ← 0	
	RET		1	6	–	PCH ← (SP + 1), PCL ← (SP), SP ← SP + 2	
	RETI		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R R R
	RETB		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1	
		rp	1	4	–	(SP – 1) ← rpH, (SP – 2) ← rpL, SP ← SP – 2	
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R R R
		rp	1	4	–	rpH ← (SP + 1), rpL ← (SP), SP ← SP + 2	
	MOVW	SP, #word	4	–	10	SP ← word	
		SP, AX	2	–	8	SP ← AX	
		AX, SP	2	–	8	AX ← SP	
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16	
		$addr16	2	6	–	PC ← PC + 2 + jdisp8	
		AX	2	8	–	PCH ← A, PCL ← X	
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or 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 (fCPU) selected by the PCC register.
2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	BT	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if(saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11 + 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 PCC register.
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.

24.3 Instructions Listed by Addressing Type

(1) 8-bit instructions
MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand / 1st Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand / First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call/instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand / First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

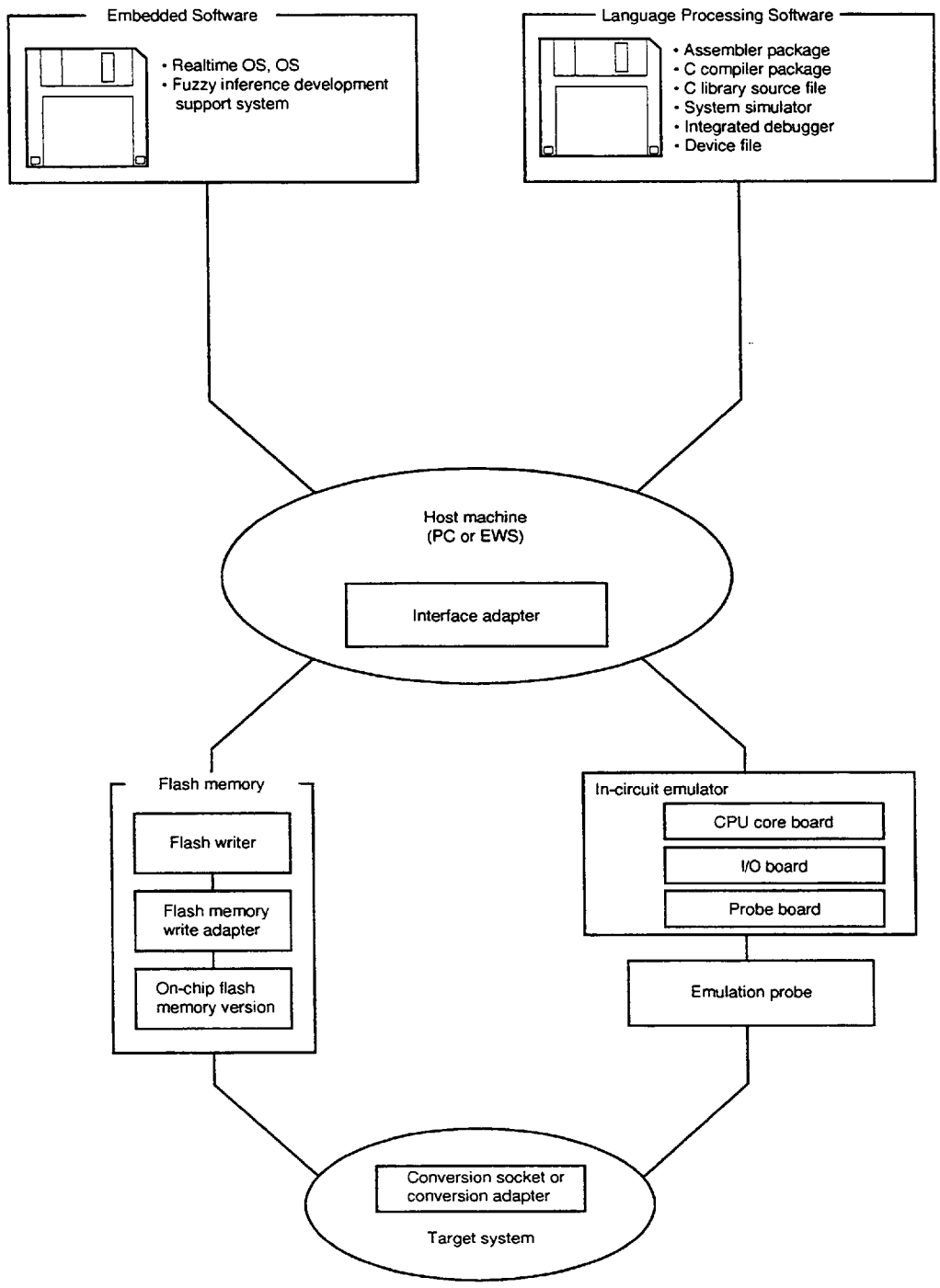
PAGE(S) INTENTIONALLY BLANK

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the μPD780024, 780034, 780024Y, and 780034Y Subseries.

Figure A-1 shows the development tool configuration.

Figure A-1. Development Tool Configuration.



A.1 Language Processing Software

RA78K/0 Assembler Package	This assembler converts programs written in mnemonics into an object code executable with a microcomputer. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler is used in combination with an optional device file (DF780034).
	Part Number: μSxxxxRA78K0
CC78K/0 C Compiler Package	This compiler converts programs written in C language into object code executable with a microcomputer. This compiler is used in combination with an optional assembler package (RA78K/0) and device file (780034).
	Part Number: μSxxxxCC78K0
DF780034[Note 1, 2] Device File	This file contains information peculiar to the device. This file is used in combination with the RA78K/0, CC78K/0, SM78K0, and ID78K0.
	Part Number: μSxxxxDF780034
CC78K/0-L C Library Source File	This is a source program of functions configuring the object library included in the C compiler package (CC78K/0). It is required for matching the object library included in the CC78K/0 with to the customer's specifications.
	Part Number: μSxxxxCC78K0-L

Notes 1. Used in common with DF780034, RA78K/0, CC78K0, SM78K0 and ID78K0.
2. Under development

Remark xxxx in the part number differs depending on the host machine and OS used.

μSxxxxRA78K0
μSxxxxCC78K0
μSxxxxDF780034
μSxxxxCC78K0-L

xxxx	Host Machine	OS	Supply Medium
5A13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2[Note])	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible	See A.4	3.5-INCH 2HD
7B10			5-inch 2HD
3H15	HP9000 Series 300™	HP-UX™ (rel.7.05B)	Cartrige tape (QIC-24)
3P16	HP9000 Series 700™	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation™	SunOS™ (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 Series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not supported by the software listed above, although it is provided in MS-DOS version 5.0 and later.

A.2 Flash Memory Writing Tools

Flashpro Flash Writer	Dedicated flash writer for microcontrollers with on-chip flash memory. Flashpro is a product of Naitoudensei Machida Seisakusho, Co., Ltd.
PA-FLASH64CW (Provisional name) Note PA-FLASH64GC (Provisional name) Note PA-FLASH64GK (Provisional name) Note Flash memory writing adapter	μPD780024, 780024Y, 780034, 780034Y Subseries flash memory writing adapter used connected to Flashpro. These are products of Naitoudensei Machida Seisakusho, Co., Ltd. • PA-FLASH64CW: 64-pin plastic shrink DIP (750 mil) • PA-FLASH64GC: 64-pin plastic QFP (14 × 14 mm) • PA-FLASH64GK: 64-pin plastic LQFP (12 × 12 mm)

Note Under development.

A.3 Debugging Tools

A.3.1 Hardware (1/2)

IE-780000-SL Note In-Circuit Emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K/0 Series. It corresponds to integrated debugger ID78K0. This emulator is used in combination with an emulation probe an interface adapter for connection to a host machine. As Ethernet™, this in-circuit emulator supports 10Base-T and 10Base-5 to enable connection to an EWS.
IE-70000-98-IF-B Interface Adapter	This adapater is required when using the PC-9800 Series computer (except notebook type) as the IE-780000-SL host machine.
IE-70000-98N-IF Interface Adapter	This adapter is required when using a PC-9800 Series computer (except notebook type) as the IE-780000-SL host machine. Connection requires a 100-pin notebook-type PC expansion bus connector.
IE-70000-PC-IF-B Interface Adapter	This adapter is required when using an IBM PC/AT or compatible as the IE-780000-SL host machine.
IE-78K0-SL-EM Note CPU Core Board	This board is used to emulate a 78K/0 Series CPU. This board is used in combination with an in-circuit emulator, I/O board, and probe board.
IE-78K0-SL-P01 Note I/O Board	This board is used to perform emulation of device specific peripheral hardware. This board is used in combination with an in-circuit emulator, CPU core board, and probe board.
IE-780034-SL-EM4 Note Probe Board	This board is used to perform mask option settings and pin connection changes.

Note Under development.

A.3.1 Hardware (2/2)

EP-64CW-SL[Note] Emulation Probe		This probe is used to connect the in-circuit emulator and the target system. It is for 64-pin plastic shrink DIP (CW type).
EP-64GC-SL[Note] Emulation Probe		This probe is used to connect the in-circuit emulator and the target system. It is for 64-pin plastic QFP (GC-AB8 type). A 64-pin conversion socket (EV-9200GC-64) is included to facilitate target system development.
	EV-9200GC-64 conversion socket	This conversion socket is used to connect a target system substrate designed to allow mounting of a 64-pin plastic QFP (GC-AB8 type) and the EP-64GC-SL.
EP-64KG-SL[Note] Emulation Probe		This probe is used to connect an in-circuit emulator and target system. It is for 64-pin plastic QFP (GK-8A8 type). A 64-pin conversion socket (TGK-064SBW) is included to facilitate target system development.
	TGK-064SBW conversion adapter	This conversion adapter is used to connect a target system substrate designed to allow mounting of a 64-pin plastic QFP (GK-8A8 type) and the EP-64KG-SL. This conversion adapter is a product of TOKYO ELETEC Corp. (Tokyo: (03)5295-1661). To purchase this product, contact an NEC dealer.

Note Under development.

Remark The EV-9200GC-64 is sold in packages of 5 units.
The TGK-064SBW is sold in single units.

A.3.2 Software (1/2)

SM78K0 System Simulator	This system simulator is used to perform debugging at C source level or assembler level while simulating the operatin of the target system on a host machine. The SM78K0 operates 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 is used in combination with the optional device file DF780034.
	Part Number: μSxxxxSM78K0-L

Remark xxxx in the part number differs depending on the host machine and OS used.

μSxxxxSM78K0

xxxx	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2[Note]) + Windows (Ver. 3.0 to Ver. 3.1)	3.5-inch 2HD
AB13	IBM PC/AT and compatible (Windows Japanese version)	See A.4	3.5-inch 2HC
BB13	IBM PC/AT and compatible (Windows English version)		3.5-inch 2HC

Note The task swap funtion is not supported by the software listed above, although it is provided in MS-DOS version 5.0 and later.

A.3.2 Software (2/2)

ID78K0 Integrated Debugger	This is a control program used to debug the 78K/0 Series. The graphical user interfaces employed are Windows for personal computers and OSF/Motif for EWSs, offering the standard appearance and operability typical of these interfaces. Further, debugging functions supporting C language are reinforced, and the trace result can be displayed in C language level by using a window integrating function that associates the source program, disassemble display, and memory display with the trace result. In addition, it can enhance the debugging efficiency of a program using a real-time OS by incorporating function expansion modules such as a task debugger and system performance analyzer. This debugger is used in combination with an optional device file (DF78014).
	Part Number: μSxxxxID78K0
DF780034[Notes 1, 2] Device File	File containing information peculiar to the device. Used in combination with optional RA78K/0, CC78K/0, SM78K0, or ID78K0.
	Part Number: μSxxxxDF780034

Notes 1. The DF780034 can be used in conjunction with the RA78K/0, CC78K/0, SM78K0, and ID78K0.

2. Under development.

Remark xxxx in the part number differs depending on the host machine and OS used.

μSxxxxID78K0
μSxxxxDF780034

xxxx	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2[Note]) + Windows (Ver. 3.1)	3.5-inch 2HD
AB13	IBM PC/AT or compatible (Japanese Windows)	See A.4	3.5-inch 2HC
BB13	IBM PC/AT or compatible (English Windows)		3.5-inch 2HC
3P16	HP9000 Series 700	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3K13			3.5-inch 2HC
3R16	NEWS™ (RISC)	NEWS-OS™ (6.1x)	1/4 inch CGMT
3R13			3.5-inch 2HC
3M15	EWS4800 Series (RISC)	EWS-UX/V (rel.4.0)	Cartridge tape (QIC-24)

Note The task swap function is not supported by the software listed above, although it is provided in MS-DOS version 5.0 and later.

A.4 OS for IBM PC

The following OSs for IBM PCs are supported.

To operate SM78K0, ID78K0, and FE9200 (see **B.2 Fuzzy Inference Development Support System**), Windows (Ver. 3.0 to Ver. 3.1) is necessary.

OS	Version
PC DOS	Ver. 5.02 to Ver. 6.3
	J6.1/V[Note] to J6.3V[Note]
IBM DOS™	J5.02/V[Note]
MS-DOS	Ver. 5.0 to Ver. 6.22
	5.0/V[Note] to 6.2/V[Note]

Note Only English mode is supported.

Caution Although Ver. 5.0 and above have a task swapping function, this function cannot be used with this software.

A.5 Development Environment when Using IE-78000-R-A

When using the IE-78000-R-A as the in-circuit emulator, the following debugging tools are required.
The software is the same as that used for the IE-780000-SL.

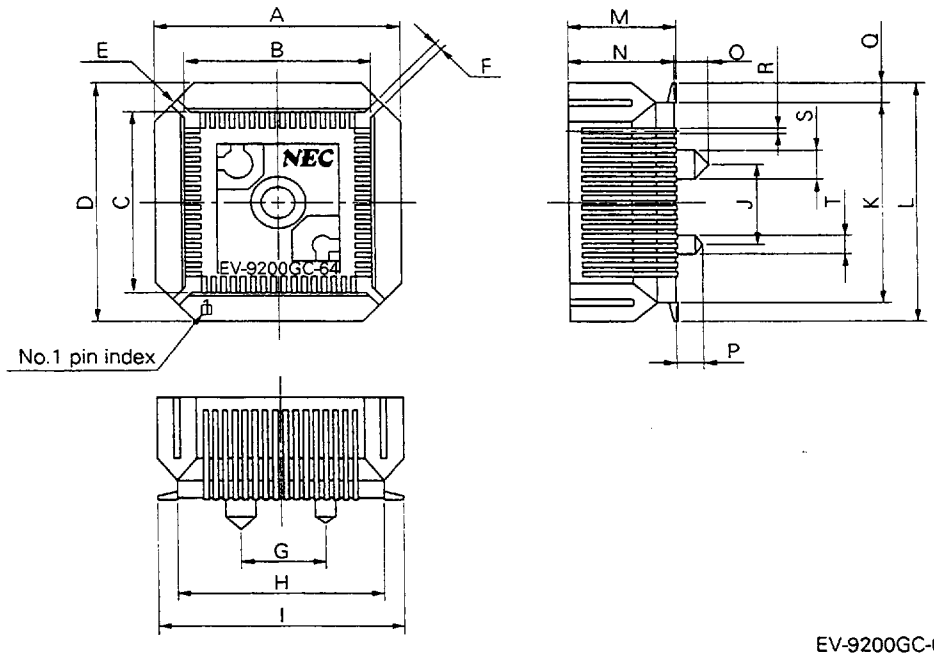
IE-78000-R-A In-Circuit Emulator	This in-circuit emulator is used to debug hardware and software when an application system using the 78K/0 Series is developed. It supports the integrated debugger (ID78K0). This emulator is used in combination with an emulation probe and an interface adapter that connects the emulator with the host machine.
IE-70000-98-IF-B IE-70000-98N-IN IE-70000-PC-IF-B Interface adapter	See **A.3.1 Hardware**.
E-78000-R-SV3 Interface Adapter	Adapter cable necessary when using an EWS as the host machine of the IE-780000-R-A. This cable is connected to the board in the IE-780000-R-A. As Ethernet™, 10Base-5 is supported. If other methods are used, a commercially available conversion adapter is necessary.
IE-780000-R-IF[Note] Interface Board	This board is used connected to the break board (IE-78000-R-BK) in the IE-78000-R-A to provide the interface with the I/O board. This board is used in combination with an in-circuit emulator, I/O board, and probe board.
IE-78K0-SL-P01[Note] I/O Board	This board is used to emulate peripheral hardware peculiar to the device. It is used in combination with an in-circuit emulator, interface board, and probe board. See also **A.3.1 Hardware**.
IE-780034-SL-EM4[Note] Probe Board	See **A.3.1 Hardware**.
EP-64CW-SL[Note] Emulation Probe	
EP-64GC-SL[Note] Emulation Probe EV-9200GC-64 Conversion Socket	
EP-64GK-SL[Note] Emulation Probe TGK-064SBW Conversion Adapter	

Note Under development.

Dimensions of Conversion Socket (EV-9200GC-64) and Recommended Footprint

Figure A-2. EV-9200GC-64 Dimensions (Reference) (Unit: mm)

Based on EV-9200GC-64
(1) Package drawing (in mm)

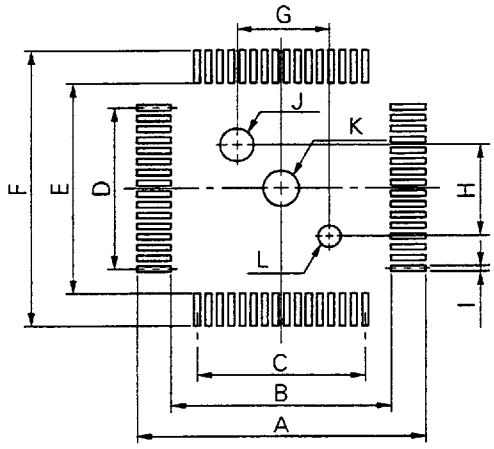


EV-9200GC-64-G0E

ITEM	MILLIMETERS	INCHES
A	18.8	0.74
B	14.1	0.555
C	14.1	0.555
D	18.8	0.74
E	4-C 3.0	4-C 0.118
F	0.8	0.031
G	6.0	0.236
H	15.8	0.622
I	18.5	0.728
J	6.0	0.236
K	15.8	0.622
L	18.5	0.728
M	8.0	0.315
N	7.8	0.307
O	2.5	0.098
P	2.0	0.079
Q	1.35	0.053
R	0.35±0.1	$0.014^{+0.004}_{-0.005}$
S	φ2.3	φ0.091
T	φ1.5	φ0.059

Figure A-3. Dimensions of EV-9200GC-64 and Recommended Footprint (Reference) (Unit: mm)

Based on EV-9200GC-64
(2) Pad drawing (in mm)



EV-9200GC-64-P1E

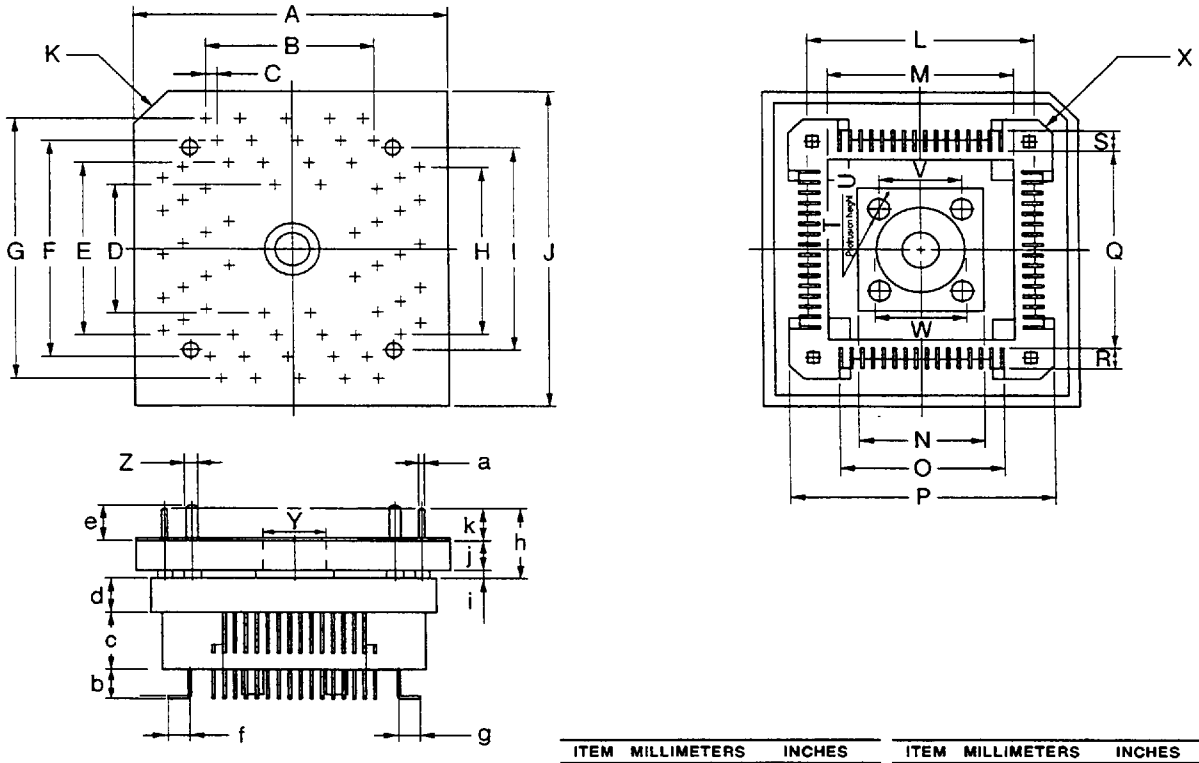
ITEM	MILLIMETERS	INCHES
A	19.5	0.768
B	14.8	0.583
C	$0.8\pm0.02\times15=12.0\pm0.05$	$0.031^{+0.002}_{-0.001}\times0.591=0.472^{+0.003}_{-0.002}$
D	$0.8\pm0.02\times15=12.0\pm0.05$	$0.031^{+0.002}_{-0.001}\times0.591=0.472^{+0.003}_{-0.002}$
E	14.8	0.583
F	19.5	0.768
G	6.00 ± 0.08	$0.236^{+0.004}_{-0.003}$
H	6.00 ± 0.08	$0.236^{+0.004}_{-0.003}$
I	0.5 ± 0.02	$0.197^{+0.001}_{-0.002}$
J	$\phi2.36\pm0.03$	$\phi0.093^{+0.001}_{-0.002}$
K	$\phi2.2\pm0.1$	$\phi0.087^{+0.004}_{-0.005}$
L	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).

Dimensions of Conversion Adapter (TGK-064SBW)

Figure A-4. Dimensions of TGK-064SBW (Reference) (Unit: mm)

Reference diagram: TGK-064SBW
Package dimension (unit: mm)



ITEM	MILLIMETERS	INCHES
A	18.4	0.724
B	0.65x15=9.75	0.026x0.591=0.384
C	0.65	0.026
D	7.75	0.305
E	10.15	0.400
F	12.55	0.494
G	14.95	0.589
H	0.65x15=9.75	0.026x0.591=0.384
I	11.85	0.467
J	18.4	0.724
K	C 2.0	C 0.079
L	12.45	0.490
M	10.25	0.404
N	7.7	0.303
O	10.02	0.394
P	14.92	0.587
Q	11.1	0.437
R	1.45	0.057
S	1.45	0.057
T	4-ϕ1.3	4-ϕ0.051
U	1.8	0.071
V	5.0	0.197
W	ϕ5.3	ϕ0.209
X	4-C 1.0	4-C 0.039
Y	ϕ3.55	ϕ0.140
Z	ϕ0.9	ϕ0.035

ITEM	MILLIMETERS	INCHES
a	ϕ0.3	ϕ0.012
b	1.85	0.073
c	3.5	0.138
d	2.0	0.079
e	3.9	0.154
f	1.325	0.052
g	1.325	0.052
h	5.9	0.232
i	0.8	0.031
j	2.4	0.094
k	2.7	0.106

TGK-064SBW-G0E

note: Product by TOKYO ELETECH CORPORATION.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

APPENDIX B EMBEDDED SOFTWARE

For efficient development and maintenance of the μPD780024, 780034, 780024Y, and 780034Y Subseries, the following embedded software products are available.

B.1 Real-Time OS (1/2)

RX78K/0 Real-time OS	RX78K/0 is a real-time OS conforming with the μITRON specifications. Tool (configurator) for generating nucleus of RX78K/0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K/0) and device file (DF780034).
	Part number: μSxxxxRX78013-ΔΔΔΔ

Caution When purchasing the RX78K/0, fill in the purchase application form in advance and sign the User Agreement.

Remark xxxx and ΔΔΔ in the part number differ depending on the host machine and OS used.

μSxxxxMX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Upper limit of mass-production quantity
001	Evaluation object	Do not use for mass-produced products.
100K	Object for mass-produced product	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

xxxx	Host Machine	OS	Supply Medium
5A13	PC-9800 Series	MS-DOS (Ver. 3.30 to Ver.6.2[Note])	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT and compatible	See A.4.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 Series 300	HP-UX (rel.7.05B)	Cartrige tape (QIC-24)
3P16	HP9000 Series 700	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 Series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not supported by the software listed above, although it is provided in MS-DOS version 5.0 and later.

B.1 Real-Time OS (2/2)

MX78K0 OS	μITRON specification subset OS. Nucleus of MX78K0 is supplied. This OS performs task management, event management, and time management. It controls the task execution sequence for task management and selects the task to be executed next.
	Part number: μSxxxxMX78K0-ΔΔΔ

Remark xxxx and ΔΔΔ in the part number differ depending on the host machine and OS used.

μSxxxxMX78K0-ΔΔΔ

ΔΔΔ	Product Outline	Note
001	Evaluation object	Use for trial product.
xx	Object for mass-produced product	Use for mass-produced product.
S01	Source program	Can be purchased only when object for mass-produced product is purchased.

xxxx	Host Machine	OS	Supply Medium
5A13	PC-9800 Series	MS-DOS (Ver. 3.30 to Ver.6.2[Note])	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT and compatible	See A.4.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 Series 300	HP-UX (rel.7.05B)	Cartrige tape (QIC-24)
3P16	HP9000 Series 700	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 Series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not supported by the software listed above, although it is provided in MS-DOS version 5.0 and later.

B.2 Fuzzy Inference Development Support System

FE9000/FE9200 Fuzzy knowledge data creation tool	Program that supports input, edit, and evaluation (simulation) of fuzzy knowledge data (fuzzy rule and membership function). FE9200 works on Windows.
	Part number: μSxxxxFE9000 (PC-9800 Series) μSxxxxFE9200 (IBM PC/AT and compatible machines)
FT9080/FT9085 Translator	Program that translates fuzzy knowledge data obtained by using fuzzy knowledge data creation tool into assembler source program for RA78K0.
	Part number: μSxxxxFT9080 (PC-9800 Series) μSxxxxFT9085 (IBM PC/AT and compatible machines)
FI78K0 Fuzzy inference module	Program that executes fuzzy inference. Executes fuzzy inference when linked with fuzzy knowledge data translated by translator.
	Part number: μSxxxxFI78K0 (PC-9800 Series, IBM PC/AT and compatible machines)
FD78K0 Fuzzy inference debugger	Support software for evaluation and adjustment of fuzzy knowledge data by using in-circuit emulator and at hardware level.
	Part number: μSxxxxFD78K0 (PC-9800 Series, PC/AT and compatible machines)

Remark xxxx in the part number differs depending on the host machine and the OS used.

μSxxxxFE9000
μSxxxxFT9080
μSxxxxFI78K0
μSxxxxFD78K0

xxxx	Host machine	OS	Supply media
5A13	PC-9800 Series	MS-DOS	3.5" 2HD
5A10		(Ver. 3.30 to Ver. 6.2[Note])	5" 2HD

Note MS-DOS Ver. 5.0 and later have the task swap function, but this function cannot be used for the above software.

μSxxxxFE9200
μSxxxxFT9085
μSxxxxFI78K0
μSxxxxFD78K0

xxxx	Host machine	OS	Supply media
7B13	IBM PC/AT and compatible machines	See A.4.	3.5" 2HC
7B10			5" 2HC

APPENDIX C REGISTER INDEX

C.1 Register Index (In Alphabetical Order with Respect to Register Names)

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

ADCR0 : A/D conversion result register 0
ADM0 : A/D converter mode register
ADS0 : Analog input channel specification register
ASIM0 : Asynchronous serial interface mode register
ASIS0 : Asynchronous serial interface status register
BRGC0 : Baud rate generator control register
CRC0 : Capture/compare control register
CR00 : Capture/compare register 00
CR01 : Capture/compare register 01
CKS : Clock output selection register
CR50 : 8-bit compare register 50
CR51 : 8-bit compare register 51
TM50 : 8-bit counter 50
TM51 : 8-bit counter 51
TMC50 : 8-bit timer mode control register 50
TMC51 : 8-bit timer mode control register 51
EGN : External interrupt falling edge enable register
EGP : External interrupt rising edge enable register
IICCL0 : IIC clock selection register
IICC0 : IIC control register
IIC0 : IIC shift register
IICS0 : IIC status register
MK0H : Interrupt mask flag register 0H
MK0L : Interrupt mask flag register 0L
MK1L : Interrupt mask flag register 1L
IF0H : Interrupt request flag register 0H
IR0L : Interrupt request flag register 0L
IF1L : Interrupt request flag register 1L
MM : Memory expansion wait setting register
IMS : Memory size switching register
MEM : Memory expansion mode register
OSTS : Oscillation stabilization time selection register
P0 : Port 0
P1 : Port 1
P2 : Port 2
P3 : Port 3
P4 : Port 4
P5 : Port 5
P6 : Port 6
P7 : Port 7
PM0 : Port mode register 0
PM2 : Port mode register 2
PM3 : Port mode register 3
PM4 : Port mode register 4
PM5 : Port mode register 5
PM6 : Port mode register 6

PM7 : Port mode register 7
PRM0 : Prescaler mode register
PR0H : Priority specify flag register 0H
PR0L : Priority specify flag register 0L
PR1L : Priority specify flag register 1L
PCC : Processor clock contrtol register
PSW : Program status word
PU0 : Pull-up resistor option register 0
PU2 : Pull-up resistor option register 2
PU3 : Pull-up resistor option register 3
PU4 : Pull-up resistor option register 4
PU5 : Pull-up resistor option register 5
PU6 : Pull-up resistor option register 6
PU7 : Pull-up resistor option register 7
RXB0 : Receive buffer register
RXS0 : Receive shift register
SIO30 : Serial I/O shift register 30
SIO31 : Serial I/O shift register 31
CSIM30 : Serial operation mode register 30
CSIM31 : Serial operation mode register 31
TMC0 : 16-bit timer mode control register
TOC0 : 16-bit timer output control register
TM0 : 16-bit timer register
SVA0 : Slave address register
TCL50 : Timer clock selection register 50
TCL51 : Timer clock selection register 51
TXS0 : Transmit shift register
WTM : Watch timer mode control register
WDTM : Watchdog timer clock selection register
ADM0 : Watchdog timer mode register