USER'S MANUAL

NEC

μPD78024 SUBSERIES

8-BIT SINGLE-CHIP MICROCOMPUTER

μPD78023
μPD78024
μPD78P024

Document No. IEU-1373A
(O. D. No. IEU-842A)
Date Published February 1995 P
Printed in Japan

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed:	μPD78P024DW
The customer must judge the need for license:	μPD78023CW-xxx, μPD78023GF-xxx-3BE μPD78024CW-xxx, μPD78024GF-xxx-3BE μPD78P024CW, μPD78P024GF-3BE

The application circuits and their parameters are for references only and are not intended for use in actual design-in's.

The information in this document is subject to change without notice.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customer must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

- Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
- Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
- Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

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

Anti-radioactive design is not implemented in this product.

M7 94.11

Major changes (1/2)

Page	Description
Throughout	• The following have been developed: μPD78023CW-xxx, μPD78024CW-xxx, μPD78024GF-xxx-3BE • μPD78022CW-xxx, μPD78022GF-xxx-3BE, and μPD78P024KL-R have been deleted. • Series name: The name μPD78024 series has been changed to μPD78024 subseries. • Main system clock oscillation frequency: 4.19 MHz has been changed to 5.0 MHz.
p.8	μPD78018F, 78018FY, 78078, 78078Y, 78083, 78098, and 780208 subseries have been added to **1.6 78K/0 Series Development**.
p.12	**1.9 Mask Option** has been added.
p.24, 26	The P110/FIP18-P117/FIP25 pins I/O circuits in Mask ROM products in **Table 2-1 Pin Input Output Circuit Types** and **Fig. 2-1 Pin Input/Output Circuit List** have been changed from 15-A to 15-C.
p.23	The description of V_{PP} in **2.2.16** has been changed from "it should be connected to Vss" to "It should be directly connected to Vss".
p.23	**2.2.17 IC (Mask ROM Product Only)** has been modified.
p.63	The caution in **4.2.3 Port 2** has been changed.
p.70	**Table 4-3 Setting of Port Mode Registers and Output Latches When Shared Functions of Port Are Used** has been added.
p.93	**Table 5-2 Maximum Time Required for CPU Clock Switchover** has been modified.
p.94	A caution concerning main system clock switching has been added to **5.6.2 System Clock and CPU Clock Switching Procedure**.
p.105	A caution concerning the case of settings for starting/stopping the 16-bit timer register operation has been added to **6.3 (2) 16-bit timer mode control register (TMC0)**.
p.132	fx/2 has been added to the 8-bit Timer Register 1, 2 Count Clock Selection Columns of **Fig. 7-4 Timer Clock Select Register 1 Format**.
p.147, 152	The interval time when operated at fx = 5.0 MHz has been added to **Tables 8-1, 8-3 Interval Timer Interval Time.**
p.150, 157, 167	The count clocks in **Figs. 8-2, 9-2, 11-2 Timer Clock Select Register 2 Format** have been changed from $f_x/2^{11}$ to $f_x/2^{20}$ to $f_x/2^3$ to $f_x/2^{12}$.
p.172	The description concerning the case that an A/D converter is not used has been added to **12.2 (8) AVss pin**.
p.182	**12.5 (6) A/D conversion end interrupt request flag (INTAD)** in the previous edition has been changed to **12.5 (6) Interrupt request flag (ADIF). 12.5 (7) A/D conversion interrupt request flag (ADIF)** has been deleted.
p.194	The clear conditions of acknowledge detection of bit 6 (ACKD) in **13.3 (3) Serial Bus Interface Control Register Format** has been modified.
p.232	The note concerning the wake-up function (C) Interrupt timing specify register in **13.4.4 2-wire Serial I/O Mode Operation** has been modified.
p.255	The description has been changed to indicate that all segment signals and digit signals in **Fig.15-8 Pin Layout for 14-Segment Display** have been requested to be ORed.
p.261 p.282	**CHAPTER 16 INTERRUPT FUNCTIONS** in the previous edition has been changed to **CHAPTER 16 INTERRUPT FUNCTIONS. 16.5 Test Functions** has been added.
p.287	Operating Status in HALT mode in **Table 17-1** has been described for 2 cases: during main system clock operation and subsystem clock operation.

Major changes (2/2)

Page	Description
p.290	Operating Status in STOP mode in **Table 17-3** has been described for 2 cases, when subsystem clock is used and when subsystem clock is not used.
p.297	Cautions 2 has been added to **19.1 Memory Size Switching Register**.
p.298	The R/W setting has been changed from W to R/W in **Fig. 19-1 Memory Size Switching Register Format**.
p.299	A caution for when writing program has been added to **19.2 PROM Programming**.
p.321 to 406 (previous edition)	**20.2 Instruction Codes** and **20.3 Instruction Description** in the previous edition have been deleted.
p.324 to 327 p.333	**APPENDIX A DEVELOPMENT TOOLS** and **APPENDIX B BUILT-IN SOFTWARE** have been modified as follows: • PC DOS version: From Ver. 3.1 to Ver. 3.3-Ver. 5.0 • 3.5-inch 2HC added to IBM PC/AT supply medium • HP9000 series 700 added
p.417, 418 (previous edition)	**APPENDIX C INSTRUCTION INDEX (in Alphabetical Order)** has been deleted.
p.373	**APPENDIX D REVISION HISTORY** has been added.

Major changes in this revision are indicated by stars(★) in the margins.

PREFACE

Intended Readership

This manual has been prepared for user engineers who want to understand the functions of the μPD78024 subseries and design and develop its application systems and programs.

Purpose

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

Organization

Two volumes of the μPD78024 subseries User's Manuals are available: this manual and Instruction Manual (common to the 78K/0 series).

μPD78024 subseries User's Manual	78K/0 series User's Manual - Instruction
• Pin function	• CPU function
• Internal block function	• Instruction set
• Interrupt	• Instruction description
• Other internal peripheral functions	

How to Read This Manual

Before reading this manual, you must have general knowledge of electric and logic circuits and microcomputer.

- □ when you want to understand the functions in general:
  → Read this manual in the order of the contents.
- □ Explanation of register format
  → For items with the bit number enclosed in a circle, that bit name is already defined as a reserved word in RA78K/0, and in a headr file named sfrbit.h in CC78K/0.
- □ For the μPD78024 subseries electrical specifications
  → See the separate **μPD78023 and 78024 Data Sheets** (in planning stage).
- □ To understand the details of the instruction function of the μPD78023, and 78024 subseries
  → Refer to the **78K/0 Series User's Manual - Instruction (IEU-1372)**.
- □ To learn the details of functions and a register whose names are known
  → Refer to **APPENDIX C REGISTER INDEX**.

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
Numeric representation	:	Binary ... xxxx or xxxxB
		Decimal ... xxxx
		Hexadecimal ... xxxxH

Related Documents

Some of the related documents listed below are preliminary versions but are not so specified here.

● Device-related Documents

Document Name	Document Number	
	Japanese	English
μPD78023, 78024 Preliminary Product Information	IP-8827	IP-3318
μPD78P024 Preliminary Product Information	IP-8833	IP-3289
μPD78024 Subseries User's Manual	This manual	IEU-1373
78K/0 Series User's Manual - Instruction	IEU-849	IEU-1372

● Development Tool-related Documents (User's Manuals)

Document Name		Document Number	
		Japanese	English
RA78K Series Assembler Package	Operation	EEU-809	EEU-1399
	Language	EEU-815	EEU-1404
RA78K Series Structured Assembler		EEU-817	EEU-1402
CC78K Series C Compiler	Operation	EEU-656	EEU-1280
	Language	EEU-655	EEU-1284
PG-1500 PROM Programmer		EEU-651	EEU-1335
PG-1500 Controller		EEU-704	EEU-1291
IE-78000-R		EEU-810	EEU-1398
IE-78000-R-BK		EEU-867	EEU-1427
IE-78044-R-EM		EEU-833	EEU-1424
EP-78024CW-R		EEU-947	EEU-1476
EP-78024GF-R		EEU-948	EEU-1477
SD78K/0 Screen Debugger	Beginner's Guide	EEU-852	EEU-1414
	Reference	EEU-816	EEU-1427

Caution The contents of the above related documents are subject to change without notice. Be sure to use the latest version of a document for designing your system.

● **Built-in Software Documents (User's Manuals)**

Document Name	Document Number	
	Japanese	English
Fuzzy Knowledge Data Creation Tool	EEU-829	EEU-1438
78K/0, 78K/II, 87AD Series Fuzzy Inference Development Support System - Translator	EEU-862	EEU-1444

● **Other Documents**

Document Name	Document Number	
	Japanese	English
Package Manual	IEI-635	IEI-1213
Semiconductor Device Mounting Technology Manual	IEI-616	IEI-1207
Quality Grade on NEC Semiconductor Devices	IEI-620	IEI-1209
Semiconductor Device Quality Guarantee Guide	MEI-603	MEI-1202

Caution The contents of the above related documents are subject to change without notice. Be sure to use the latest version of a document for designing your system.

TABLE OF CONTENTS

LIST OF FIGURES (1/4)

LIST OF FIGURES (2/4)

LIST OF FIGURES (3/4)

LIST OF FIGURES (4/4)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL DESCRIPTION

1.1 Features

- **Built-in large-capacity ROM and RAM**

Item / Product Name	Program Memory (ROM)	Data Memory	
		Internal High-Speed RAM	FIP display RAM
μPD78023	24K bytes	512 bytes	32 bytes
μPD78024	32K bytes		
μPD78P024	32K bytes[Note]	512 bytes	

★

Note 24K or 32K bytes selectable by means of the memoy size switching register.

- Instruction execution time changeable from high speed (0.4μs: at main system clock 5.0 MHz operation) to ultra-low speed (122 μs: at subsystem clock 32.768 kHz operation)
- I/O ports: 54
- FIP® controller/driver: Total of 26 display outputs
  - Number of segments: 9 to 16
  - Number of digits : 2 to 16
- 8-bit resolution A/D converter: 8 channels
  - Operating power supply voltage range (V_{DD} = 4.5 to 5.5 V) ★
- Serial interface: 2 channels
  - 3-wire/SBI/2-wire mode: 1 channel
  - 3-wire mode : 1 channel
- Timer: 5 channels
  - 16-bit timer/event counter: 1 channel
  - 8-bit timer/event counter : 2 channels
  - Watch timer : 1 channel
  - Watchdog timer : 1 channel
- 15 vectored interrupts
- 1 test input
- 2 types of built-in clock oscillator circuits (main system clock and subsystem clock)
- Operating power supply voltage range: 2.7 to 6.0 V

1.2 Applications

VCRs, audio equipment, etc.

1.3 Ordering Information

★

Ordering Code	Package	Internal ROM
μPD78023CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78023GF-xxx-3BE[Note]	64-pin plastic QFP (14 × 20 mm)	Mask ROM
μPD78024CW-xxx	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78024GF-xxx-3BE	64-pin plastic QFP (14 × 20 mm)	Mask ROM
μPD78P024CW[Note]	64-pin plastic shrink DIP (750 mil)	One-time PROM
μPD78P024DW[Note]	64-pin ceramic shrink DIP (750 mil)	EPROM
μPD78P024GF-3BE[Note]	64-pin plastic QFP (14 × 20 mm)	One-time PROM

Note Under development

Remark xxx indicates a ROM code number.

1.4 Quality Grade

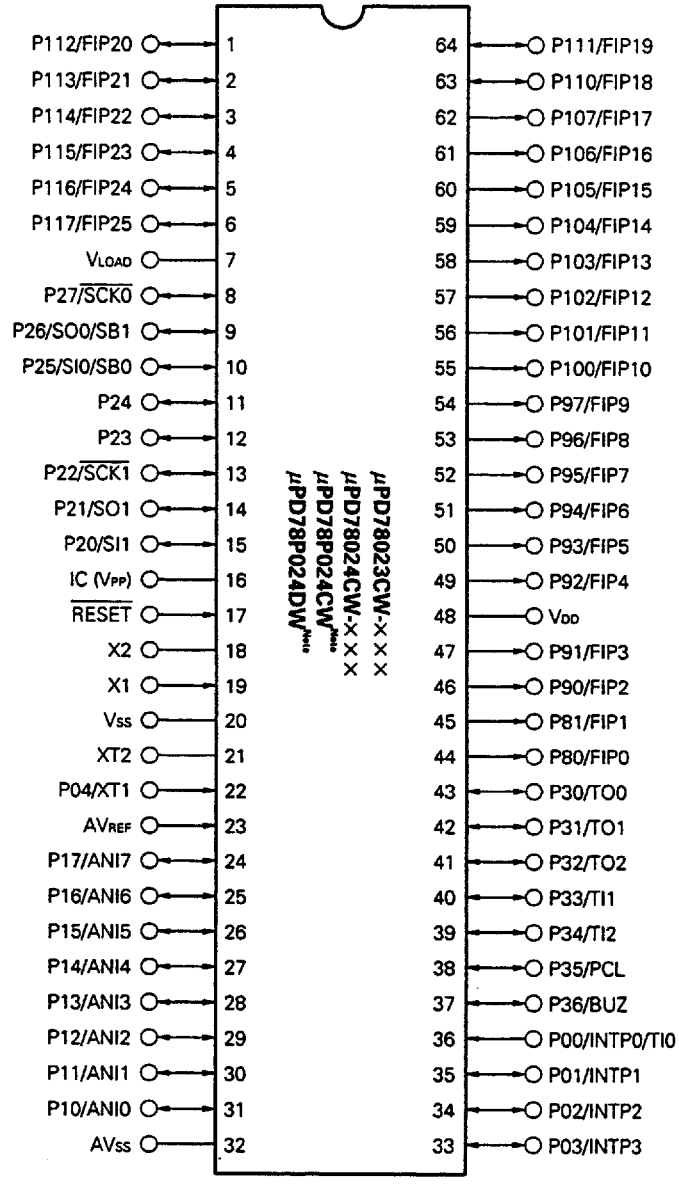
Standard

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)

(1) Normal operation mode

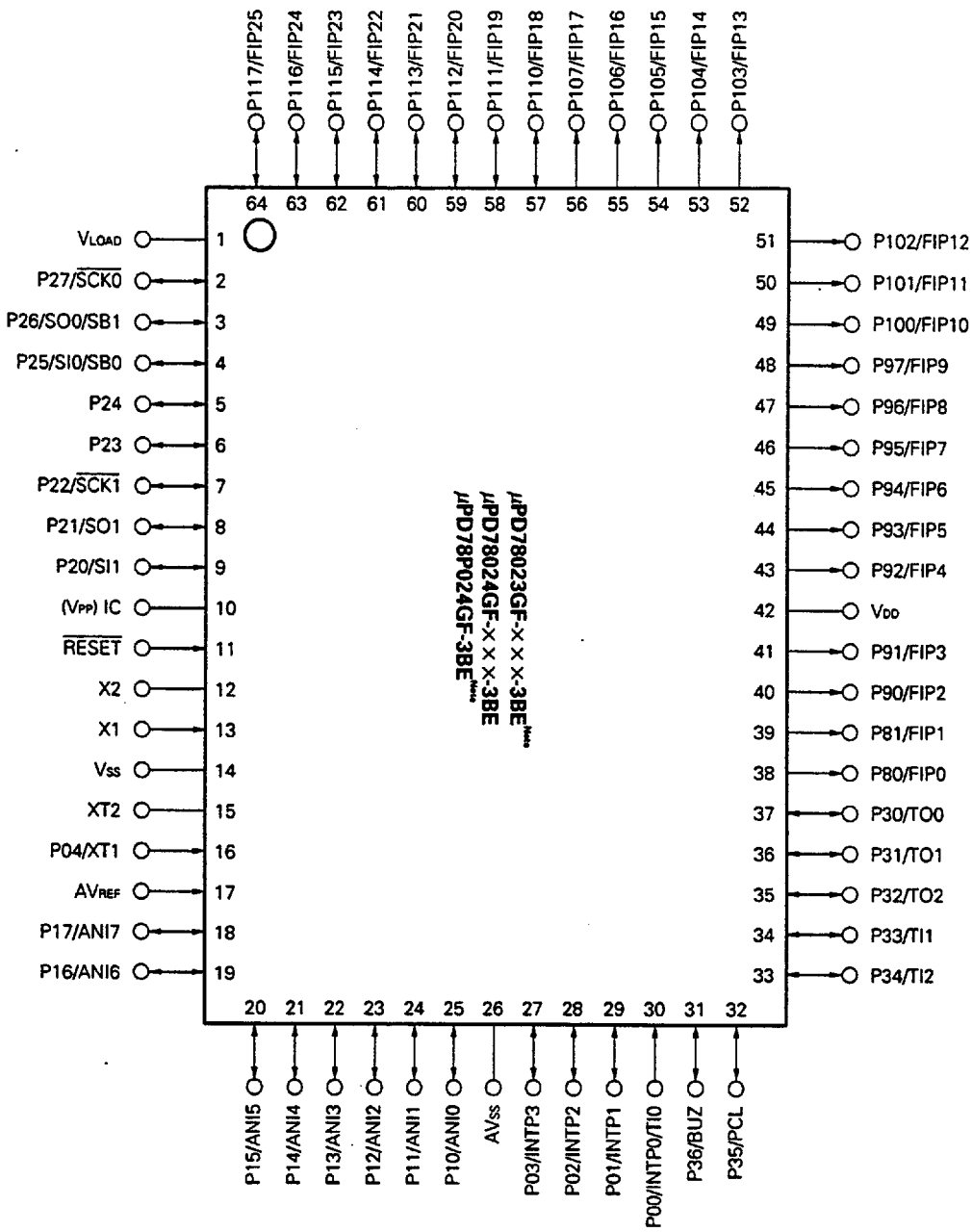
64-pin plastic shrink DIP (750 mil)
64-pin ceramic shrink DIP (750 mil)



Note Under Development

Remarks 1. IC (Internally Connected) should be connected to Vss directly.
2. AVss should be connected to Vss.
3. Connects of ( ) apply to the μPD78P024.

64-pin plastic QFP (14 × 20 mm)



Note Under Development

Remarks 1. IC (Internally Connected) should be connected to Vss directly.

2. AVss should be connected to Vss.

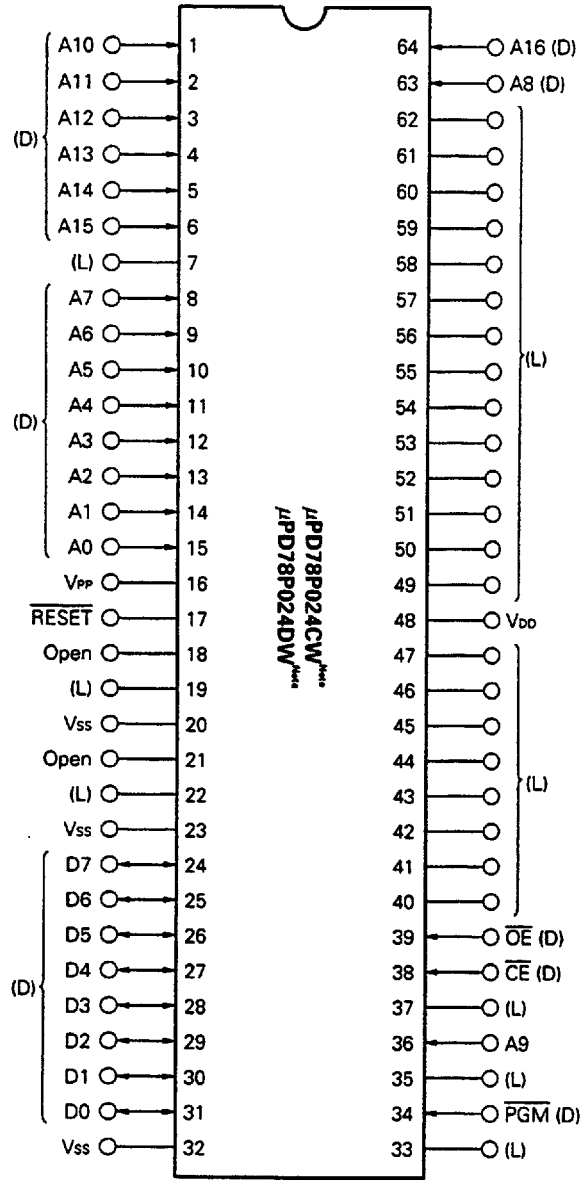
3. Connects of ( ) apply to the μPD78P024.

P00 to P04 : Port 0
P10 to P17 : Port 1
P20 to P27 : Port 2
P30 to P36 : Port 3
P80 and P81 : Port 8
P90 to P97 : Port 9
P100 to P107 : Port 10
P110 to P117 : Port 11
INTP0 to INTP3 : Interrupt From Peripherals
TI0 to TI2 : Timer Input
TO0 to TO2 : Timer Output
SB0, SB1 : Serial Bus
SI0, SI1 : Serial Input
SO0, SO1 : Serial Output
$\overline{SCK0}$, $\overline{SCK1}$: Serial Clock

PCL : Programmable Clock
BUZ : Buzzer Clock
FIP0 to FIP25 : Fluorescent Indicator Panel
V_{LOAD} : Negative Power Supply
X1, X2 : Crystal (Main System clock)
XT1, XT2 : Crystal (Subsystem Clock)
$\overline{RESET}$: Reset
ANI0 to ANI7 : Analog Input
AV_{SS} : Analog Ground
AV_{REF} : Analog Reference Voltage
V_{DD} : Power Supply
V_{PP} : Programming Power Supply
V_{SS} : Ground
IC : Internally Connected

(2) PROM programming mode

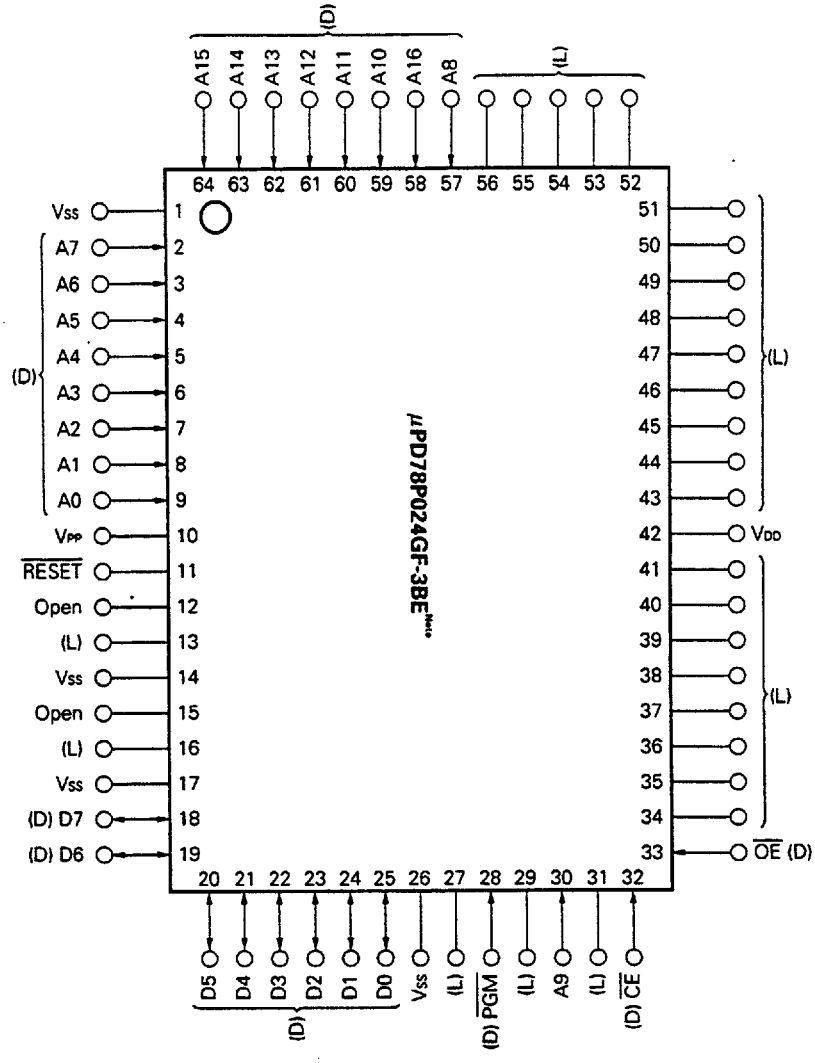
64-pin plastic shrink DIP (750 mil)
64-pin ceramic shrink DIP (750 mil)



Note Under development

Cautions 1. (L) : Connect to Vss individually with a pull-down resistor.
2. (D) : Connect it via the driver.
3. Vss : Connect to ground.
4. RESET: Drive low.
5. Open : Do not make any connection.

64-pin plastic QFP (14 × 20 mm)

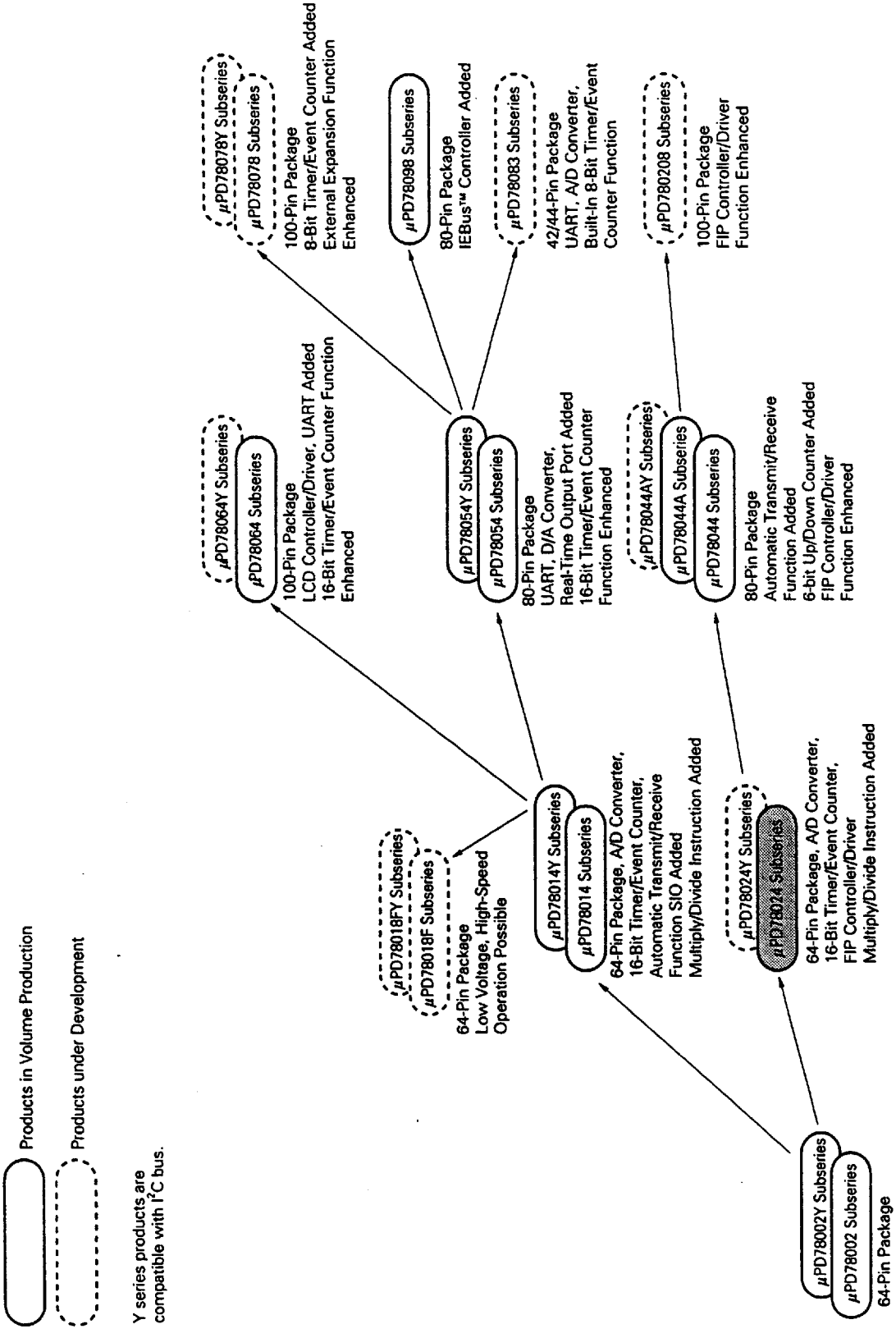


Note Under development

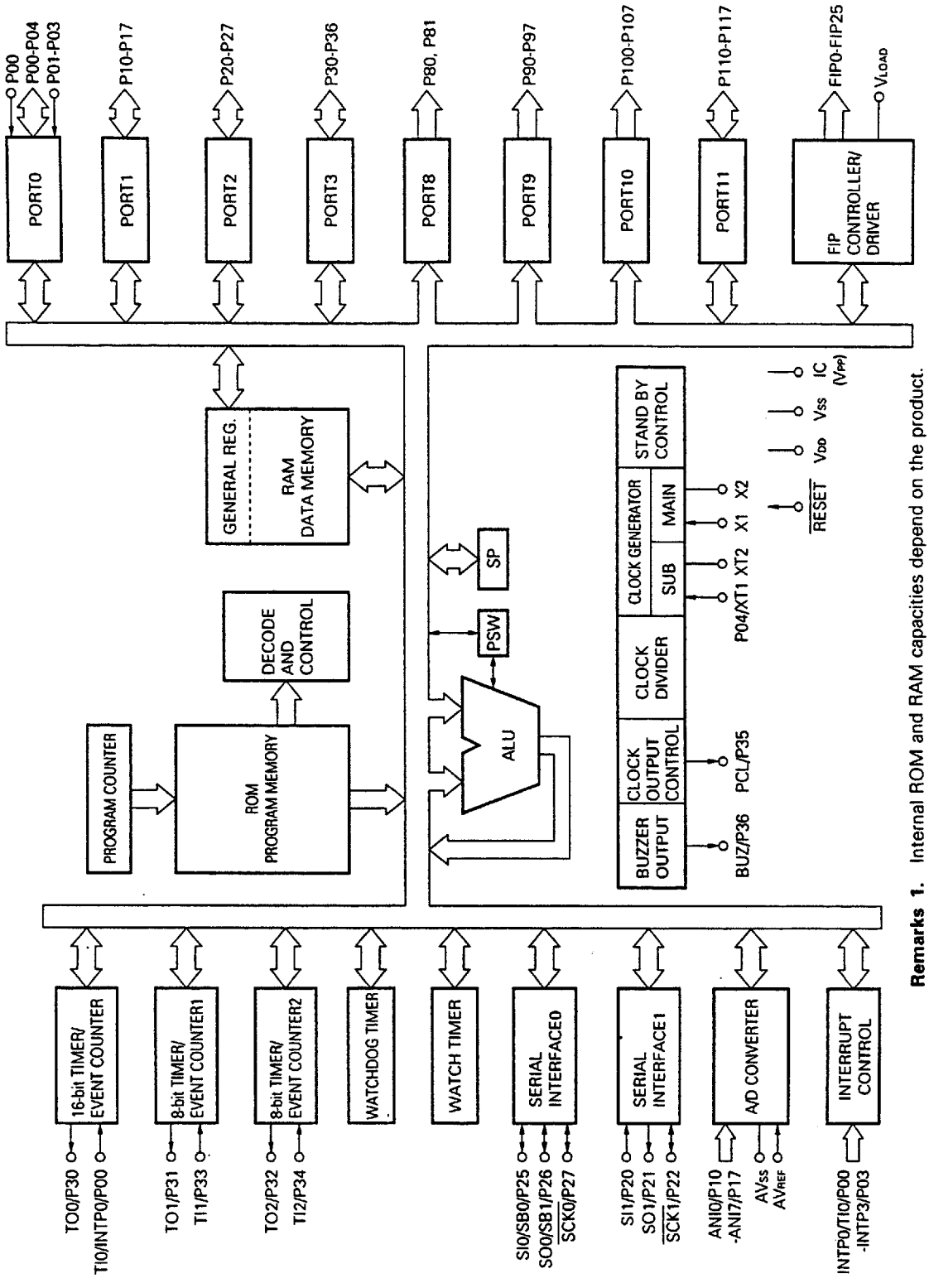
Cautions 1. (L) : Connect to Vss individually with a pull-down resistor.
2. (D) : Connect it via the driver.
3. Vss : Connect to ground.
4. RESET : Drive low.
5. Open : Do not make any connection.

A0 to A16	: Address Bus	RESET	: Reset
D0 to D7	: Data Bus	VDD	: Power Supply
CE	: Chip Enable	VPP	: Programming Power Supply
OE	: Output Enable	Vss	: Ground
		PGM	: Program

★ 1.6 78K/0 Series Development



1.7 Block Diagram



Remarks 1. Internal ROM and RAM capacities depend on the product.

2. Connects of ( ) apply to the μPD78P024.

1.8 Outline of Function

Item \ Product Name			μPD78023	μPD78024	μPD78P024[Note 1]
Internal memory	ROM	Structure	Mask ROM		One-time PROM/EPROM
		Capacity	24K bytes	32K bytes	32K bytes[Note 2]
	Internal high-speed RAM	Capacity	512 bytes		512 bytes
	FIP display RAM	Capacity	32 bytes		
General register			8 bits × 8 registers × 4 banks		
Instruction cycle	with main system clock selected		0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs (when operated at 5.0 MHz)		
	with subsystem clock selected		122 μs (when operated at 32.768 kHz)		
Instruction set			• 8-bit operation, transfer • 16-bit transfer • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, and Boolean operation) • BCD adjustment, etc.		
I/O ports (including FIP dual-function pins)			• Total : 54 • CMOS input : 2 • CMOS input/output : 26 • P-ch open-drain input/output : 8 • N-ch open-drain output : 18		
FIP controller/driver			• Total display outputs : 26 • Number of segments : 9 to 16 • Number of digits : 2 to 16		
A/D Converter			• 8-bit resolution × 8 channels • Operating power supply voltage range : V_{DD}=4.5 to 5.5 V		
Serial interface			• 3-wire/SBI/2-wire mode selection possible : 1 channel • 3-wire 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			3 outputs (14-bit PWM generation possible from one output)		

Notes 1. Under development

2. 24K or 32K bytes selectable by means of the memory size switching register.

Item \ Product Name		μPD78023	μPD78024	μPD78P024[Note]
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz (when operated at 5.0 MHz with main system clock) 32.768 kHz (when operated at 32.768 kHz with subsystem clock)		
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz (when operated at 5.0 MHz with main system clock)		
Vector interrupt	Maskable interrupt	Internal: 9, external: 4		
	Non-maskable interrupt	Internal: 1		
	Software interrupt	Internal: 1		
Test input		Internal: 1		
Operating power supply voltage range		V_{DD}=2.7 to 6.0 V		
Package		• 64-pin plastic shrink DIP (750 mil) • 64-pin plastic QFP (14 x 20 mm) • 64-pin ceramic shrink DIP (750 mil): μPD78P024[Note] only		

Note Under development

★ 1.9 Mask Option

The mask ROM models (μPD78023 and 78024) are provided with mask options. By specifying a mask option when placing your order, pull-up resistors or pull-down resistors can be internally connected in these models as shown in Table 1-1. If these mask options are used when pull-up or pull-down resistors are necessary, the number of external components can be decreased and therefore, the mounting area on a printed wiring board can be reduced.

Table 1-1 lists the mask options available for the μPD78024 subseries.

Table 1-1 Mask Options for Mask ROM Models

Pin Name	Mask Option
P30/TO0-P32/TO2, P33/TI1, P34/TI2, P35/PCL, P36/BUZ, P37	Pull-down resistor can be connected in 1-bit units.
P80/FIP0, P81/FIP1	Pull-down resistor can be connected in 1-bit units. Pull-down resistor can be connected in 2-bit units from P80 to V_{LOAD} or Vss.
P90/FIP2-P97/FIP9, P100/FIP10-P107/FIP17, P110/FIP18-P117/FIP25	Pull-down resistor can be connected in 1-bit units . Pull-down resistor can be connected in 4-bit units from P90 to V_{LOAD} or Vss.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

2.1.1 Normal operation mode pins

(1) Port pins (1/2)

Pin Name	Input/Output	Function		After Reset	Dual-Function Pin
P00	Input	Port 0.	Input only	Input	INTP0/TI0
P01	Input/ output	5-bit input/output port	Input/output specifiable bit-wise. If used as an input port, a pull-up resistor can be connected by software.	Input	INTP1
P02					INTP2
P03					INTP3
P04[Note1]	Input		Input only	Input	XT1
P10 to P17	Input/ output	Port 1. 8-bit input/output port. Input/output can be specified bit-wise. If used as an input port, a pull-up resistor can be connected by software[Note 2].		Input	ANI0 to ANI7
P20	Input/ output	Port 2. 8-bit input/output port. Input/output specifiable bit-wise. If used as an input port, a pull-up resistor can be connected by software.		Input	SI1
P21					SO1
P22					$\overline{SCK1}$
P23					—
P24					—
P25					SI0/SB0
P26					SO0/SB1
P27					$\overline{SCK0}$
P30	Input/ output	Port 3. 7-bit input/output port. Direct LED drive capability. Input/output specifiable bit-wise. If used as an input port, a pull-up resistor can be connected by software. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option.		Input	TO0
P31					TO1
P32					TO2
P33					TI1
P34					TI2
P35					PCL
P36					BUZ

Notes 1. When using the P04/XT1 pin as an input port pin, set the bit 6 (FRC) of the processor clock control register should be set to 1 (do not use one-chip feedback resistor of the subsystem clock oscillator circuit).

2. When using the P10/ANI0 to P17/ANI7 pins as an analog input of the A/D converter, port 1 should be set in input mode. The pull-up resistor cannot be used automatically.

(1) Port pins (2/2)

Pin Name	Input/Output	Function	After Reset	Dual-Function Pin
P80, P81	Output	Port 8. P-ch open-drain 2-bit high-voltage output port. Direct LED drive capability. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option (connection to V_{LOAD} or Vss specifiable in 2-bit units). μPD78P024: pull-down resistor incorporated (connected to V_{LOAD}).	Output	FIP0, FIP1
P90 to P97	Output	Port 9. P-ch open-drain 8-bit high-voltage output port. Direct LED drive capability. Mask ROM products only: bit-wise incorporation of pull-down resistor V_{LOAD} or Vss specifiable in 4-bit units). μPD78P024: pull-down resistor incorporated (connected to V_{LOAD}).	Output	FIP2 to FIP9
P100 to P107	Output	Port 10. P-ch open-drain 8-bit high-voltage output port. Direct LED drive capability. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option (connection to V_{LOAD} or Vss specifiable in 4-bit units). μPD78P024: pull-down resistor incorporated in P100 to P105 (connected to V_{LOAD}).	Output	FIP10 to FIP17
P110 to P117	Input/ output	Port 11. P-ch open-drain 8-bit high-voltage input/output port. Direct LED drive capability. Input/output specifiable bit-wise. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option (connection to V_{LOAD} or Vss specifiable in 4-bit units).	Input	FIP18 to FIP25

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

Pin Name	Input/Output	Function	After Reset	Dual-Function Pin
INTP0	Input	Valid edges (rising edge, falling edge and both rising and falling edges) specifiable	Input	P00/TI0
INTP1				P01
INTP2		External interrupt input		P02
INTP3		Falling edge detected external interrupt input	Input	P03
SI0	Input	Serial interface serial data input	Input	P25/SB0
SI1				P20
SO0	Output	Serial interface serial data output	Input	P26/SB1
SO1				P21
SB0	Input/ output	Serial interface serial data input/output	Input	P25/SI0
SB1				P26/SO0
$\overline{SCK0}$	Input/ output	Serial interface serial clock input/output	Input	P27
$\overline{SCK1}$				P22
TI0	Input	External count clock input to 16-bit timer (TM0)	Input	P00/INTP0
TI1		External count clock input to 8-bit timer (TM1)		P33
TI2		External count clock input to 8-bit timer (TM2)		P34
TO0	Output	16-bit timer output (Dual-function for 14-bit PWM output)	Input	P30
TO1		8-bit timer output		P31
TO2				P32
PCL	Output	Clock output (for main system clock and subsystem clock timing)	Input	P35
BUZ	Output	Buzzer output	Input	P36
FIP0, FIP1	Output	High-voltage, high-current output for FIP controller/driver digit output.	Output	P80, P81
FIP2 to FIP9				P90 to P97
FIP10 to FIP15	Output	High-voltage, high-current outputs for FIP controller/driver digit/segment output.	Output	P100 to P105
FIP16, FIP17	Output	High-voltage outputs for FIP controller/driver segment output.	Output	P106, P107
FIP18 to FIP25			Input	P100-P117
V_{LOAD}	—	FIP controller/driver pull-down resistor connection.	—	—
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
AV_{REF}	Input	A/D converter reference voltage input	—	—
AV_{SS}	—	A/D converter ground potential connected to Vss	—	—
$\overline{RESET}$	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	Input	P04
XT2	—		—	—

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

Pin Name	Input/Output	Function	After Reset	Dual-Function Pin
V_{DD}	—	Positive power supply	—	—
★ V_{PP}	—	Connected directly to V_{SS}.	—	—
V_{SS}	—	Ground potential	—	—
IC	—	Internally connection connected to V_{SS} directly	—	—

2.1.2 PROM programming mode pins (μPD78P024 only)

Pin Name	Input/Output	Function
$\overline{RESET}$	Input	PROM programming mode setting. When +5 V or +12.5 V is applied to the V_{PP} pin or a low level voltage is applied to the $\overline{RESET}$ pin, the PROM programming mode is set.
V_{PP}	Input	High-voltage application for PROM programming mode setting and program write/verify.
A0 to A16	Input	Address bus
D0 to D7	Input/output	Data bus
$\overline{CE}$	Input	PROM enable input/program pulse input
$\overline{OE}$	Input	Read strobe input to PROM
$\overline{PGM}$	Input	Program/program inhibit input in PROM programming mode
V_{DD}	—	Positive power supply
V_{SS}	—	Ground potential

2.2 Description of Pin Functions

2.2.1 P00 to P04 (Port 0)

These ports are 5-bit input/output ports. Besides serving as input/output ports, they function for an external interrupt input, a external count clock input to the timer, a capture trigger signal input and crystal connection for subsystem clock oscillation.

The following operating modes can be specified bit-wise.

(1) Port mode

P00 and P04 function as input-only ports and P01 to P03 function as input/output ports.

P01 to P03 can be specified for input or output ports bit-wise with port mode register 0. When they are used as input ports, a pull-up resistor can be connected to them with a pull-up resistor option register.

(2) Control mode

In this mode, these ports function for an external count clock input to the timer and crystal function for subsystem clock oscillation.

(a) INTP0 to INTP3

INTP0 to INTP2 are external interrupt input pins which can specify valid edges (rising edge, falling edge and both rising and falling edges). INTP0 becomes a 16-bit timer/event counter capture trigger signal input pin with a valid edge input. INTP3 serves as a falling edge-detected external interrupt input pin.

(b) TI0

TI0 is a pin for external count clock input to 16-bit timer/event counter.

(c) XT1

Crystal connect pin for subsystem clock oscillation

2.2.2 P10 to P17 (Port 1)

8-bit input-only ports. Besides serving as input ports, they function for 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. By port mode register 1, these ports can be specified as input port or output port. When these are used as input port, a pull-up resistor can be connected to these ports with a pull-up resistor option register.

(2) Control mode

These ports function as A/D converter analog input pins (ANI0-ANI7). The pull-up resistor is not automatically used for the pins specified for analog input.

2.2.3 P20 to P27 (Port 2)

8-bit input/output ports. Besides serving as input/output ports, they function for data input/output to/from the serial interface and clock input/output.

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 for input or output ports bit- wise with port mode register 2. When they are used as input ports, a pull-up resistor can be connected to them with a pull-up resistor option register.

(2) Control mode

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

(a) SI0, SI1, SO0 and SO1

Serial interface serial data input/output pins

(b) $\overline{\text{SCK0}}$ and $\overline{\text{SCK1}}$

Serial interface serial clock input/output pins

(c) SB0 and SB1

NEC standard serial bus interface input/output pins

Caution When the P22/$\overline{\text{SCK1}}$ and P27/$\overline{\text{SCK0}}$ pins are used for serial clock output, set PM22 and PM27 to 0 and the output latch to 1.

2.2.4 P30 to P36 (Port 3)

7-bit input/output ports. Beside serving as input/output ports, they function for timer input/output, clock output and buzzer output.

The mask ROM products can incorporate a pull-down resistor by mask option.

It can also 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 for input or output ports with port mode register 3. When they are used as input ports, a pull-up resistor can be connected to them with a pull-up resistor option register.

(2) Control mode

These ports function for timer input/output, clock output and buzzer output.

(a) TI1, TI2

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

(b) TO0-TO2

Timer output pins

(c) PCL

Clock output pin

(d) BUZ

Buzzer output pin

2.2.5 P80 and P81 (Port 8)

2-bit dedicated output port. In addition to the output port function, also has an FIP controller/driver digit output function.

Direct LED drive capability.

The following operating modes can be specified bit-wise.

(1) Port mode

Functions as a 2-bit dedicated output port.

P80 and P81 function as P-ch open-drain outputs. On mask ROM products, incorporation of a pull-down resistor is possible with a mask option. The μPD78P024 incorporates a pull-down resistor.

(2) Control mode

P80 and P81 function as FIP controller/driver digit output pins (FIP0, FIP1).

2.2.6 P90 to P97 (Port 9)

8-bit dedicated output port. In addition to the output port function, also has an FIP controller/driver digit output function.

Direct LED drive capability.

The following operating modes can be specified bit-wise.

(1) Port mode

Functions as 8-bit dedicated output port.

P90 to P97 function as P-ch open-drain outputs. On mask ROM products, incorporation of a pull-down resistor is possible with a mask option. The μPD78P024 incorporates a pull-down resistor.

(2) Control mode

P90 to P97 function as FIP controller/driver digit output pins (FIP2-FIP9).

2.2.7 P100 to P107 (Port10)

8-bit dedicated output port. In addition to the output port function, also has an FIP controller/driver digit/segment output function.

P100 to P105 have direct LED drive capability.

The following operating modes can be specified bit-wise.

(1) Port mode

Functions as an 8-bit dedicated output port.

P100 to P107 function as P-ch open-drain outputs. On mask ROM products, incorporation of a pull-down resistor is possible with a mask option. The μPD78P024 incorporates a pull-down resistor in P100 to P105.

(2) Control mode

P100 to P105 function as FIP controller/driver digit/segment dual-function output pins (FIP10-FIP15). P106 and P107 function as FIP controller/driver segment output pins (FIP16, FIP17).

2.2.8 P110 to P117 (Port 11)

8-bit input/output port. In addition to the input/output port function, also has an FIP controller/driver segment output function.

P110 to P117 have direct LED drive capability.

The following operating modes can be specified bit-wise.

(1) Port mode

Functions as an 8-bit input/output port.

P110 to P117 function as P-ch open-drain pins. On mask ROM products, incorporation of a pull-down resistor is possible with a mask option.

(2) Control mode

P110 to P117 function as FIP controller/driver segment output pins (FIP18-FIP25).

2.2.9 AV_{REF}

A/D converter reference voltage input pin.

2.2.10 AVss

A/D converter ground potential pin.

This pin should always be used at the same potential as the Vss pin even when the A/D converter is not used.

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

Low-level active system reset input pin.

2.2.12 X1 and X2

Crystal resonator connect pins for main system clock.

For external clock supply, input it to X1 and the reverse signal to X2.

2.2.13 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

For external clock supply, input it to X1 and the reverse signal to X2.

2.2.14 V_{DD}

Positive power supply pin.

2.2.15 V_{SS}

Ground potential pin.

2.2.16 V_{PP} (μPD78P024 only)

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

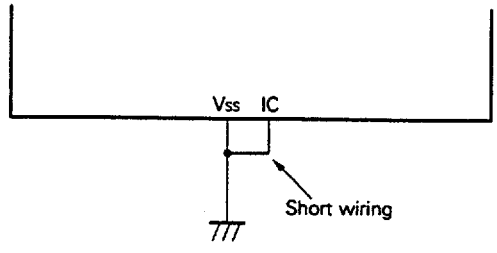
It should be directly connected to Vss in normal operating mode. ★

2.2.17 IC (Mask ROM products only) ★

The IC (Internally Connected) pin sets a test mode when the μPD78023 or 78024 is tested before shipment. Directly connect this pin to the Vss pin in the ordinary operation mode. At that time, keep the wiring length as short as possible.

If the wiring length between the IC and Vss pins is too long, or if external noise is superimposed on the IC pin, a voltage difference occurs between the IC and Vss pins. If this happens, your program may not run correctly.

- **Directly connect the IC pin to the Vss pin.**



2.3 Input/Output Circuit and Recommended Connection of Unused Pins

Table 2-1 shows the input/output circuit types of pins and the recommended connection of unused pins. Refer to Fig. 2-1 for the configuration of the input/output circuit of each type.

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0/TI0	2	Input	Connect to V_{SS}.
P01/INTP1	8-A	Input/output	Input : Connect to V_{SS}. Output: Leave open
P02/INTP2			
P03/INTP3			
P04/XT1	16	Input	Connect to V_{SS}.
P10/ANI0 to P17/ANI7	11	Input/output	Input : Connect to V_{DD} or V_{SS}. Output: Leave Open
P20/SI1	8-A	Input/output	Input : Connect to V_{DD} or V_{SS}. Output: Leave Open
P21/SO1	5-A		
P22/$\overline{SCK1}$	8-A		
P23, P24	5-A		
P25/SI0/SB0	10-A		
P26/SO0/SB1			
P27/$\overline{SCK0}$			
Mask ROM product			
P30/TO0	5-C	Input/output	Input : Connect to V_{DD} or V_{SS}. Output: Leave Open
P31/TO1			
P32/TO2			
P33/TI1	8-B		
P34/TI2			
P35/PCL	5-C		
P36/BUZ			
μPD78P024			
P30/TO0	5-A	Input/output	Input : Connect to V_{DD} or V_{SS}. Output: Leave Open
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			

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

Pin Name		Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
Mask ROM products				
	P80/FIP0, P81/FIP1	14-A	Output	Leave Open
	P90/FIP2 to P97/FIP9			
	P100/FIP10 to P107/FIP17			
	P110/FIP18 to P117/FIP25	15-C	Input/output	Input : Connect to V_{DD} or V_{SS}. Output: Leave open
	IC	—	—	Connect to V_{SS} directly.
μPD78P024				
	P80/FIP0, P81/FIP1	14	Output	Leave open
	P90/FIP2 to P97/FIP9			
	P100/FIP10 to P105/FIP15			
	P106/FIP16, P107/FIP17	14-B	Output	Leave open
	P110/FIP18 to P117/FIP25	15	Input/output	Input : Connect to V_{DD} or V_{SS}. Output: Leave open
	V_{PP}	—	—	Connect directly to V_{SS}.
$\overline{RESET}$		2	Input	—
XT2		16	—	Leave open
AV_{REF}		—		Connect to V_{SS}.
AV_{SS}				
V_{LOAD}				

★

Fig. 2-1 Pin Input/Output Circuit List (1/3)

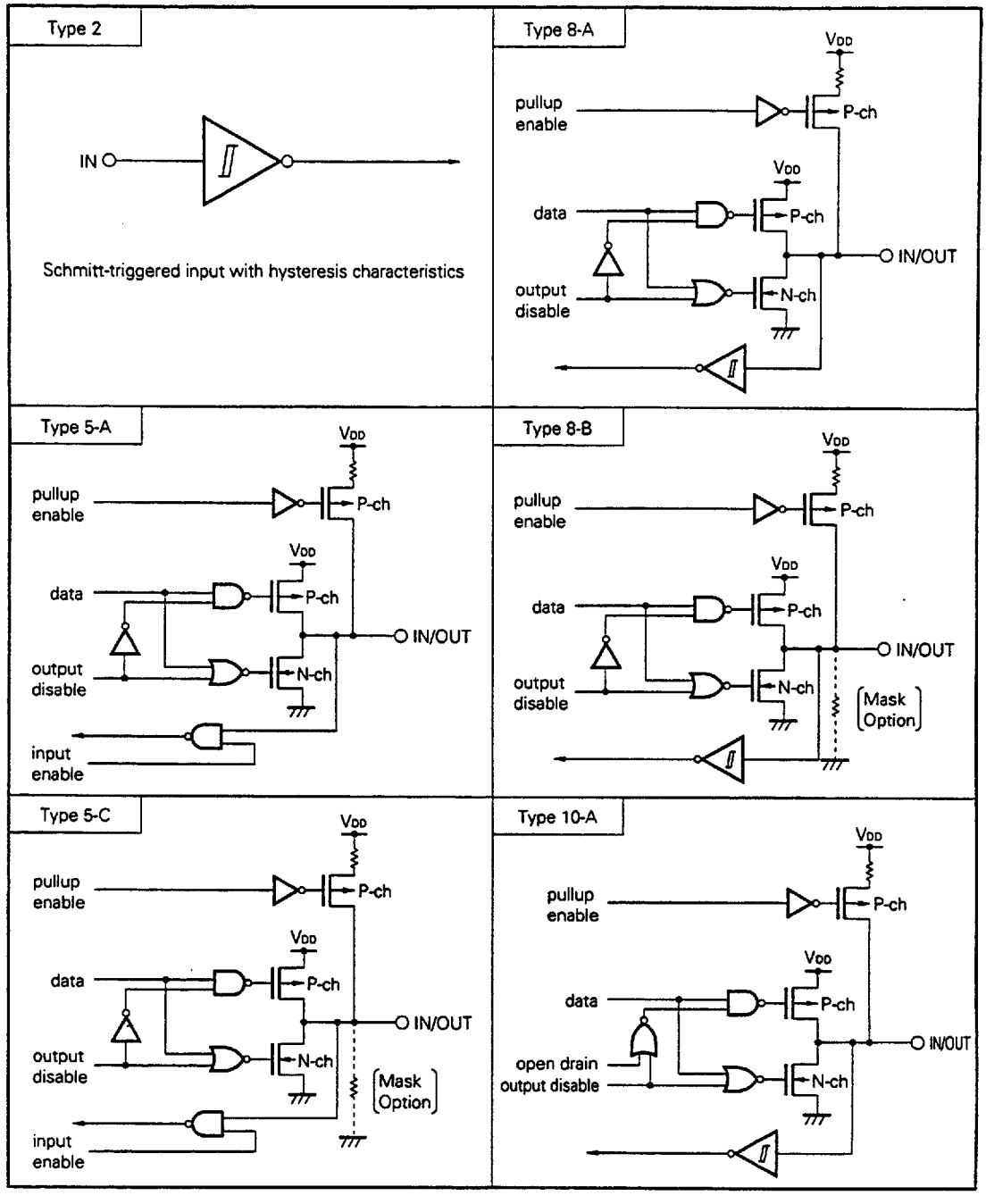


Fig. 2-1 Pin Input/Output Circuit List (2/3)

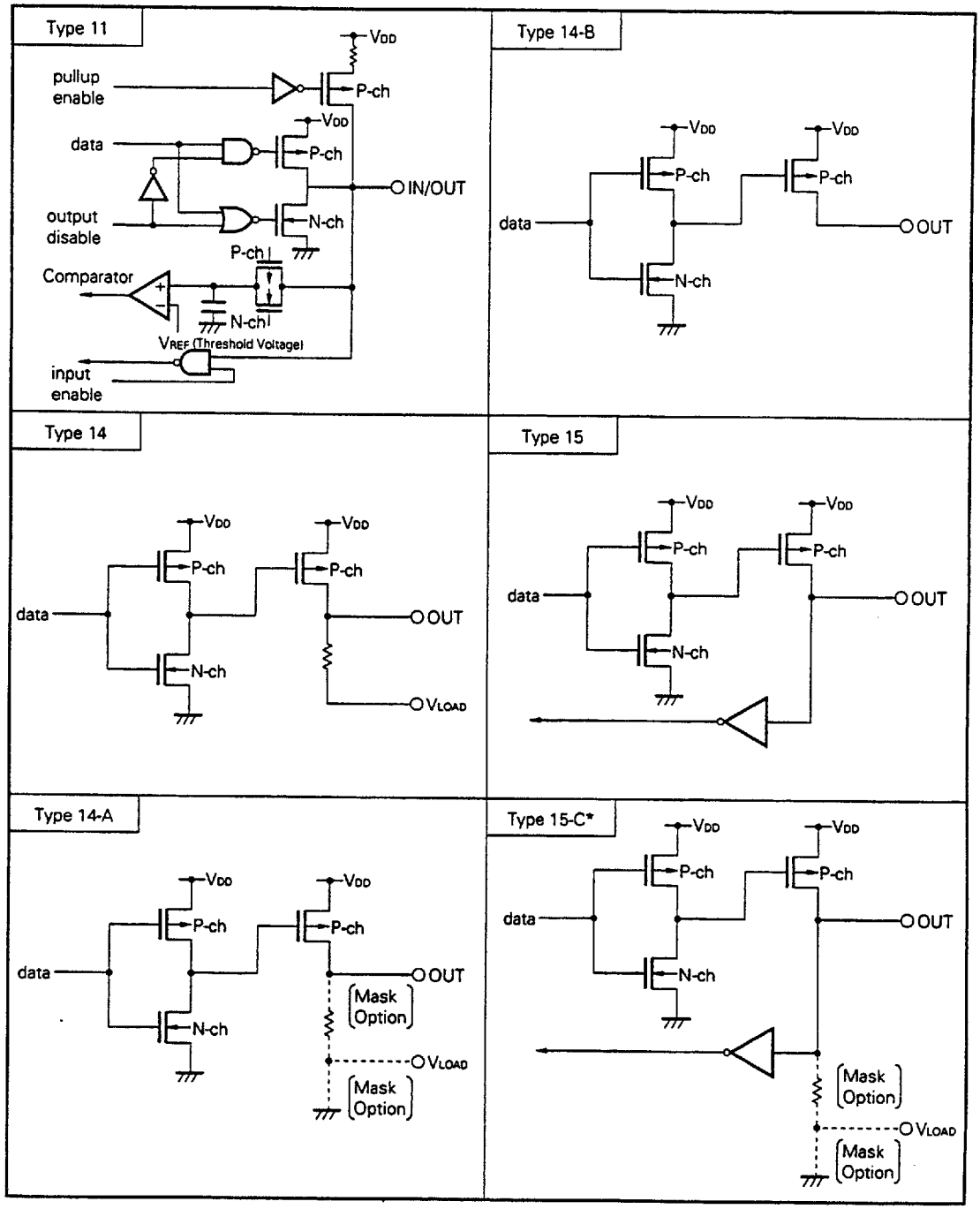
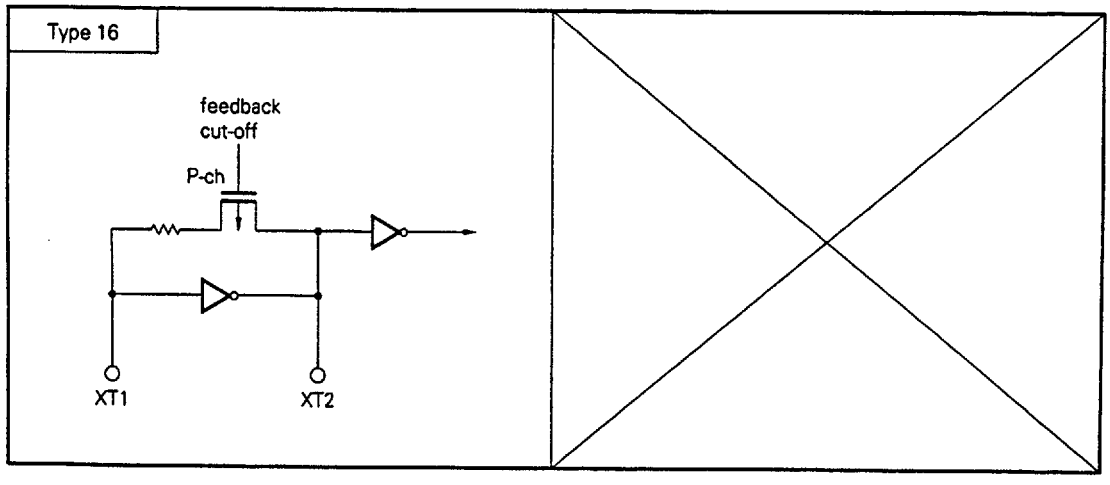


Fig. 2-1 Pin Input/Output Circuit List (3/3)



3.1 Memory Space

Figs. 3-1 to 3-3 show memory maps.

Fig. 3-1 Memory Map (μPD78023)

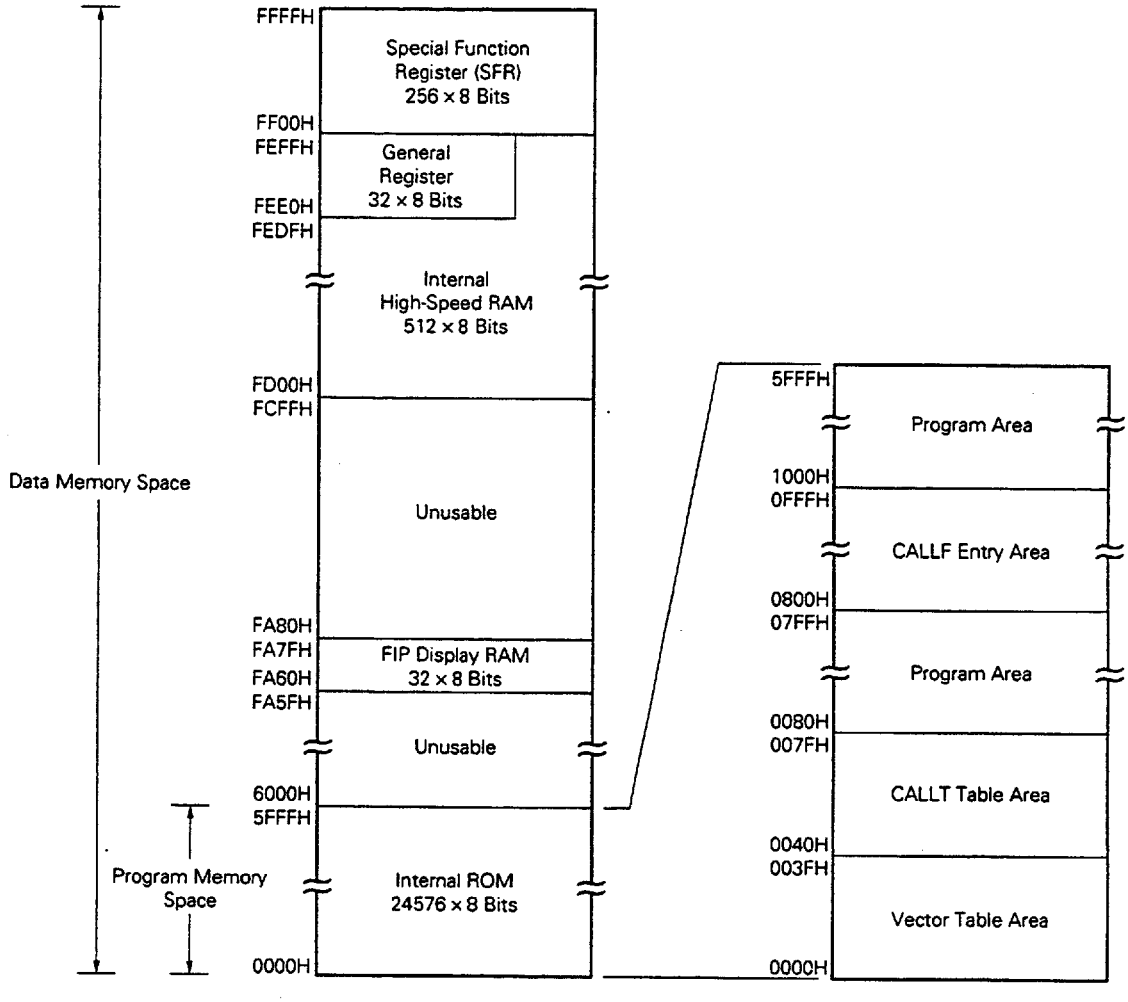


Fig. 3-2 Memory Map (μPD78024)

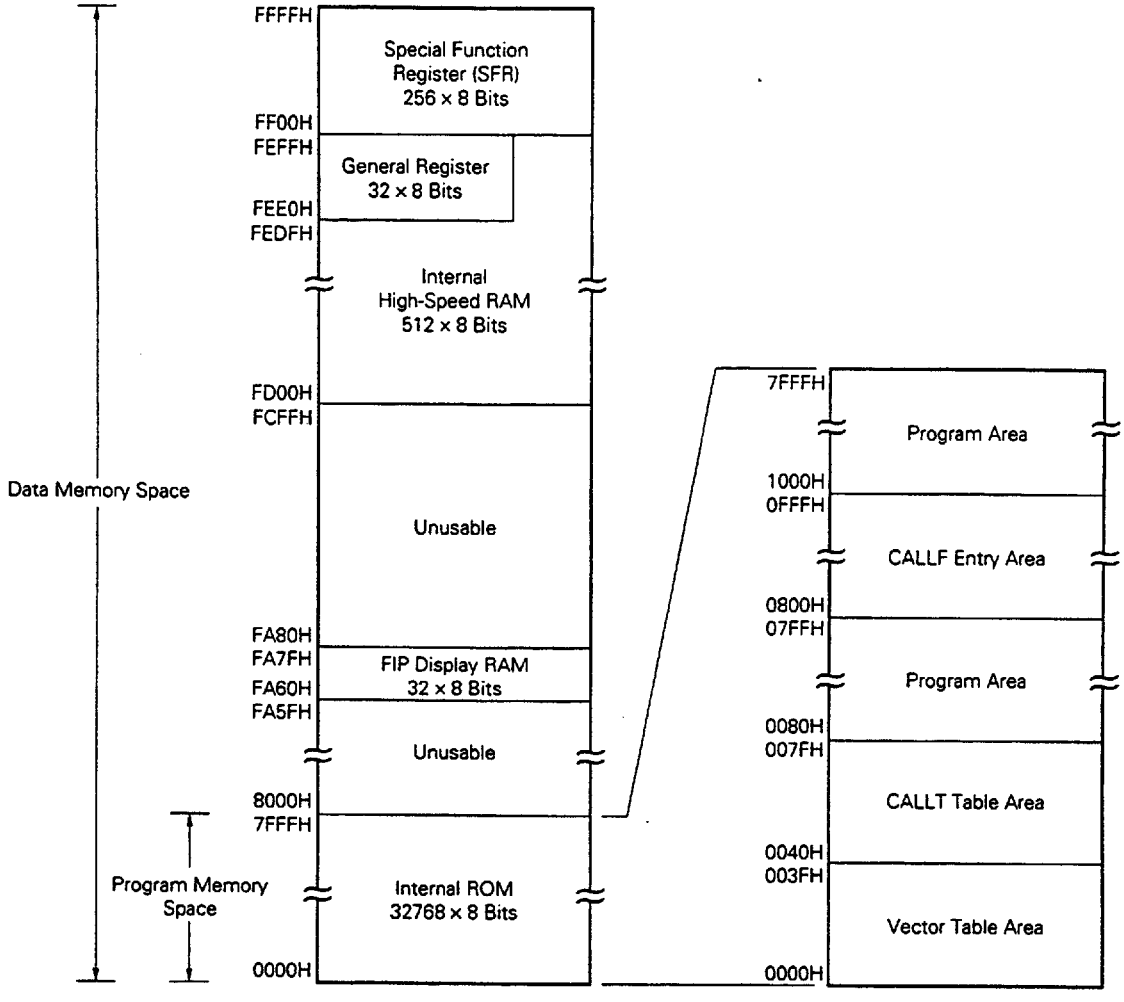
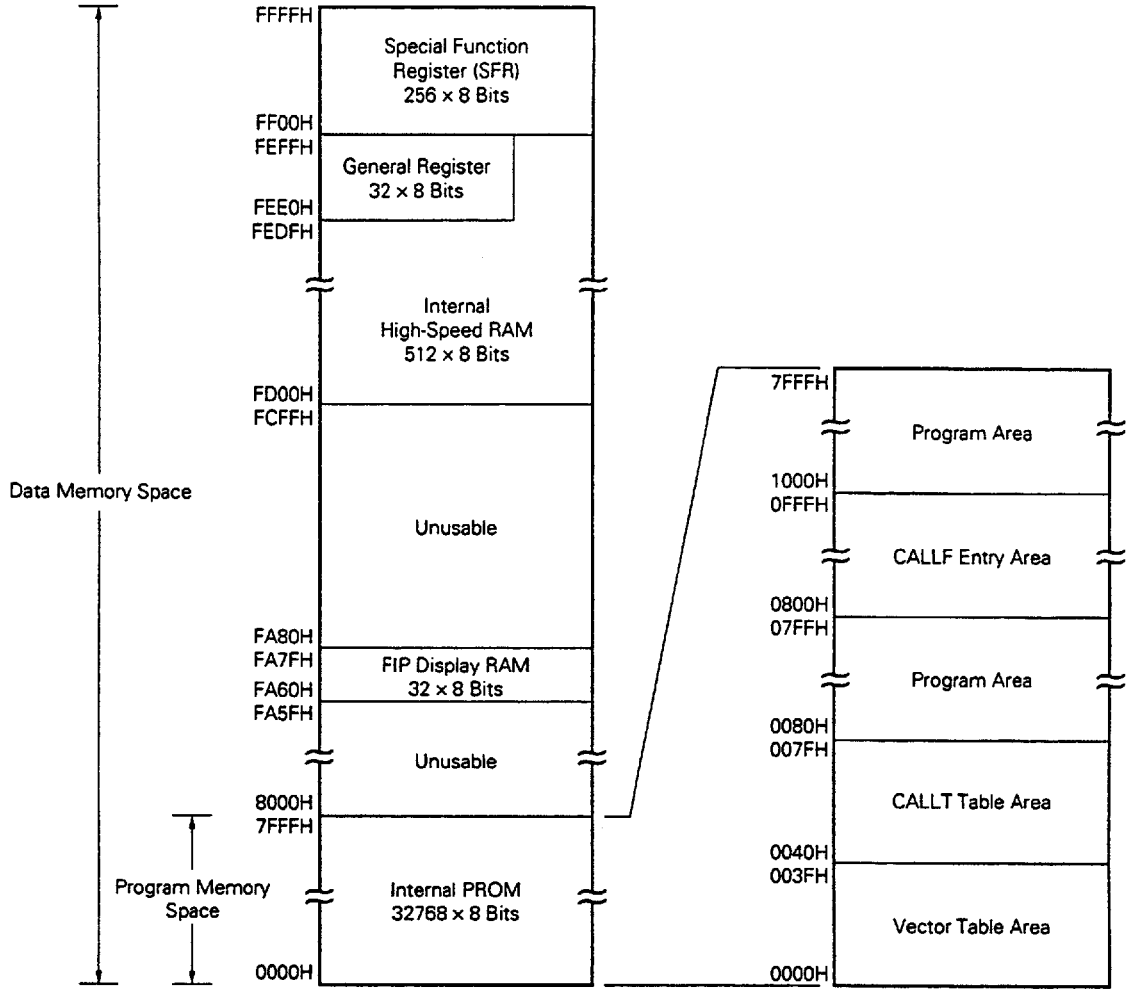


Fig. 3-3 Memory Map (μPD78P024)



3.1.1 Internal program memory space

The μPD78023 is a mask ROM with a configuration of 16384 × 8 bits, and PROMs with configurations of 24576 × 8 bits and 32768 × 8 bits, respectively. The μPD78P024 is a PROM with a configuration of 32768 × 8 bits. They store programs and table data. Normally, they are addressed with a program counter (PC).

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as 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 3-1 Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	0010H	INTCSI1
0004H	INTWDT	0012H	INTTM3
0006H	INTP0	0014H	INTTM0
0008H	INTP1	0016H	INTTM1
000AH	INTP2	0018H	INTTM2
000CH	INTP3	001AH	INTAD
000EH	INTCSI0	001CH	INTKS
		003EH	BRK instruction

(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 can with a 2-byte call instruction (CALLF).

3.1.2 Internal data memory space

The μPD78024 subseries units incorporate the following RAMs.

(1) Internal high-speed RAM

The μPD78024 subseries units have a 512 × 8-bit configuration. 4 banks of general registers, each bank consisting of eight 8-bit registers are allocated in the 32-byte area FEE0H to FEFFH.

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

(2) FIP display RAM

FIP display RAM is allocated to the 32-byte area from FA60H to FA7FH. The FIP display RAM can also be used as ordinary RAM.

3.1.3 Special function register (SFR) area

An on-chip peripheral hardware special-function register (SFR) is allocated in the area FF00H to FFFFH. (Refer to **Table 3-2**).

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

3.1.4 Data memory addressing

The μPD78024 subseries units are equipped with a variety of addressing modes considering memory maneuverability. Specific addressing operations are possible in accordance with the functions of the special function register (SFR) and general registers. The data memory space is an entire 64K-byte space of the 0000H to FFFFH. Figs. 3-4 to 3-6 show data memory addressing.

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

Fig. 3-4 Data Memory Addressing (μPD78023)

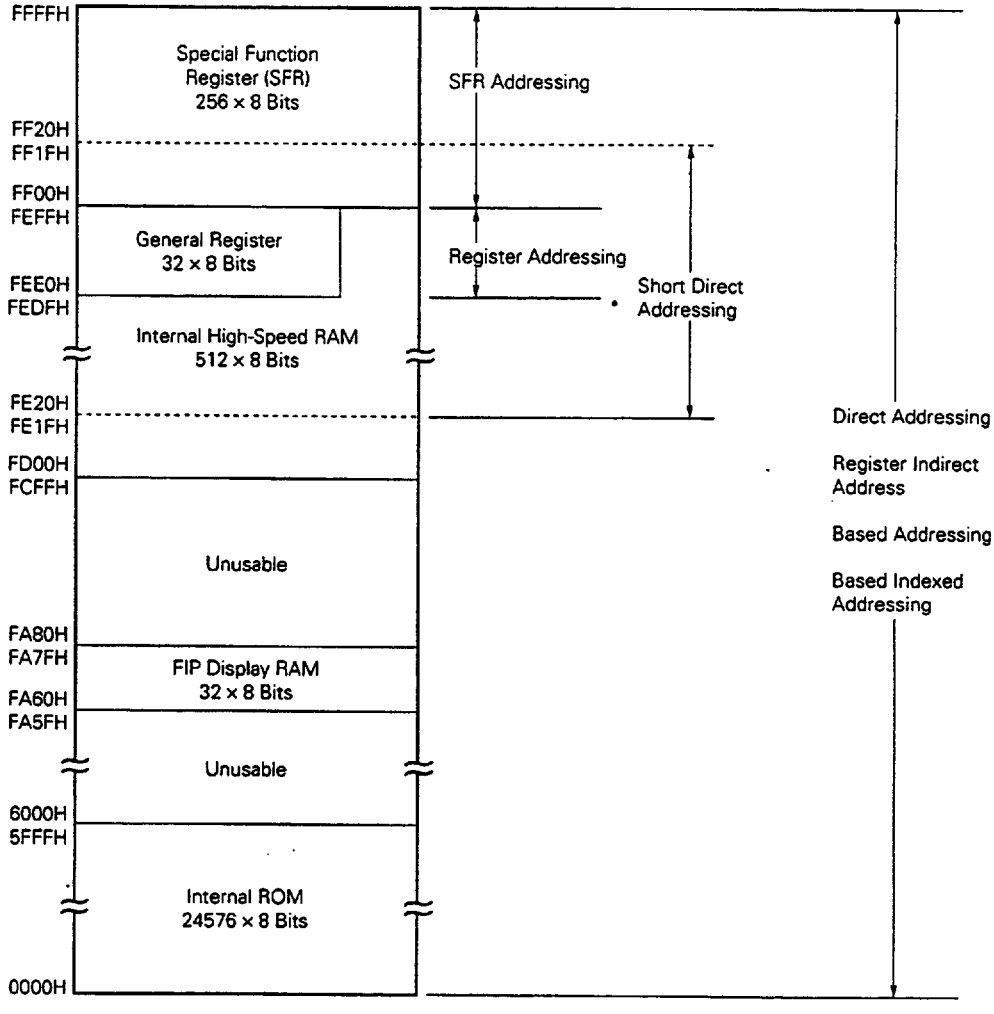


Fig. 3-5 Data Memory Addressing (μPD78024)

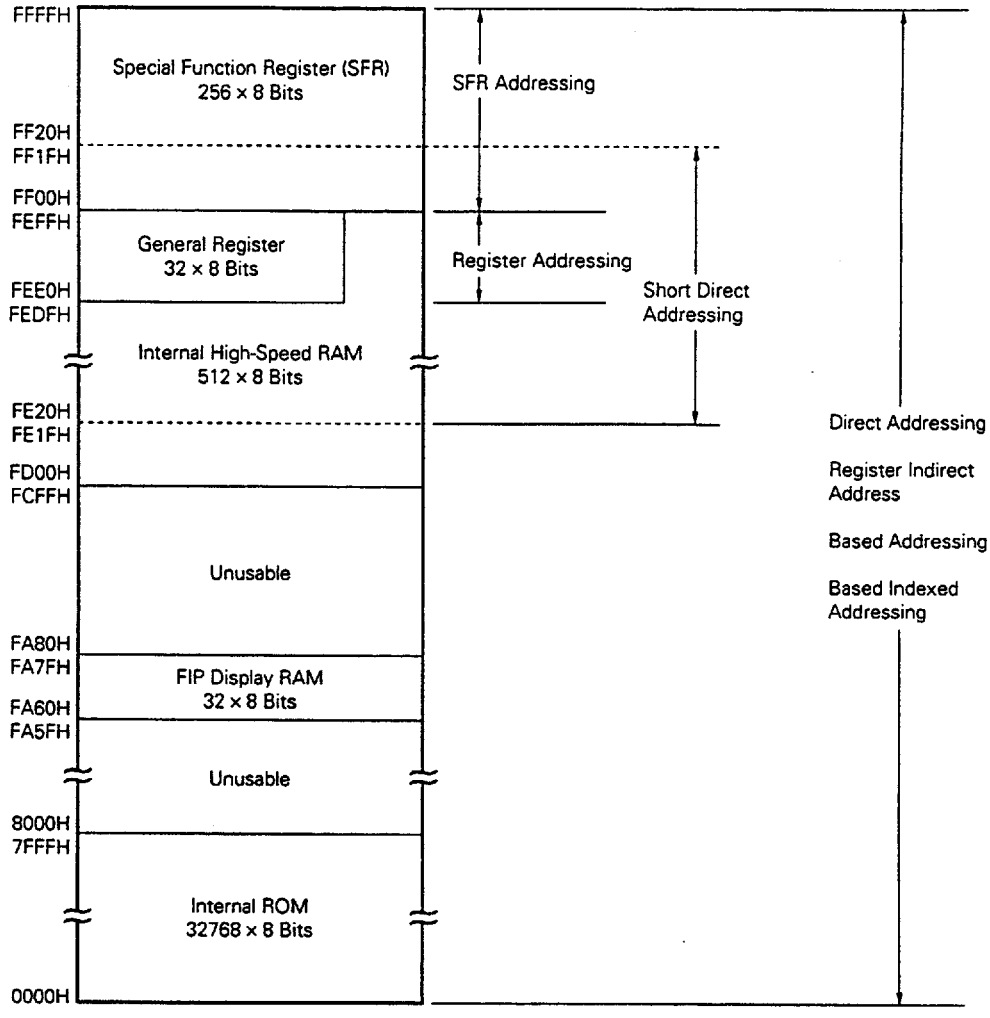
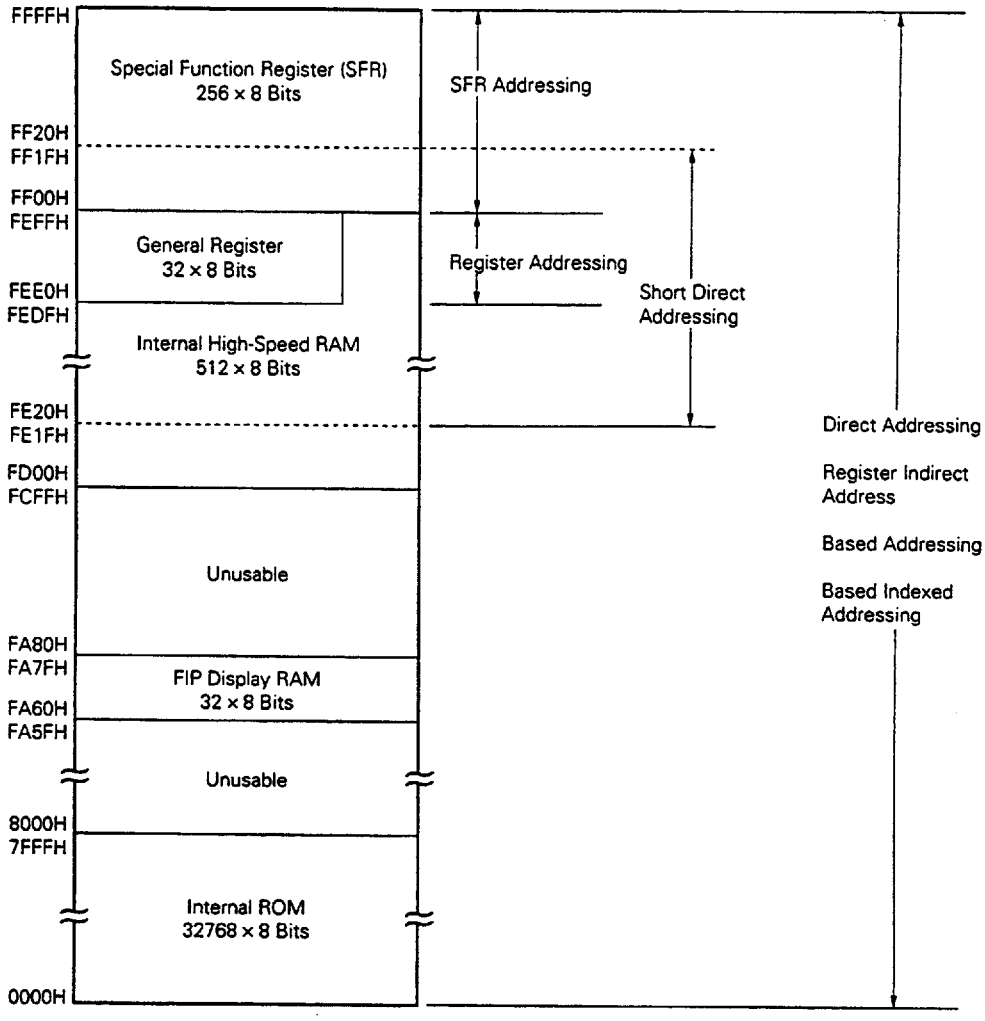


Fig. 3-6 Data Memory Addressing (μPD78P024)



3.2 Processor Registers

The μPD78024 subseries units incorporate the following processor registers.

3.2.1 Control registers

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

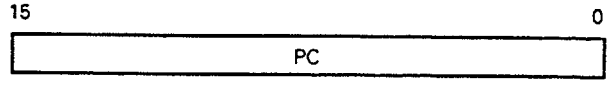
(1) Program counter (PC)

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.

Fig. 3-7 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.

Fig. 3-8 Program Status Word Configuration

7							0
IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledge operations of 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 in-service 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 acknowledgment 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, 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 SBL 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 (PR) are disabled for acknowledgment. When it is 1, all interrupts are acknowledgeable regardless of the interrupt priority order. Actual acknowledgment is controlled with interrupt enable flag (IE).

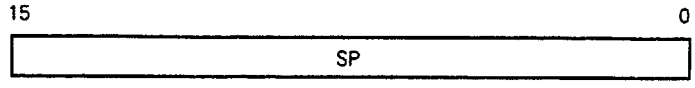
(f) Carry flag (CY)

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 (FD00H to FEFFH) can be set as the stack area.

Fig. 3-9 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 Figs. 3-10 and 3-11.

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

Fig. 3-10 Data to be Saved to Stack Memory

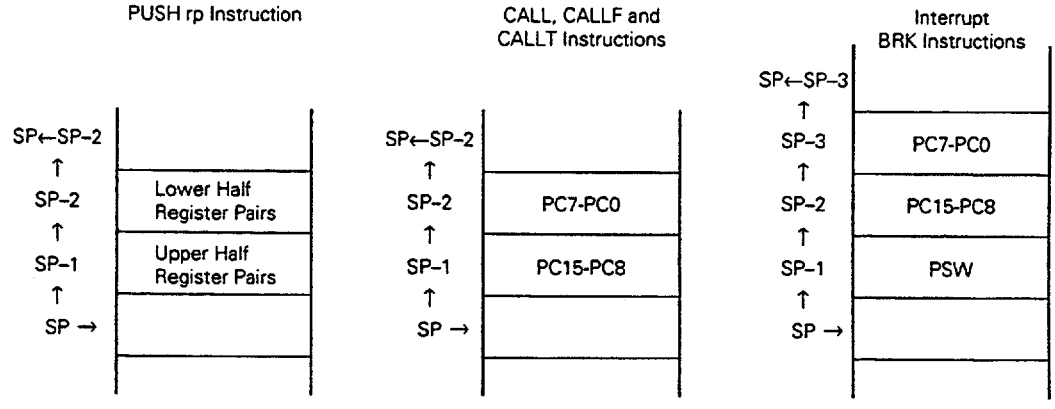
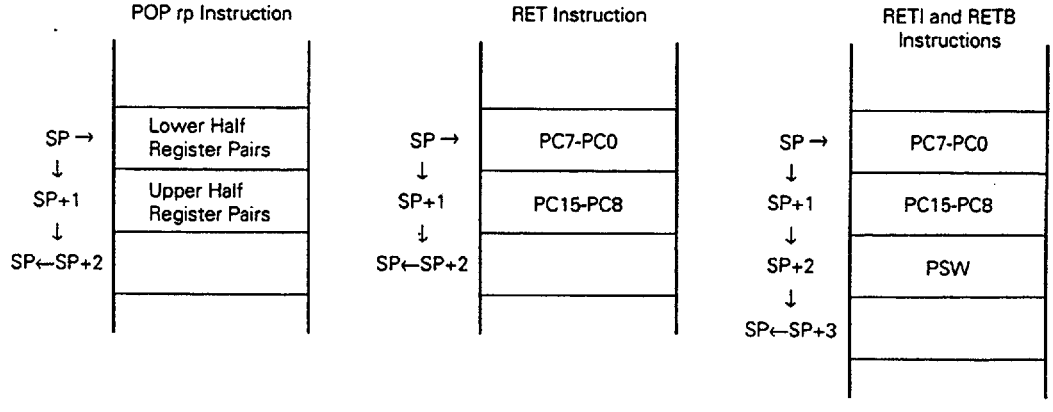


Fig. 3-11 Data to be Reset from Stack Memory



3.2.2 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).

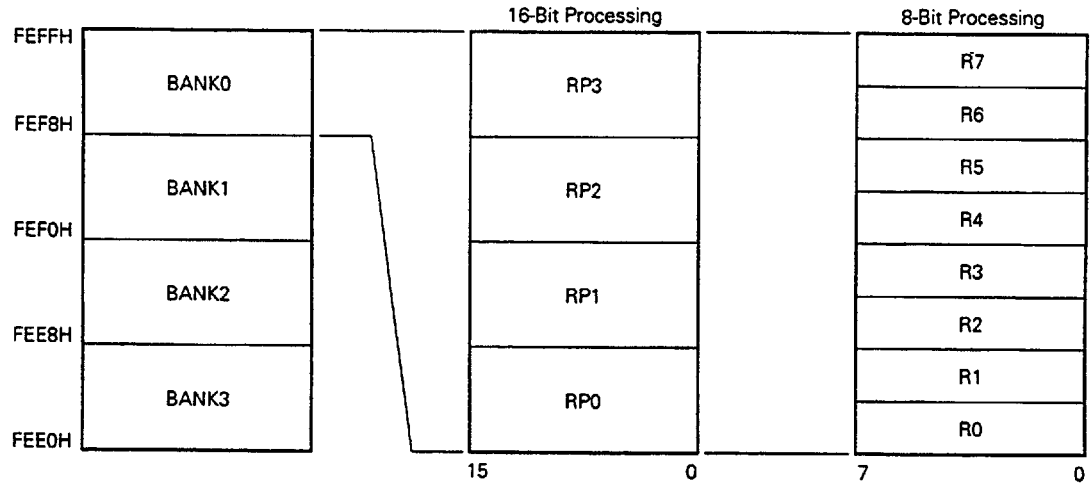
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE and HL).

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

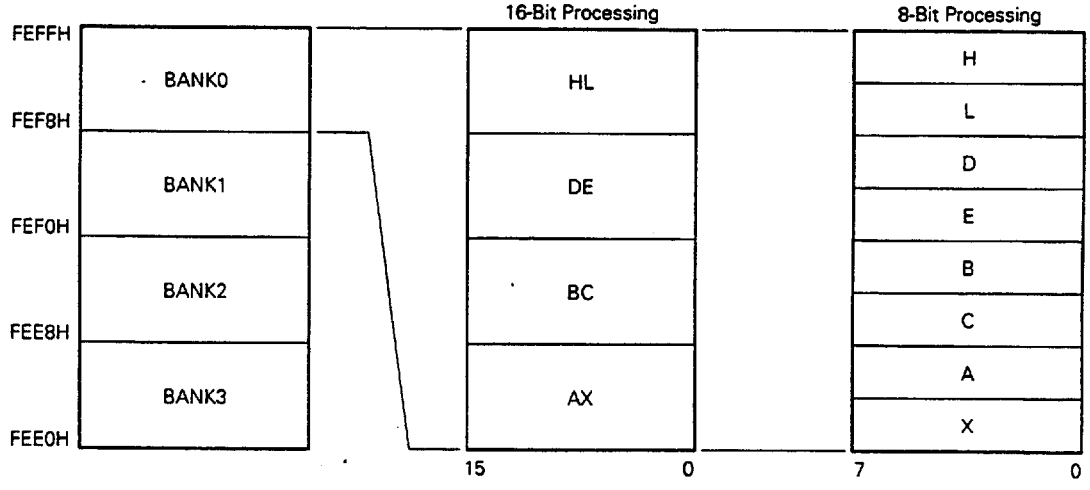
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interruption for each bank.

Fig. 3-12 General Register Configuration

(a) Absolute Names



(b) Functional Names



3.2.3 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
  Describes a 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
  Describes a symbol reserved with assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describes a symbol reserved with assembler for the 16-bit manipulation instruction operand (sfrp). When addressing an address, describe an event address.

Table 3-2 gives a list of special-function registers.

- Symbol
  This is a symbol to use in the assembler (RA78K/0) which indicates an address of the built-in special-function register. It can be described as an instruction operand.
- R/W
  Indicates whether the corresponding special-function register can be read or written.
  R/W : Read/write enable
  R : Read only
  W : Write only
- Manipulatable bit units
  Manipulatable bit units, 1, 8 and 16, are indicated.
- After reset
  Indicates each register status upon $\overline{\text{RESET}}$ input.

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

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 bit	8 bits	16 bits	
FF00H	Port 0	P0	R/W	O	O	-	00H
FF01H	Port 1	P1		O	O	-	
FF02H	Port 2	P2		O	O	-	
FF03H	Port 3	P3		O	O	-	
FF08H	Port 8	P8	W	O	O	-	
FF09H	Port 9	P9		O	O	-	
FF0AH	Port 10	P10		O	O	-	
FF0BH	Port 11	P11	R/W	O	O	-	
FF10H FF11H	16-bit compare register	CR00		-	-	O	Undefined
FF12H FF13H	16-bit capture register	CR01	R	-	-	O	
FF14H FF15H	16-bit timer register	TM0		-	-	O	00H
FF16H	8-bit compare register	CR10	R/W	-	O	-	Undefined
FF17H	8-bit compare register	CR20		-	O	-	
FF18H	8-bit timer register 1	TMS TM1	R	-	O	O	00H
FF19H	8-bit timer register 2	TM2		-	O		
FF1AH	Serial I/O shift register 0	SIO0	R/W	-	O	-	Undefined
FF1BH	Serial I/O shift register 1	SIO1		-	O	-	
FF1FH	A/D conversion result register	ADCR	R	-	O	-	
FF20H	Port mode register 0	PM0	R/W	O	O	-	1FH
FF21H	Port mode register 1	PM1		O	O	-	FFH
FF22H	Port mode register 2	PM2		O	O	-	
FF23H	Port mode register 3	PM3		O	O	-	
FF2BH	Port mode register 11	PM11		O	O	-	
FF40H	Timer clock select register 0	TCL0		O	O	-	00H
FF41H	Timer clock select register 1	TCL1		-	O	-	
FF42H	Timer clock select register 2	TCL2		-	O	-	
FF43H	Timer clock select register 3	TCL3		-	O	-	88H
FF47H	Sampling clock select register	SCS		-	O	-	00H
FF48H	16-bit timer mode control register	TMC0		O	O	-	
FF49H	8-bit timer mode control register	TMC1		O	O	-	
FF4AH	Watch timer mode control register	TMC2		O	O	-	

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

Address	Special-Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 bit	8 bits	16 bits	
FF4EH	16-bit timer output control register	TOC0		R/W	O	O	–	00H
FF4FH	8-bit timer output control register	TOC1			O	O	–	
FF60H	Serial operating mode register 0	CSIM0			O	O	–	
FF61H	Serial bus interface control register	SBIC			O	O	–	
FF62H	Slave address register	SVA			–	O	–	Undefined
FF63H	Interrupt timing specify register	SINT			O	O	–	00H
FF68H	Serial operating mode register 1	CSIM1			O	O	–	
FF80H	A/D converter mode register	ADM			O	O	–	01H
FF84H	A/D converter input select register	ADIS			–	O	–	00H
FFA0H	Display mode register 0	DSPM0			Δ[Note]	O	–	
FFA1H	Display mode register 1	DSPM1			–	O	–	
FFE0H	Interrupt request flag register L	IF0	IF0L		O	O	O	
FFE1H	Interrupt request flag register H		IF0H		O	O		
FFE4H	Interrupt mask flag register L	MK0	MK0L		O	O	O	FFH
FFE5H	Interrupt mask flag register H		MK0H		O	O		
FFE8H	Priority order specify flag register L	PR0	PR0L		O	O	O	
FFE9H	Priority order specify flag register H		PR0H		O	O		
FFECH	External interrupt mode register	INTM0			–	O	–	00H
FFF0H	Memory size switch register[Note 2]	IMS			–	O	–	48H ★
FFF7H	Pull-up resistor option register	PUO			O	O	–	00H
FFF9H	Watchdog timer mode register	WDTM			O	O	–	
FFFAH	Oscillation stabilizing time select register	OSTS			–	O	–	04H
FFFBH	Processor clock control register	PCC			O	O	–	

Notes 1. Bit 7 only can be manipulated as a read-only bit.

2. The μPD78P024 only is incorporated.

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents and is normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing (For details of each instruction, refer to **78K/0 Series User's Manual - Instruction (IEU-1372)**.

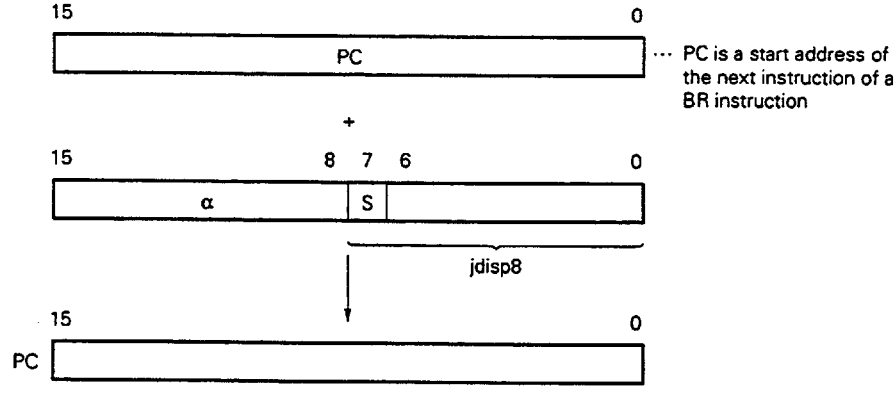
3.3.1 Relative addressing

[Function]

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

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

[Illustration]



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

3.3.2 Immediate addressing

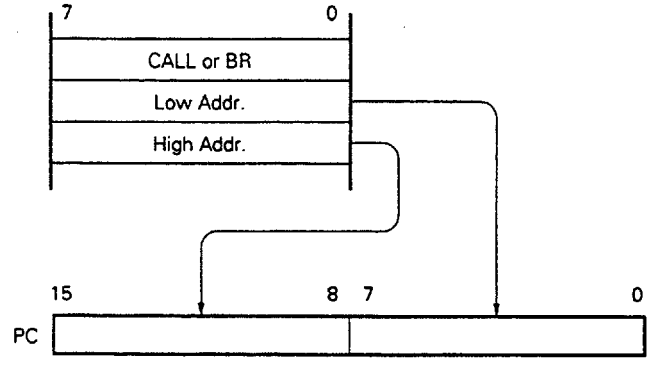
[Function]

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

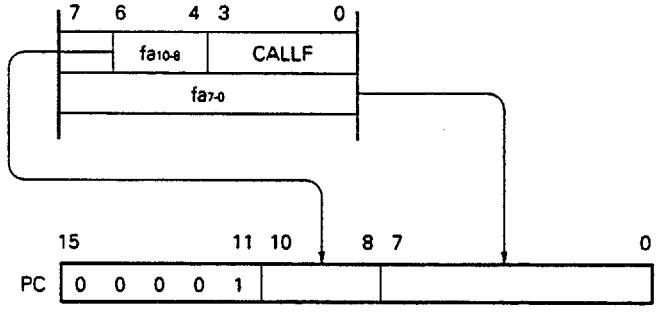
This function is carried out when the "CALL !addr16" or "BR !addr16" or "CALLF !addr11" instruction is executed.

[Illustration]

In case of CALL !addr16, BR !addr16 instruction



In case of CALLF !addr11 instruction



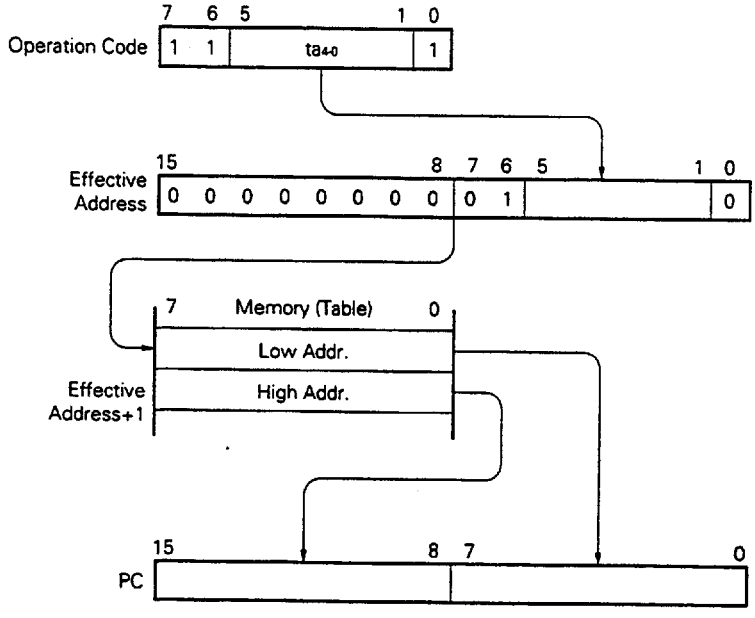
3.3.3 Table indirect addressing

[Function]

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

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

[Illustration]



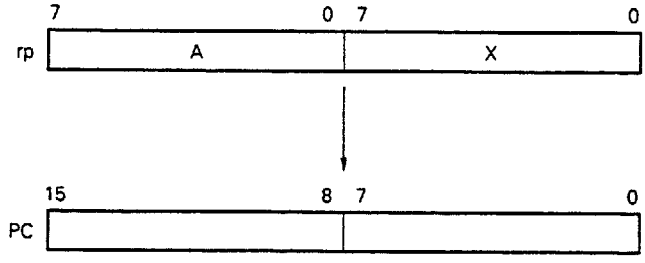
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

The register which functions as an accumulator (A and AX) in the general register is automatically addressed. Of the μPD78024 subseries instruction words, the following instructions employs 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 divided and quotient storage
ADJBA/ADJBS	A register for storage of numeric values which become decimal correction targets
ROR4/ROL4	A register for storage of digit data which undergoes digit rotation

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

The general 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 instruction code.

[Operand format]

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

'r' and 'rp' can be described 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]

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

Instruction code	0	1	1	0	0	0	1	0

When selecting the DE register pair for INCW DE;rp

Instruction code	1	0	0	0	0	1	0	0

3.4.3 Direct addressing

[Function]

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

[Operand format]

Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

3.4.4 Short direct addressing

[Function]

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

This addressing is applied to the 256-byte space FE20H to FF1FH. An internal high-speed 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 and a capture register 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 valid address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1.

[Operand format]

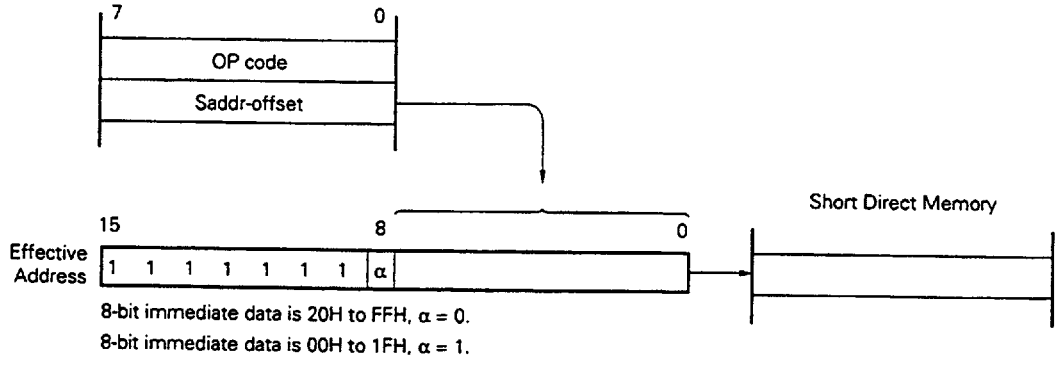
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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

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

[Illustration]



3.4.5 Special-function register (SFR) addressing

[Function]

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

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

[Operand format]

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

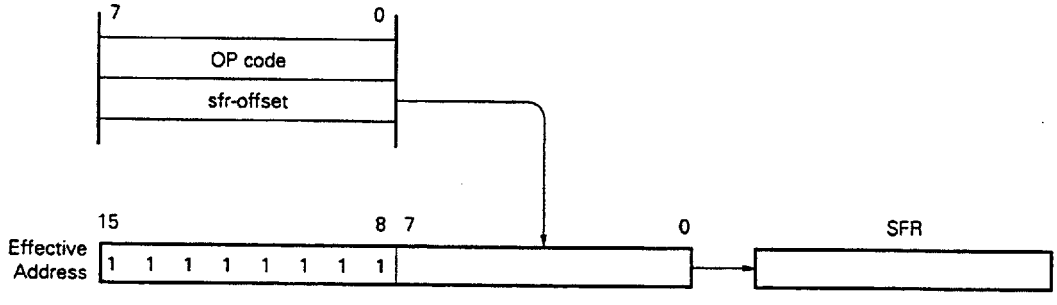
[Description example]

When selecting PM0 for MOV PM0, A;sfr

Instruction code 1 1 1 1 0 1 1 0

0 0 1 0 0 0 0 0

[Illustration]



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

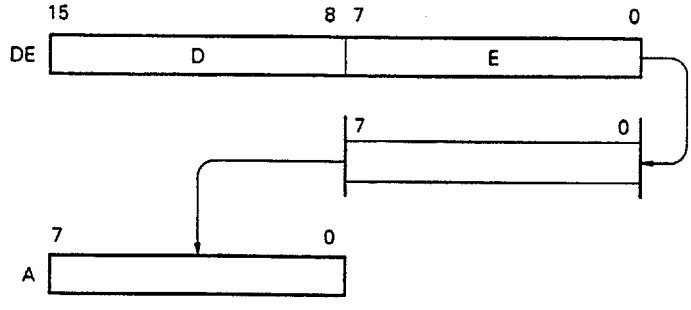
Identifier	Description
—	[DE], [HL]

[Description example]

When selecting [DE] for MOV A, [DE]: register pair.

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

[Illustration]



3.4.7 Based addressing

[Function]

8-bit immediate data is added as offset data to the contents of the base re 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]

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

Instruction code

1	0	1	0	1	1	1	0

0	0	0	1	0	0	0	0

3.4.8 Based indexed addressing

[Function]

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 to 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]

Instruction code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Istruction code	1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD78024 subseries units incorporate 2 input ports, 18 output ports, and 34 input/output ports. Fig. 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as built-in hardware input/output pins.

Fig. 4-1 Port Types

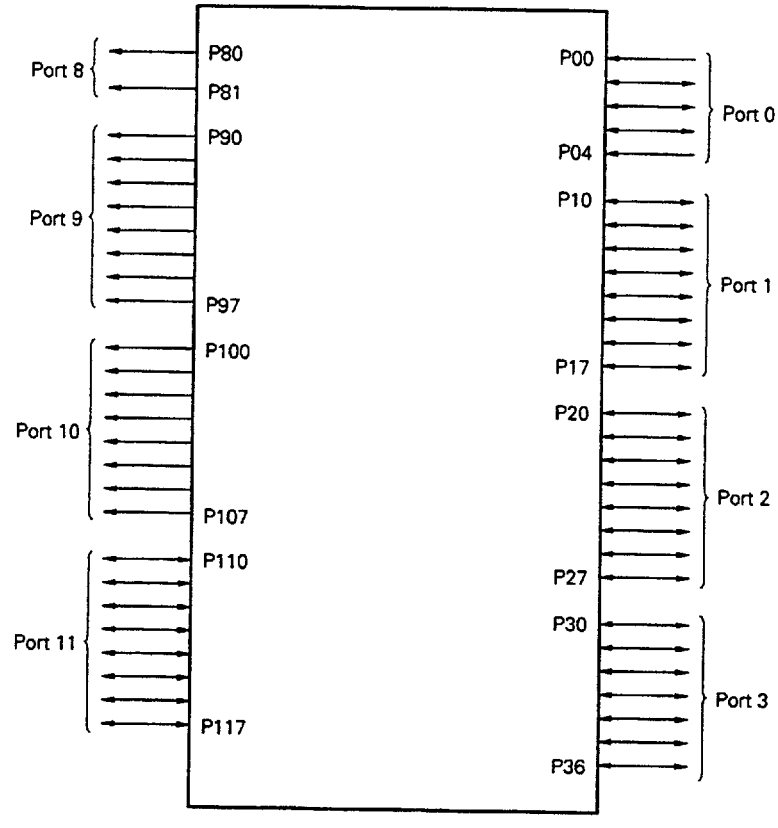


Table 4-1 Port Functions (1/2)

Pin Name	Function		Dual-Function Pin
P00	Port 0.	Input only	INTP0/TI0
P01	5-bit input/output port	Input/output specifiable bit-wise. If used as an input port, a pull-up resistor can be specified by software.	INTP1
P02			INTP2
P03			INTP3
P04		Input only	XT1
P10 to P17	Port 1. 8-bit input/output port. Input/output specifiable bit-wise. If used as an input port, a pull-up resistor can be specified by software.		ANI0 to ANI7
P20	Port 2.		SI1
P21	8-bit input/output port.		SO1
P22	Input/output specifiable bit-wise.		$\overline{SCK1}$
P23	If used as an input port, a pull-up resistor can be specified by software.		—
P24			—
P25			SI0/SB0
P26			SO0/SB1
P27			$\overline{SCK0}$
P30	Port 3.		TO0
P31	7-bit input/output port.		TO1
P32	Input/output specifiable bit-wise.		TO2
P33	Direct LED drive capability.		TI1
P34	If used as an input port, a pull-up resistor can be specified by software.		TI2
P35	Mask ROM products only: bit-wise incorporation of pull-up resistor possible by mask		PCL
P36	option.		BUZ
P80 and P81	Port 8. P-ch open-drain 2-bit high-voltage output port. Direct LED drive capability. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option (connection to V_{LOAD} or Vss specifiable as 2-bit unit). μPD78P024: pull-down resistor incorporated (connected to V_{LOAD}).		FIP0, FIP1

Table 4-1 Port Functions (2/2)

Pin Name	Function	Dual-Function Pin
P90 to P97	Port 9. P-ch open-drain 8-bit high-voltage output port. Directly LED drive capability. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option (connection to V_{LOAD} or Vss specifiable in 4-bit units). μPD78P024: pull-down resistor incorporated (connected to V_{LOAD}).	FIP2 to FIP9
P100 to P107	Port 10. P-ch open-drain 8-bit high-voltage output port. Directly LED drive capability. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option (connection to V_{LOAD} or Vss specifiable in 4-bit units). μPD78P024: pull-down resistor incorporated in P100 to P105 (connected to V_{LOAD}).	FIP10 to FIP17
P110 to P117	Port 11. P-ch open-drain 8-bit high-voltage input/output port. Directly LED drive capability. Input/output specifiable bit-wise. Mask ROM products only: bit-wise incorporation of pull-down resistor possible by mask option (connection to V_{LOAD} or Vss specifiable in 4-bit units).	FIP18 to FIP25

4.2 Port Configuration

A port consists of the following hardware.

Table 4-2 Port Configuration

Item	Configuration
Control registers	Port mode registers (PMm: m = 0, 1, 2, 3, 11) Pull-up resistor option register (PUO)
Ports	Total: 54 (input: 2, output: 18, input/output: 34)
Pull-up resistors	• Mask ROM product total: 26 (software control: 26) • μPD78P024 total: 26
Pull-down resistor	• Mask ROM product total: 33 (mask option control: 33) • μPD78P024 total: 16

4.2.1 Port 0

Port 0 is a 5-bit input/output port with output latch. P01 to P03 pins can specify the input mode/output mode in 1-bit units with port mode register 0. P00 to P04 pins are input-only ports. When P01 to P03 pins are used as input ports, a pull-up resistor can be connected to them in 3-bit units with a pull-up resistor option register.

Dual-functions include external interrupt input, external count clock input to the timer and crystal connection for subsystem clock oscillation.

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

Figs. 4-2 and 4-3 show block diagrams of port 0.

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.

Fig. 4-2 P00 and P04 Configurations

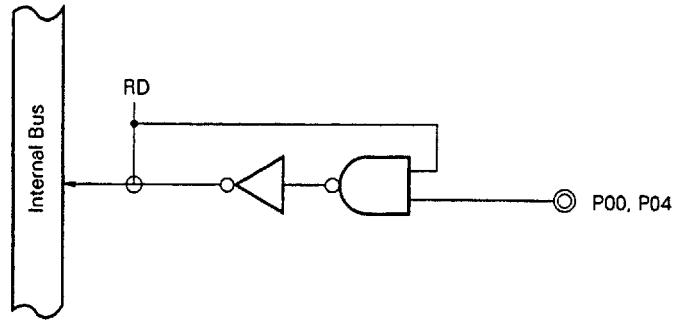
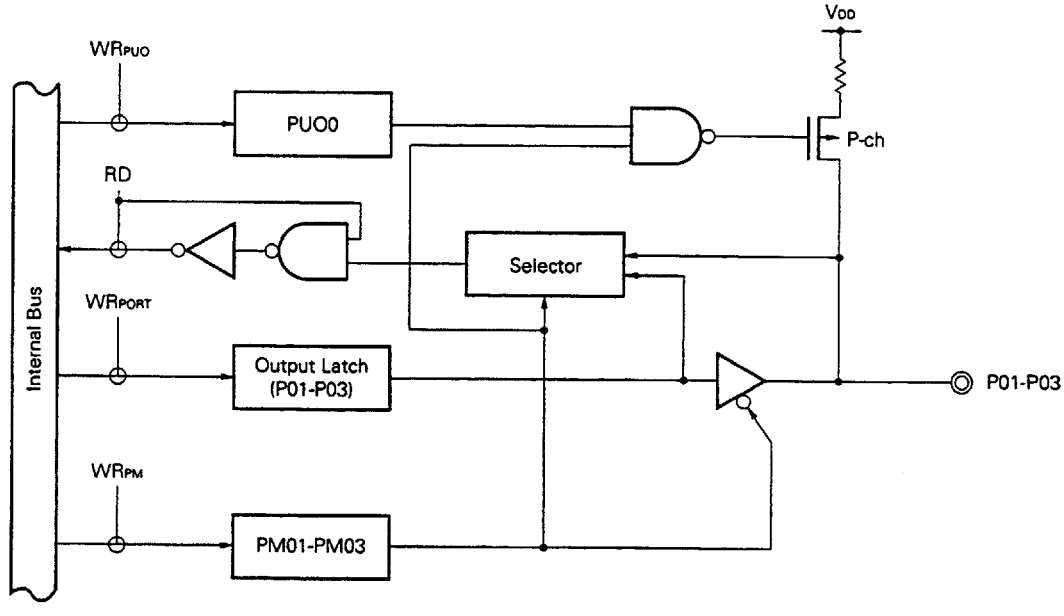


Fig. 4-3 P01 to P03 Configurations



PUO : Pull-up resistor option register
PM : Port mode register
RD : Read signal of port 0
WR : Write signal of port 0

4.2.2 Port 1

This is an 8-bit I/O port with an output latch. The pins P10 to P17 can be specified to the input mode/output mode bit-wise by port mode register 1. When pins P10 to P17 are used as input port pins, the pull-up register can be connected in 8-bit units by the pull-up resistor option register.

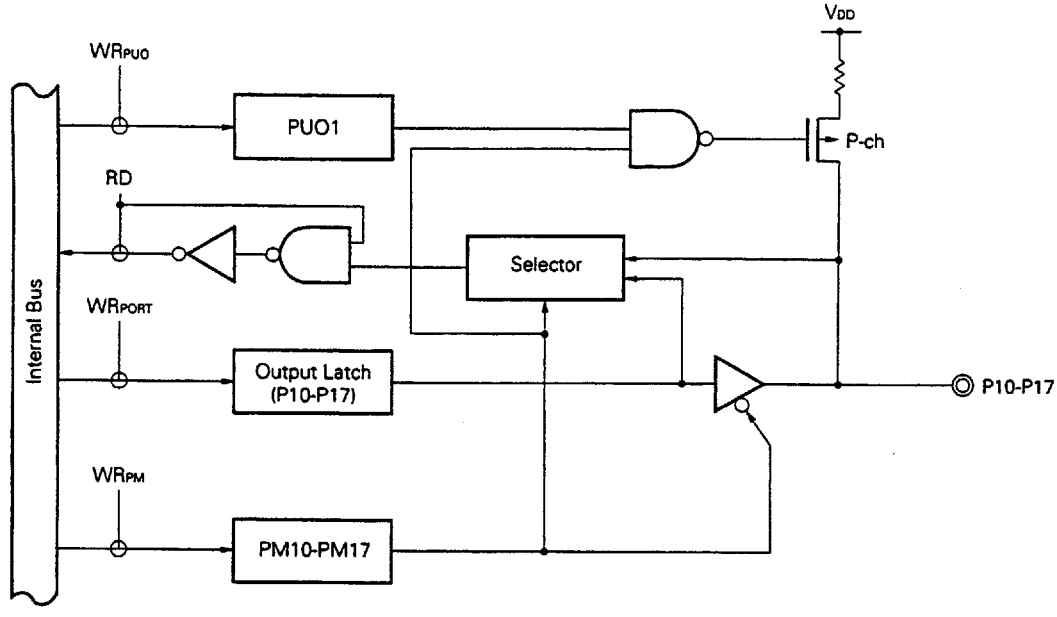
Dual-functions include an A/D converter analog input.

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

Fig. 4-4 shows a block diagram of port 1.

Caution On-chip pull-up register cannot be used for the pin used as an A/D converter analog input.

Fig. 4-4 P10 to P17 Configurations



PUO : Pull-up resistor option register
PM : Port mode register
RD : Read signal of port 1
WR : Write signal of port 1

4.2.3 Port 2

Port 2 is an 8-bit input/output port with output latch. P20 to P27 pins can specify the input mode/output mode in 1-bit units with port mode register 2. When P20 to P27 pins are used as input ports, a pull-up resistor can be connected to them in 8-bit units with a pull-up resistor option register.

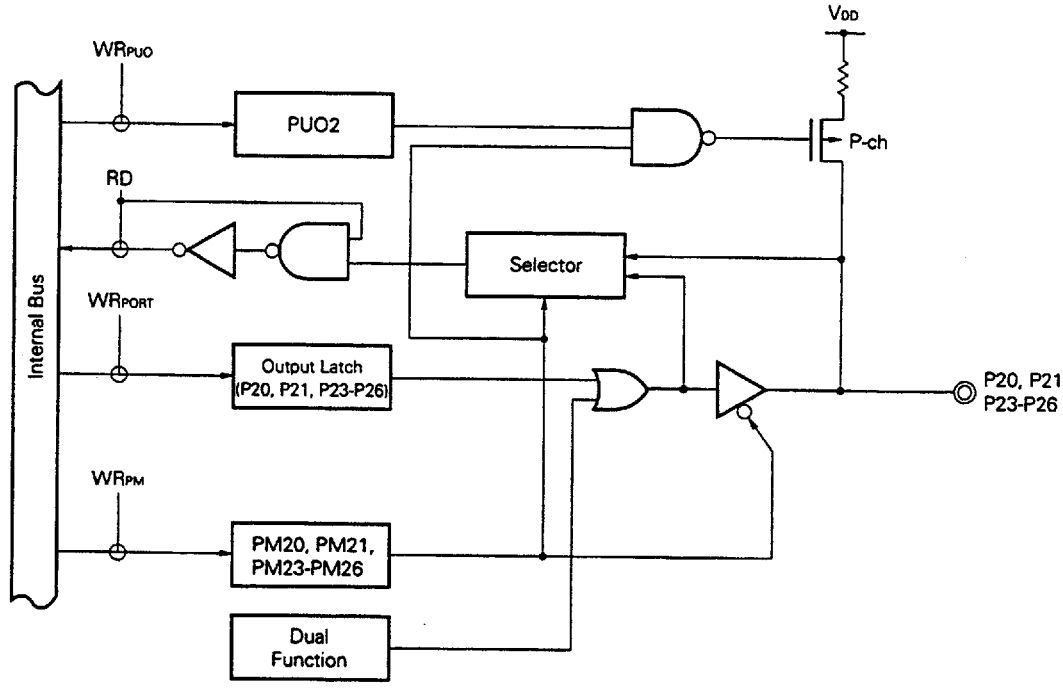
Dual-functions include serial interface data input/output and clock input/output.

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

Fig. 4-5 and 4-6 show block diagrams of port 2.

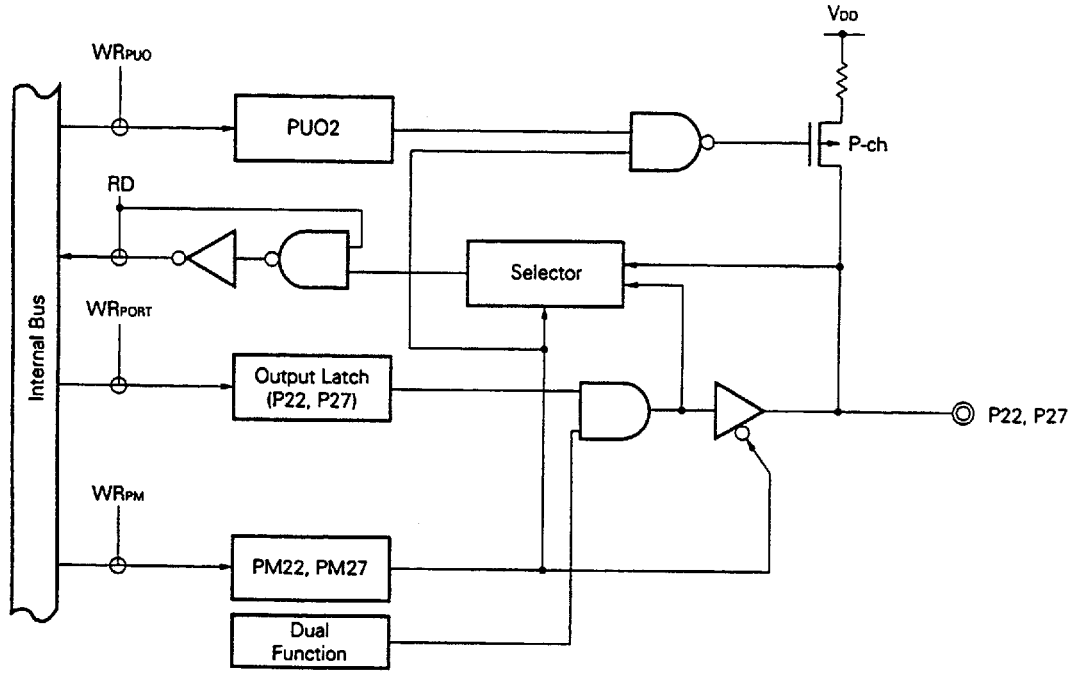
Caution When using port 2 as the serial interface, the input/output mode and output latch must be set according to the function required. For details, refer to Fig. 13-4 Format of Serial Operation Mode Register 0, and Fig. 14-3 Format of Serial Operation Mode Register 1.

Fig. 4-5 P20, P21, and P23 to P26 Configurations



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

Fig. 4-6 P22 and P27 Configurations



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

4.2.4 Port 3

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 port mode register 3. When P30 to P36 pins are used as input ports, a pull-up resistor can be connected to them in 7-bit units with a pull-up resistor option register.

In mask ROM products, bit-wise incorporation of a pull-down resistor is possible by means of a mask option. The μPD78P024 does not incorporate pull-down resistors.

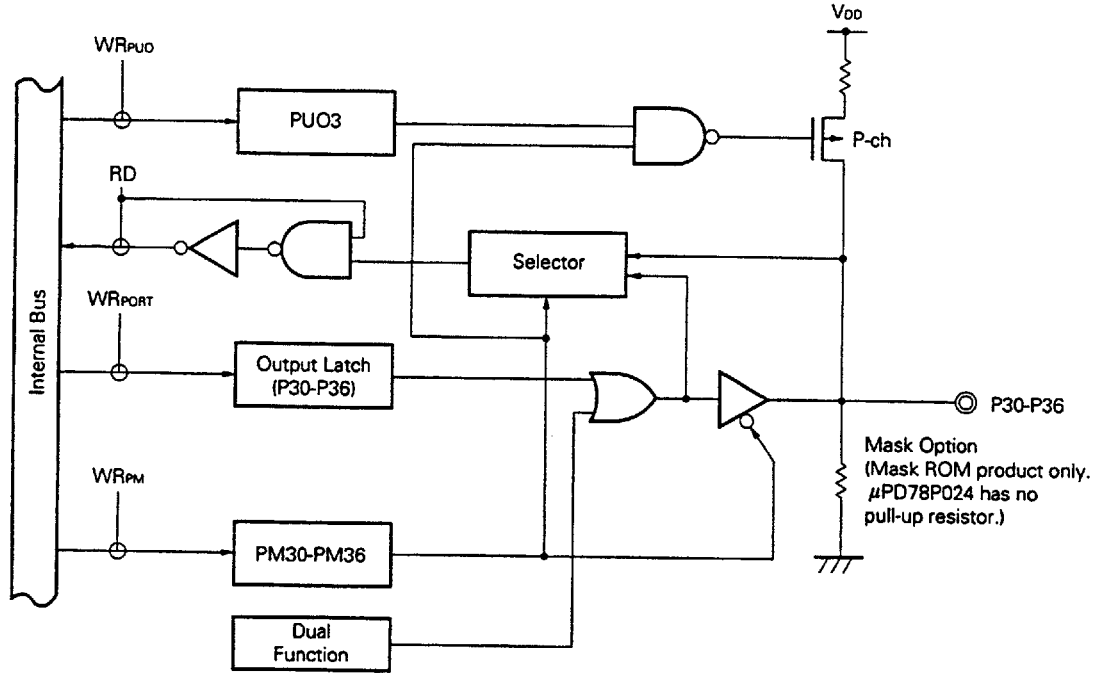
Port 3 has direct LED drive capability.

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

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

Fig. 4-7 shows a block diagram of port 3.

Fig. 4-7 P30 to P36 Configurations



PUO : Pull-up resistor option register
PM : Port mode register
RD : Read signal of port 3
WR : Write signal of port 3

4.2.5 Port 8

2-bit dedicated output port. In mask ROM products, bit-wise incorporation of a pull-down resistor is possible by means of a mask option. The pull-down resistor can be specified for connection to V_{LOAD} or Vss as a 2-bit unit. The μPD78P024 incorporates pull-down resistors connected to V_{LOAD}.

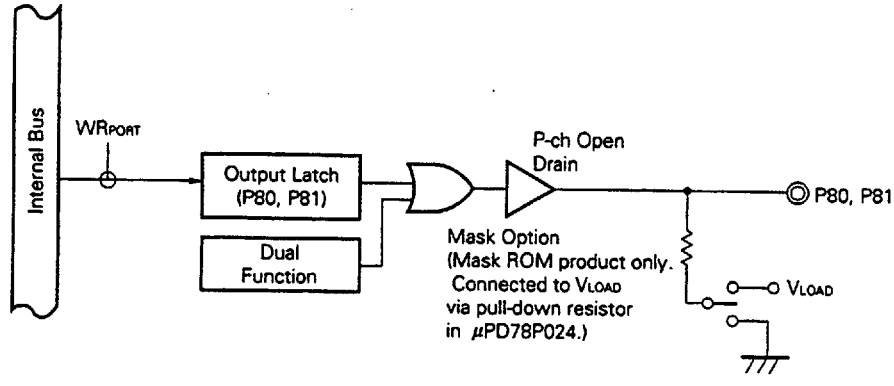
Port 8 has direct LED drive capability.

These pins have a dual function as FIP controller/driver digit outputs.

$\overline{RESET}$ input sets port 8 to the output mode.

The block diagram of port 8 is shown in Fig. 4-8.

Fig. 4-8 P80 and P81 Configurations



WR : Write signal of port 8

4.2.6 Port 9

8-bit dedicated output port. In mask ROM products, bit-wise incorporation of a pull-down resistor is possible by means of a mask option. The pull-down resistor can be specified for connection to V_{LOAD} or Vss in 4-bit units. The μPD78P024 incorporates pull-down resistors connected to V_{LOAD}.

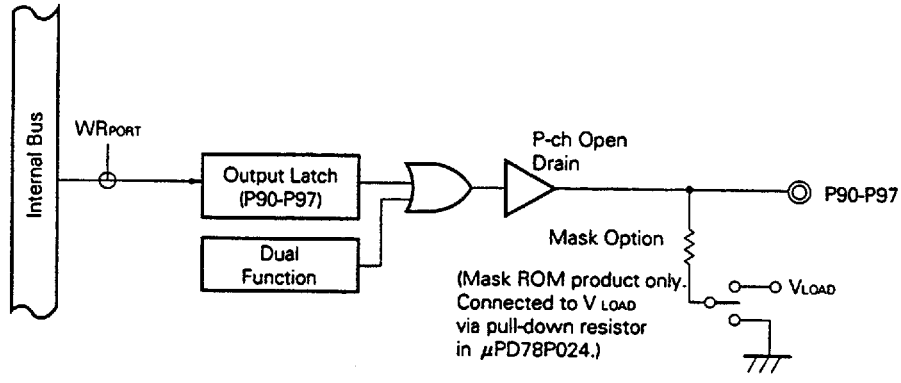
Port 9 has direct LED drive capability.

These pins have a dual function as FIP controller/driver digit outputs.

$\overline{RESET}$ input sets port 9 to the output mode.

The block diagram of port 9 is shown in Fig. 4-9.

Fig. 4-9 P90 to P97 Configurations



WR : Write signal of port 9

4.2.7 Port 10

8-bit dedicated output port. In mask ROM products, bit-wise incorporation of a pull-down resistor is possible by means of a mask option. The pull-down resistor can be specified for connection to V_{LOAD} or Vss in 4-bit units. The μPD78P024 incorporates pull-down resistors connected to V_{LOAD} in pins P100 to P105.

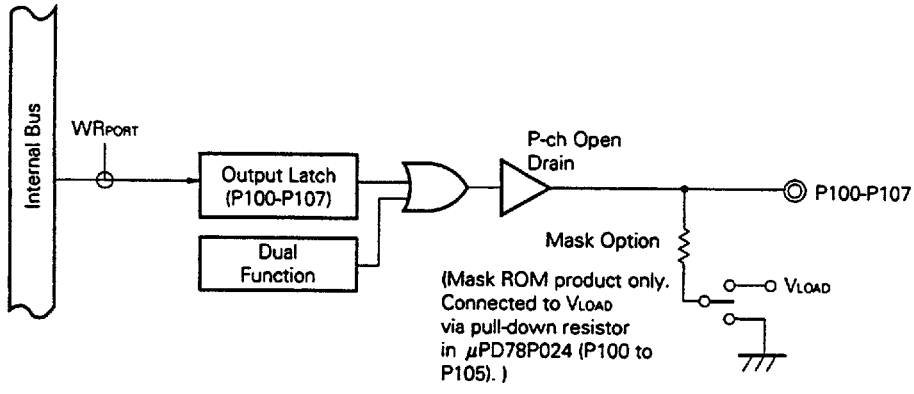
Port 10 has direct LED drive capability.

Port 10 pins have a dual function as FIP controller/driver segment/digit outputs.

$\overline{\text{RESET}}$ input sets port 10 to the output mode.

The block diagram of port 10 is shown in Fig. 4-10.

Fig. 4-10 P100 to P107 Configurations



WR : Write signal of port 10

4.2.8 Port 11

8-bit input/output port with an output latch. Input mode or output mode can be specified bit-wise for pins P110 to P117 by means of port mode register 11. In mask ROM products, bit-wise incorporation of a pull-down resistor is possible by means of a mask option. The pull-down resistor can be specified for connection to V_{LOAD} or V_{SS} in 4-bit units. The μPD78P024 does not incorporate pull-down resistors.

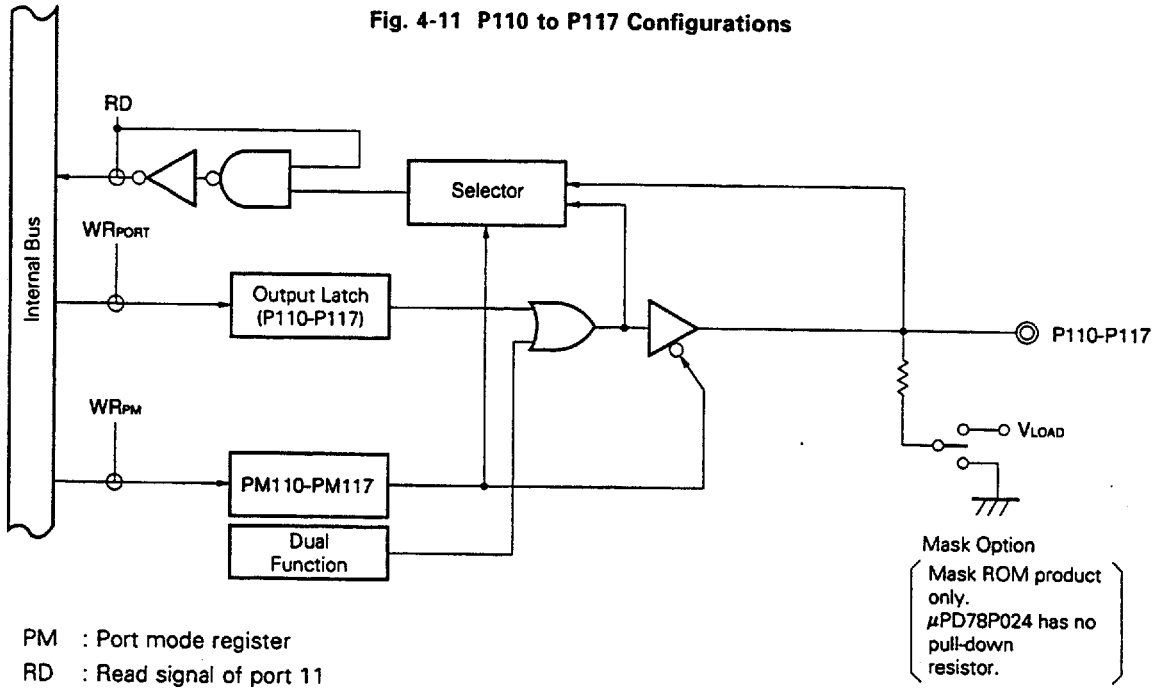
Port 11 has direct LED drive capability.

Port 11 pins have a dual function as FIP controller/driver segment outputs.

$\overline{RESET}$ input sets port 11 to the input mode.

The block diagram of port 11 is shown in Fig. 4-11.

Fig. 4-11 P110 to P117 Configurations



PM : Port mode register
RD : Read signal of port 11
WR : Write signal of port 11

4.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM1, PM2, PM3, PM11)
- Pull-up resistor option register (PUO)

(1) Port mode registers (PM0, PM1, PM2, PM3, PM11)

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

PM0, PM1, PM2, PM3 and PM11 each are set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets PM0 to 1FH and all other port mode registers to FFH.

When using the shared function of each port, set the corresponding port mode register and output latch as shown in Table 4-3.

Cautions
1. **P00 and P04 pins are input-only pins.**
2. **Pins P80, P81, P90 to P97, and P100 to P107 are dedicated output pins.**
3. **As port 0 has a dual function as an 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.**
4. **In case of byte access to PM3 (FF23H), be sure to set 1 in the most significant bit.**

★ **Table 4-3 Setting of Port Mode Registers and Output Latches When Shared Functions of Ports Are Used**

Pin Name	Shared Function		PMxx	Pxx	Pin Name	Shared Function		PMxx	Pxx
	Name	I/O				Name	I/O		
P00	INTP0	Input	1 (fixed)	None	P35	PCL	Output	0	0
	TI0	Input	1 (fixed)	None	P36	BUZ	Output	0	0
P01 to P03	INTP1-INTP3	Input	1	×	P80, P81	FIP0, FIP1	Output	0	0
P04[Note]	XT1	Input	1 (fixed)	None	P90 to P97	FIP2-FIP9	Output	0	0
P10-P17[Note]	ANI0-ANI7	Input	1	×	P100 to P107	FIP10-FIP17	Output	0	0
P30-P32	TO0-TO2	Output	0	0	P110 to P117	FIP18-FIP25	Output	0	0
P33, P34	TI1-TI2	Input	1	×					

Note Becomes undefined when the shared function is used.

Caution When using port 2 as the serial interface, the input/output mode and output latch must be set according to the function required. For details, refer to Fig. 13-4 Format of Serial Operation Mode Register 0 and Fig. 14-3 Format of Serial Operation Mode Register 1.

Remarks × : Don't care (needs not to be set)
PMxx : Port mode register
Pxx : Output latch of port

Fig. 4-12 Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM0	0	0	0	1	PM03	PM02	PM01	1	FF20H	1FH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FF2BH	FFH	R/W

PMmn	Pmn Pin Input/Output Mode Selection (m = 0, 1, 2, 3, 11 : n = 0-7)
0	Output mode (Output buffer ON)
0	Input mode (Output buffer OFF)

(2) Pull-up resistor option register (PUO)

This register is used to set connection of a on-chip pull-up resistor at each port. A pull-up resistor internally can be used only for bits which are set to the input mode at a port where the use of pull-up resistor has been specified with PUO. No pull-up resistors can be used at bits set to the output mode or bits used for analog input pins, irrespective of PUO setting.

PUO is set with a 1-bit or 8-bit memory manipulate instruction.

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

Cautions 1. P00 and P04 pins do not incorporate a pull-up resistor.

2. When using port 1 as A/D converter analog input, pull-up resistor cannot be connected if set PUO1 in 1.

Fig. 4-13 Pull-up Resistor Option Register Format

Symbol	7	6	5	4	③	②	①	⓪	Address	After Reset	R/W
PUO	0	0	0	0	PUO3	PUO2	PUO1	PUO0	FFF7H	00H	R/W

PUOm	Pm Pull-Up Resistor Connection Selection (m=0, 1, 2, 3)
0	Pull-up resistor disconnected
1	Pull-up resistor connected

4.4 Port Function Operations

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

4.4.1 Writing to input/output port

(1) Output port

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

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

(2) Input port

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.

4.4.2 Reading from input/output port

(1) Output port

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

(2) Input port

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

4.4.3 Operation on input/output port

(1) Output port

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 port

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 except for the manipulated bit.

4.5 Mask Option Selection

The following mask options are provided for mask ROM products. The μPD78P024 has no mask options.

Table 4-4 Comparison of Mask ROM Product Mask Options and μPD78P024

Pin Name	Mask Option of Mask ROM Model	μPD78P024
P30/TO0-P32/TO2, P33/TI1 P34/TI2, P35/PCL, P36/BUZ	Pull-down resistor can be incorporated bit-wise.	Pull-down resistor is not incorporated.
P80/FIP0, P81/FIP1	Pull-down resistor can be incorporated bit-wise (connection to V_{LOAD} or Vss can be specified as 2-bit unit).	Pull-down resistor is incorporated (connected to V_{LOAD}).
P90/FIP2-P97/FIP9	Pull-down resistor can be incorporated bit-wise (connection to V_{LOAD} or Vss can be specified as 4-bit unit).	Pull-down resistor is incorporated (connected to V_{LOAD}).
P100/FIP10-P107/FIP17	Pull-down resistor can be incorporated bit-wise (connection to V_{LOAD} or Vss can be specified as 4-bit unit).	Pull-down resistor is incorporated (connected to V_{LOAD}) in pins P100/FIP10-P105/FIP15. Pull-down resistor is incorporated (connected to V_{LOAD}) in P106/FIP16, P107/FIP17.
P110/FIP18-P117/FIP25	Pull-down resistor can be incorporated bit-wise (connection to V_{LOAD} or Vss can be specified as 4-bit unit).	Pull-down resistor is not incorporated.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

(1) Main system clock oscillation

This circuit oscillates a frequency of 1 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register. ★

(2) Subsystem clock oscillation

The circuit generates a 32.768 kHz signal. Oscillation cannot be stopped. If the subsystem clock oscillator is not used, non-use of the feedback resistance can be set the processor clock control register. This enables to decrease power consumption in the STOP mode.

During FIP display the noise elimination circuit functions automatically, and the effect of switching noise is reduced.

5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1 Clock Generator Configuration

Item	Configuration
Control register	Processor clock control register (PCC) Display mode register 0 (DSPM0) Display mode register 1 (DSPM1)
Oscillator	Main system clock oscillator Subsystem clock oscillator

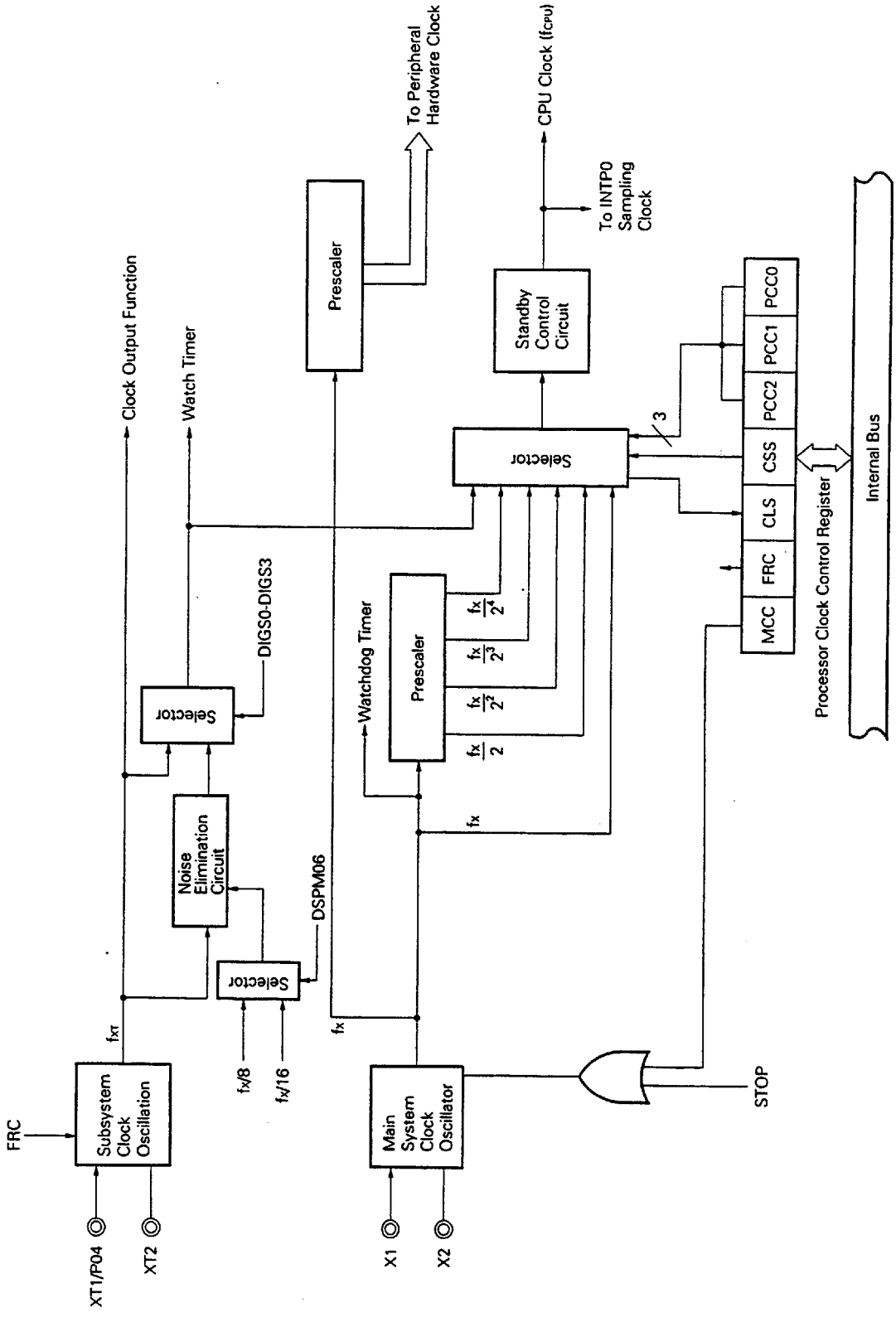


Fig. 5-1 Clock Generator Block Diagram

5.3 Clock Generator Control Register

The clock generator is controlled by the following 3 registers:

- Processor clock control register (PCC)
- Display mode register 0 (DSPM0)
- Display mode register 1 (DSPM1)

(1) Processor clock control register (PCC)

The PCC sets CPU clock selection, the ratio of division, main system clock oscillator operation/stop and whether subsystem clock oscillator feedback resistor is used or not.

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

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

Fig. 5-2 Subsystem Clock Feedback Resistor

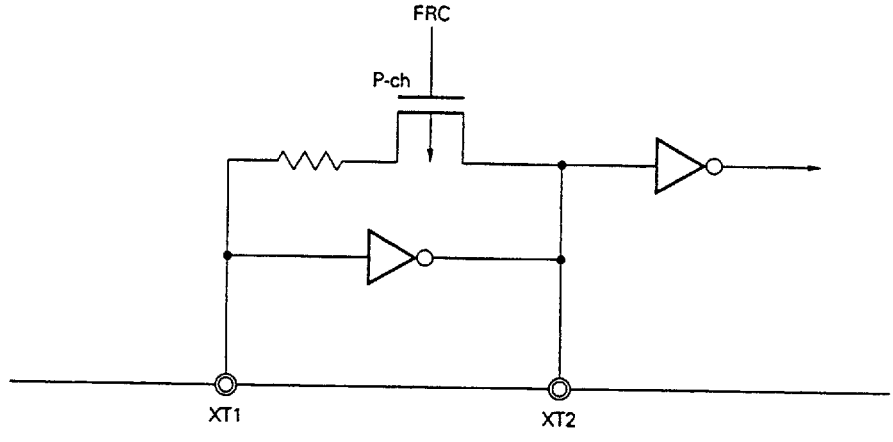


Fig. 5-3 Processor Clock Control Register Format

Symbol	⑦	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W[Note 1]

R/W

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection[Note 2]
0	0	0	0	fx (0.4 μs)
	0	0	1	fx/2 (0.8 μs)
	0	1	0	fx/2^2(1.6 μs)
	0	1	1	fx/2^3(3.2 μs)
	1	0	0	fx/2^4(6.4 μs)
1	0	0	0	f_{XT}(122 μs)
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than the above				Setting prohibited

R

CLS	CPU Clock Status
0	Main system clock
1	Subsystem clock

R/W

FRC	Subsystem Clock Feedback Resistor Selection
0	Feedback resistor connected
1	Feedback resistor disconnected

R/W

MCC	Main System Clock Oscillation Control[Note 3]
0	Oscillation possible
1	Oscillation stop

Notes 1. Bit 5 is Read Only.

2. FIP display is possible only when CSS is 0 and PCC2 through PCC0 are 000 or 001.

3. When stopping main system clock oscillation while the CPU is in operation with subsystem clocks, use the MCC. Do not use the STOP instruction.

Caution Be sure to set bit 3 to 0.

Remarks 1. fx : Main system clock oscillation frequency

2. f_{XT} : Subsystem clock oscillation frequency execution time operating at fx = 5.0 MHz and f_{XT} = 32.768 kHz : 2/f_{CPU}

★ **3.** Figs. in parentheses denote the instruction execution time operating at fx = 5.0 MHz and f_{XT} = 32.768 kHz : 2/f_{CPU}

(2) Display mode register 0 (DSPM0)

This register sets the mode for the subsystem clock noise elimination circuit.

DSPM0 is set by an 8-bit memory manipulation instruction. Bit 7 (KSF) only can be read by a 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DSPM0 to 00H.

Remark In addition to setting the mode for the subsystem clock noise elimination circuit, DSPM0 has the additional functions of setting the number of display segments and indicating the key scan timing.

Fig. 5-4 Display Mode Register 0 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM0	KSF	DSPM06	0	0	SEGS3	SEGS2	SEGS1	SEGS0	FFA0H	00H	R/W[Note 1]

SEGS3	SEGS2	SEGS1	SEGS0	Number of Display Segments
0	0	0	0	9 Segments
0	0	0	1	10 Segments
0	0	1	0	11 Segments
0	0	1	1	12 Segments
0	1	0	0	13 Segments
0	1	0	1	14 Segments
0	1	1	0	15 Segments
0	1	1	1	16 Segments
Other than above				Setting prohibited

DSPM06	Subsystem Clock Noise Elimination Circuit Mode Setting[Note 2]
0	2.5 MHz < fx ≤ 5.0 MHz
1	1.25 MHz ≤ fx ≤ 2.5 MHz

KSF	Timing Status
0	Display timing
1	Key scan timing

Notes 1. Bit 7 (KSF) is Read Only.

2. A value should be set to match the main system clock frequency (fx) used. The noise elimination circuit is effective during an FIP display operation.

Remark fx : Main system clock oscillation frequency

(3) Display mode register 1 (DSPM1)

This register sets display operation/display stoppage.

DSPM1 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DSPM1 to 00H.

Remark In addition to setting display operation/display stoppage, DSPM1 has the additional functions of setting the number of display digits, the digit signal cut width and the display cycle.

Fig. 5-5 Display Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM1	DIGS3	DIGS2	DIGS1	DIGS0	DIMS3	DIMS2	DIMS1	DIMS0	FFA1H	00H	R/W

DIMS0	Display Cycle Selection
0	1024/fx used as display cycle (1 display cycle = 204.8 μs : 5.0 MHz operation)
1	2048/fx used as display cycle (1 display cycle = 409.6 μs : 5.0 MHz operation)

DIMS3	DIMS2	DIMS1	Digit Signal Cut Width
0	0	0	1/16
0	0	1	2/16
0	1	0	4/16
0	1	1	6/16
1	0	0	8/16
1	0	1	10/16
1	1	0	12/16
1	1	1	14/16

DIGS3	DIGS2	DIGS1	DIGS0	Number of Display Digits
0	0	0	0	Display stopped (Static display)[Note]
0	0	0	1	2 digits
0	0	1	0	3 digits
0	0	1	1	4 digits
0	1	0	0	5 digits
0	1	0	1	6 digits
0	1	1	0	7 digits
0	1	1	1	8 digits
1	0	0	0	9 digits
1	0	0	1	10 digits
1	0	1	0	11 digits
1	0	1	1	12 digits
1	1	0	0	13 digits
1	1	0	1	14 digits
1	1	1	0	15 digits
1	1	1	1	16 digits

Note When the display stoppage is set, the static display is possible by operating the port output latch.

Remark fx : Main system clock oscillation frequency

5.4 System Clock Oscillation

5.4.1 Main system clock oscillation

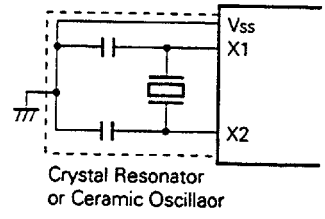
The main system clock oscillator oscillates with a crystal resonator (standard: 5.0 MHz) or a ceramic resonator ★ connected to the X1 and X2 pins.

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

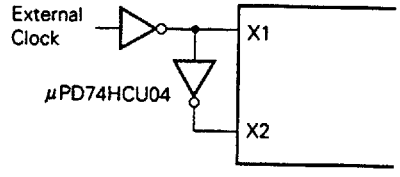
Fig. 5-6 shows an external circuit of the main system clock oscillator.

Fig. 5-6 External Circuit of Main System Clock Oscillator

(a) Crystal and ceramic oscillation



(b) External clock



Caution The STOP mode cannot be set while an external clock is input. This is because the X1 pin is short-circuited to Vss in the STOP mode.

5.4.2 Subsystem clock oscillator

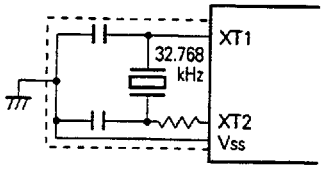
The subsystem clock oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the X1 and X2 pins.

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

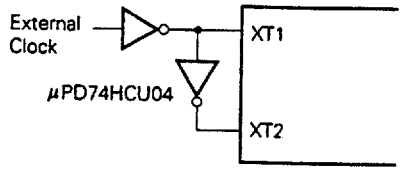
Fig. 5-7 shows an external circuit of the subsystem clock oscillator.

Fig. 5-7 External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



Refer to the next page for Caution.

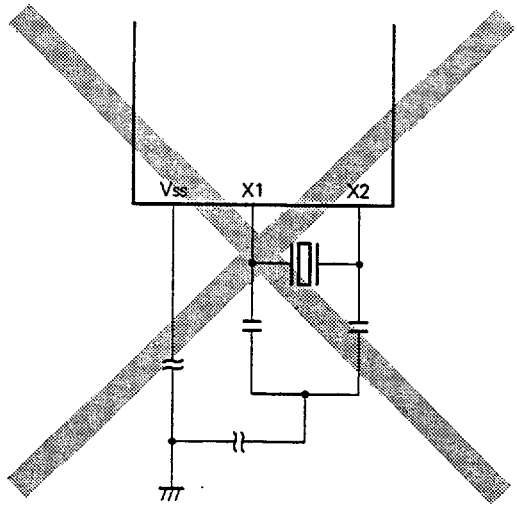
Caution When using a main system clock oscillator and a subsystem clock oscillator, carry out wiring in the dotted-line area in Figs. 5-6 and 5-7 as follows to prevent any effects from wiring capacities.

- Minimize the wiring length.
- Do not allow wiring to intersect with other signal conductors. Do not allow wiring to come near abruptly changing high current.
- Set the potential of the grounding position of the oscillator capacitor to that of Vss. Do not ground to any ground pattern where high current is present.
- Do not fetch signals from the oscillator.

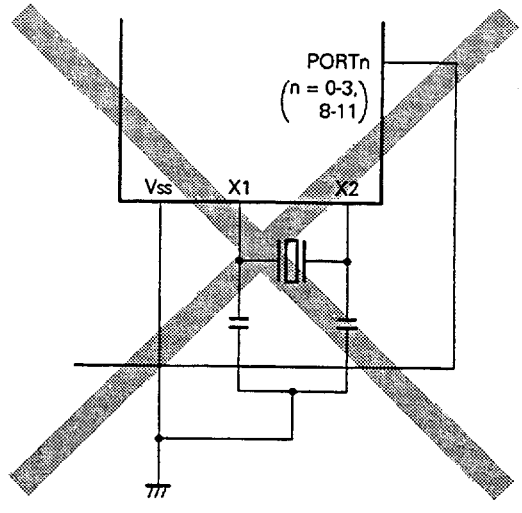
Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.
Fig. 5-8 shows examples of oscillator having bad connection.

Fig. 5-8 Examples of Resonator with Bad Connection (1/2)

(a) Wiring or connection circuits is too long



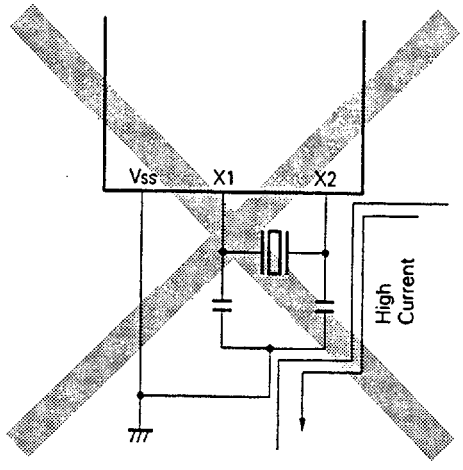
(b) Signal conductors intersect with each other



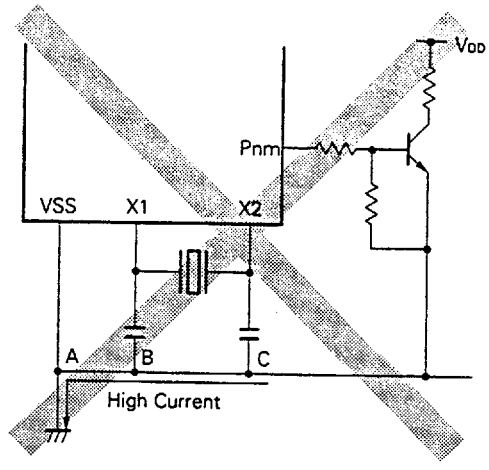
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Fig. 5-8 Examples of Resonator with Bad Connection (1/2)

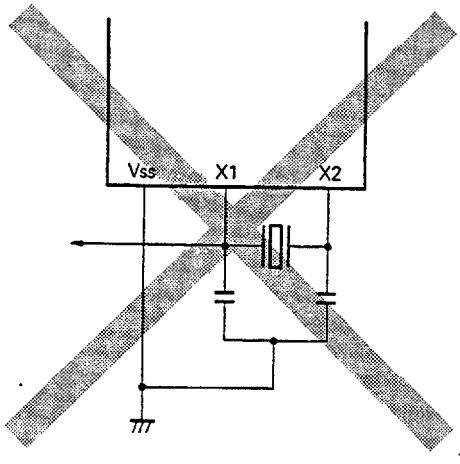
(c) Changing high current is too near a signal conductor



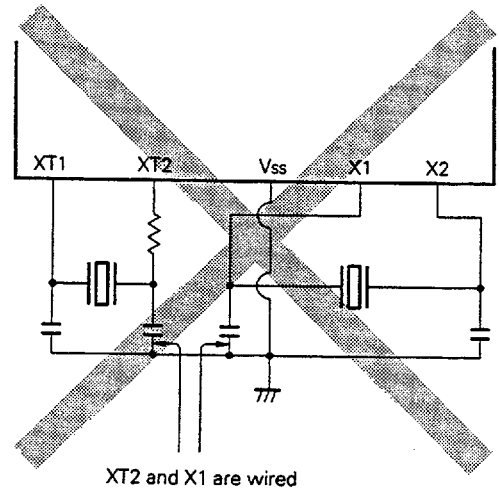
(b) Current flows through the grounding line of the oscillator (Potential at points A, B and C fluctuate)



(e) Signals are fetched



(f) Signal conductors of the main and subsystem clocks are parallel and near 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.

5.4.3 Frequency divider

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

5.4.4 When no subsystem clock 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 Vss
XT2: Leave 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 suppress this, the above on-chip feedback resistor can be removed with bit 6 (FRC) of the processor clock control register. In this case also, connect the XT1 and XT2 pins as described above.

5.5 Clock Generator Operations

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

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

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

(a) Upon generation of $\overline{\text{RESET}}$ signal, the lowest speed of the main system clock (6.4 μs when operated at 5.0 MHz) is selected (PCC = 04H). The main system clock oscillation is halted while low level signal is input at $\overline{\text{RESET}}$ pin. ★

(b) With the main system clock selected, one of the five CPU clocks (0.4 μs, 0.8 μs, 1.6 μs, 3.2 μs and 6.4 μs when operated at 5.0 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, it can be specified that the internal feedback resistor of the subsystem clock is not used, and the subsystem clock can be stopped.

(d) The PCC can be used to select the subsystem clock and to operate the system with low current consumption (minimum instruction execution time of 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 the clock timer and clock output functions only. Thus, the clock 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 for external clock input).

5.5.1 Main system clock operations

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

(a) Because the operation guarantee instruction execution speed depends on the power supply voltage, the 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 **Fig. 5-9**).

Fig. 5-9 Main System Clock Stop Function (1/2)

(a) Operation to be carried out when MCC is set after CSS setting with the main system clock in operation

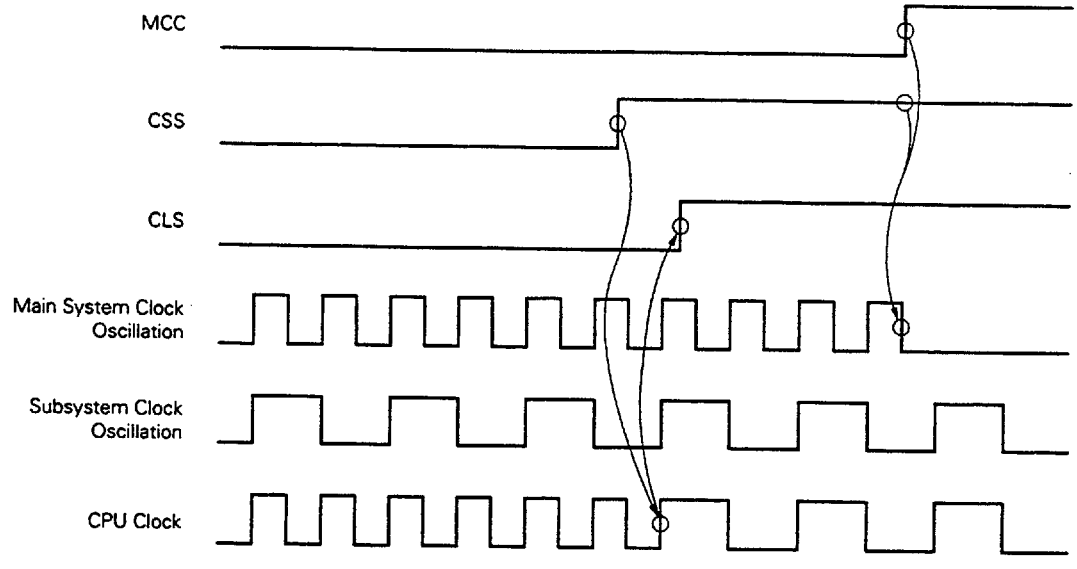
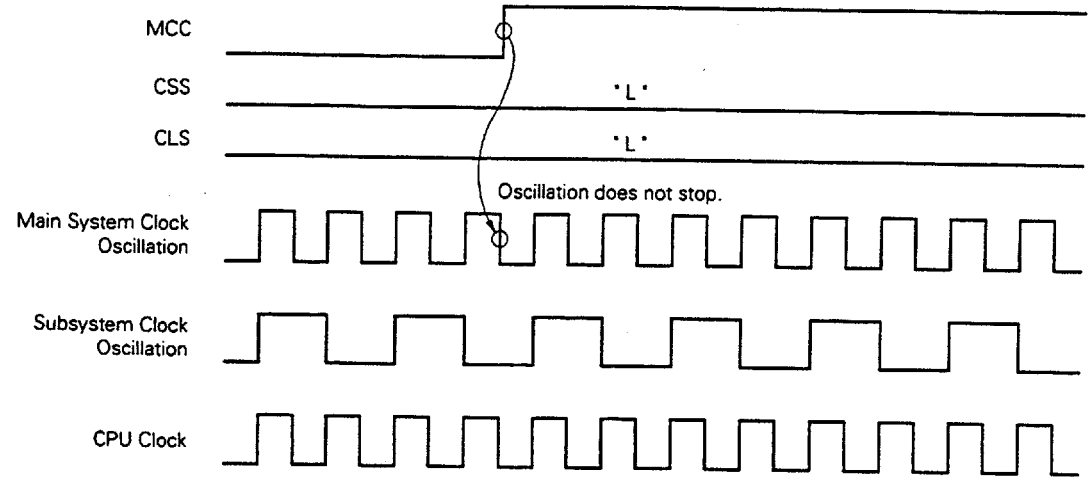
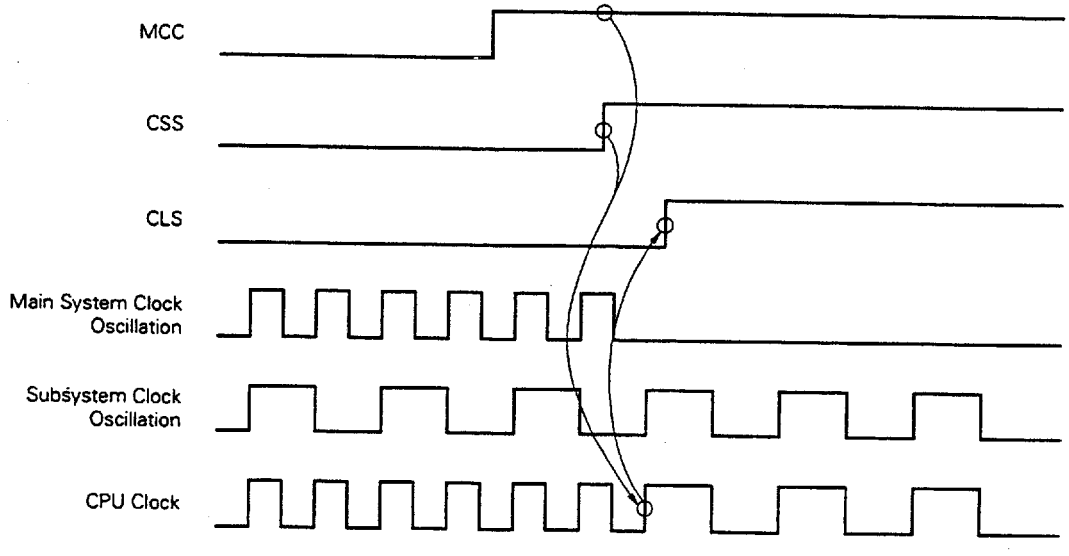


Fig. 5-9 Main System Clock Stop Function (2/2)

(b) Operation to be carried out when MCC is set with the main system clock



(c) Operation to be carried out when CSS is set after MCC is set with the main system clock in operation



5.5.2 Subsystem 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) 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 he subsystem clock is in operation.

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

The actual switchover operation is not performed directly after writing to the PCC, and operation continues on the pre-switchover clock for a number of clock cycles (see **Table 5-2**).

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 5-2 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 instructions				16 instructions				16 instructions				16 instructions				$f_X/2f_{XT}$ instructions (64 instructions)			
	0	0	1	8 instructions								8 instructions				8 instructions				8 instructions				$f_X/4f_{XT}$ instructions (32 instructions)			
	0	1	0	4 instructions				4 instructions								4 instructions				4 instructions				$f_X/8f_{XT}$ instructions (16 instructions)			
	0	1	1	2 instructions				2 instructions				2 instructions								2 instructions				$f_X/16f_{XT}$ instructions (8 instructions)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								$f_X/32f_{XT}$ instructions (4 instructions)			
1	×	×	×	1 instruction				1 instruction				1 instruction				1 instruction				1 instruction							

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 switchover from the subsystem clock to the main system clock (changing CSS from 1 to 0).

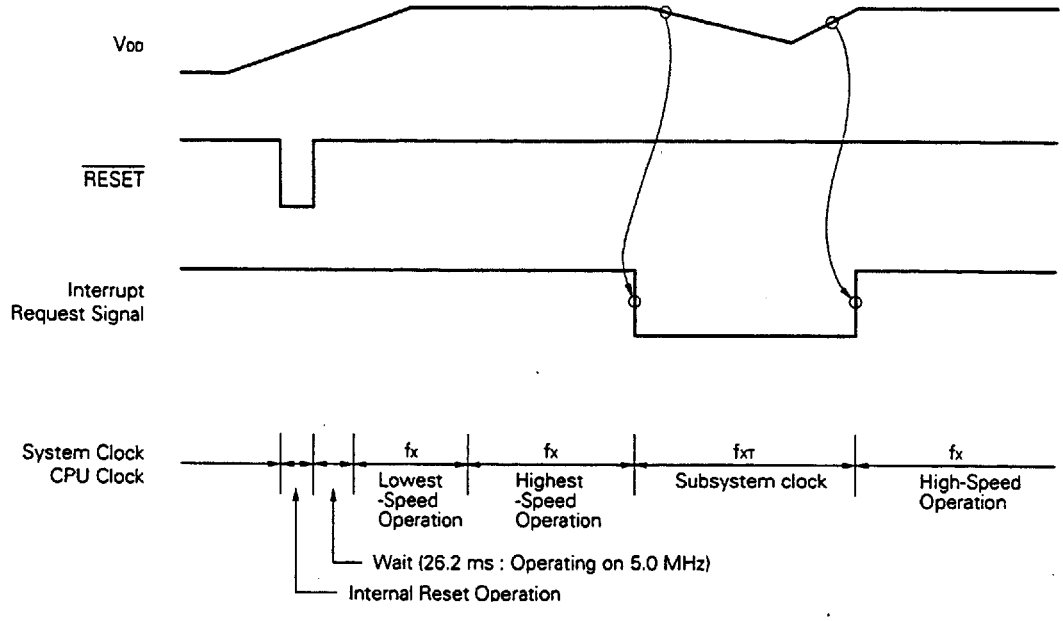
Remarks 1. 1 instruction is the minimum instruction execution time with the pre-switchover CPU clock.

2. Figs. in parentheses apply to operation with fx =5.0 MHz and f_{XT} = 32.768 MHz. ★

5.6.2 System clock and CPU clock switching procedure

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

Fig. 5-10 System Clock and CPU Clock Switching



① The CPU is reset by setting the $\overline{\text{RESET}}$ pin to low after power-on. After that, when the $\overline{\text{RESET}}$ pin is set to high, reset is released and the main system clock starts oscillating. Then, automatically the oscillation stabilization time ($2^{17}/f_X$) is secured.

★ After that, the CPU starts executing the instruction at the lowest speed of the main system clock (6.4 μs when operated at 5.0 MHz).

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

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

④ Upon detection of V_{DD} 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 highest-speed operation is resumed.

★ **Caution After the system clock has been changed to the subsystem clock from the main system clock, before returning to the main system clock again be sure that the program ensures elapsing of the oscillation stabilization time of the main system clock.**

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

The times incorporated into the μPD78024 subseries units arened below.

(1) 16-bit timer/event counter (TM0)

The TM0 can be used to serve as an interval timer and an external event counter, to output PWM and square waves with any selected frequency and to measure pulse widths (infrared remote controlled receive function).

(2) 8-bit timer/event counters (TM1 and TM2)

TM1 and TM2 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 7 8-BIT TIMER/EVENT COUNTER**).

(3) Watch timer (TM3)

This timer can set a flag every 0.5 sec and simultaneously generates interrupts at the preset time intervals (see **CHAPTER 8 WATCH TIMER**).

(4) Watchdog timer (WDTM)

WDTM can performs the watchdog timer function or generate non-maskable interrupts, maskable interrupts and $\overline{\text{RESET}}$ at the preset time intervals (see **CHAPTER 9 WATCHDOG TIMER**).

(5) Clock output control circuit

This circuit supplies other devices with the divided main system clock and the subsystem clock (see **CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT**).

(6) Buzzer output control circuit

This circuit outputs the buzzer frequency obtained by dividing the main system clock (see **CHAPTER 11 BUZZER OUTPUT CONTROL CIRCUIT**).

Table 6-1 Timer/Event Counter Types and Functions

		16-Bit Timer/Event Counter	8-Bit Timer/Event Counter	Watch Timer	Watchdog Timer
Type	Interval timer	1 channel	2 channels	1 channel Note 1	1 channel Note 2
	External event counter	O	O	-	-
Functions	Timer output	O	O	-	-
	PWM output	O	-	-	-
	Pulse width measurement	O	-	-	-
	Square wave output	O	O	-	-
	Interrupt request	O	O	O	O

Notes 1. TM3 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.

6.1 16-Bit Timer/Event Counter Functions

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

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

(1) Interval timer

TM0 generates an interrupt at the preset time intervals.

Table 6-2 16-Bit Timer/Event Counter Interval Times

Minimum Interval Time	Maximum Interval Time	Resolution
2 x TI0 Input Cycle	2^{16} x TI0 Input Cycle	TI0 Input Edge Cycle
2 x 1/fx (400 ns)	2^{16} x 1/fx (13.1 ms)	1/fx (200 ns)
2^2 x 1/fx (800 ns)	2^{17} x 1/fx (26.2 ms)	2 x 1/fx (400 ns)
2^3 x 1/fx (1.6 μs)	2^{18} x 1/fx (52.4 ms)	2^2 x 1/fx (800 ns)
2^4 x 1/fx (3.2 μs)	2^{19} x 1/fx (104.9 ms)	2^3 x 1/fx (1.6 μs)

Remarks 1. fx : Main system clock oscillation frequency

2. ( ) : Values in parentheses when operating at fx = 5.0 MHz ★

(2) PWM output

TM0 can g14 bits.

(3) Pulse width measurement

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

(4) External event counter

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

(5) Square-wave output

TM0 can output a square wave with any frequency.

Table 6-3 16-Bit Timer/Event Counter Square-Wave Output Range

Minimum Pulse Time	Maximum Pulse Time	Resolution
2 x TI0 Input Cycle	2^{16} x TI0 Input Cycle	TI0 Input Edge Cycle
2 x 1/fx (400 ns)	2^{16} x 1/fx (13.1 ms)	1/fx (200 ns)
2^2 x 1/fx (800 ns)	2^{17} x 1/fx (26.2 ms)	2 x 1/fx (400 ns)
2^3 x 1/fx (1.6 μs)	2^{18} x 1/fx (52.4 ms)	2^2 x 1/fx (800 ns)
2^4 x 1/fx (3.2 μs)	2^{19} x 1/fx (104.9 ms)	2^3 x 1/fx (1.6 μs)

Remarks 1. fx : Main system clock oscillation frequency

2. ( ) : Values in parentheses when operated at fx = 5.0 MHz ★

6.2 16-Bit Timer/Event Counter Configuration

The 16-bit timer/event counter consists of the following hardware:

Table 6-4 16-Bit Timer/Event Counter Configuration

Item	Configuration
Timer register	16 bits x 1 (TM0)
Register	16-bit compare register : 1 (CR00) 16-bit capture register : 1 (CR01)
Timer output	1 (TO0)
Control register	Timer clock select register 0 (TCL0) 16-bit timer mode control register (TMC0) 16-bit timer output control register (TOC0) Port mode register 3 (PM3) External interrupt mode register (INTM0) Sampling clock select register (SCS)

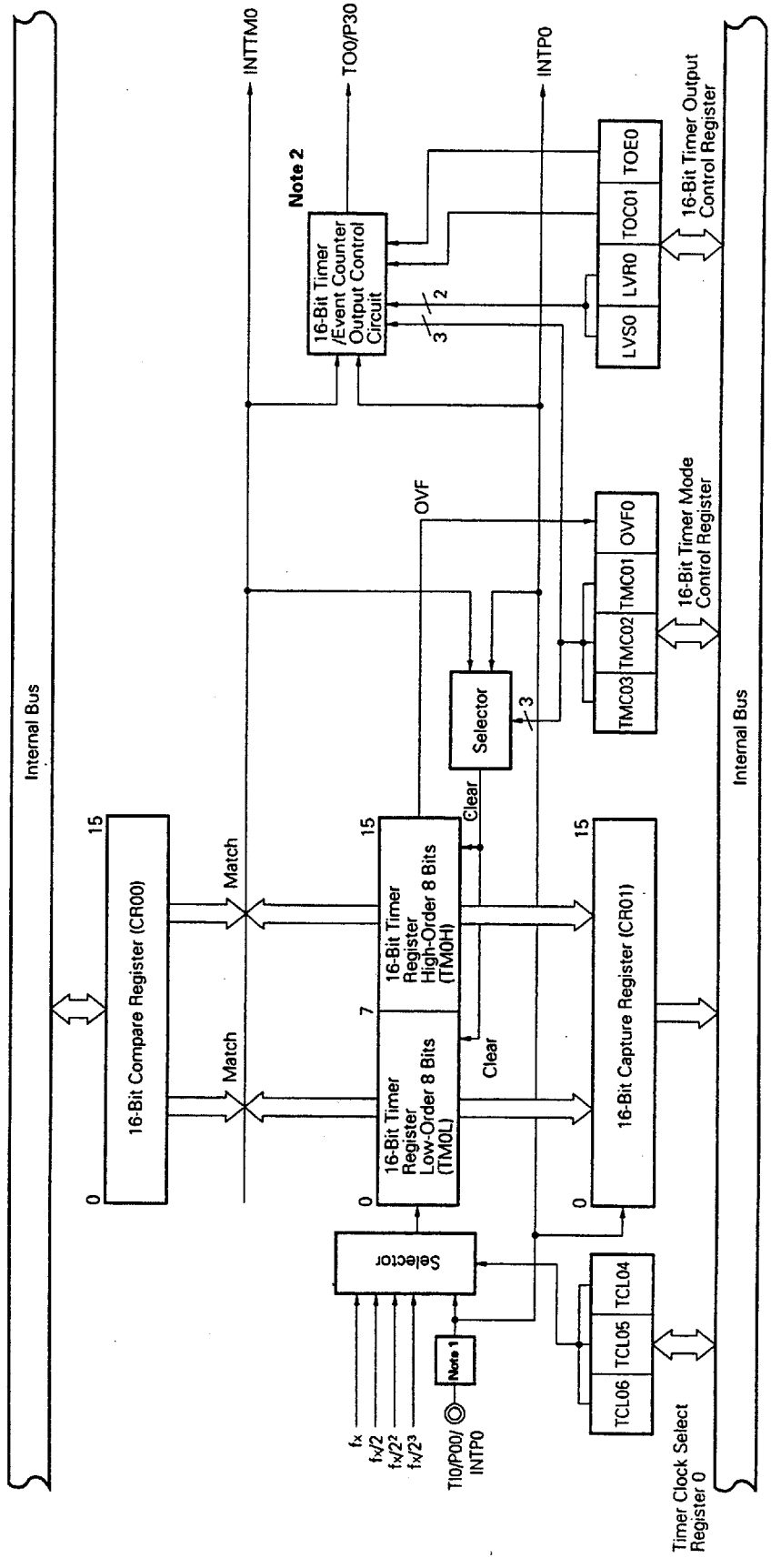


Fig. 6-1 16-Bit Timer/Event Counter (Timer Mode) Block Diagram

Notes 1. Edge detector

2. Refer to Fig. 6-3 for details of the 16-bit timer/event counter output control circuit configuration.

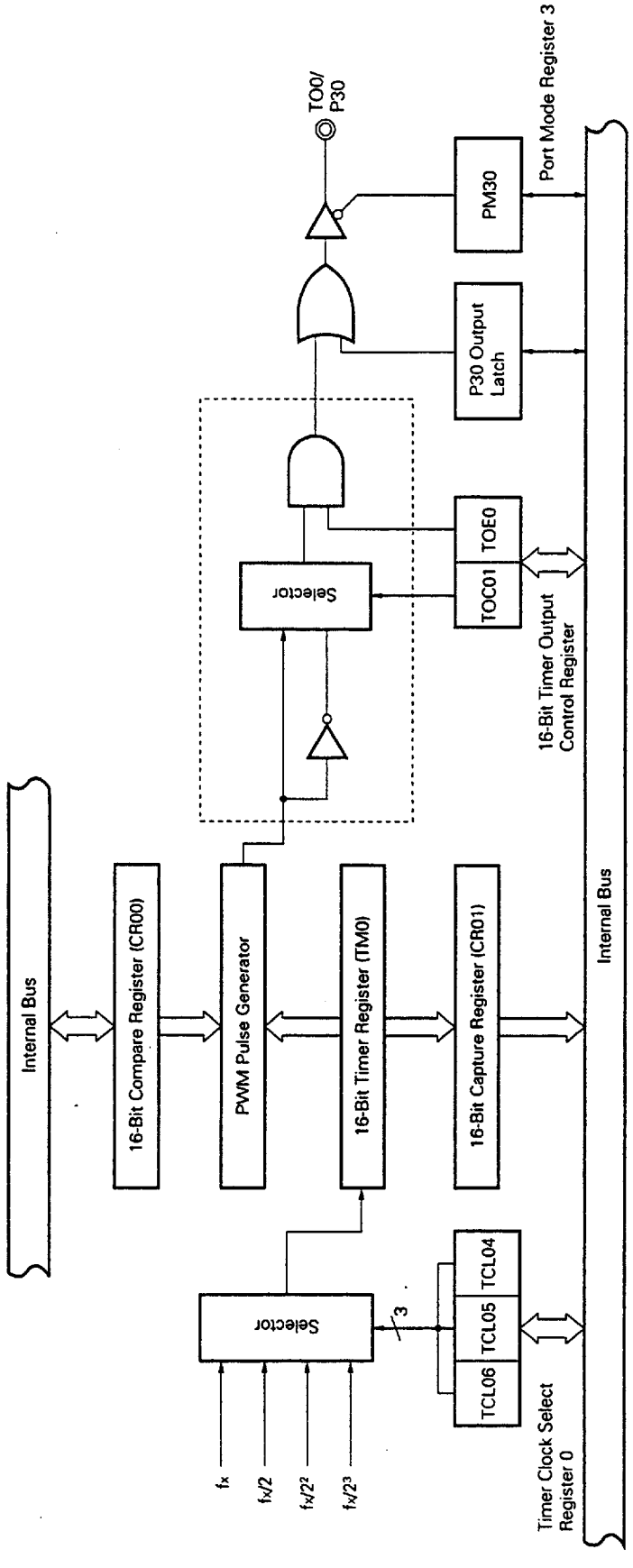


Fig. 6-2 16-Bit Timer/Event Counter (PWM Mode) Block Diagram

Remark The section within the dotted line is included in the output control circuit.

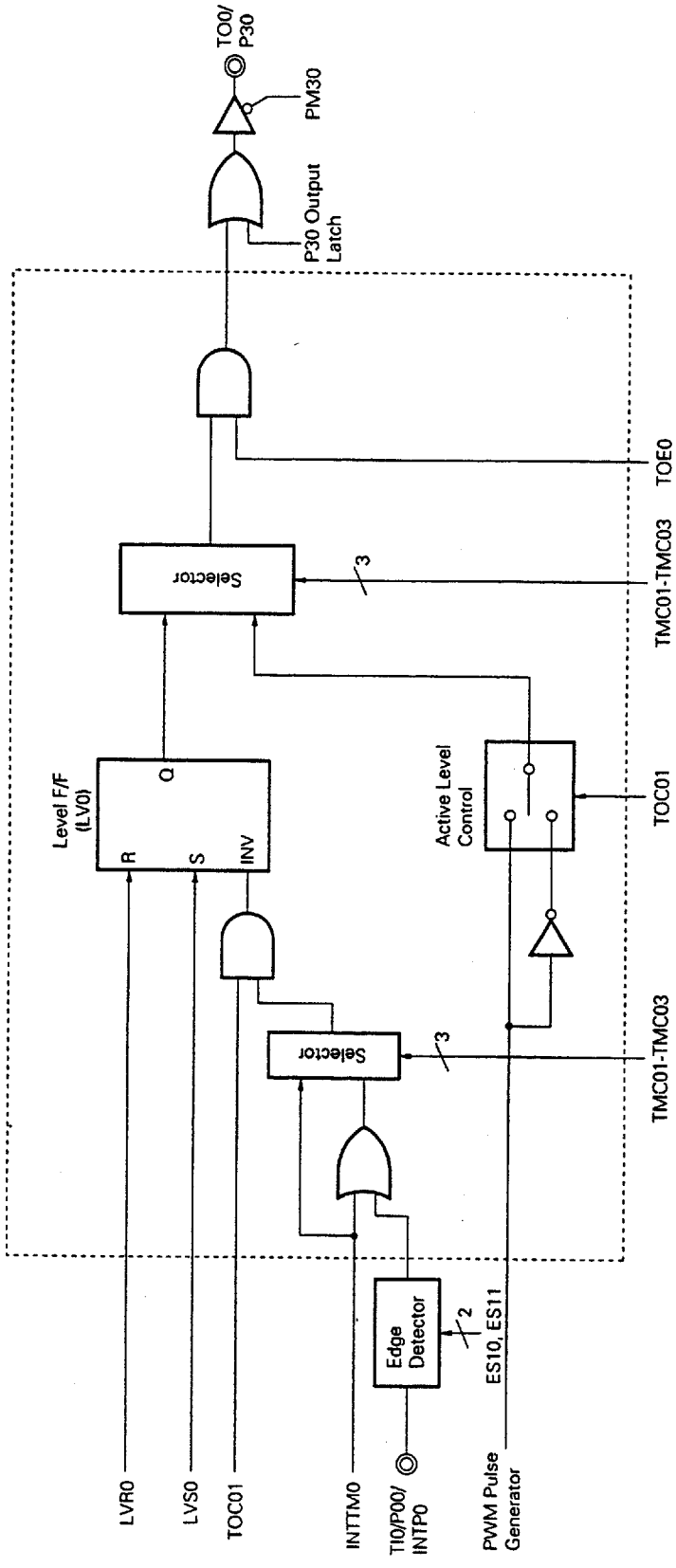


Fig. 6-3 16-Bit Timer/Event Counter Output Control Circuit Block Diagram

Remark The section within the dotted line is the output control circuit.

(1) 16-bit compare register (CR00)

This is a 16-bit register which compares the values set to CR00 and the count value of the 16-bit timer register (TM0) and, when the two values match, it generates an interrupt request (INTTM0).

When TM0 is set for interval timer operation, CR00 can also be used to hold the interval times and to set the pulse widths the PWM operation mode.

CR00 is set with a 16-bit memory manipulate instruction. 0001H to FFFFH values can be set.

$\overline{\text{RESET}}$ input makes CR00 undefined.

Caution Set PWM data (14 bits) to the high-order 14 bits of CR00. Set 00 to the low-order 2 bits.

(2) 16-bit capture register (CR01)

This is a 16-bit register which captures the contents of 16-bit timers (TM0).

Capture trigger is an valid edge input of the INTP0/TI0 pin. The INTP0 valid edge is set with the external interrupt mode register.

CR01 is read with a 16-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input makes CR01 undefined.

(3) 16-bit timer register (TM0)

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

TM0 is read with a 16-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TM0 to 0000H.

Caution When the TM0 value is read, the CR01 value is destroyed.

6.3 16-Bit Timer/Event Counter Control Registers

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

- Timer clock select register 0 (TCL0)
- 16-bit timer mode control register (TMC0)
- 16-bit timer output control register (TOC0)
- Port mode register 3 (PM3)
- External interrupt mode register (INTM0)
- Sampling clock select register (SCS)

(1) Timer clock select register 0 (TCL0)

This register is used to set the count clock of the 16-bit timer register.

TCL0 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TCL0 to 00H.

Remark TCL0 has the function of setting the PCL output clock in addition to that of setting the count clock of the 16-bit timer register.

Fig. 6-4 Timer Clock Select Register 0 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL0	CLOE	TCL06	TCL05	TCL04	TCL03	TCL02	TCL01	TCL00	FF40H	00H	R/W

TCL03	TCL02	TCL01	TCL00	PCL Output Clock Selection
0	0	0	0	f_{XT} (32.768 kHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
Other than the above				Setting prohibited

TCL06	TCL05	TCL04	16-Bit Timer Register Count Clock Selection
0	0	0	TI0 (valid edge specifiable)
0	0	1	f_X (5.0 MHz)
0	1	0	$f_X/2$ (2.5 MHz)
0	1	1	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_X/2^3$ (6.25 kHz)
Other than the above			Setting prohibited

CLOE	PCL Output Control
0	Output disabled
1	Output enabled

Cautions 1. Set the effective edge of the TI0/INTP0 pin with an external interrupt mode register. The sampling clock frequency is selected with a sampling clock select register.

2. When enabling PCL output, first set TCL00 to TCL03 and then set CLOE to 1 with a 1-bit memory manipulate instruction.

3. When reading the count value with TM0 count clock set ti TI0, do so from TM0 instead of CR01 capture register.

★ **4. To write different data to TCL0, stop the timer operation once.**

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XT} : Subsystem clock oscillation frequency

3. TI0 : 16-bit timer/event counter input pin

4. TM0 : 16-bit timer register

★ **5.** Values in parentheses when operated at f_X = 5.0 MHz and f_{XT} = 32.768 kHz.

6. Refer to **CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT** for details of PCL.

(2) 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 to 00H.

Caution The 16-bit timer register starts operating when a value other than 0, 0, 0 (operation stop mode) is set to TMC01-TMC03. To stop the operation, set 0, 0, 0 to TMC01-TMC03. ★

Fig. 6-5 16-Bit Timer Mode Control Register Format

Symbol	7	6	5	4	3	2	1	⓪	Address	After Reset	R/W
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0	FF48H	00H	R/W

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

TMC03	TMC02	TMC01	Operation Mode Clear Mode Selection	TO0 Output Timing Selection	Interrupt Occurrence
0	0	0	Operation stop (TM0 cleared to 0)	No change	No interrupt
0	0	1	PWM mode (free running)	PWM pulse output	Interrupt generation upon TM0 and CR00 match
0	1	0	Free running mode	TM0 and CR00 match	
0	1	1		TM0 and CR00 match or TI0 valid edge	
1	0	0	Clear and start at TI0 valid edge	TM0 and CR00 match	
1	0	1		TM0 and CR00 match or TI0 valid edge	
1	1	0	Clear and start upon TM0 and CR00 match	TM0 and CR00 match	
1	1	1		TM0 and CR00 match or TI0 valid edge	

Cautions 1. Switch the clear mode and the TO0 output timing after stoppping the timer operation (by setting TMC01 to TMC03 to 0, 0, 0).

2. Set the valid edge of the TI0/INTP0 pin with an external interrupt mode register and select the sampling clock frequency with a sampling clock select register.

3. When using the PWM mode, set the PWM and then set data to CR00.

Remarks 1. TO0 : 16-bit timer/event counter output pin

2. TI0 : 16-bit timer/event counter input pin

3. TM0 : 16-bit timer register

4. CR00: Compare register 00

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

This register controls operations of the 16-bit timer/event counter output control circuit. It sets/resets the R-S flip-flop (LV0), sets the active level in the PWM mode, enables/disables output inversion except in the PWM mode and sets the data output mode.

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

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

Fig. 6-6 16-Bit Timer Output Control Register Format

Symbol	7	6	5	4	③	②	1	⓪	Address	After Reset	R/W
TOC0	0	0	0	0	LVS0	LVR0	TOC01	TOE0	FF4EH	00H	R/W

TOE0	16-Bit Timer/Event Counter Output Control
0	Output disabled (Port mode)
1	Output enabled

TOC01	PWM Mode	Other than PWM Mode
	Active Level Selection	Timer Output F/F Control
0	High active	Inversion operation disabled
1	Low active	Inversion operation enabled

LVS0	LVR0	16-Bit Timer/Event Counter Timer Output F/F Status Set
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

Cautions 1. Be sure to set TOC0 after stopping timer operation.

2. After data has been set, 0 can be read from LVS0 and LVR0.

(4) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P30/TO0 pin for timer output, set PM30 and the P30 output latch to 0. PM3 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Fig. 6-7 Port Mode Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM3	1	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 6)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Caution Set 1 in bit 7.

(5) External interrupt mode register (INTM0)

This register is used to set INTP0 to INTP2 valid edges.

INTM0 is set with an 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets INTM0 to 00H.

Remarks 1. INTP0 pin and TI0/P00 serve dual functions.

2. INTP3 is fixed to the falling edge.

Fig. 6-8 External Interrupt Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM0	ES31	ES30	ES21	ES20	ES11	ES10	0	0	FFECH	00H	R/W

ES11	ES10	INTP0 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES21	ES20	INTP1 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	INTP2 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

(6) Sampling clock select register (SCS)

This register sets clocks which undergo clock sampling of valid edges to be input to INTP0. When remote controlled reception is carried out using INTP0, digital noise is removed with sampling clock.

SCS is set with an 8-bit memory manipulate instruction.

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

Fig. 6-9 Sampling Clock Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	INTP0 Sampling Clock Selection
0	0	$fx/2^{N+1}$
0	1	Setting prohibited
1	0	$fx/2^6$ (78.1 kHz)
1	1	$fx/2^7$ (39.1 kHz)

Caution $fx/2^{N+1}$ is a clock to be supplied to CPU. $fx/2^6$, $fx/2^7$ are clocks to be supplied to the peripheral hardware. $fx/2^{N+1}$ remains at reset in the HALT mode.

Remarks 1. N : Value set to bits 0 to 2 (PCC0 to PCC2) of the processor lock control register (N = 0 to 4)

2. fx: Main system clock oscillation frequency

★ **3.** Values in parentheses when operated at fx = 5.0 MHz.

6.4 16-Bit Timer/Event Counter Operation

6.4.1 Interval timer operations

When bits 2 and 3 (TMC02 and TMC03) of the 16-bit timer mode control register (TMC0) are set to 1 and 1, respectively, the 16-bit timer/event counter operates as interval timer. Interrupts are repeatedly generated at intervals of the count value preset to the 16-bit compare register (CR00).

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 (INTTM0) is generated.

Count clock of the 16-bit timer/event counter can be selected with bits 4 to 6 (TCL04 to TCL06) of the timer clock select register (TCL0).

Fig. 6-10 Interval Timer Configuration

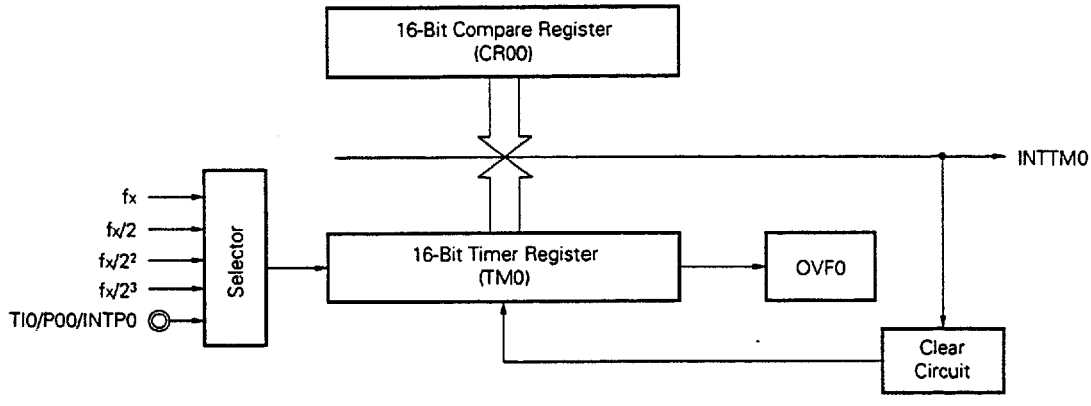
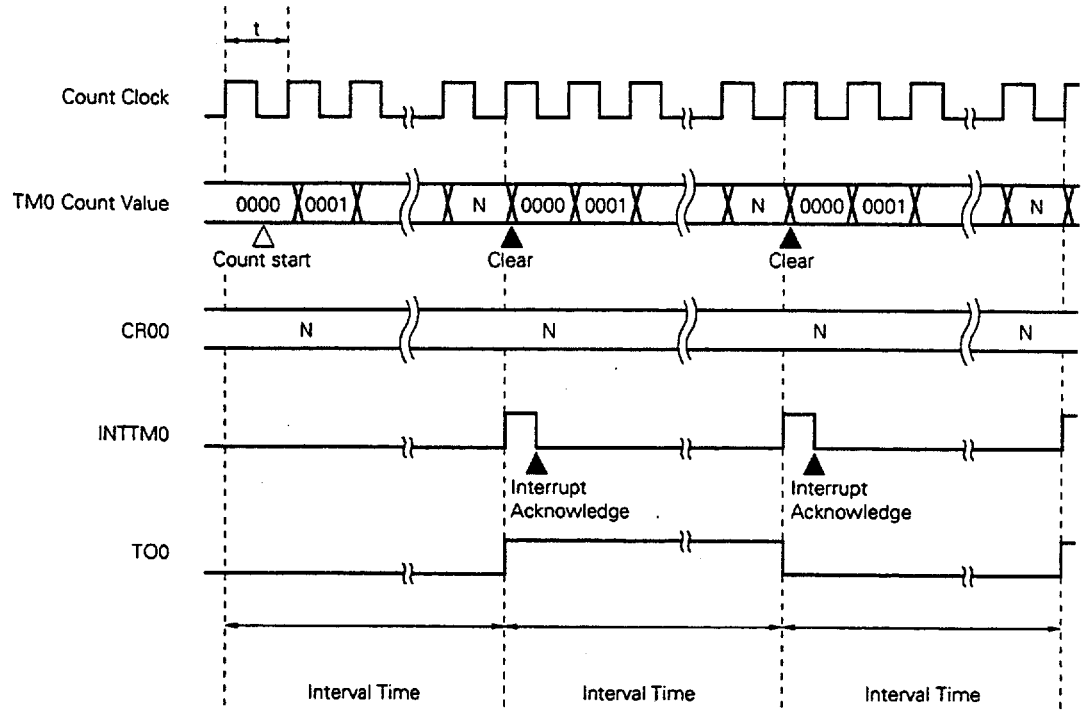


Fig. 6-11 Interval Timer Operation Timings



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

Table 6-5 16-Bit Timer/Event Counter Interval Times

TCL06	TCL05	TCL04	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
0	0	1	2 × 1/fx (400 ns)	2^{16} × 1/fx (13.1 ms)	1/fx (200 ns)
0	1	0	2^2 × 1/fx (800 ns)	2^{17} × 1/fx (26.2 ms)	2 × 1/fx (400 ns)
0	1	1	2^3 × 1/fx (1.6 μs)	2^{18} × 1/fx (52.4 ms)	2^2 × 1/fx (800 ns)
1	0	0	2^4 × 1/fx (3.2 μs)	2^{19} × 1/fx (104.9 ms)	2^3 × 1/fx (1.6 μs)
Other than the above			Setting prohibited		

Remarks 1. fx: Main system clock oscillation frequency

★ **2.** Values in parentheses when operated at 5.0 MHz

6.4.2 PWM output operation

When bits 1 to 3 (TMC01 to 03) of the 16-bit timer mode control register (TMC0) are set to 1, 0 and 1, respectively, the 16-bit timer/event counter operates as PWM output. Pulses with a duty ratio to be determined with the value set to the 16-bit compare register (CR00) are output from the TO0/P30 pin.

Set the active level width of the PWM pulse to the high-order 14 bits of CR00. Select the active level with bit 1 (TOC01) of the 16-bit timer output control register (TOC0).

This PWM pulse has a 14-bit resolution. The pulse can be converted to an analog voltage by integrating it with an external low pulse filter (LPF). The PWM pulse has a combination of the basic cycle determined by $2^8/\Phi$ and the sub cycle determined by $2^{14}/\Phi$ so that the time constant of the external LPF can be shortened. Count clock can be selected with bits 4 to 6 (TCL04 to TCL06) of the timer clock select register (TCL0).

PWM output enable/disable can be selected with bit 0 (TOE0) of TOC0.

Cautions 1. Setting to CR00 should be performed after PWN operating mode selection.

2. Be sure to write 0 to bits 0 and 1 of CR00.

3. Do not select the PWM operation mode for external clock input form the TI0/P00 pin.

By integrating 14-bit resolution PWM pulses with an external low-pass filter, they can be converted to an analog voltage and used for electronic tuning and D/A converter applications, etc.

The analog output voltage (V_{AN}) used for D/A conversion with the configuration shown in Fig. 6-12 is as follows.

$$V_{AN} = V_{REF} \times \frac{\text{Compare register (CR00) value}}{2^{16}}$$

V_{REF}: External switching circuit reference voltage

Fig. 6-12 Example of D/A Converter Configuration Using PWM Output

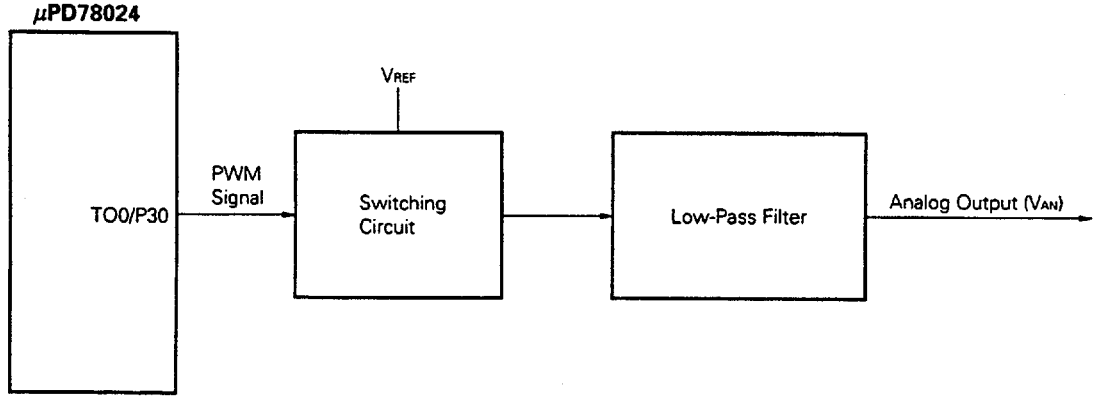
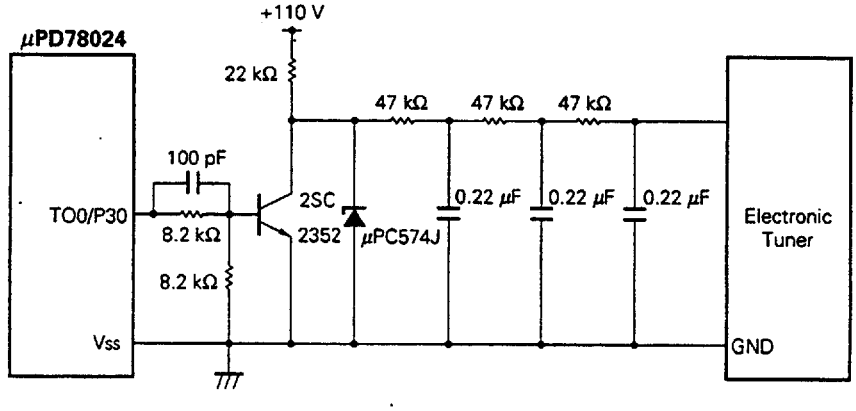


Fig. 6-13 shows an example in which PWM output is converted to an analog voltage and used in a voltage synthesizer type TV tuner.

Fig. 6-13 Example of TV Tuner Application Circuit



6.4.3 Pulse width measurement operation

The width of an external pulse to be input to the TI0/P00 pin is measured.

Two measurement methods are available. One is to measure the pulse width while free-running the 16-bit timer register (TM0) and the other is to measure the pulse width after restarting the timer at the effective edge of a signal to be input to the TI0/P00 pin.

(1) Pulse width measurement by means of free-running

If the edge specified with the external interrupt mode register (INTM0) is input to the TI0/P00 pin while TM0 is in operation, the TM0 value is fetched into the capture register (CR01) and the external interrupt request signal (INTP0) is set.

The rising edge, the falling edge or both edges can be selected with bits 2 and 3 (ES10 and ES11) of INTM0.

Because capture operation is carried out only after the valid edge is detected twice by sampling at the cycle selected with the sampling clock select register (SCS), noise with short pulse widths can be removed.

Fig. 6-14 Configuration of Pulse Width Measurement by Way of Free Running

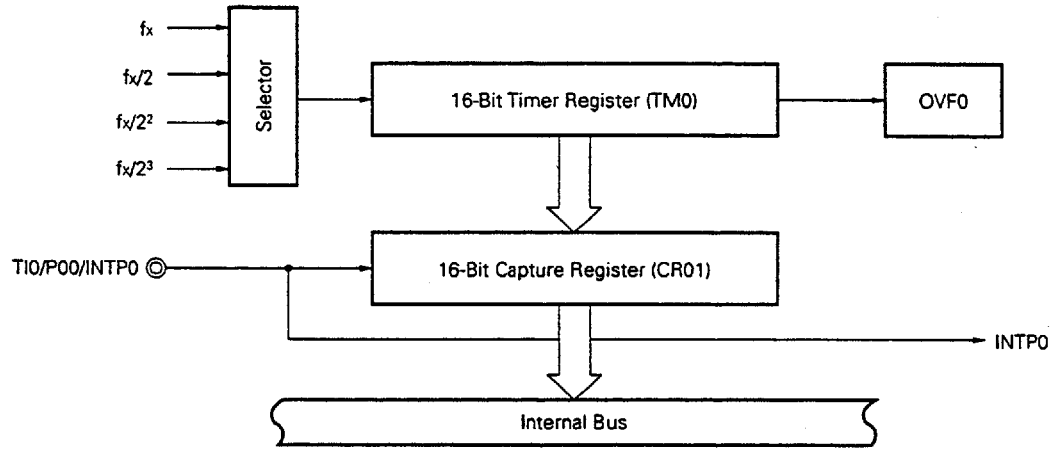
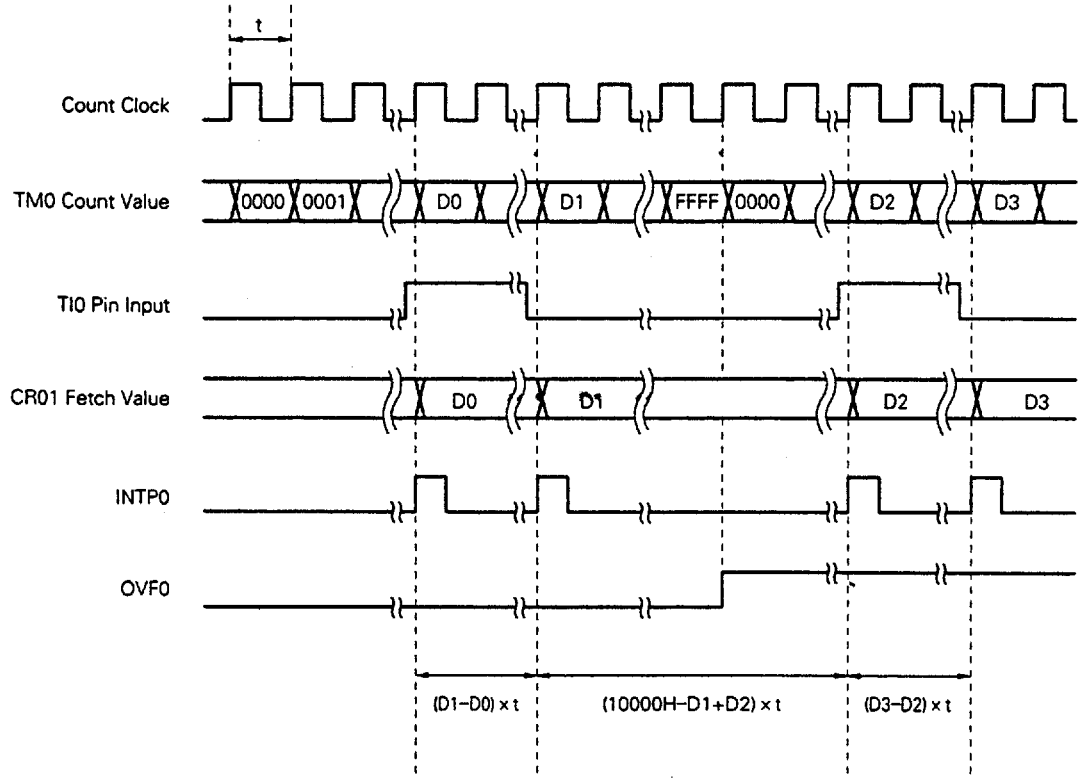


Fig. 6-15 Pulse Width Measurement Timings in Free Running (with Both Edges Specified)

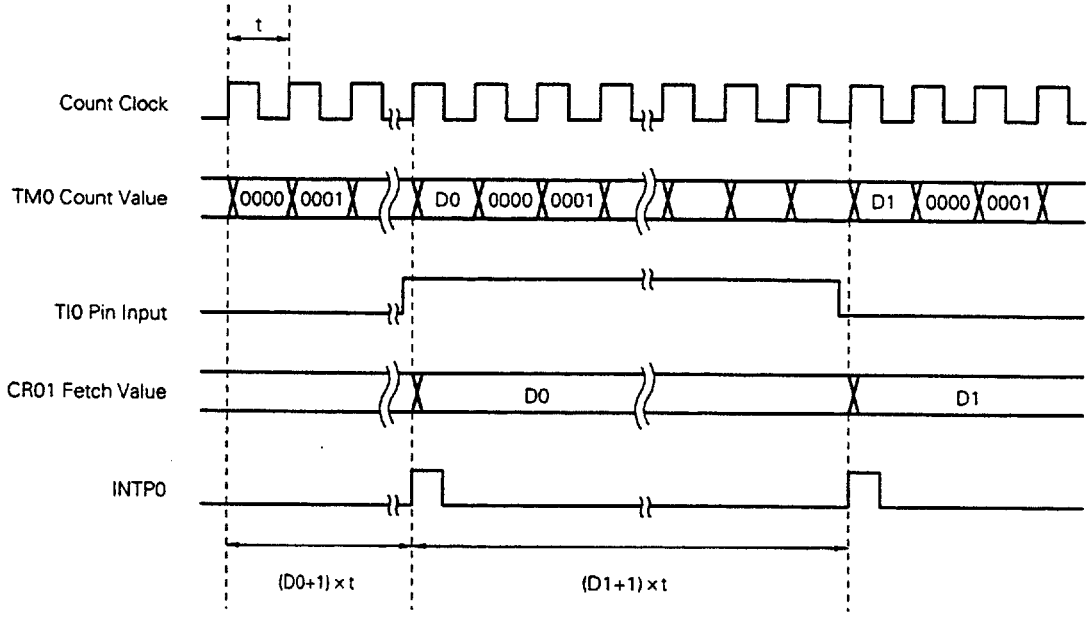


(2) Pulse width measurement by restart

Upon detection of a valid edge to the TI0/P00 pin, the pulse width of a signal input to the TI0/P00 pin is measured by first fetching the count value of the 16-bit timer register (TM0) into the capture register (CR01), then clearing TM0 and restarting counting.

The rising edge, the falling edge or both edges can be selected with bits 2 and 3 (ES10 and ES11) of INTM0. Because capture operation is carried out only after the valid edge is detected twice by sampling at the cycle selected with the sampling clock select register (SCS), noise with short pulse widths can be removed.

Fig. 6-16 Timings of Pulse Width Measurement by Restart (with Both Edges Specified)



6.4.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI0/P00 pin with the 16-bit timer register (TM0).

TM0 is incremented each time the valid edge specified with the external interrupt mode register (INTM0) is input.

When the TM0 counted value matches the 16-bit compare register (CR00) value, TM0 is cleared to 0 and the interrupt request signal (INTTM0) is generated.

The rising edge, the falling edge or both edges can be selected with bits 2 and 3 (ES10 and ES11) of INTM0.

Because operation is carried out only after the valid edge is detected twice by sampling at the cycle selected with the sampling clock select register (SCS), noise with short pulse widths can be removed.

Fig. 6-17 External Event Counter Configuration

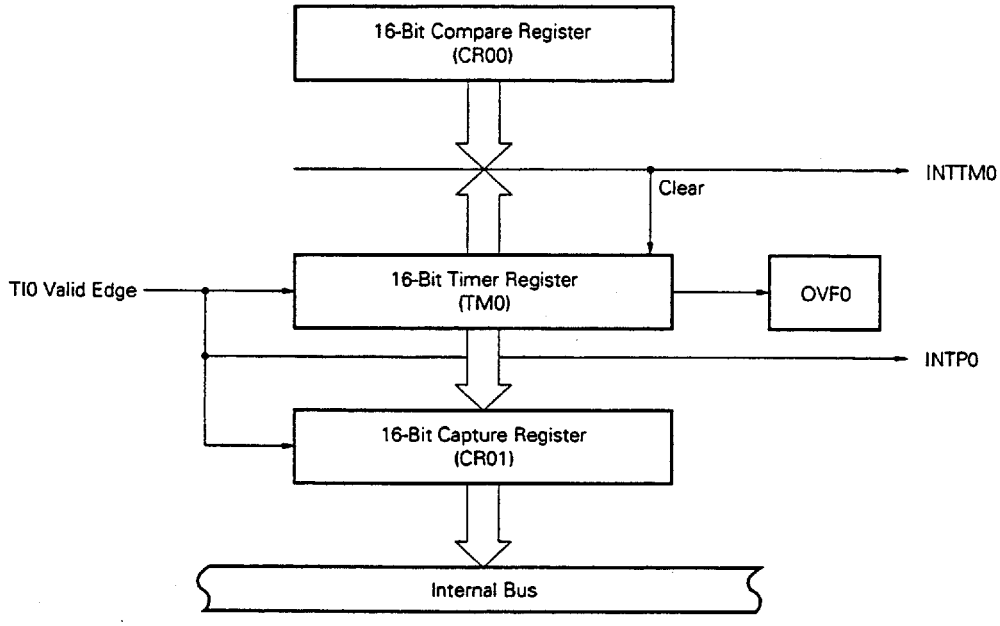
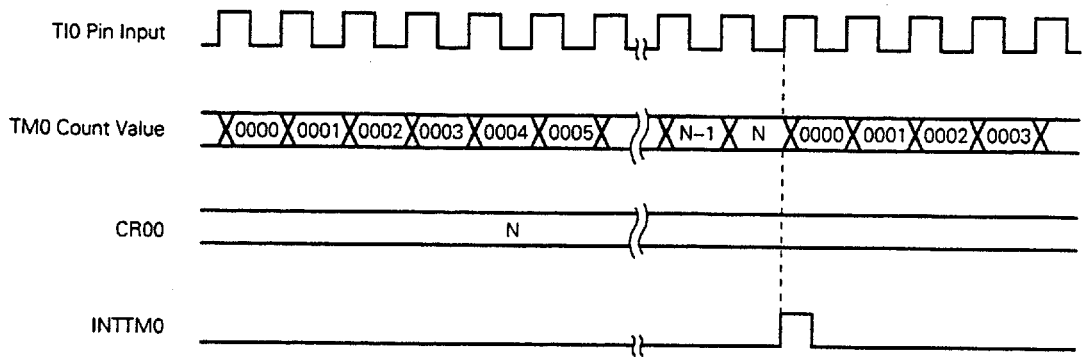


Fig. 6-18 External Event Counter Operation Timings (with Rising Edge Specified)



6.4.5 Square wave output operation

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

The TO0/P00 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 with any selected frequency to be output.

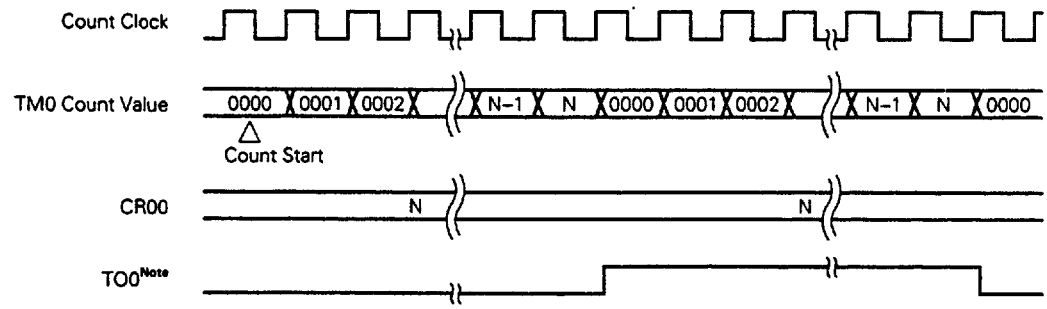
Table 6-6 16-Bit Timer/Event Counter Square-Wave Output Ranges

TCL06	TCL05	TCL04	Minimum Pulse Time	Maximum Pulse Time	Resolution
0	0	0	2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
0	0	1	2 × 1/fx (400 ns)	2^{16} × 1/fx (13.1 ms)	1/fx (200 ns)
0	1	0	2^2 × 1/fx (800 ns)	2^{17} × 1/fx (26.2 ms)	2 × 1/fx (400 ns)
0	1	1	2^3 × 1/fx (1.6 μs)	2^{18} × 1/fx (52.4 ms)	2^2 × 1/fx (800 ns)
1	0	0	2^4 × 1/fx (3.2 μs)	2^{19} × 1/fx (104.9 ms)	2^3 × 1/fx (1.6 μs)

Remarks 1. fx: Main system clock oscillation frequency

★ 2. Values in parentheses when operated at fx = 5.0 MHz

Fig. 6-19 Timing of Square Wave Output Operation



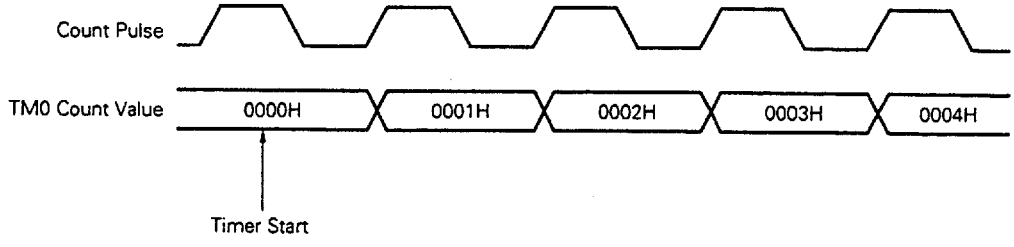
Note The initial value of TO0 output can be set with LVS0 and LVR0.

6.5 16-Bit Timer/Event Counter 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.

Fig. 6-20 16-Bit Timer Register Start Timings



(2) 16-bit compare register set

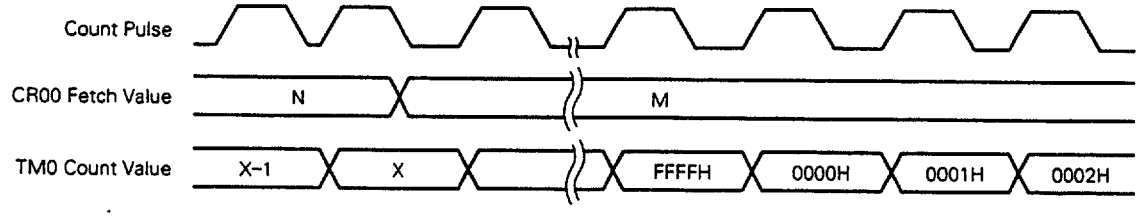
Set a value other than 0000H to the 16-bit compare register (CR00).

Thus, when using the 16-bit compare register as event counter, one-pulse count operation cannot be carried out.

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

If the value after the 16-bit 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.

Fig. 6-21 Timings after Compare Register Change during Timer Count Operation

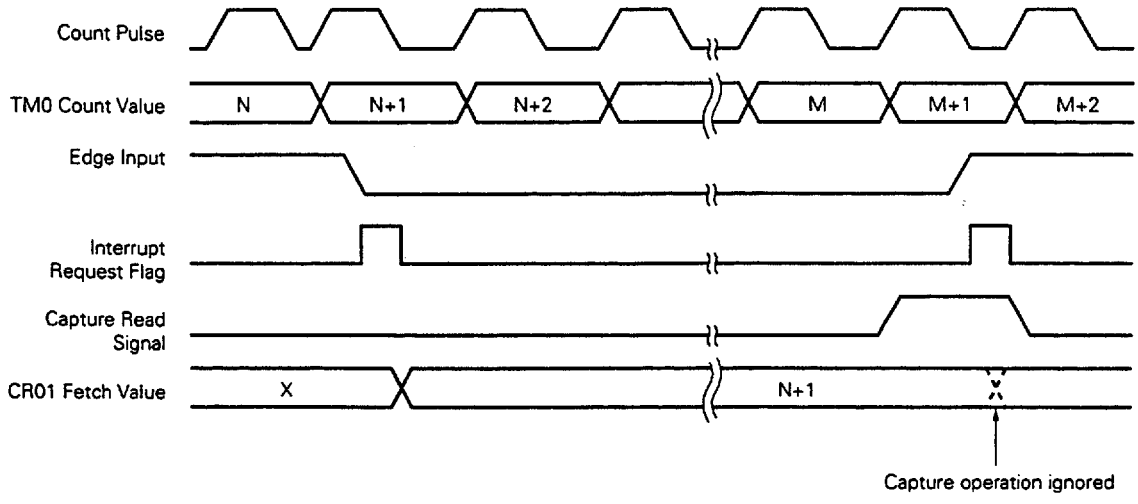


Remark N > X > M

(4) Capture register data hold timings

If the valid edge of the TI0/P00 pin is input during 16-bit capture register (CR01) read, CR01 holds data without carrying out capture operation. However, the interrupt request flag (PIF0) is set upon detection of the valid edge.

Fig. 6-22 Capture Register Data Hold Timings



(5) Valid edge set

Set the valid edge of the TI0/INTP0 pin after setting bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register to 0, 0 and 0, respectively, and then stopping timer operation. Valid edge setting is carried out with bits 2 and 3 (ES10 and ES11) of the external interrupt mode register.

CHAPTER 7 8-BIT TIMER/EVENT COUNTER

7.1 8-Bit Timer/Event Counter Functions

The 8-bit timers/event counters provided in the μPD78024 subseries have the following two modes:

- 8-bit timer/event counter mode:
  In this mode, the 8-bit timers/event counters for two channels are separately used.
- 16-bit timer/event counter mode:
  The two channels of the 8-bit timers/event counters are used in combination as a 16-bit timer/event counter.

7.1.1 8-bit timer/event counter mode

The 8-bit timer/event counters 1 and 2 (TM1 and TM2) have the following functions.

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

(1) 8-bit interval timer

When TM1 or TM2 is used as an interval timer, it generates an interrupt at preset time intervals.

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

Minimum Interval Time		Maximum Interval Time		Resolution	
$2 \times 1/fx$	(400 ns)	$2^9 \times 1/fx$	(102.4 μs)	$2 \times 1/fx$	(400 ns)
$2^2 \times 1/fx$	(800 ns)	$2^{10} \times 1/fx$	(204.8 μs)	$2^2 \times 1/fx$	(800 ns)
$2^3 \times 1/fx$	(1.6 μs)	$2^{11} \times 1/fx$	(409.6 μs)	$2^3 \times 1/fx$	(1.6 μs)
$2^4 \times 1/fx$	(3.2 μs)	$2^{12} \times 1/fx$	(819.2 μs)	$2^4 \times 1/fx$	(3.2 μs)
$2^5 \times 1/fx$	(6.4 μs)	$2^{13} \times 1/fx$	(1.64 ms)	$2^5 \times 1/fx$	(6.4 μs)
$2^6 \times 1/fx$	(12.8 μs)	$2^{14} \times 1/fx$	(3.28 ms)	$2^6 \times 1/fx$	(12.8 μs)
$2^7 \times 1/fx$	(25.6 μs)	$2^{15} \times 1/fx$	(6.55 ms)	$2^7 \times 1/fx$	(25.6 μs)
$2^8 \times 1/fx$	(51.2 μs)	$2^{16} \times 1/fx$	(13.1 ms)	$2^8 \times 1/fx$	(51.2 μs)
$2^9 \times 1/fx$	(102.4 μs)	$2^{17} \times 1/fx$	(26.2 ms)	$2^9 \times 1/fx$	(102.4 μs)
$2^{10} \times 1/fx$	(204.8 μs)	$2^{18} \times 1/fx$	(52.4 ms)	$2^{10} \times 1/fx$	(204.8 μs)
$2^{12} \times 1/fx$	(819.2 μs)	$2^{20} \times 1/fx$	(209.7 ms)	$2^{12} \times 1/fx$	(819.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz.

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 7-2 8-Bit Timer/Event Counter Square-Wave Output Ranges

Minimum Interval Time		Maximum Interval Time		Resolution	
$2 \times 1/fx$	(400 ns)	$2^9 \times 1/fx$	(102.4 μs)	$2 \times 1/fx$	(400 ns)
$2^2 \times 1/fx$	(800 ns)	$2^{10} \times 1/fx$	(204.8 μs)	$2^2 \times 1/fx$	(800 ns)
$2^3 \times 1/fx$	(1.6 μs)	$2^{11} \times 1/fx$	(409.6 μs)	$2^3 \times 1/fx$	(1.6 μs)
$2^4 \times 1/fx$	(3.2 μs)	$2^{12} \times 1/fx$	(819.2 μs)	$2^4 \times 1/fx$	(3.2 μs)
$2^5 \times 1/fx$	(6.4 μs)	$2^{13} \times 1/fx$	(1.64 ms)	$2^5 \times 1/fx$	(6.4 μs)
$2^6 \times 1/fx$	(12.8 μs)	$2^{14} \times 1/fx$	(3.28 ms)	$2^6 \times 1/fx$	(12.8 μs)
$2^7 \times 1/fx$	(25.6 μs)	$2^{15} \times 1/fx$	(6.55 ms)	$2^7 \times 1/fx$	(25.6 μs)
$2^8 \times 1/fx$	(51.2 μs)	$2^{16} \times 1/fx$	(13.1 ms)	$2^8 \times 1/fx$	(51.2 μs)
$2^9 \times 1/fx$	(102.4 μs)	$2^{17} \times 1/fx$	(26.2 ms)	$2^9 \times 1/fx$	(102.4 μs)
$2^{10} \times 1/fx$	(204.8 μs)	$2^{18} \times 1/fx$	(52.4 ms)	$2^{10} \times 1/fx$	(204.8 μs)
$2^{12} \times 1/fx$	(819.2 μs)	$2^{20} \times 1/fx$	(209.7 ms)	$2^{12} \times 1/fx$	(819.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz.

7.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

Interrupts can be generated at the preset time intervals.

Table 7-3 Interval Times when 8-Bit Timer/Event Counter is Used as a 16-Bit Timer/Event Counter

Minimum Interval Time		Maximum Interval Time		Resolution	
★ $2 \times 1/fx$	(400 ns)	$2^{17} \times 1/fx$	(26.2 ms)	$2 \times 1/fx$	(400 ns)
$2^2 \times 1/fx$	(800 ns)	$2^{18} \times 1/fx$	(52.4 ms)	$2^2 \times 1/fx$	(800 ns)
$2^3 \times 1/fx$	(1.6 μs)	$2^{19} \times 1/fx$	(104.9 ms)	$2^3 \times 1/fx$	(1.6 μs)
$2^4 \times 1/fx$	(3.2 μs)	$2^{20} \times 1/fx$	(209.7 ms)	$2^4 \times 1/fx$	(3.2 μs)
$2^5 \times 1/fx$	(6.4 μs)	$2^{21} \times 1/fx$	(419.4 ms)	$2^5 \times 1/fx$	(6.4 μs)
$2^6 \times 1/fx$	(12.8 μs)	$2^{22} \times 1/fx$	(838.9 ms)	$2^6 \times 1/fx$	(12.8 μs)
$2^7 \times 1/fx$	(25.6 μs)	$2^{23} \times 1/fx$	(1.7 s)	$2^7 \times 1/fx$	(25.6 μs)
$2^8 \times 1/fx$	(51.2 μs)	$2^{24} \times 1/fx$	(3.4 s)	$2^8 \times 1/fx$	(51.2 μs)
$2^9 \times 1/fx$	(102.4 μs)	$2^{25} \times 1/fx$	(6.7 s)	$2^9 \times 1/fx$	(102.4 μs)
$2^{10} \times 1/fx$	(204.8 μs)	$2^{26} \times 1/fx$	(13.4 s)	$2^{10} \times 1/fx$	(204.8 μs)
$2^{12} \times 1/fx$	(819.2 μs)	$2^{28} \times 1/fx$	(53.7 s)	$2^{12} \times 1/fx$	(819.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

★ **2.** Values in parentheses when operated at fx = 5.0 MHz.

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 7-4 Square-Wave Output Ranges When 8-Bit Timer/Event Counter is Used as 16-Bit Timer/Event Counter

Minimum Interval Time		Maximum Interval Time		Resolution	
$2 \times 1/fx$	(400 ns)	$2^{17} \times 1/fx$	(26.2 ms)	$2 \times 1/fx$	(400 ns)
$2^2 \times 1/fx$	(800 ns)	$2^{18} \times 1/fx$	(52.4 ms)	$2^2 \times 1/fx$	(800 ns)
$2^3 \times 1/fx$	(1.6 μs)	$2^{19} \times 1/fx$	(104.9 ms)	$2^3 \times 1/fx$	(1.6 μs)
$2^4 \times 1/fx$	(3.2 μs)	$2^{20} \times 1/fx$	(209.7 ms)	$2^4 \times 1/fx$	(3.2 μs)
$2^5 \times 1/fx$	(6.4 μs)	$2^{21} \times 1/fx$	(419.4 ms)	$2^5 \times 1/fx$	(6.4 μs)
$2^6 \times 1/fx$	(12.8 μs)	$2^{22} \times 1/fx$	(838.9 ms)	$2^6 \times 1/fx$	(12.8 μs)
$2^7 \times 1/fx$	(25.6 μs)	$2^{23} \times 1/fx$	(1.7 s)	$2^7 \times 1/fx$	(25.6 μs)
$2^8 \times 1/fx$	(51.2 μs)	$2^{24} \times 1/fx$	(3.4 s)	$2^8 \times 1/fx$	(51.2 μs)
$2^9 \times 1/fx$	(102.4 μs)	$2^{25} \times 1/fx$	(6.7 s)	$2^9 \times 1/fx$	(102.4 μs)
$2^{10} \times 1/fx$	(204.8 μs)	$2^{26} \times 1/fx$	(13.4 s)	$2^{10} \times 1/fx$	(204.8 μs)
$2^{12} \times 1/fx$	(819.2 μs)	$2^{28} \times 1/fx$	(53.7 s)	$2^{12} \times 1/fx$	(819.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz.

7.2 8-Bit Timer/Event Counter Configuration

The 8-bit timer/event counter consists of the following hardware.

Table 7-5 8-Bit Timer/Event Counter Configuration

Item	Configuration
Timer register	8 bits x 2 (TM1, TM2)
Register	8-bit compare register: 2 (CR10, CR20)
Timer output	2 (TO1, TO2)
Control registers	Timer clock select register 1 (TCL1) 8-bit timer mode control register (TMC1) 8-bit timer output control register (TOC1) Port mode register 3 (PM3)

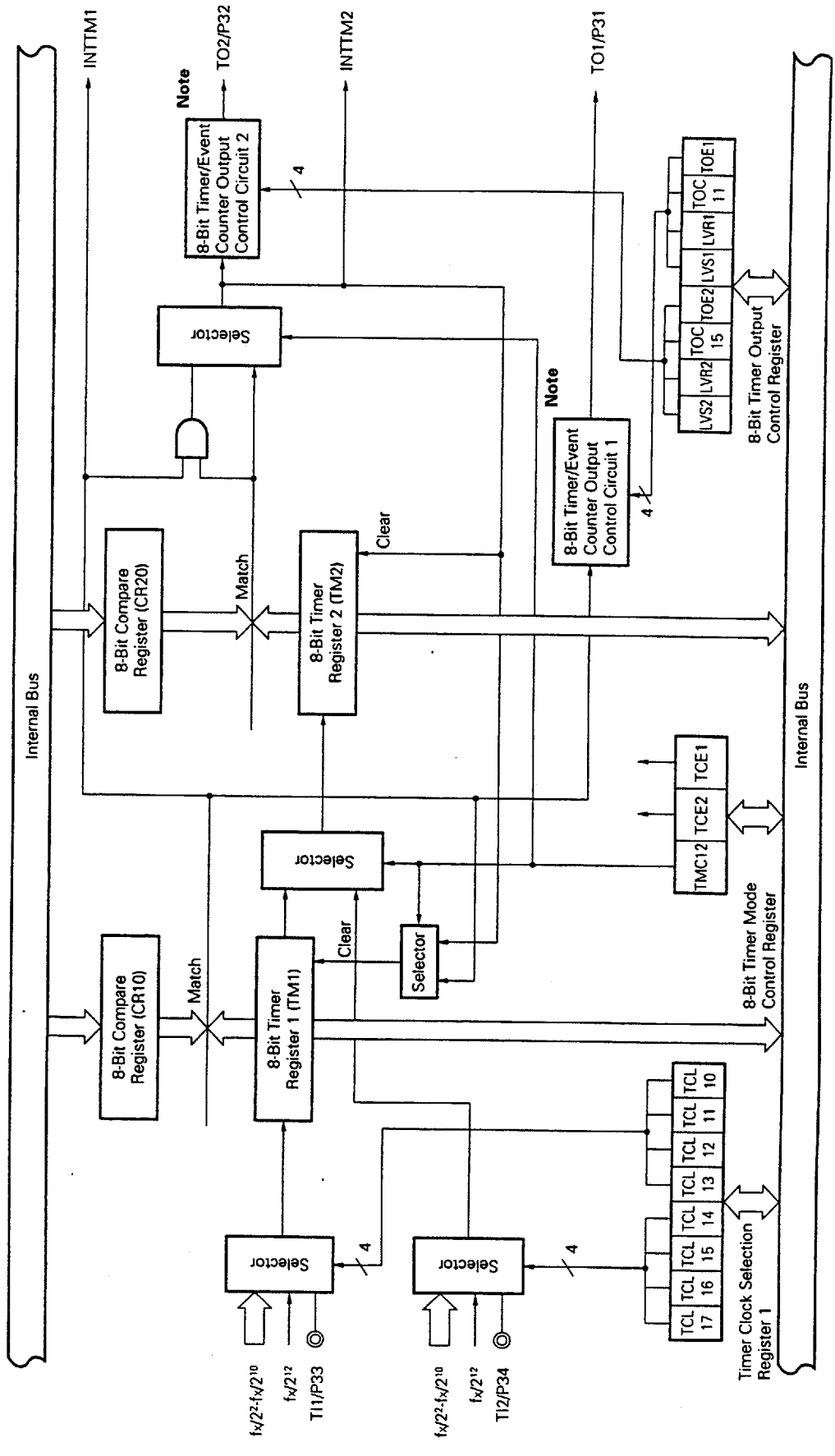
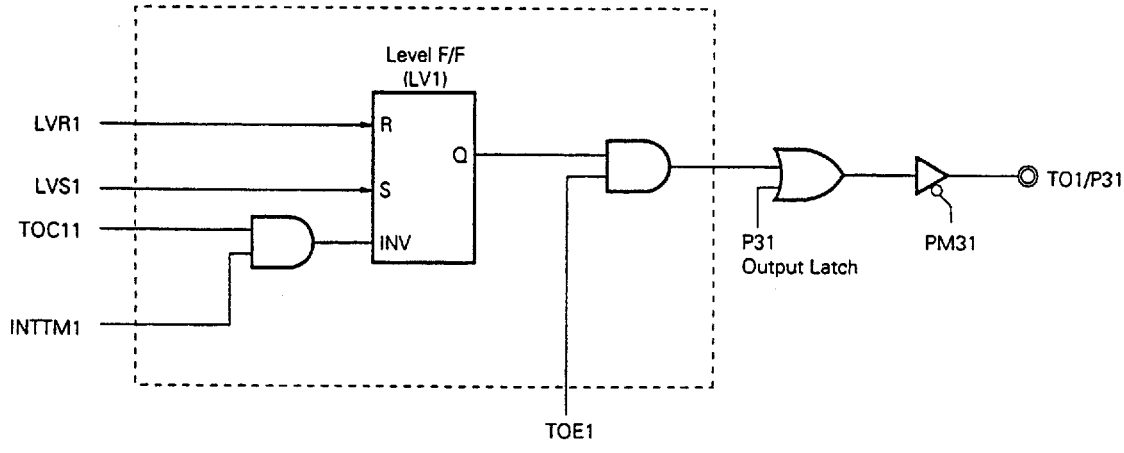


Fig. 7-1 8-Bit Timer/Event Counter Block Diagram

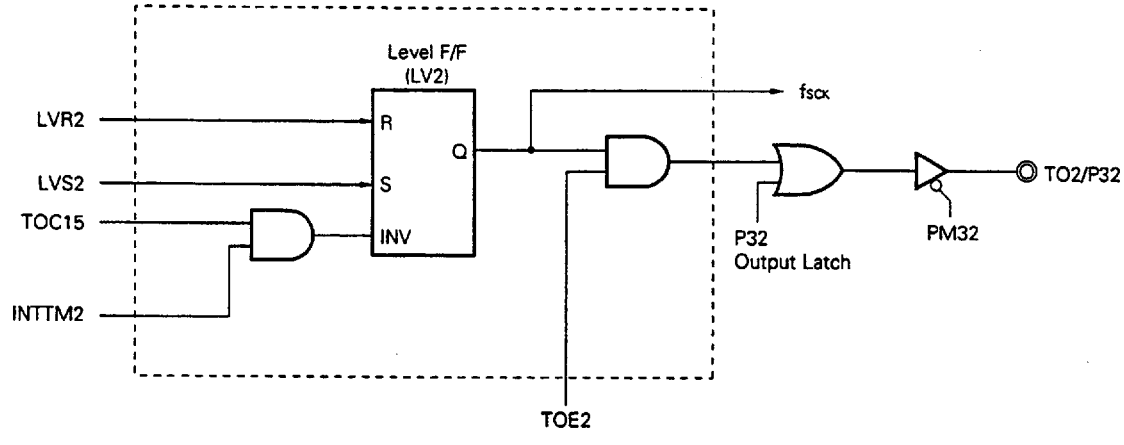
Note With the configuration of 8-bit timer/event counter output control circuits 1 and 2, refer to Figures 7-2 and 7-3.

Fig. 7-2 Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 1



Remark The section in the broken line is an output control circuit.

Fig. 7-3 Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 2



Remarks 1. The section in the broken line is an output control circuit.

2. fscx: Serial clock frequency

(1) 8-bit compare registers (CR10, CR20)

This is an 8-bit register to compare the value set to CR10 to the 8-bit timer register 1 (TM1) count value, and the value set to CR20 to the 8-bit timer register 2 (TM2) count value, and, if they match, generates an interrupt request (INTTM1 and INTTM2, respectively).

CR10 and CR20 are set with an 8-bit memory manipulate instruction. They cannot be set with a 16-bit memory manipulate instruction. When the compare register is used as 8-bit timer/event counter, the 00H to FFH values can be set. When the compare register is used as 16-bit timer/event counter, the 0000H to FFFFH values can be set.

$\overline{\text{RESET}}$ input makes CR10 and CR20 undefined.

Caution When using the compare register as 16-bit timer/event counter, be sure to set data after stopping timer operation.

(2) 8-bit timer registers 1 and 2 (TM1, TM2)

These are 8-bit registers to count count pulses.

When TM1 and TM2 are used in the 8-bit timer × 2-channel mode, they are read with an 8-bit memory manipulate instruction. When TM1 and TM2 are used as 16-bit timer × 1-channel mode, they are read with a 16-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TM1 and TM2 to 00H.

7.3 8-Bit Timer/Event Counter Control Registers

The following four types of registers are used to control the 8-bit timer/event counter.

- Timer clock select register 1 (TCL1)
- 8-bit timer mode control register (TMC1)
- 8-bit timer output control register (TOC1)
- Port mode register 3 (PM3)

(1) Timer clock select register 1 (TCL1)

This register sets count clocks of 8-bit timer registers 1 and 2.

TCL1 is set with an 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TCL1 to 00H.

Fig. 7-4 Timer Clock Select Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL1	TCL17	TCL16	TCL15	TCL14	TCL13	TCL12	TCL11	TCL10	FF41H	00H	R/W

TCL13	TCL12	TCL11	TCL10	8-Bit Timer Register 1 Count Clock Selection
0	0	0	0	TI1 falling edge
0	0	0	1	TI1 rising edge
0	1	0	1	fx/2 (2.5 MHz)
0	1	1	0	$fx/2^2$ (1.25 MHz)
0	1	1	1	$fx/2^3$ (625 kHz)
1	0	0	0	$fx/2^4$ (313 kHz)
1	0	0	1	$fx/2^5$ (156 kHz)
1	0	1	0	$fx/2^6$ (78.1 kHz)
1	0	1	1	$fx/2^7$ (39.1 kHz)
1	1	0	0	$fx/2^8$ (19.5 kHz)
1	1	0	1	$fx/2^9$ (9.8 kHz)
1	1	1	0	$fx/2^{10}$ (4.9 kHz)
1	1	1	1	$fx/2^{12}$ (1.2 kHz)
Other than the above				Setting prohibited

TCL17	TCL16	TCL15	TCL14	8-Bit Timer Register 2 Count Clock Selection
0	0	0	0	TI2 falling edge
0	0	0	1	TI2 rising edge
0	1	0	1	fx/2 (2.5 MHz)
0	1	1	0	$fx/2^2$ (1.25 MHz)
0	1	1	1	$fx/2^3$ (625 kHz)
1	0	0	0	$fx/2^4$ (313 kHz)
1	0	0	1	$fx/2^5$ (156 kHz)
1	0	1	0	$fx/2^6$ (78.1 kHz)
1	0	1	1	$fx/2^7$ (39.1 kHz)
1	1	0	0	$fx/2^8$ (19.5 kHz)
1	1	0	1	$fx/2^9$ (9.8 kHz)
1	1	1	0	$fx/2^{10}$(4.9 kHz)
1	1	1	1	$fx/2^{12}$(1.2 kHz)
Other than the above				Setting prohibited

★ **Caution Before rewriting the data of TCL1, stop the timer once.**

Remarks 1. fx : Main system clock oscillation frequency

2. TI1 : 8-bit timer register 1 input pin

3. TI2 : 8-bit timer register 2 input pin

★ **4.** Values in parentheses operated at fx = 5.0 MHz.

(2) 8-bit timer mode control register (TMC1)

This register enables/halts operation of 8-bit timer registers 1 and 2 and sets the operating mode of 8-bit timer register 2.

TMC1 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TMC1 to 00H.

Fig. 7-5 8-Bit Timer Mode Control Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
TMC1	0	0	0	0	0	TMC12	TCE2	TCE1	FF49H	00H	R/W

TCE1	8-Bit Timer Register 1 Operation Control
0	Operation Stop (TM1 is cleared to 0)
1	Operation enabled

TCE2	8-Bit Timer Register 2 Operation Control
0	Operation Stop (TM2 is cleared to 0)
1	Operation enabled

TMC12	Operating Mode Selection
0	8-bit timer register x 2-channel mode (TM1, TM2)
1	16-bit timer register x 1-channel mode (TMS)

Cautions 1. Switch the operation mode after stopping timer operation.

2. Operation enable/halt should be performed by TCE1 when used as 16-bit timer register.

(3) 8-bit timer output control register (TOC1)

This register controls operation of 8-bit timer/event counter output control circuits 1 and 2.

Its sets/resets the R-S flip-flops (LV1 and LV2) and enables/disables inversion and 8-bit timer output of 8-bit timer registers 1 and 2.

TOC1 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TOC1 to 00H.

Fig. 7-6 8-Bit Timer Output Control Register Format

Symbol	⑦	⑥	5	④	③	②	1	⓪	Address	After Reset	R/W
TOC1	LVS2	LVR2	TOC15	TOE2	LVS1	LVR1	TOC11	TOE1	FF4FH	00H	R/W

TOE1	8-Bit Timer/Event Counter 1 Output Control
0	Output disabled (Port mode)
1	Output enabled

TOC11	8-Bit Timer/Event Counter 1 Timer Output F/F Control
0	Inversion operation disabled
1	Inversion operation enabled

LVS1	LVR1	8-Bit Timer/Event Counter 1 Timer Output F/F Status Set
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOE2	8-Bit Timer/Event Counter 2 Output Control
0	Output disabled (Port mode)
1	Output enabled

TOC15	8-Bit Timer/Event Counter 2 Timer Output F/F Control
0	Inversion operation disabled
1	Inversion operation enabled

LVS2	LVR2	8-Bit Timer/Event Counter 2 Timer Output F/F Status Set
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

Cautions 1. Be sure to set TOC1 after stopping timer operation.

2. After data setting, 0 can be read from LVS1, LVS2, LVR1 and LVR2.

(4) Port mode register 3 (PM3)

This register sets port 3 in the input or output mode in 1-bit units.

When using the P31/TO1 and P32/TO2 pins for timer output, set PM31, PM32 output latches to 0.

PM3 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Fig. 7-7 Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM3	1	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 6)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Caution Set 1 in bit 7.

7.4 8-Bit Timer/Event Counter Operations

7.4.1 8-bit timer/event counter mode

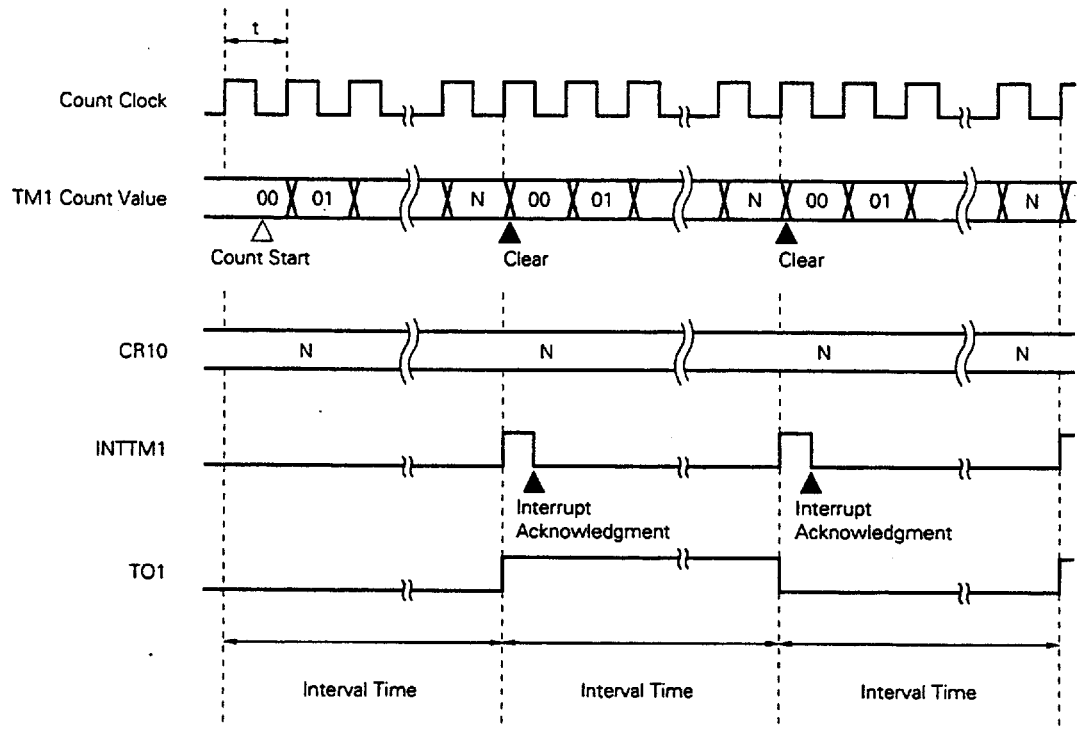
(1) Interval timer operations

The 8-bit timer/event counter operates as interval timer which generates interrupts repeatedly at intervals of the count value preset to 8-bit compare registers (CR10 and CR20).

When the count values of the 8-bit timer registers 1 and 2 (TM1 and TM2) match the values set to CR10 and CR20, counting continues with the TM1 and TM2 values cleared to 0 and the interrupt request signals (INTTM1 and INTTM2) are generated.

Count clock of the 8-bit timer register 1 (TM1) can be selected with bits 0 to 3 (TCL10 to TCL13) of the timer clock select register 1 (TCL1). Count clock of the 8-bit timer register 2 (TM2) can be selected with bits 4 to 7 (TCL14 to TCL17) of the 8-bit timer register 2 (TM2).

Fig. 7-8 Interval Timer Operation Timings



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

Table 7-6 8-Bit Timer/Event Counters 1 Interval Times

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI1 input cycle	2^8 x TI1 input cycle	TI1 input edge cycle
0	0	0	1	TI1 input cycle	2^8 x TI1 input cycle	TI1 input edge cycle
0	1	0	1	2 x 1/fx (400 ns)	2^9 x 1/fx (102.4 μs)	2 x 1/fx (400 ns)
0	1	1	0	2^2 x 1/fx (800 ns)	2^{10} x 1/fx (204.8 μs)	2^2 x 1/fx (800 ns)
0	1	1	1	2^3 x 1/fx (1.6 μs)	2^{11} x 1/fx (409.6 μs)	2^3 x 1/fx (1.6 μs)
1	0	0	0	2^4 x 1/fx (3.2 μs)	2^{12} x 1/fx (819.2 μs)	2^4 x 1/fx (3.2 μs)
1	0	0	1	2^5 x 1/fx (6.4 μs)	2^{13} x 1/fx (1.64 ms)	2^5 x 1/fx (6.4 μs)
1	0	1	0	2^6 x 1/fx (12.8 μs)	2^{14} x 1/fx (3.28 ms)	2^6 x 1/fx (12.8 μs)
1	0	1	1	2^7 x 1/fx (25.6 μs)	2^{15} x 1/fx (6.55 ms)	2^7 x 1/fx (25.6 μs)
1	1	0	0	2^8 x 1/fx (51.2 μs)	2^{16} x 1/fx (13.1 ms)	2^8 x 1/fx (51.2 μs)
1	1	0	1	2^9 x 1/fx (102.4 μs)	2^{17} x 1/fx (26.2 ms)	2^9 x 1/fx (102.4 μs)
1	1	1	0	2^{10} x 1/fx (204.8 μs)	2^{18} x 1/fx (52.4 ms)	2^{10} x 1/fx (204.8 μs)
1	1	1	1	2^{12} x 1/fx (819.2 μs)	2^{20} x 1/fx (209.7 ms)	2^{12} x 1/fx (819.2 μs)
Other than the above				Setting prohibited		

Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz

Table 7-7 8-Bit Timer/Event Counters 2 Interval Times

TCL17	TCL16	TCL15	TCL14	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI2 input cycle	2^8 x TI2 input cycle	TI2 input edge cycle
0	0	0	1	TI2 input cycle	2^8 x TI2 input cycle	TI2 input edge cycle
★ 0	1	0	1	2 x 1/fx (400 ns)	2^9 x 1/fx (102.4 μs)	2 x 1/fx (400 ns)
0	1	1	0	2^2 x 1/fx (800 ns)	2^{10} x 1/fx (204.8 μs)	2^2 x 1/fx (800 ns)
0	1	1	1	2^3 x 1/fx (1.6 μs)	2^{11} x 1/fx (409.6 μs)	2^3 x 1/fx (1.6 μs)
1	0	0	0	2^4 x 1/fx (3.2 μs)	2^{12} x 1/fx (819.2 μs)	2^4 x 1/fx (3.2 μs)
1	0	0	1	2^5 x 1/fx (6.4 μs)	2^{13} x 1/fx (1.64 ms)	2^5 x 1/fx (6.4 μs)
1	0	1	0	2^6 x 1/fx (12.8 μs)	2^{14} x 1/fx (3.28 ms)	2^6 x 1/fx (12.8 μs)
1	0	1	1	2^7 x 1/fx (25.6 μs)	2^{15} x 1/fx (6.55 ms)	2^7 x 1/fx (25.6 μs)
1	1	0	0	2^8 x 1/fx (51.2 μs)	2^{16} x 1/fx (13.1 ms)	2^8 x 1/fx (51.2 μs)
1	1	0	1	2^9 x 1/fx (102.4 μs)	2^{17} x 1/fx (26.2 ms)	2^9 x 1/fx (102.4 μs)
1	1	1	0	2^{10} x 1/fx (204.8 μs)	2^{18} x 1/fx (52.4 ms)	2^{10} x 1/fx (204.8 μs)
1	1	1	1	2^{12} x 1/fx (819.2 μs)	2^{20} x 1/fx (209.7 ms)	2^{12} x 1/fx (819.2 μs)
Other than the above				Setting prohibited		

Remarks 1. fx: Main system clock oscillation frequency

★ **2.** Values in parentheses when operated at fx = 5.0 MHz

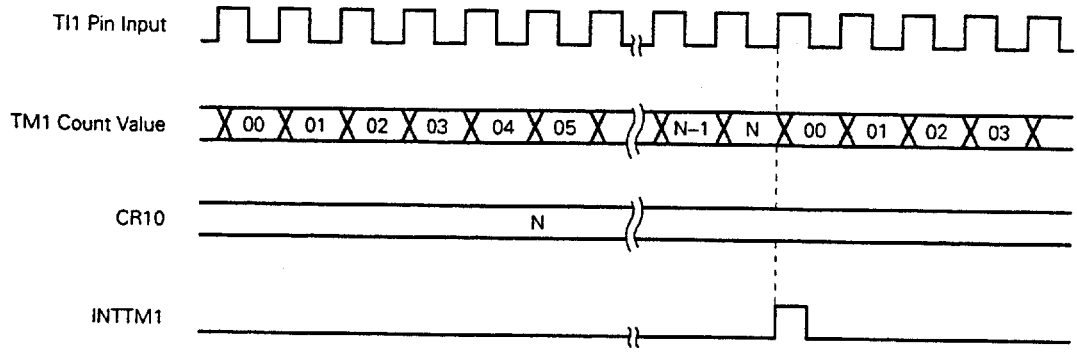
(2) External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI1/P33 and TI2/P34 pins with 8-bit timer registers 1 and 2 (TM1 and TM2).

TM1 and TM2 are incremented each time the valid edge specified with the timer clock select register (TCL1) is input. Either the rising or falling edge can be selected.

When the TM1 and TM2 counted values match the values of 8-bit compare registers (CR10 and CR20), TM1 and TM2 are cleared to 0 and the interrupt request signals (INTTM1 and INTTM2) are generated.

Fig. 7-9 External Event Counter Operation Timings (with Rising Edge Specified)



Remark N = 00H to FFH

(3) Square-wave output

A square wave with any selected frequency is output at intervals of the value preset to 8-bit compare registers (CR10 and CR20).

The TO1/P31 or TO2/P32 pin output status is reserved at intervals of the count value preset to CR10 or CR20 by setting bit 0 (TOE1) or bit 4 (TOE2) of the 8-bit timer output control register (TOC1) to 1. This enables a square with any selected frequency to be output.

Table 7-8 8-Bit Timer/Event Counter Square-Wave Output Ranges

TCL13	TCL12	TCL11	TCL10	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	1	0	1	$2 \times 1/fx$ (400 ns)	$2^9 \times 1/fx$ (102.4 μs)	$2 \times 1/fx$ (400 ns)
0	1	1	0	$2^2 \times 1/fx$ (800 ns)	$2^{10} \times 1/fx$ (204.8 μs)	$2^2 \times 1/fx$ (800 ns)
0	1	1	1	$2^3 \times 1/fx$ (1.6 μs)	$2^{11} \times 1/fx$ (409.6 μs)	$2^3 \times 1/fx$ (1.6 μs)
1	0	0	0	$2^4 \times 1/fx$ (3.2 μs)	$2^{12} \times 1/fx$ (819.2 μs)	$2^4 \times 1/fx$ (3.2 μs)
1	0	0	1	$2^5 \times 1/fx$ (6.4 μs)	$2^{13} \times 1/fx$ (1.64 ms)	$2^5 \times 1/fx$ (6.4 μs)
1	0	1	0	$2^6 \times 1/fx$ (12.8 μs)	$2^{14} \times 1/fx$ (3.28 ms)	$2^6 \times 1/fx$ (12.8 μs)
1	0	1	1	$2^7 \times 1/fx$ (25.6 μs)	$2^{15} \times 1/fx$ (6.55 ms)	$2^7 \times 1/fx$ (25.6 μs)
1	1	0	0	$2^8 \times 1/fx$ (51.2 μs)	$2^{16} \times 1/fx$ (13.1 ms)	$2^8 \times 1/fx$ (51.2 μs)
1	1	0	1	$2^9 \times 1/fx$ (102.4 μs)	$2^{17} \times 1/fx$ (26.2 ms)	$2^9 \times 1/fx$ (102.4 μs)
1	1	1	0	$2^{10} \times 1/fx$ (204.8 μs)	$2^{18} \times 1/fx$ (52.4 ms)	$2^{10} \times 1/fx$ (204.8 μs)
1	1	1	1	$2^{12} \times 1/fx$ (819.2 μs)	$2^{20} \times 1/fx$ (209.7 ms)	$2^{12} \times 1/fx$ (819.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz

7.4.2 16-bit timer/event counter mode

When bit 2 (TMC12) of 8-bit timer mode control register (TMC1) is set to 1 and the 16-bit timer/counter mode is selected, the overflow signal of 8-bit timer/event counter 1 (TM1) becomes a count clock of 8-bit timer/event counter 2 (TM2).

When a 2-channel 8-bit timer/event counter is used in the 16-bit timer/event counter mode, the count clock is selected with bits 0 to 3 (TCL10 to TCL13) of TCL1. Count operation enable/disable is selected with bits 1 and 2 (TCE1 and TCE2) of TMC1.

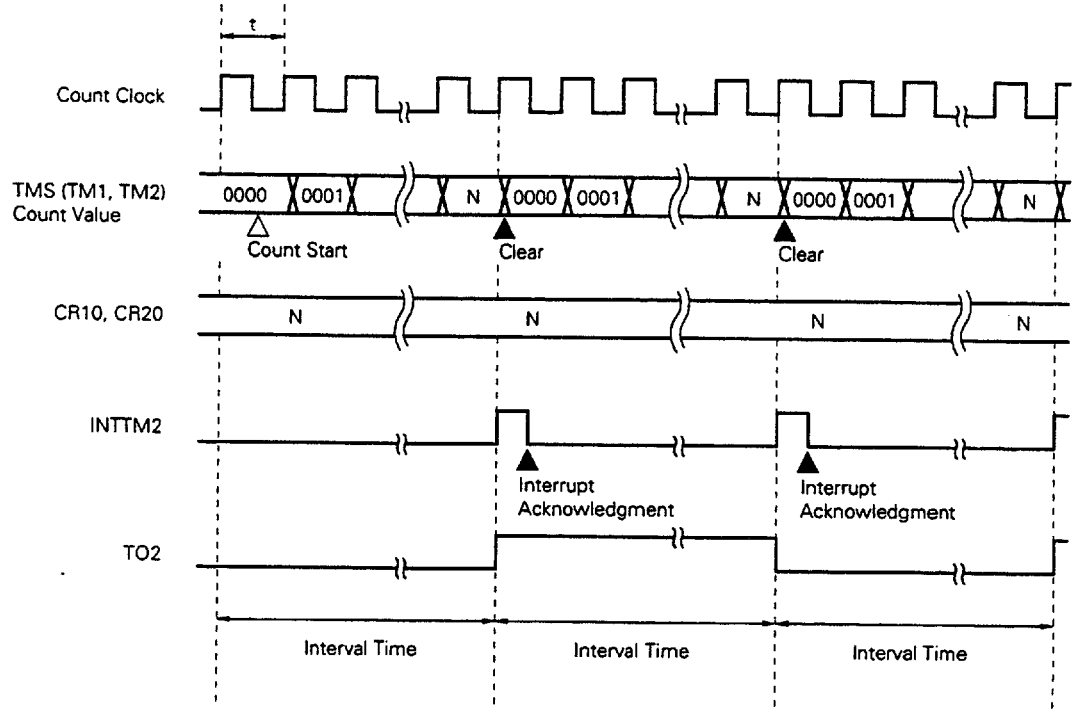
(1) Interval timer

The 8-bit timer/event counter operates as interval timer which generates interrupts repeatedly at intervals of the count value preset to 2-channel 8-bit compare registers (CR10 and CR20).

When the 8-bit timer register (TM1) and CR10 values match and the 8-bit timer register 2 (TM2) and CR20 values match, counting continues with the TM1 and TM2 values cleared to 0 and the interrupt request signal (INTTM2) is generated.

Count clock can be selected with bits 0 to 3 (TCL10 to TCL13) of the timer clock select register 1 (TCL1).

Fig. 7-10 Interval Timer Operation Timings



Remark Interval time = (N + 1) × t: N = 0000H to FFFFH

Caution Even if the 16-bit timer/event counter mode is used, when the TM1 count value matches the CR10 value, interrupt request (INTTM1) is generated and the F/F of 8-bit timer/event counter output control circuit 1 is reversed. Thus, when using 8-bit timer/event counter as 16-bit internal timer, set the INTTM1 acknowledge inhibit mask flag TMMK1 to 1 to disable INTTM1 acknowledgment. In the case of reading the timer count value, the 16-bit timer (TMS) should be read by the 16-bit memory manipulation instruction.

Table 7-9 Interval Times when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI1 input cycle	2^8 × TI1 input cycle	TI1 input edge cycle
0	0	0	1	TI1 input cycle	2^8 × TI1 input cycle	TI1 input edge cycle
★ 0	1	0	1	2 × 1/fx (400 ns)	2^{17} × 1/fx (26.2 ms)	2 × 1/fx (400 ns)
0	1	1	0	2^2 × 1/fx (800 ns)	2^{18} × 1/fx (52.4 ms)	2^2 × 1/fx (800 ns)
0	1	1	1	2^3 × 1/fx (1.6 μs)	2^{19} × 1/fx (104.9 ms)	2^3 × 1/fx (1.6 μs)
1	0	0	0	2^4 × 1/fx (3.2 μs)	2^{20} × 1/fx (209.7 ms)	2^4 × 1/fx (3.2 μs)
1	0	0	1	2^5 × 1/fx (6.4 μs)	2^{21} × 1/fx (419.4 ms)	2^5 × 1/fx (6.4 μs)
1	0	1	0	2^6 × 1/fx (12.8 μs)	2^{22} × 1/fx (838.9 ms)	2^6 × 1/fx (12.8 μs)
1	0	1	1	2^7 × 1/fx (25.6 μs)	2^{23} × 1/fx (1.7 s)	2^7 × 1/fx (25.6 μs)
1	1	0	0	2^8 × 1/fx (51.2 μs)	2^{24} × 1/fx (3.4 s)	2^8 × 1/fx (51.2 μs)
1	1	0	1	2^9 × 1/fx (102.4 μs)	2^{25} × 1/fx (6.7 s)	2^9 × 1/fx (102.4 μs)
1	1	1	0	2^{10} × 1/fx (204.8 μs)	2^{26} × 1/fx (13.4 s)	2^{10} × 1/fx (204.8 μs)
1	1	1	1	2^{12} × 1/fx (819.2 μs)	2^{28} × 1/fx (53.7 s)	2^{12} × 1/fx (819.2 μs)
Other than the above				Setting prohibited		

Remarks 1. fx: Main system clock oscillation frequency

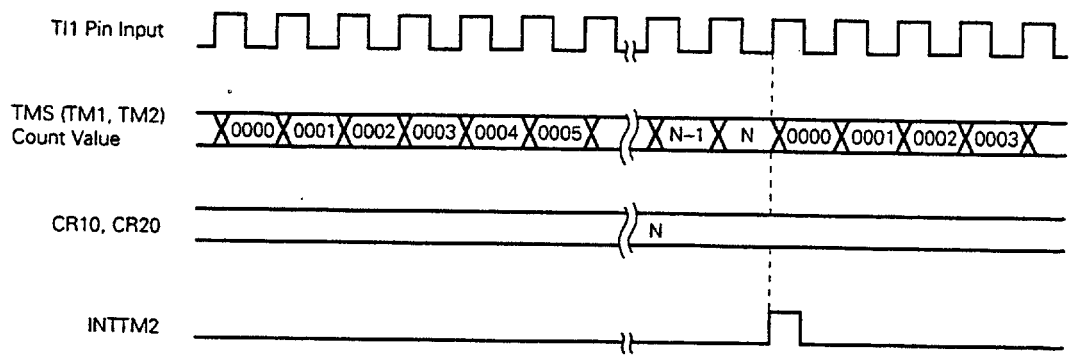
★ **2.** Values in parentheses when operated at fx = 5.0 MHz

(2) External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI1/P33 pin with 2-channel 8-bit timer registers 1 and 2 (TM1 and TM2).

TM1 and TM2 are incremented each time the valid edge specified with the timer clock select register (TCL1). Either the rising or falling edge can be selected. When the TM1 and TM2 counted values match the values of 8-bit compare registers (CR10 and CR20), TM1 and TM2 are cleared to 0 and the interrupt request signal (INTTM2) is generated.

Fig. 7-11 External Event Counter Operation Timings (with Rising Edge Specified)



Caution Even if the 16-bit timer/event counter mode is used, when the TM1 count value matches the CR10 value, interrupt request (INTTM1) is generated and the F/F of 8-bit timer/event counter output control circuit 1 is reversed. Thus, when using 8-bit timer/event counter as 16-bit interval timer, set the INTTM1 acknowledge inhibit mask flag TMMK1 to 1 to disable INTTM1 acknowledgment. When the timer count value is read, the 16-bit timer (TMS) should be read by the 16-bit memory manipulation instruction.

(3) Square-wave output operation

A square wave with any selected frequency is output at intervals of the count value preset to 8-bit compare registers (CR10 and CR20).

The TO2/P32 pin output status is reversed at intervals of the count value preset to CR10 and CR20 by setting bit4 (TOE2) of the 8-bit timer output control register (TOC1) to 1. This enables a square with any selected frequency to be output.

Table 7-10 Square Wave Output Range When 2 Channels of 8-Bit Timers/Event Counters (TM1 and TM2) are Used as a 16-Bit Timer/Event Counter

TCL13	TCL12	TCL11	TCL10	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	1	0	1	$2 \times 1/fx$ (400 ns)	$2^{17} \times 1/fx$ (26.2 ms)	$2 \times 1/fx$ (400 ns)
0	1	1	0	$2^2 \times 1/fx$ (800 ns)	$2^{18} \times 1/fx$ (52.4 ms)	$2^2 \times 1/fx$ (800 ns)
0	1	1	1	$2^3 \times 1/fx$ (1.6 μs)	$2^{19} \times 1/fx$ (104.9 ms)	$2^3 \times 1/fx$ (1.6 μs)
1	0	0	0	$2^4 \times 1/fx$ (3.2 μs)	$2^{20} \times 1/fx$ (209.7 ms)	$2^4 \times 1/fx$ (3.2 μs)
1	0	0	1	$2^5 \times 1/fx$ (6.4 μs)	$2^{21} \times 1/fx$ (419.4 ms)	$2^5 \times 1/fx$ (6.4 μs)
1	0	1	0	$2^6 \times 1/fx$ (12.8 μs)	$2^{22} \times 1/fx$ (838.9 ms)	$2^6 \times 1/fx$ (12.8 μs)
1	0	1	1	$2^7 \times 1/fx$ (25.6 μs)	$2^{23} \times 1/fx$ (1.7 s)	$2^7 \times 1/fx$ (25.6 μs)
1	1	0	0	$2^8 \times 1/fx$ (51.2 μs)	$2^{24} \times 1/fx$ (3.4 s)	$2^8 \times 1/fx$ (51.2 μs)
1	1	0	1	$2^9 \times 1/fx$ (102.4 μs)	$2^{25} \times 1/fx$ (6.7 s)	$2^9 \times 1/fx$ (102.4 μs)
1	1	1	0	$2^{10} \times 1/fx$ (204.8 μs)	$2^{26} \times 1/fx$ (13.4 s)	$2^{10} \times 1/fx$ (204.8 μs)
1	1	1	1	$2^{12} \times 1/fx$ (819.2 μs)	$2^{28} \times 1/fx$ (53.7 s)	$2^{12} \times 1/fx$ (819.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

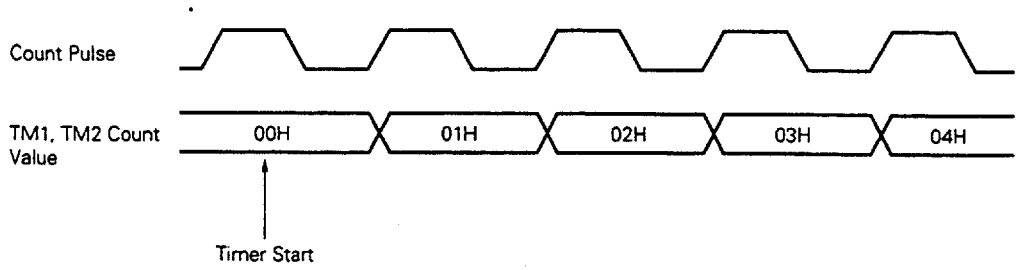
2. Values in parentheses when operated at fx = 5.0 MHz

7.5 8-Bit Timer/Event Counter 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 8-bit timer registers 1 and 2 (TM1 and TM2) are started asynchronously with the count pulse.

Fig. 7-12 8-Bit Timer Register Start Timing



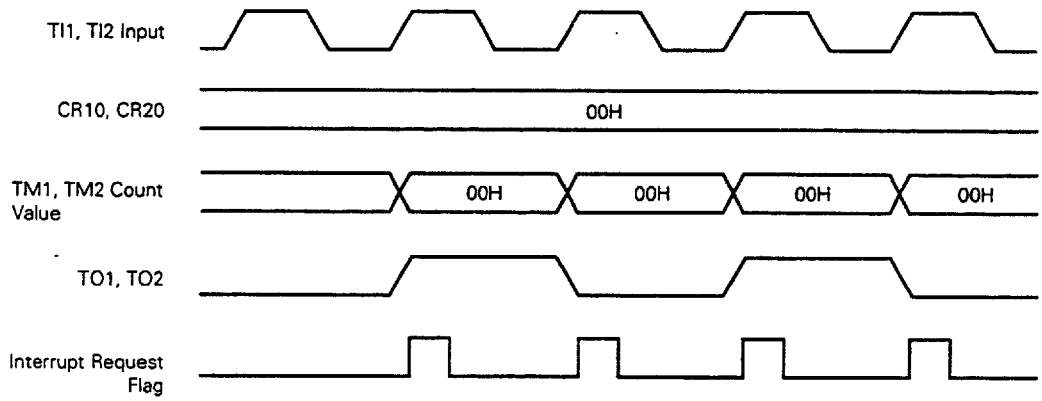
(2) 8-bit compare registers 1 and 2 setting

The 8-bit compare registers (CR10 and CR20) can be set to 00H.

Thus, when the 8-bit compare register is used as event counter, one-pulse count operation can be carried out.

When the 8-bit compare register is used as 16-bit timer/event counter, write data to CR10 and CR20 should be performed after first setting bits 0 (TCE1) of 8-bit timer mode control register to 0, and stopping timer operation.

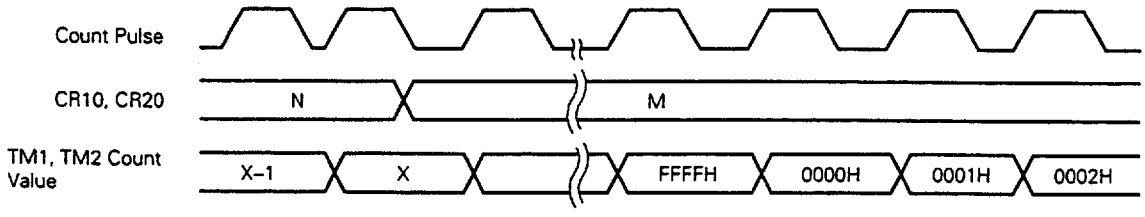
Fig. 7-13 External Event Counter Operation Timings



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

If the values after the 8-bit compare registers (CR10 and CR20) are changed are smaller than those of 8-bit timer registers (TM1 and TM2), TM1 and TM2 continue counting, overflow and then restarts counting from 0. Thus, if the value (M) after CR10 and CR20 change is smaller than that (N) before change, it is necessary to restart the timer after changing CR10 and CR20.

Fig. 7-14 Timing after Compare Register Change during Timer Count Operation



Remark N > X > M

CHAPTER 8 WATCH TIMER

8.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and interval timer can be used simultaneously.

(1) Watch timer

Flags (WTIF) are set at a timer interval of 0.5 or 0.25 sec by using a 32.768 kHz subsystem clock.
Flags (WTIF) are also at a time interval of 0.5 or 1 sec by using a 4.19 MHz (4.194304 MHz typ.) main system clock.

Caution The time interval of 0.5 second cannot be created with a main system clock of 5.0 MHz. Change the system clock to a subsystem clock of 32.768 kHz to create the time interval of 0.5 second. ★

(2) Interval timer

Interrupt requests (INTTM3) are generated at the preset time interval.

Table 8-1 Interval Timer Interval Time ★

Interval Time	When Operated at fx = 5.0 MHz	When Operated at fx = 4.19 MHz	When Operated at fxT = 32.768 kHz
$2^{12} \times 1/fx$	819 μs	978 μs	488 μs
$2^{13} \times 1/fx$	1.64 ms	1.96 ms	978 μs
$2^{14} \times 1/fx$	3.28 ms	3.91 ms	1.96 ms
$2^{15} \times 1/fx$	6.55 ms	7.82 ms	3.91 ms
$2^{16} \times 1/fx$	13.1 ms	15.6 ms	7.82 ms
$2^{17} \times 1/fx$	26.2 ms	31.3 ms	15.6 ms

Remarks 1. fx : Main system clock oscillation frequency
2. fxT : Subsystem clock oscillation frequency

8.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 8-2 Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Control register	Timer clock select register 2 (TCL2) Watch timer mode control register (TMC2)

8.3 Watch Timer Control Registers

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

- Timer clock select register 2 (TCL2)
- Watch timer mode control register (TMC2)

(1) Timer clock select register 2 (TCL2)

This register sets the watch timer count clock.

TCL2 is set with an 8-bit memory manipulate instruction.

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

Remark Besides setting the watch timer count clock, TCL2 sets the watchdog timer count clock and buzzer output frequency.

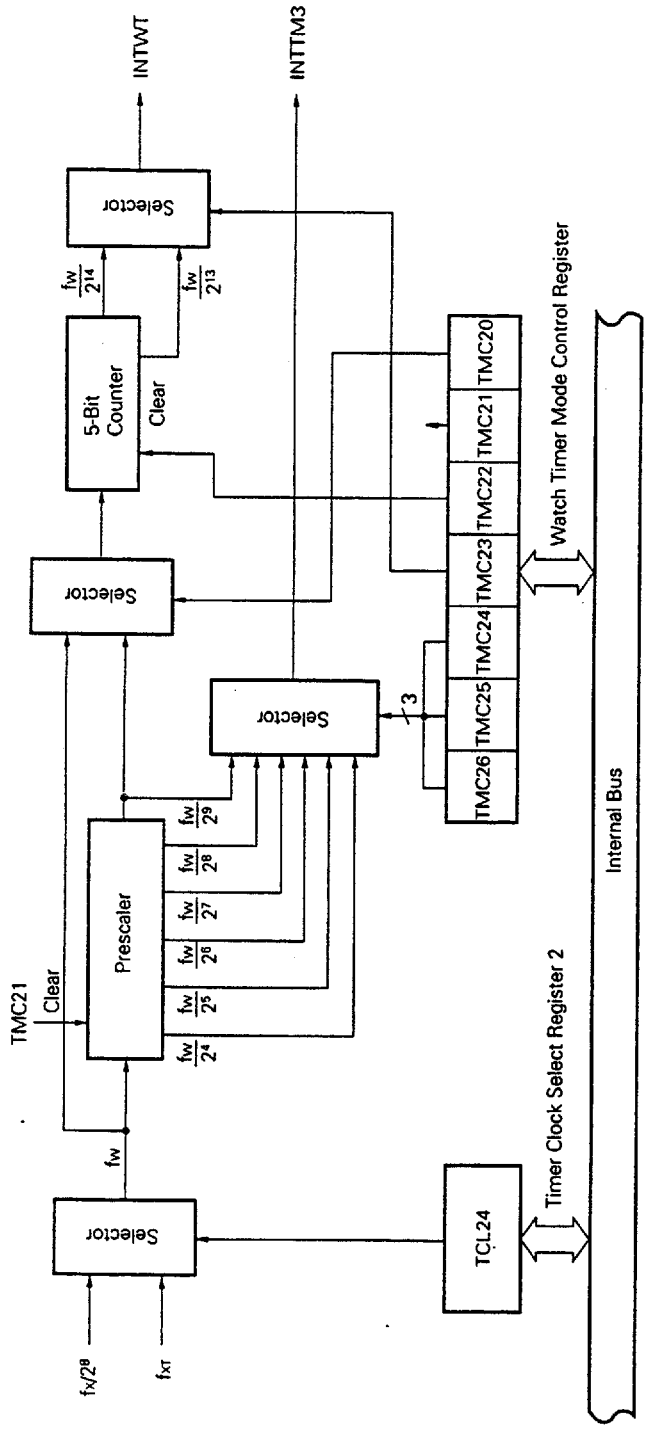


Fig. 8-1 Watch Timer Block Diagram

Fig. 8-2 Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Count Clock Selection	
			Watchdog Timer Mode	Interval Timer Mode
0	0	0	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)
0	0	1	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)
0	1	0	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)
0	1	1	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)
1	0	0	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	0	1	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)
1	1	0	$f_x/2^9$ (9.8 kHz)	$f_x/2^{10}$ (4.9 kHz)
1	1	1	$f_x/2^{11}$ (2.4 kHz)	$f_x/2^{12}$ (1.2 kHz)

TCL24	Watch Timer Count Clock Selection
0	$f_x/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disabled
1	0	0	$f_x/2^{10}$ (4.9 kHz)
1	0	1	$f_x/2^{11}$ (2.4 kHz)
1	1	0	$f_x/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

★ **Caution Before rewriting the data of TCL1, stop the timer once.**

Remarks 1. f_x : Main system clock oscillation frequency

2. f_{XT} : Subsystem clock oscillation frequency

3. × : Don't care

★ **4.** Values in parentheses when operated at f_x = 5.0 MHz and f_{XT} = 32.768 kHz.

(2) Watch timer mode control register (TMC2)

This register sets the watch timer operation mode, clock flag set time and prescaler interval time and enables/disables prescaler and 5-bit counter operations.

TMC2 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TMC2 to 00H.

Fig. 8-3 Watch Timer Mode Control Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TMC2	0	TMC26	TMC25	TMC24	TMC23	TMC22	TMC21	TMC20	FF4AH	00H	R/W

TMC23	TMC20	Watch Flag Set Time Selection
0	0	2^{14}/fw (0.5 s)
1		2^{13}/fw (0.25 s)
0	1	2^{5}/fw (977 μs)
1		2^{4}/fw (488 μs)

TMC21	Prescaler Operation Control[Note]
0	Clear after Operation stop
1	Operation enabled

TMC22	5-Bit Counter Operation Control
0	Clear after Operation stop
1	Operation enabled

TMC26	TMC25	TMC24	Prescaler Interval Time Selection
0	0	0	2^{4}/fw (489 μs)
0	0	1	2^{5}/fw (978 μs)
0	1	0	2^{6}/fw (1.96 ms)
0	1	1	2^{7}/fw (3.91 ms)
1	0	0	2^{8}/fw (7.82 ms)
1	0	1	2^{9}/fw (15.6 ms)
Other than the above			Setting prohibited

Note When using a watch timer, do not clear the prescaler frequently.

Remarks 1. fw: Watch timer clock frequency ($f_X/2^8$ or f_{XT})

2. Values in parentheses when operated at fw = 32.768 kHz

8.4 Watch Timer Operations

8.4.1 Watch timer operation

The watch timer operates at a time interval of 0.5 or 0.25 sec with a 32.768 kHz subsystem clock. It can also operate at a time interval of 0.5 or 1 sec with a 4.19 MHz main system clock.

The watch timer sets the test input flag (WTIF) to 1 at the constant time interval. The standby state (STOP mode/ HALT mode) can be cleared by setting WTIF to 1.

When bit 2 (TMC22) of the watch timer mode control register is set to 0, 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 TMC22 to 0 (maximum error: 31.3 ms when operated at fx = 4.19 MHz).

8.4.2 Interval timer operation

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

The interval time can be selected with bits 4 to 6 (TMC24 to TMC26) of the watch timer mode control register.

★ **Table 8-3 Interval Timer Interval Time**

TMC26	TMC25	TMC24	Interval Time	When Operated at fx = 5.0 MHz	When Operated at fx = 4.19 MHz	When Operated at fxT = 32.768 kHz
0	0	0	2^4 x 1/fw	819 μs	978 μs	489 μs
0	0	1	2^5 x 1/fw	1.64 ms	1.96 ms	978 μs
0	1	0	2^6 x 1/fw	3.28 ms	3.91 ms	1.96 ms
0	1	1	2^7 x 1/fw	6.55 ms	7.82 ms	3.91 ms
1	0	0	2^8 x 1/fw	13.1 ms	15.6 ms	7.82 ms
1	0	1	2^9 x 1/fw	26.2 ms	31.3 ms	15.6 ms
Othrer than the above			Setting prohibited			

Remarks 1. fx : Main system clock oscillation frequency
2. fxT : Subsystem clock oscillation frequency
3 fw : Watch timer clock frequency

CHAPTER 9 WATCHDOG TIMER

9.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(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 9-1 Watchdog Timer Inadvertent Program Overrun Detection Times ★

Inadvertent Detected Time	When Operated at fx = 5.0 MHz	Inadvertent Program Overrun Detected Time	When Operated at fx = 5.0 MHz
$2^{11} \times 1/fx$	410 μs	$2^{15} \times 1/fx$	6.55 ms
$2^{12} \times 1/fx$	819 μs	$2^{16} \times 1/fx$	13.1 ms
$2^{13} \times 1/fx$	1.64 ms	$2^{17} \times 1/fx$	26.2 ms
$2^{14} \times 1/fx$	3.28 ms	$2^{19} \times 1/fx$	104.9 ms

Remark fx: Main system clock oscillation frequency

(2) Interval timer mode

Interrupts are generated at the preset time intervals.

Table 9-2 Interval Time ★

Interval Time	When Operated at fx = 5.0 MHz	Interval Time	When Operated at fx = 5.0 MHz
$2^{11} \times 1/fx$	410 μs	$2^{15} \times 1/fx$	6.55 ms
$2^{12} \times 1/fx$	819 μs	$2^{16} \times 1/fx$	13.1 ms
$2^{13} \times 1/fx$	1.64 ms	$2^{17} \times 1/fx$	26.2 ms
$2^{14} \times 1/fx$	3.28 ms	$2^{19} \times 1/fx$	104.9 ms

Remark fx: Main system clock oscillation frequency

9.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 9-3 Watchdog Timer Configuration

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

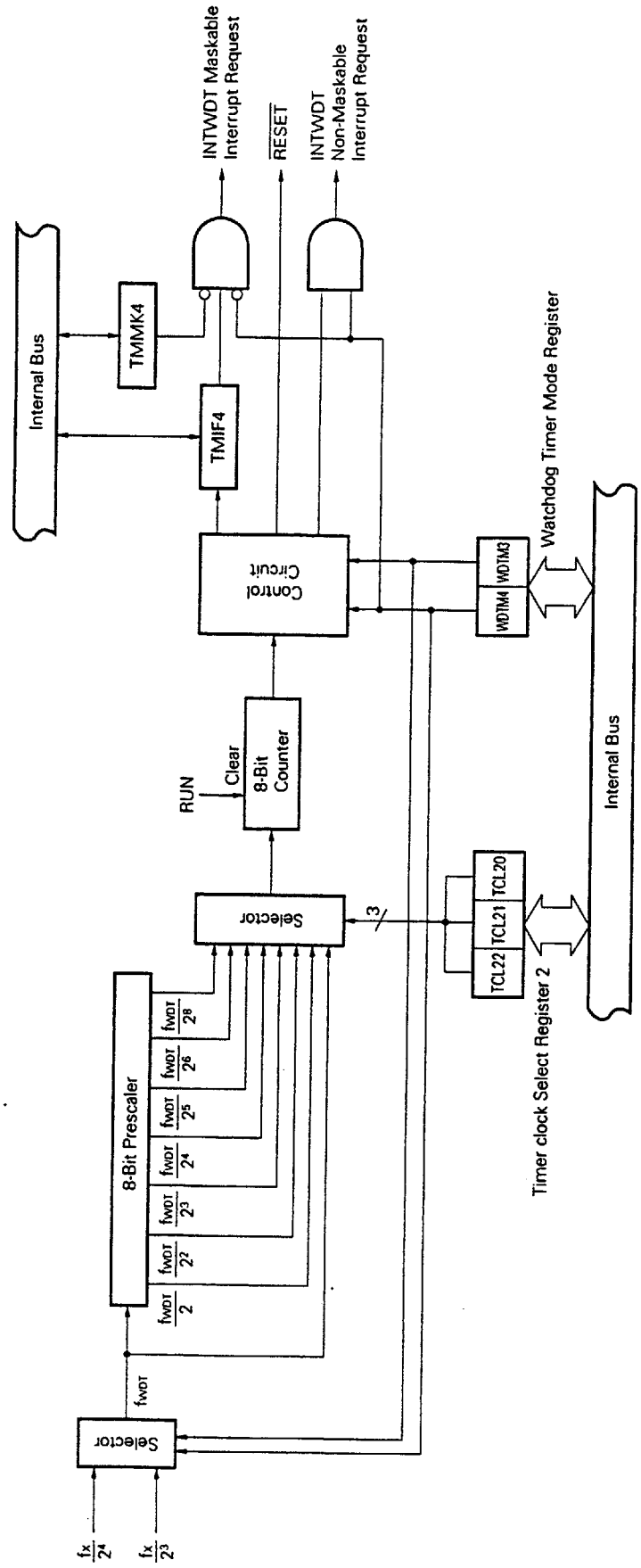


Fig. 9-1 Watchdog Timer Block Diagram

9.3 Watchdog Timer Control Registers

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

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

(1) Timer clock select register 2 (TCL2)

This register sets the watchdog timer count clock.

TCL2 is set with a 1-bit or 8-bit memory manipulate instruction.

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

Remark Besides setting the watchdog timer count clock, TCL2 sets the watch timer count clock and buzzer output frequency.

Fig. 9-2 Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Count Clock Selection	
			Watchdog Timer Mode	Interval Timer Mode
0	0	0	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
0	0	1	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
0	1	0	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
0	1	1	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	0	0	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	1	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	0	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)

TCL24	Watch Timer Count Clock Selection
0	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disabled
1	0	0	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

Caution Before rewriting the data of TCL1, stop the timer once. ★

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XT} : Subsystem clock oscillation frequency

3. × : Don't care

4. Values in parentheses when operated at f_X = 5.0 MHz and f_{XT} = 32.768 kHz ★

(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 manipulate instruction.

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

Fig. 9-3 Watchdog Timer Mode Register Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Watchdog Timer Operating Mode Selection Note 1
0	0	Operation stop
0	1	Interval timer mode (When overflow occurs, maskable interrupt is generated)
1	0	Watchdog timer mode 1 (When overflow occurs, non-maskable interrupt is generated)
1	1	Watchdog timer mode 2 (When overflow occurs, reset operation is activated)

RUN	Watchdog Timer Operation Mode Selection Note 2
0	Count stop
1	Counter cleared, and counting starts

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

2. 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.

Caution When RUN is set to 1 and the watchdog timer is cleared, the actual overflow time is shorter a maximum of 0.5% than the time set by timer clock selection register 2.

9.4 Watchdog Timer Operations

9.4.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer is operated to detect any 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). Setting 1 in bit 7 (RUN) of WDTM starts the watchdog timer. After the watchdog timer starts, 1 should be set in RUN within the set runaway time interval. Setting 1 in RUN clears the watchdog timer and enables the count to be started. If bit 7 (RUN) is not set to 1 and the inadvertent program loop detection time is past, system reset or a non-maskable interrupt is generated according to the WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode bus 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 9-4 Watchdog Timer Overrun Detection Time ★

TCL22	TCL21	TCL20	Overrun Detection Time	When Operated at fx = 5.0 MHz
0	0	0	$2^{11} \times 1/fx$	410 μs
0	0	1	$2^{12} \times 1/fx$	819 μs
0	1	0	$2^{13} \times 1/fx$	1.64 ms
0	1	1	$2^{14} \times 1/fx$	3.28 ms
1	0	0	$2^{15} \times 1/fx$	6.35 ms
1	0	1	$2^{16} \times 1/fx$	13.1 ms
1	1	0	$2^{17} \times 1/fx$	26.2 ms
1	1	1	$2^{19} \times 1/fx$	104.9 ms

Remark fx: Main system clock oscillation frequency

9.4.2 Interval timer operation

The watchdog timer operates as interval timer which generates interrupts repeatedly at an interval of the preset count value when bit 3 (WDTM3) of the watchdog timer mode register (WDTM) is set to 1.

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specify flag (TMPR4) are valided and the maskable interrupt (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 the 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 9-5 Interval Timer Interval Time**

TCL22	TCL21	TCL20	Interval Time	fx = 5.0 MHz
0	0	0	$2^{12} \times 1/fx$	819 μs
0	0	1	$2^{13} \times 1/fx$	1.64 ms
0	1	0	$2^{14} \times 1/fx$	3.28 ms
0	1	1	$2^{15} \times 1/fx$	6.55 ms
1	0	0	$2^{16} \times 1/fx$	13.1 ms
1	0	1	$2^{17} \times 1/fx$	26.2 ms
1	1	0	$2^{18} \times 1/fx$	52.4 ms
1	1	1	$2^{20} \times 1/fx$	209.7 ms

Remark fx: Main system clock oscillation frequency

CHAPTER 10 CLOCK OUTPUT CONTROL CIRCIT

10.1 Clock Output Control Circuit Functions

The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSI. Clocks selected with the timer clock select register 0 (TCL0) are output from the PCL/P35 pin.

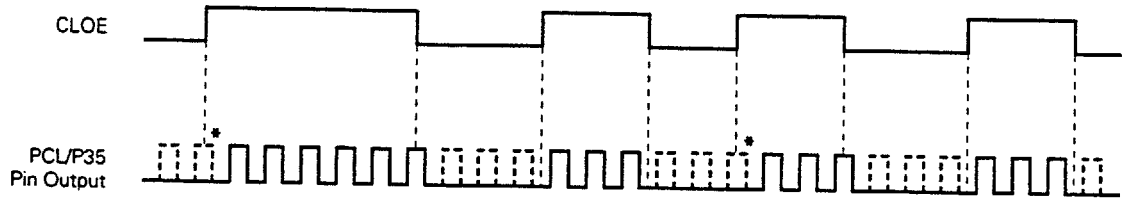
Follow the procedure below to output clock pulses.

① Select the clock pulse output frequency (with clock pulse output disabled) with bits 0 to 3 of TCL0 (TCL00 to TCL03).
② Set the P35 output latch to 0.
③ Set bit 5 (PM35) of port mode register 3 to 0 (Set in output mode).
④ Set bit 7 (CLOE) of timer clock select register 0 to 1.

Caution When setting the P35 output latch to 1, the clock output cannot be used.

Remark When clock output enable/disable is switched, the clock output control circuit does not output pulses with small widths (Refer to * in **Fig. 10-1**).

Fig. 10-1 Remote Controlled Output Application Example



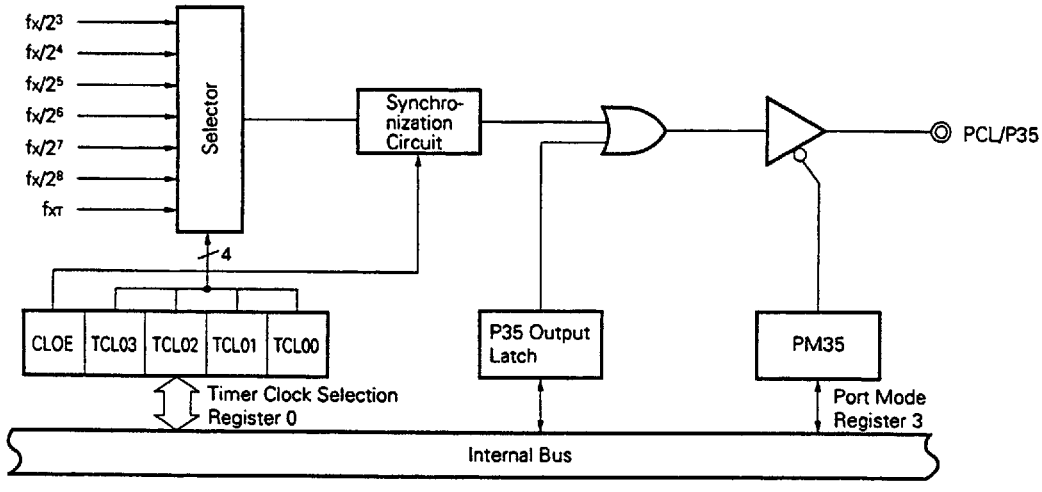
10.2 Clock Output Control Circuit Configuration

The clock output control circuit consists of the following hardware.

Table 10-1 Clock Output Control Circuit Configuration

Item	Configuration
Control register	Timer clock select register 0 (TCL0) Port mode register 3 (PM3)

Fig. 10-2 Clock Output Control Circuit Block Diagram



10.3 Clock Output Function Control Registers

The following two types of registers are used to control the clock output function.

- Timer clock select register 0 (TCL0)
- Port mode register 3 (PM3)

(1) Timer clock select register 0 (TCL0)

This register sets PCL output clock.

TCL0 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TCL0 to 00H.

Remark Besides setting PCL output clock, TCL0 sets the 16-bit timer register count clock.

Fig. 10-3 Timer Clock Select Register 0 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL0	CLOE	TCL06	TCL05	TCL04	TCL03	TCL02	TCL01	TCL00	FF40H	00H	R/W

TCL03	TCL02	TCL01	TCL00	PCL Output Clock Selection
0	0	0	0	f_{XT} (32.768 kHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
Other than the above				Setting prohibited

TCL06	TCL05	TCL04	16-Bit Timer Register Count Clock Selection
0	0	0	TI0 (effective edge specifiable)
0	0	1	f_X (5.0 MHz)
0	1	0	$f_X/2$ (2.5 MHz)
0	1	1	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_X/2^3$ (625 kHz)
Other than the above			Setting prohibited

CLOE	PCL Output Control
0	Output disabled
1	Output enabled

Cautions
1. **Set the valid edge of the TI0/INTP0 pin with an external interrupt mode register and select the sampling clock frequency with a sampling clock select register.**
2. **When enabling PCL output, set TCL00 to TCL03 and then set CLOE to 1 with a 1-bit memory manipulate instruction.**
3. **When reading the count value with TM0 count value set to TI0, read it from TM0 instead of CR01 capture register.**
4. **Before rewriting the data of TCL1, stop the timer once.** ★

Remarks
1. f_X : Main system clock oscillation frequency
2. f_{XT} : Subsystem clock oscillation frequency
3. TI0 : 16-bit timer/event counter input pin
4. TM0: 16-bit timer register
5. Values in parentheses when operated at f_X = 5.0 MHz and f_{XT} = 32.768 kHz ★

(2) Port mode register 3 (PM3)

This register set port 3 input/output in 1-bit units.

When using the P35/PCL pin for clock output function, set PM35 and P35 output latches to 0.

PM3 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Fig. 10-4 Port Mode Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM3	1	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 6)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Caution Set 1 in bit 7.

CHAPTER 11 BUZZER OUTPUT CONTROL CIRCUIT

11.1 Buzzer Output Control Circuit Functions

The buzzer output control circuit outputs 1.2 kHz, 2.4 kHz and 4.9 kHz divided signals. The buzzer frequency ★
selected with timer clock select register 2 (TCL2) is output from the BUZ/P36 pin.

Follow the procedure below to output the buzzer frequency.

① Select the buzzer output frequency with bits 5 to 7 (TCL25 to TCL27) of TCL2.
② Set the P36 output latch to 0.
③ Set bit 6 (PM36) of port mode register 3 to 0.

Caution When set 1 in P36 output latch, the buzzer output cannot be used.

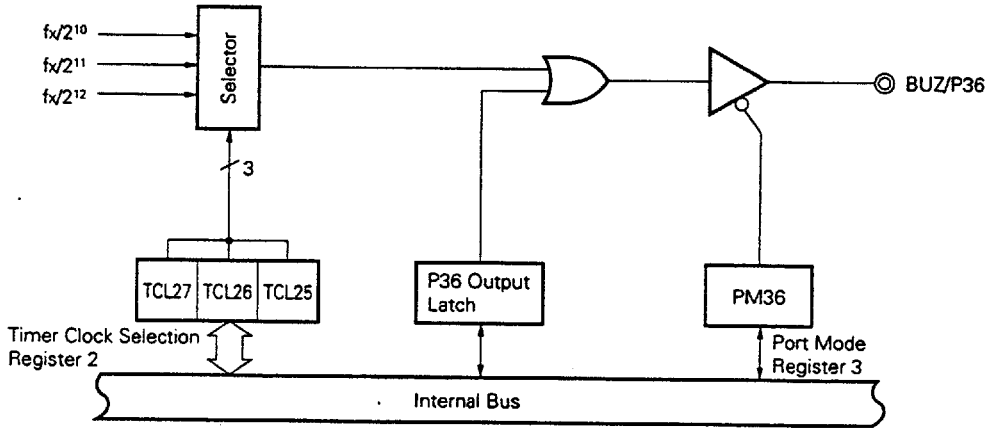
11.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 11-1 Buzzer Output Control Circuit Block Diagram

Item	Configuration
Control register	Timer clock select register 2 (TCL2) Port mode register 3 (PM3)

Fig. 11-1 Buzzer Output Control Circuit Block Diagram



11.3 Buzzer Output Function Control Registers

The following two types of registers are used to control the buzzer output function.

- Timer clock select register 2 (TCL2)
- Port mode register 3 (PM3)

(1) Timer clock select register 2 (TCL2)

This register sets the buzzer output frequency.

TCL2 is set with an 8-bit memory manipulate instruction.

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

Remark Besides setting the buzzer output frequency, TCL2 set the clock timer count clock and the watchdog timer count clock.

Fig. 11-2 Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Count Clock Selection	
			Watchdog Timer Mode	Interval Timer Mode
0	0	0	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
0	0	1	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
0	1	0	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
0	1	1	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	0	0	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	1	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	0	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)

TCL24	Watch Timer Count Clock Selection
0	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disabled
1	0	0	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

Caution Before rewriting the data of TCL1, stop the timer once. ★

Remarks 1. f_X : Main system clock oscillation frequency
2. f_{XT} : Subsystem clock oscillation frequency
3. × : Don't care
4. Values in parentheses when operated at f_X = 5.0 MHz and f_{XT} = 32.768 kHz ★

(2) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P36/BUZ pin for buzzer output function, set PM36 and P36 output latches to 0.

PM3 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Fig. 11-3 Port Mode Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM3	1	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 6)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Caution Set 1 in bit 7.

CHAPTER 12 A/D CONVERTER

12.1 A/D Converter Functions

The A/D converter converts an analog input into a digital value. It consists of 8 channels (ANI0 to ANI7) with an 8-bit resolution.

The conversion method is based on successive approximation and the conversion result is held in the 8-bit A/D conversion result register (ADCR).

The following two ways are available to start A/D conversion.

(1) Hardware start

Conversion is started by trigger input (INTP3).

(2) Software start

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

One channel of analog input is selected from among ANI0 to ANI7 and A/D conversion is carried out. In the case of hardware start, A/D conversion operation stops when it terminates. In the case of software start, the A/D conversion operation repeats. Each time A/D conversion is terminated, the interrupt request (INTAD) is generated.

12.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 12-1 A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Control register	A/D converter mode register (ADM) A/D converter input select register (ADIS)
Register	Successive approximation register (SAR) A/D conversion result register (ADCR)

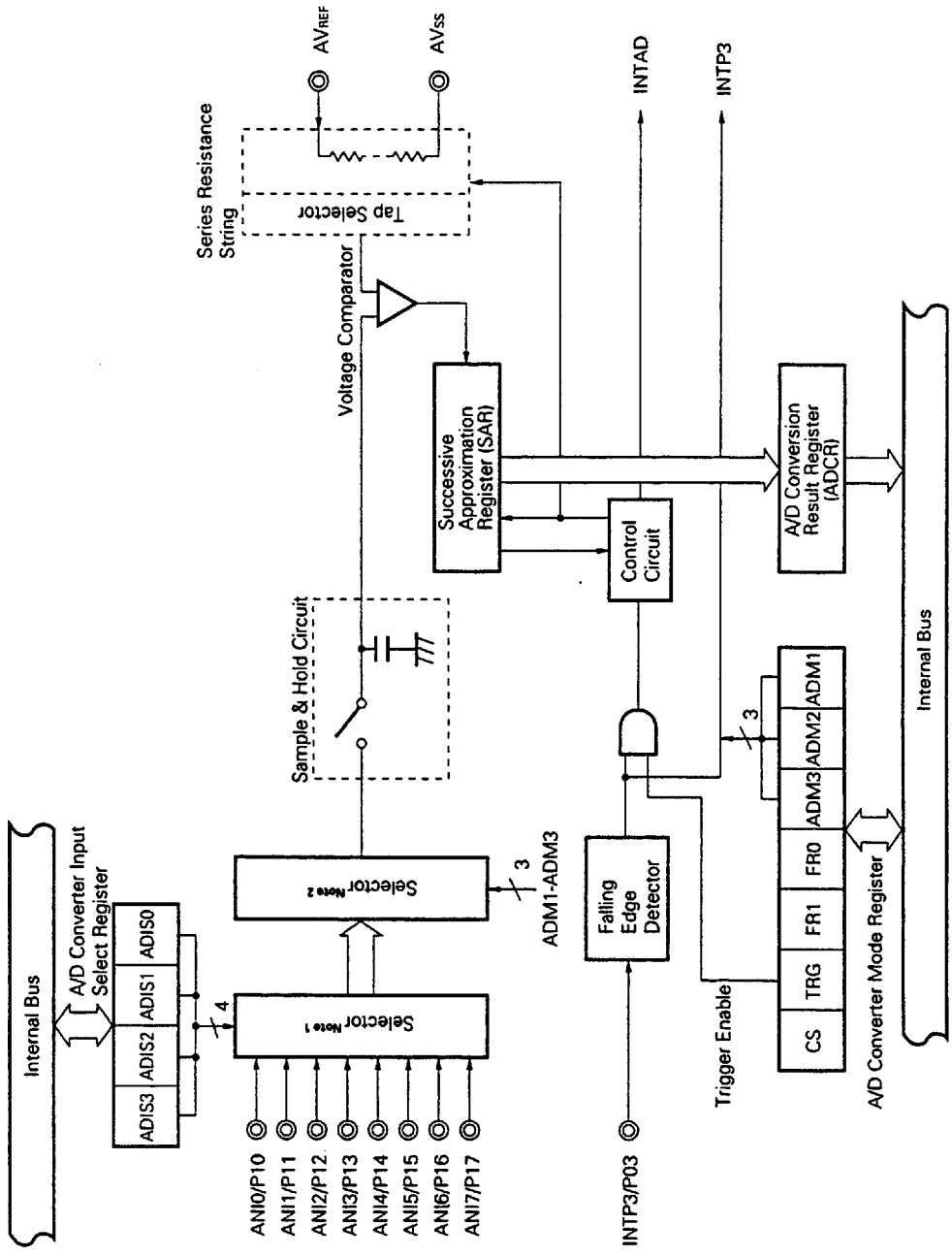


Fig. 12-1 A/D Converter Block Diagram

Notes 1. Selector to select the number of channels to be used for analog input

2. Selector to select the channel for A/D conversion

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistance string and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is set (termination of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

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

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

ADCR is read with an 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input makes ADCR indeterminate.

(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 resistance string output voltage.

(5) Series resistance string

The series resistance string which is inserted between AV_{REF} and AVss generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are 8-channel analog input pins to input analog signals to undergo A/D conversion to the A/D converter. These pins, other than the one selected by the A/D converter input select register (ADIS) as an analog input pin, can be used as input/output port pins. ★

Caution Use ANI0 to ANI7 input voltages within the specified range. If a voltage higher than AV_{REF} or lower than AVss is applied (even if within the absolute maximum ratings), the converted value of the corresponding channel becomes indeterminate and may adversely affect the converted values of other channels.

(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 AVss.

(8) AVss pin

★ This is an A/D converter grounding pin. Make sure that this pin is always at the same potential as the V_{DD} pin, even when the A/D converter is not used.

12.3 A/D Converter Control Registers

The following two types of registers are used to control the A/D converter.

- A/D converter mode register (ADM)
- A/D converter input select register (ADIS)

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

This register sets the analog input channel for A/D conversion, conversion start/stop and external trigger.
ADM is set with a 1-bit or 8-bit memory manipulate instruction.
$\overline{\text{RESET}}$ input sets ADM to 01H.

Fig. 12-2 A/D Converter Mode Register Format

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	1	FF80H	01H	R/W

ADM3	ADM2	ADM1	Analog Input Channel Selection
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

FR1	FR0	A/D Conversion Time Selection[Note 1]	
		When operated at f_x = 5.0 MHz	When operated at f_x = 4.19 MHz
0	0	160/f_x (32.0 μs)	160/f_x (38.1 μs)
0	1	80/f_x (Setting prohibited[Note2])	80/f_x (19.1 μs)
1	0	200/f_x (40.0 μs)	200/f_x (47.7 μs)
1	0	Setting prohibited	

TRG	External Trigger Selection
0	No external trigger (Software start mode)
1	Conversion started by external trigger (Hardware start mode)

CS	A/D Conversion Operation Control
0	Operation stop
1	Operation start

Notes 1. Set A/D conversion time 19.1 μs or more.

2. This setting is prohibited because the A/D conversion time is set to less than 19.1 μs.

Cautions 1. Set 1 to bit 0.

2. Before executing the STOP instruction, clear the CS bit to 0 to stop A/D conversion operation.

3. To resume an A/D conversion operation that has once been stopped, clear interrupt request flag (ADIF), and then start the operation.

Remark fx : Main system clock oscillation frequency

(2) A/D converter input select register (ADIS)

This register determines whether the ANI0/P10 to ANI7/P17 should be used for analog input channels or ports.

★ These pins, other than the one that is selected as an analog input channel, can be used as I/O port pins.

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

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

Cautions 1. Set the analog input channel in the following order.

① Set the number of analog input channels with ADIS.

② Using ADM, select one channel to undergo A/D conversion from among the channels set for analog input with ADIS.

2. Internal pull-up resistor is not used for the channels set for analog input with ADIS, irrespective of the value of bit 1 (PUO1) of the pull-up resistor option register.

Fig. 12-3 A/D Convertor Input Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	Analog Input Channel Number Selection
0	0	0	0	No analog input channel (P10 to P17)
0	0	0	1	1 channel (ANI0, P11 to P17)
0	0	1	0	2 channels (ANI0, ANI1, P12 to P17)
0	0	1	1	3 channels (ANI0 to ANI2, P13 to P17)
0	1	0	0	4 channels (ANI0 to ANI3, P14 to P17)
0	1	0	1	5 channels (ANI0 to ANI4, P15 to P17)
0	1	1	0	6 channels (ANI0 to ANI5, P16 to P17)
0	1	1	1	7 channels (ANI0 to AN16, P17)
1	0	0	0	8 channels (ANI0 to ANI7)
Other than the above				Setting prohibited

12.4 A/D Converter Operations

12.4.1 Basic operations of A/D converter

① Set the number of analog input channels with A/D converter input select register (ADIS).

② From among the analog input channels set with ADIS, select one channel for A/D conversion with A/D converter mode register (ADM).

③ Sample the voltage input to the selected analog input channel with the sample & hold circuit.

④ Sampling for the specified period of time sets the sample & hold circuit to the hold state so that the circuit hold the input analog voltage until termination of A/D conversion.

⑤ Bit 7 of successive approximation register (SAR) is set and the tap selector sets the series resistance string voltage tap to (1/2) AV_{REF}.

⑥ The voltage difference between the series input string voltage tap and analog input is compared with a voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set. If the input is smaller than (1/2) AV_{REF}, the MSB is reset.

⑦ Next, bit 6 of SAR is automatically set and the operation proceeds to the next comparison. In this case, the series resistance string voltage tap is selected according to the preset value of bit 7 as described below.

- Bit 7 = 1: (3/4) AV_{REF}
- Bit 7 = 0: (1/4) AV_{REF}

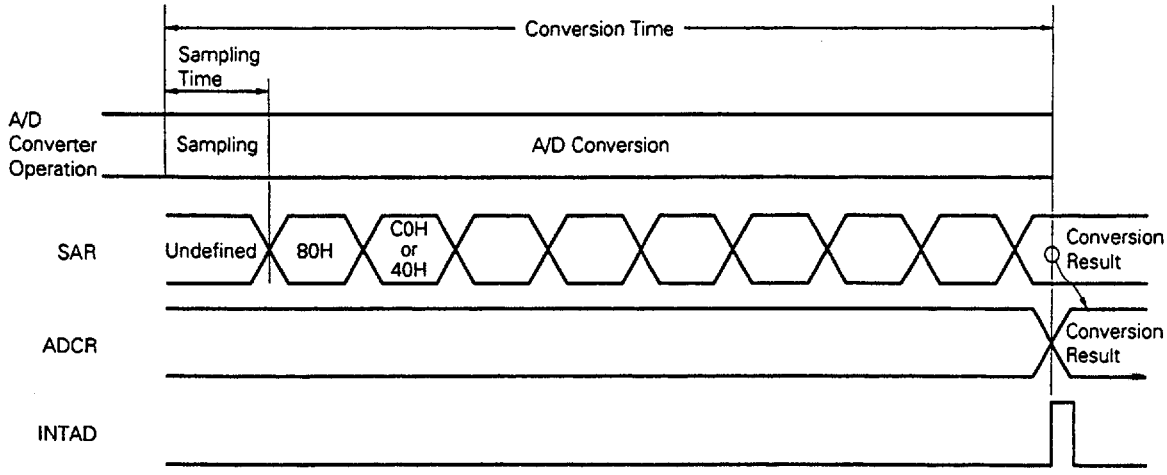
The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated with the result as follows.

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

⑧ Comparison of this sort continues up to bit 0 of SAR.

⑨ Upon completion of the comparison of 8 bits, any effective digital resultant value remains in SAR and the resultant value is transferred to and latched in the A/D conversion result register (ADCR). Simultaneously, the A/D conversion end interrupt request (INTAD) can be generated.

Fig. 12-4 A/D Converter Basic Operation



The A/D conversion is performed consecutively until the CS bit is reset (0) by software.

During A/D conversion, if write operation to the ADM register is performed, the conversion is initialized, and if the CS bit has been set (1), the conversion starts from the beginning.

The ADCR register becomes undefined by $\overline{\text{RESET}}$ input.

12.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (AN0 to AN7) and the A/D conversion results (value stored in the ADCR) is described in the following expression.

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

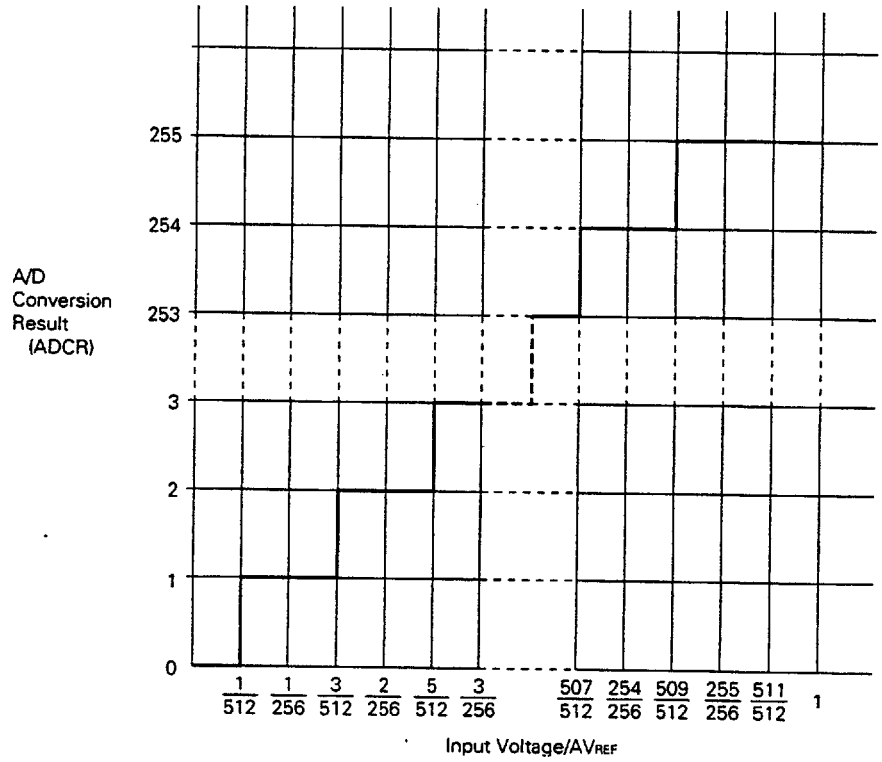
or

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

Remark INT ( ) : Function which returns integer parts of value in parentheses.
V_{IN} : Analog input voltage
AV_{REF} : AV_{REF} pin voltage
ADCR : ADCR register value

Fig. 12-5 shows the relation between the analog input voltage and the A/D conversion result.

Fig. 12-5 Relation between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operating mode

The operating mode is a select mode. One analog input channel is selected from among ANI0 to ANI7 with the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and A/D conversion is executed.

The following two ways are available to start A/D conversion.

- Hardware start : Conversion is started by trigger input (INTP3).
- Software start : Conversion is started by setting ADM.

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

(1) A/D conversion by hardware start

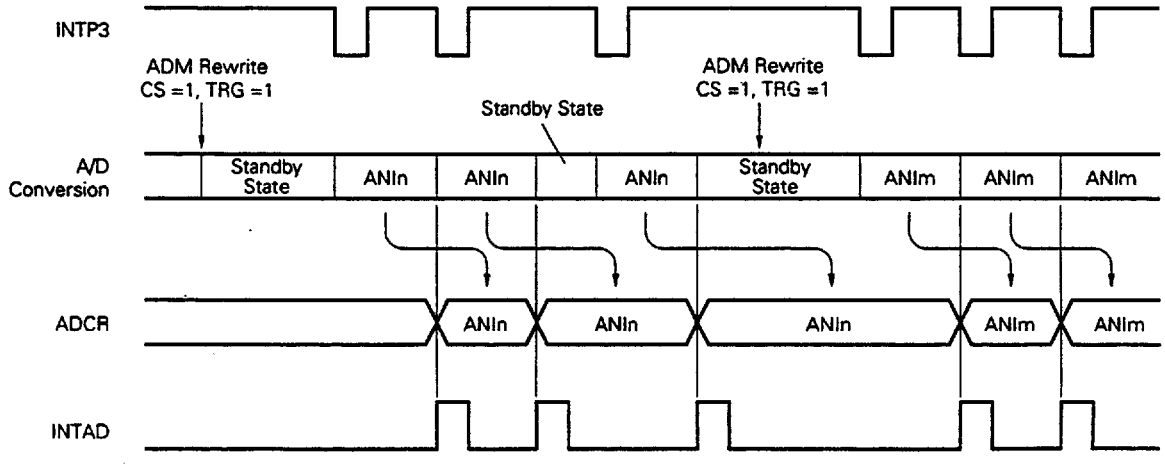
When bit 6 (TRG) and bit 7 (CS) of ADM are set to 1, the A/D conversion standby state is generated. When the external trigger signal (INTP3) is input, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, another operation is not started until a new external trigger signal is input.

If data with CS set to 1 is written 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 CS set to 0 is written to ADM during A/D conversion, the A/D conversion operation stops immediately.

Fig. 12-6 A/D Conversion by Hardware Start



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

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

(2) A/D conversion by software start

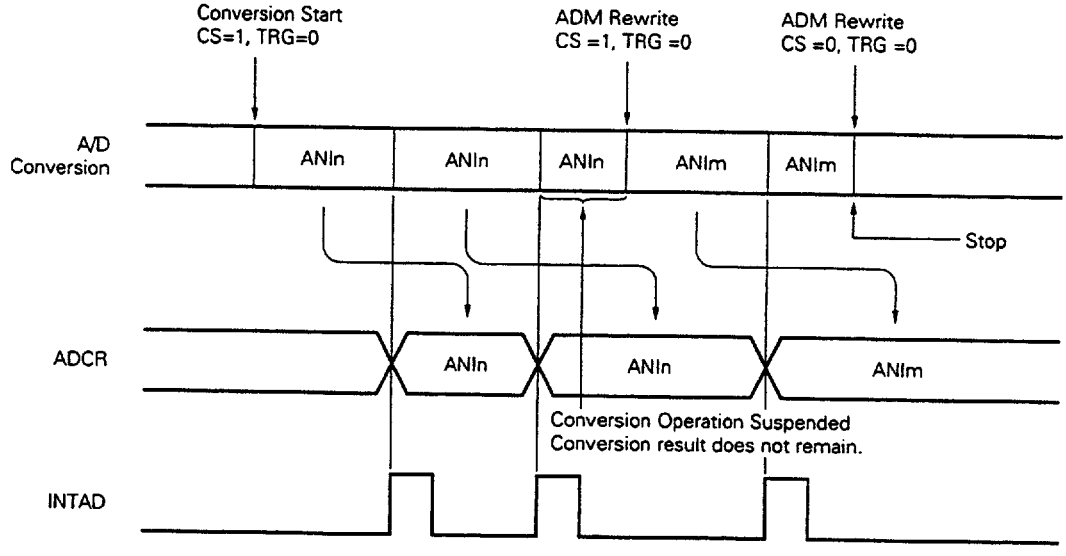
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, the next A/D conversion operation starts immediately. The A/D conversion operation continues repeatedly until new data is written to ADM.

If data with CS set to 1 is written again during A/D conversion, the converter suspends its A/D conversion operation and starts A/D conversion on the newly written data.

If data with CS set to 0 is written to AMD during A/D conversion, the A/D conversion operation stops immediately.

Fig. 12-7 A/D Conversion by Software Start



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

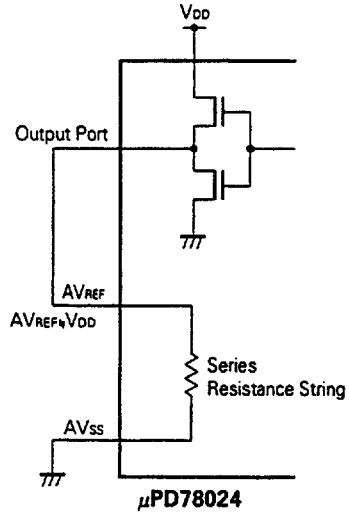
2. m= 0, 1, ... 7

12.5 A/D Converter Cautions

(1) Note on consumption current in standby mode

The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode or in HALT mode with the subsystem clock. As a current still flows in the AV_{REF} pin at this time, this current must be cut in order to minimize the overall system power dissipation. In Fig. 12-8, in standby mode, the power dissipation can be reduced by outputting a low-level signal to the output port. However, there is no precision to the actual AV_{REF} voltage, and therefore the conversion values themselves lack precision and can only be used for relative comparison.

Fig. 12-8 Example of Method of Reducing Power Dissipation in Standby Mode



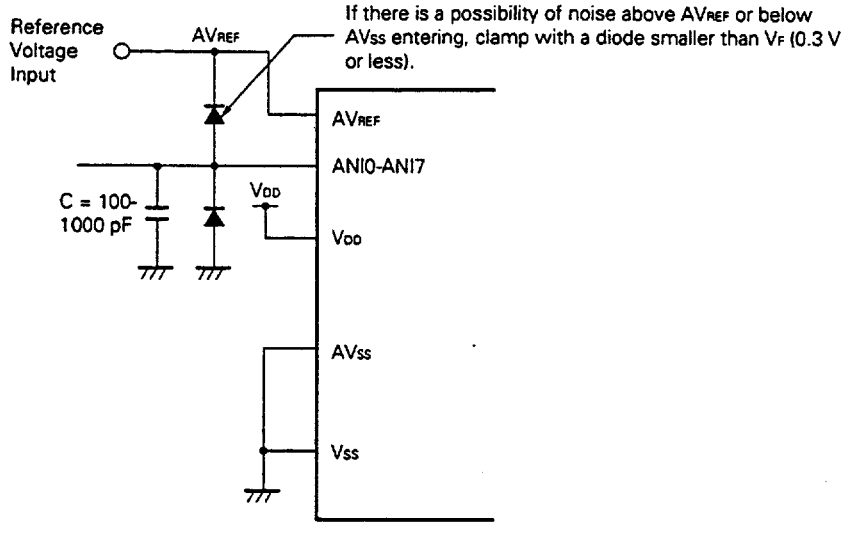
(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage above AVREF or below AVss is input (even if within the absolute maximum rating range), the conversion value for that channel will be indeterminate and the conversion values of the other channels may also be affected.

(3) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on pins AVREF and ANI0 to ANI7. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that C be connected externally as shown in Fig. 12-9 in order to reduce noise.

Fig. 12-9 Analog Input Pin Disposition



(4) Pins ANI0/P10 to ANI7/P17

The analog input pins ANI0 to ANI7 also function as input/output port (PORT1) pins.

A pin used as an analog input should be specified in input mode.

When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute a PORT1 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.

(5) AVREF pin input high-impedance

The series resistor string of approx. 10 kΩ is connected between the AVREF pin and AVss pin.

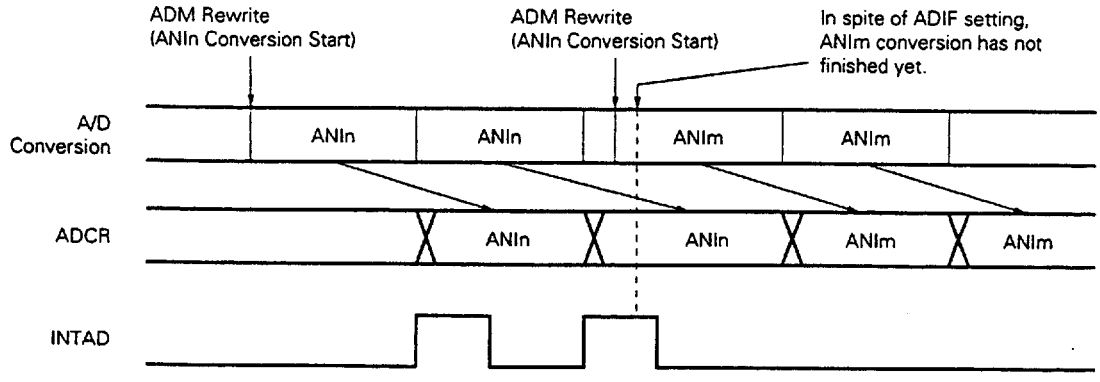
Therefore, when the output impedance of the reference voltage source is high, it is regarded as being connected in parallel with the series resistor string between the AVREF pin and AVss pin, resulting in a greater reference voltage error.

★ **(6) Interrupt request flag (ADIF)**

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM) is changed. Therefore, when the analog input pin is changed during the A/D conversion, the result of A/D conversion on the analog input before changing and the conversion end interrupt request flag may be set immediately before the ADM rewrite, and if ADIF is read immediately after ADM write, ADIF may be set even if the A/D conversion is not complete. Thus, care is needed (see **Fig. 12-10**).

When the A/D conversion is once stopped and restarts, the A/D conversion interrupt request flag (ADIF) should be cleared before the restart.

Fig. 12-10 A/D Conversion End Interrupt Generation Timing



CHAPTER 13 SERIAL INTERFACE CHANNEL 0

The μPD78024 subseries incorporates two channels of clock synchronous serial interfaces. Differences between channels 0 and 1 are as follows. For the details of the serial interface channel 1, refer to **CHAPTER 14 SERIAL INTERFACE CHANNEL 1**).

Table 13-1 Differences between Channels 0 and 1

Serial Transfer Mode		Channel 0	Channel 1
3-wire serial I/O	Clock selection	$fx/2^2$, $fx/2^3$, $fx/2^4$, $fx/2^5$, $fx/2^6$, $fx/2^7$, $fx/2^8$, $fx/2^9$, external clock, TO2 output	$fx/2^2$, $fx/2^3$, $fx/2^4$, $fx/2^5$, $fx/2^6$, $fx/2^7$, $fx/2^8$, $fx/2^9$, external clock, TO2 output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit
	Transfer end flag	Serial transfer end interrupt request flag (INTCSI0)	Serial transfer end interrupt request flag (INTCSI1)
SBI (serial bus interface)		Can be used	Not provided
2-wire serial I/O			

13.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following four modes.

- Operation stop mode
- 3-wire serial I/O mode
- SBI (serial bus interface) mode
- 2-wire serial I/O mode

(1) Operation stop mode

This mode is used serial transfer is not carried out. Power consumption can be reduced.

(2) 3-wire serial I/O mode

This mode is used for 8-bit data transfer using three lines, each for serial clock ($\overline{SCK0}$), serial output (SO0) and serial input (SI0).

The 3-wire serial I/O mode enables simultaneous transmission/reception and so decreases the data transfer processing time.

Since the start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB, connection is enabled with either start bit device.

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface as is the case with the 75X, 78K and 17K series.

(3) SBI(serial bus interface) mode

This mode is used for 8-bit data transfer with two or more devices using two lines of serial clock ($\overline{SCK0}$) and serial data bus (SB0 or SB1).

The SBI mode is in compliance with the NEC serial bus format. In the SBI mode, the transmitter can output "addresses" for serial communication target device selection, "commands" to give instructions to the target devices and actual "data", all onto the serial data bus. The receiver can automatically distinguish the received data into "address", "command" or "data", by hardware.

This function enables the input/output ports to be used effectively and the application program control portions to be simplified.

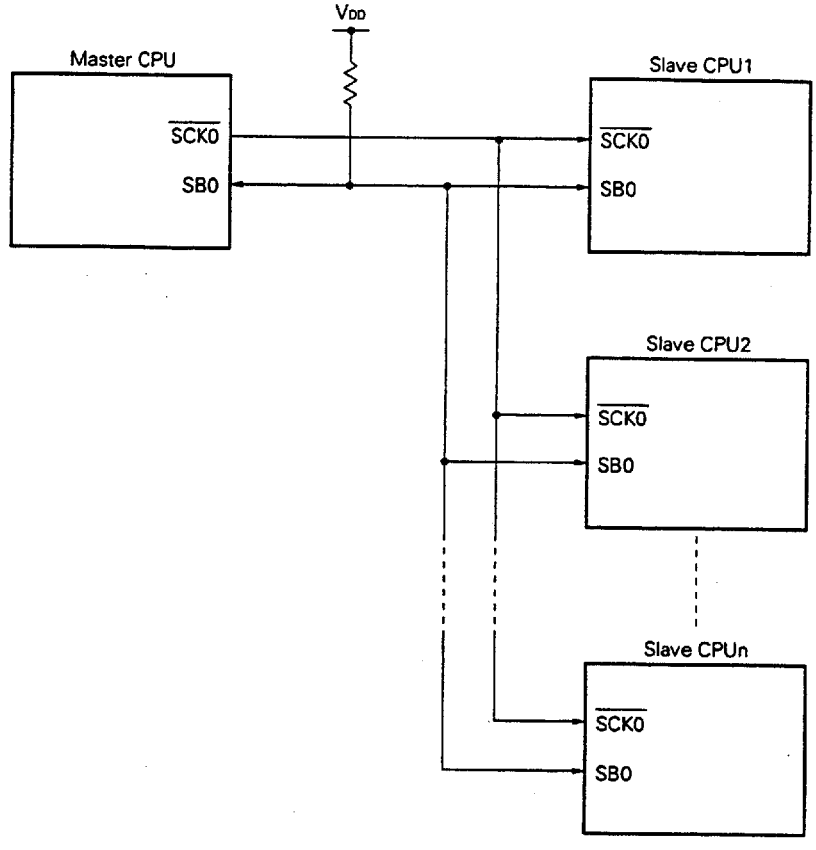
In this mode, the wake-up function for handshake and the acknowledge signal and busy signal output function can also be used.

(4) 2-wire serial I/O mode

This mode is used for 8-bit data transfer using two lines of serial clock ($\overline{SCK0}$) and serial data bus (SB0 or SB1).

This mode enables to cope with any data transfer format by controlling $\overline{SCK0}$ and the SB0 or SB1 output level. Thus, the handshake lines previously necessary for connection of two or more devices can be removed and the input/outputs can be used effectively.

Fig. 13-1 Serial Bus Interface (SBI) System Configuration Example



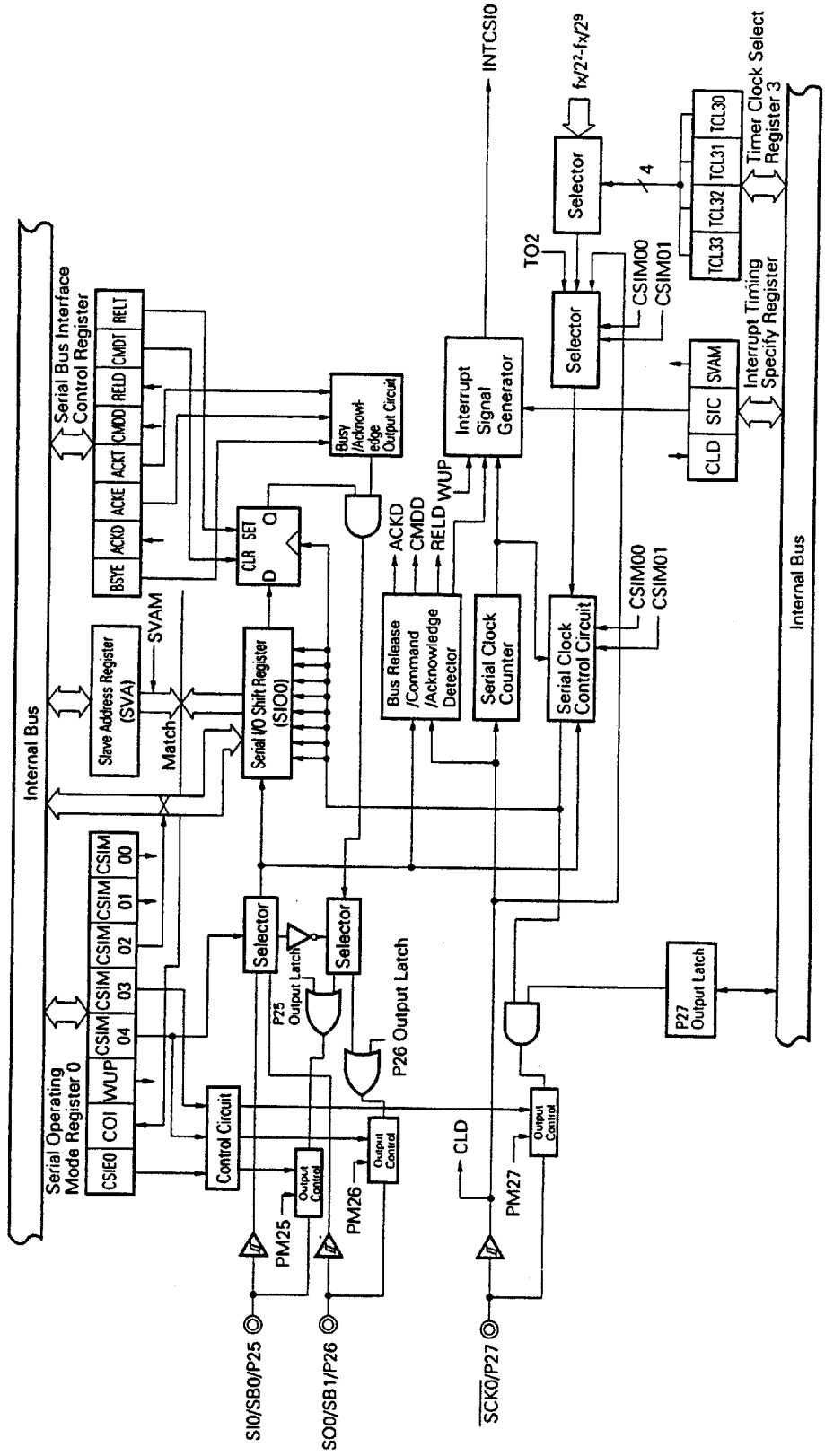
13.2 Serial Interface Channel 0 Configuration

Serial interface channel 0 consists of the following hardware.

Table 13-2 Serial Interface Channel 0 Configuration

Item	Configuration
Register	Serial I/O shift register 0 (SIO0) Slave address register (SVA)
Control register	Timer clock select register 3 (TCL3) Serial operating mode register 0 (CSIM0) Serial bus interface control register (SBIC) Interrupt timing specify register (SINT)

Fig. 13-2 Serial Interface Channel 0 Block Diagram



Remark Output control selects CMOS output or N-ch open-drain output.

(1) Serial I/O shift register 0 (SIO0)

This is an 8-bit register to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO0 is set with an 8-bit memory manipulate instruction.

When bit 7 (CSIE0) of serial operating mode register 0 (CSIM0) is 1, writing data to SIO0 starts serial operation.

In transmission, data written to SIO0 is output to the serial output (SO0) or serial data bus (SB0/SB1). In reception, data is read from the serial input (SIO) or SB0/SB1 to SIO0.

The SBI mode and 2-wire serial I/O mode bus configurations enables the pin to serve for both input and output. Thus, in the case of a device for reception, write FFH to SIO0 in advance (except when address reception is carried out by setting bit 5 (WUP) of CSIM0 to 1).

In the SBI mode, the busy state can be cleared by writing data to SIO0. In this case, bit 7 (BSYE) of the serial bus interface control register (SBIC) is not cleared to 0.

$\overline{\text{RESET}}$ input makes SIO0 undefined.

(2) Slave address register (SVA)

This is an 8-bit register to set the slave address value for connection of a slave device to the serial bus.

SVA is set with an 8-bit memory manipulate instruction.

The master device outputs a slave address for selection of a particular slave device to the connected slave device. These two data (the slave address output from the master device and the SVA value) are compared with an address comparator. If they match, the slave device has been selected. In that case, bit 6 (COI) of serial operating mode register 0 (CSIM0) becomes 1.

Address comparison can also be executed on the data of LSB-masked high-order 7 bits with bit 4 (SVAM) of the interrupt timing specify register (SINT).

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. When bit 5 (WUP) of CSIM0 is 1, the interrupt request signal (CSIIF0) is generated only if the matching is detected. This interrupt request enables to recognize the generation of the communication request from the master device.

Further, when SVA transmits data as master or slave device in the SBI or 2-wire serial I/O mode, errors are detected if any.

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

(3) SO0 latch

This latch holds SI0/SB0/P25 and SO0/SB1/P26 pin levels. It can be directly controlled by software. In the SBI mode, this latch is set upon termination of the 8th serial clock.

(4) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

(5) Serial clock control circuit

This circuit controls serial clock supply to the serial I/O shift register 0 (SIO0). When the internal system clock is used, the circuit also controls clock output to the $\overline{SCK0}$/P27 pin.

(6) Interrupt signal generator

This circuit control interrupt request signal generation. It generates the interrupt request signal in the following cases.

- In the 3-wire serial I/O mode and 2-wire serial I/O mode
  This circuit generates an interrupt request signal every eight serial clocks.
- In the SBI mode
  When WUP[Note] is 0:
  Generates an interrupt request signal every eight serial clocks.
  When WUP[Note] is 1:
  Generates an interrupt request signal when the serial I/O shift register 0 (SIO0) value matches the slave address register (SVA) value after address reception.

Note WUP is a wake-up function specify bit. It is bit 5 of serial operation mode register 0 (CSIMO).

(7) Busy/acknowledge output circuit and bus release/command/acknowledge detector

These two circuits output and detect various control signals in the SBI mode.
These do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

(8) P27 output latch

This latch generates a serial clock by software after termination of eight serial clocks.
When using serial interface channel 0, set the P27 output latch to 1.
$\overline{RESET}$ input sets the latch to 0.

13.3 Serial Interface Channel 0 Control Registers

The following four types of registers are used to control serial interface channel 0.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 0 (CSIMO)
- Serial bus interface control register (SBIC)
- Interrupt timing specify register (SINT)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 0.

TCL3 is set with an 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Remark Besides setting the serial clock of serial interface channel 0, TCL3 sets the serial clock of serial interface channel 1.

Fig. 13-3 Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection
0	1	1	0	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_x/2^8$ (19.5 kHz)
1	1	0	1	$f_x/2^9$ (9.8 kHz)
Other than the above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection
0	1	1	0	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_x/2^8$ (19.5 kHz)
1	1	0	1	$f_x/2^9$ (9.8 kHz)
Other than the above				Setting prohibited

Caution Before rewriting the data of TCL3, stop the timer once. ★

Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz ★

(2) Serial operating mode register 0 (CSIM0)

This register sets serial interface channel 0 serial clock, operating mode, operation enable/stop, wake-up function and displays the address comparator match signal.

CSIM0 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Fig. 13-4 Serial Operating Mode Register 0 Format (1/2)

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W Note 1

R/W

CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
0	×	Clock externally input to $\overline{SCK0}$ pin
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/P25 Pin Function	SO0/P26 Pin Function	$\overline{SCK0}$/P27 Pin Function
0	×	0	1	×	0	0	0	1	3-wire serial I/O mode	MSB	SI0 Note 2 (Input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS I/O)
		1								LSB			
1	0	0	Note 3 ×	Note 3 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{SCK0}$ (CMOS I/O)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	
1	1	0	Note 3 ×	Note 3 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{SCK0}$ (N-ch open-drain I/O)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

R/W

WUP	WUP Wake-Up Function Control
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register data in SBI mode

Notes 1. Bit 6 (COI) is a Read-Only bit.
2. Can be used as P25 (CMOS input/output) when using transmission only.
3. Can be used freely as port function.

Remark ×: Don't care

Fig. 13-4 Serial Operating Mode Register 0 Format (2/2)

R

COI	Slave Address Comparison Result Flag[Note]
0	Slave address register not equal to serial I/O shift register 0 data
1	Slave address register equal to serial I/O shift register 0 data

R/W

CSIE0	Serial Interface 0 Operation Control
0	Operation disabled
1	Operation enabled

Note When CSIE = 0, COI becomes 0.

Remark ×: Don't care

(3) Serial bus interface control register (SBIC)

This register sets serial bus interface operation and displays statuses.

SBIC is set with a 1-bit or 8-bit memory manipulate instruction.

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

Fig. 13-5 Serial Bus Interface Control Register Format (1/2)

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W[Note]

R/W

RELT	Use for bus release signal output. When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W

CMDT	Use for command signal output. When CMDT = 1, SO latch is set to (0). After SO latch setting, automatically cleared to (0). Also cleared to 0 when CSIE0 = 0.

R

RELD	Bus Release Detection	
Clear Conditions (RELD = 0)		Set Conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When bus release signal (REL) is detected

Note Bits 2, 3, and 6 (RELD, CMDD, ACKD) are Read-Only bits.

Fig. 13-5 Serial Bus Interface Control Register Format (2/2)

R

CMDD	Command Detection	
Clear Conditions (CMDD = 0)		Set Conditions (CMDD = 1)
• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When command signal (CMD) is detected (CMD)

R/W

ACKT	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction to be set to 1 and, after acknowledge signal output, automatically cleared to 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

R/W

ACKE	Acknowledge Signal Output Control	
0	Acknowledge signal automatic output disable (output with ACKT enable)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with the falling edge of 9th $\overline{\text{SCK0}}$ clock (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

R

ACKD	Acknowledge Detection	
Clear Conditions (ACKD = 0)		Set Conditions (ACKD = 1)
• When $\overline{\text{SCK0}}$ falls after busy mode has been released after execution of transfer start instruction • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When acknowledge signal ($\overline{\text{ACK}}$) is detected at the rising edge of $\overline{\text{SCK0}}$ clock after completion of transfer

★

R/W

Note BSYE	Synchronizing Busy Signal Output Control
0	Busy signal in output is disabled at the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction to be cleared to 0.
1	Busy signal is output at the falling edge of $\overline{\text{SCK0}}$ clock following the acknowledge signal.

Note The serial interface transfer start or the address signal reception release a busy mode. However, the BSYE flag is not cleared to 0.

(4) Interrupt timing specify register (SINT)

This register sets the bus release interrupt and address mask functions and displays the P27 pin level status.

SINT is set with a 1-bit or 8-bit memory manipulate instruction.

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

Fig. 13-6 Interrupt Timing Specify Register Format

Symbol	7	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W Note 1

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interruput Source Selection Note 2
0	CSIIF0 is set upon termination of serial interface channel 0 transfer.
1	CSIIF0 is set upon termination of serial interface channel 0 transfer or upon bus release detection.

R

CLD	$\overline{\text{SCK0}}$ Pin Level Note 3
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is a Read-Only bit.

2. When the wake-up function is used, set 0 to SIC.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register

13.4 Serial Interface Channel 0 Operations

The following four operating modes are available to the serial interface channel 0.

- Operation stop mode
- 3-wire serial I/O mode
- SBI mode
- 2-wire serial I/O mode

13.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 0 (SIO0) does not carry out shift operation either and thus it can be used as ordinary 8-bit register.

In the operation stop mode, the P25/SI0/SB0, P26/SO0/SB1 and P27/$\overline{SCK0}$ pins can be used as ordinary input/output ports.

(1) Register set

The operation stop mode is set with the serial operation mode register 0 (CSIM0).

CSIM0 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{RESET}$ input sets CSIM0 to 00H.

The shaded area is used in the operation stop mode.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stop
	1	Operation enabled

13.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface as is the case with the 75X, 78K and 17K series.

Communication is carried out with three lines of serial clock ($\overline{SCK0}$), serial output (SO0) and serial input (SI0).

(1) Register set

The 3-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{RESET}$ input sets CSIM0 to 00H.

The shaded area is used in the 3-wire serial I/O mode.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W Note 1

R/W

CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
0	×	Clock externally input to $\overline{SCK0}$ pin
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/P25 Pin Function	SO0/P26 Pin Function	$\overline{SCK0}$/P27 Pin Function
0	×	0	1	×	0	0	0	1	3-line serial I/O mode	MSB	SI0 Note 2 (Input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS I/O)
		1								LSB			
1	0	SBI mode (Refer to **13.4.3 SBI mode operation.**)											
1	0	2-wire serial I/O mode (Refer to **13.4.4 2-wire serial I/O mode operation.**)											

R/W

WUP	Wake-Up Function Control
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register data in SBI mode

R

COI	Slave Address Compare Result Flag Note 3
0	Slave address register not equal to serial I/O shift register 0 data
1	Slave address register equal to serial I/O shift register 0 data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation disable
1	Operation enable

Notes 1. Bit 6 (COI) is a Read-Only bit.
2. Can be used as P25 (CMOS input/output) when using transmission only.
3. When CSIEO = 0, COI becomes 0.

Remark × : Don't care

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulate instruction.

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

The shaded area is used in the 3-wire serial I/O mode.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W		
R/W	RELT	When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleated to 0 when CSIE0 = 0.
R/W	CMDT	When CMDT = 1, SO latch is set to 0. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

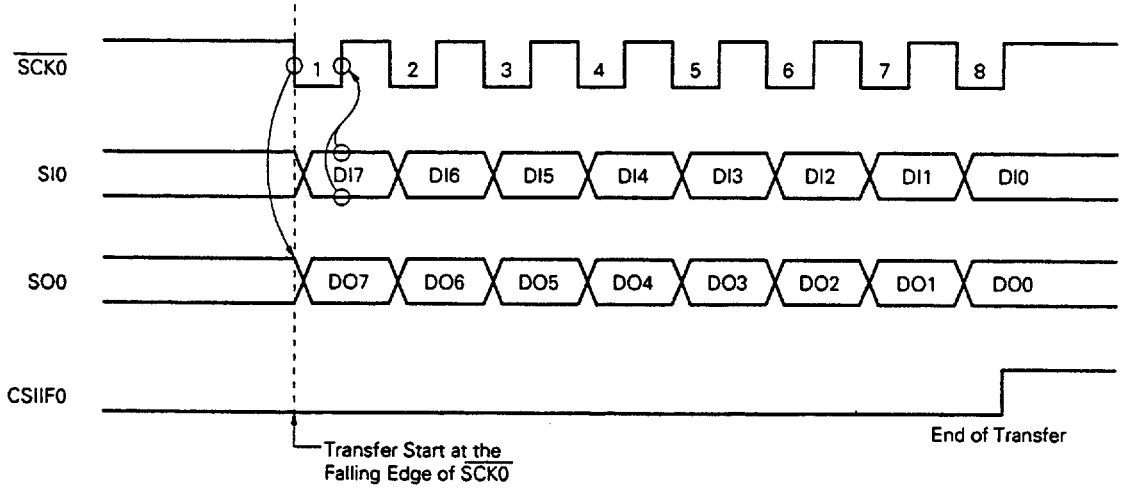
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization of the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of the serial clock ($\overline{SCK0}$). The transmitted data is held in the SO0 latch and is output from the SO0 pin. The received data input to the SIO pin is latched in SIO0 at the rising edge of $\overline{SCK0}$.

Upon termination of 8-bit transfer, SIO0 operation stops automatically and the interrupt request flag (CSIIF0) is set.

Fig. 13-7 3-Wire Serial I/O Mode Timings



The SO0 pin serves for CMOS output and generates the SO0 latch status. Thus, the SO0 pin output status can be manipulated by setting the RELT and CMDT bits.

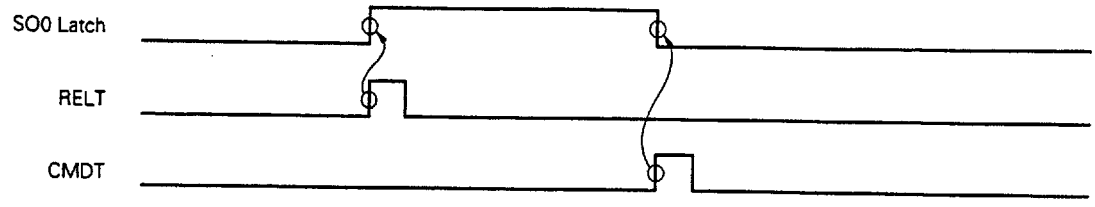
However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **13.4.5 $\overline{SCK0}$ pin output manipulation**).

(3) Various signals

Fig. 13-8 shows RELT and CMDT operations.

Fig. 13-8 RELT and CMDT Operations



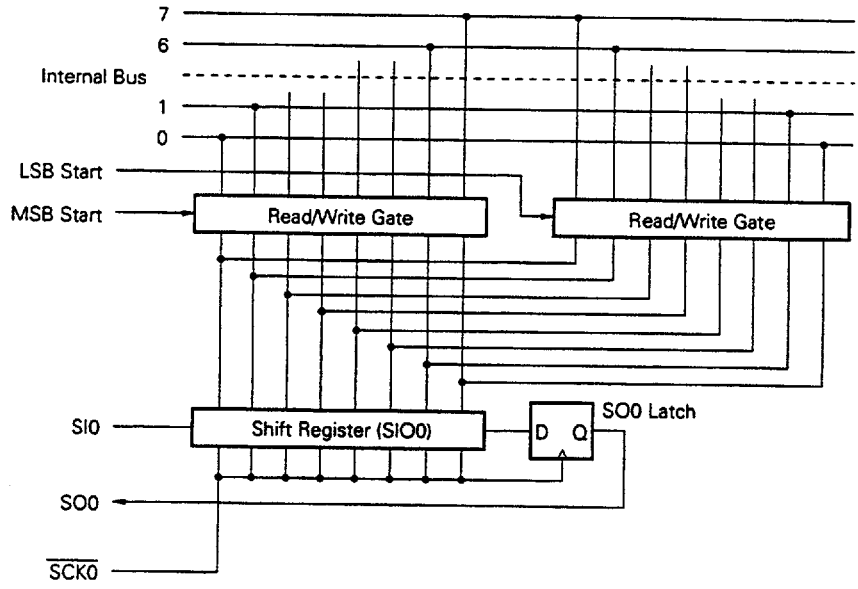
(4) MSB/LSB switching as the start bit

The 3-wire serial I/O mode enables to select transfer to start at MSB or LSB.

Fig. 13-9 shows the configuration of the serial I/O shift register 0 (SIO0) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 2 (CSIM02) of the serial operating mode register 0 (CSIM0).

Fig. 13-9 Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO0. The SIO0 shift order remains unchanged.

Thus, switch the MSB/LSB start bit before writing data to the shift register.

(5) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{SCK0}$ is a high level after 8-bit serial transfer.

Caution If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

13.4.3 SBI mode operation

SBI (Serial Bus Interface) is a high-speed serial interface in compliance with the NEC serial bus format.

SBI has a format with the bus configuration function added to the clocked serial I/O method so that it can carry out communication with two or more devices with two signal conductors on the single-master high-speed serial bus. Thus, when making up a serial bus with two or more microcomputers and peripheral ICs, the number of ports to be used and the number of wires on the board can be decreased.

The master device can output to the serial data bus of the slave device "addresses" for selection of the serial communication target device, "commands" to instruct the target device and actual "data". The slave device can identify the received data into "address", "command" or "data", by hardware. This function enables the application program serial interface (channel 0) control portions to be simplified.

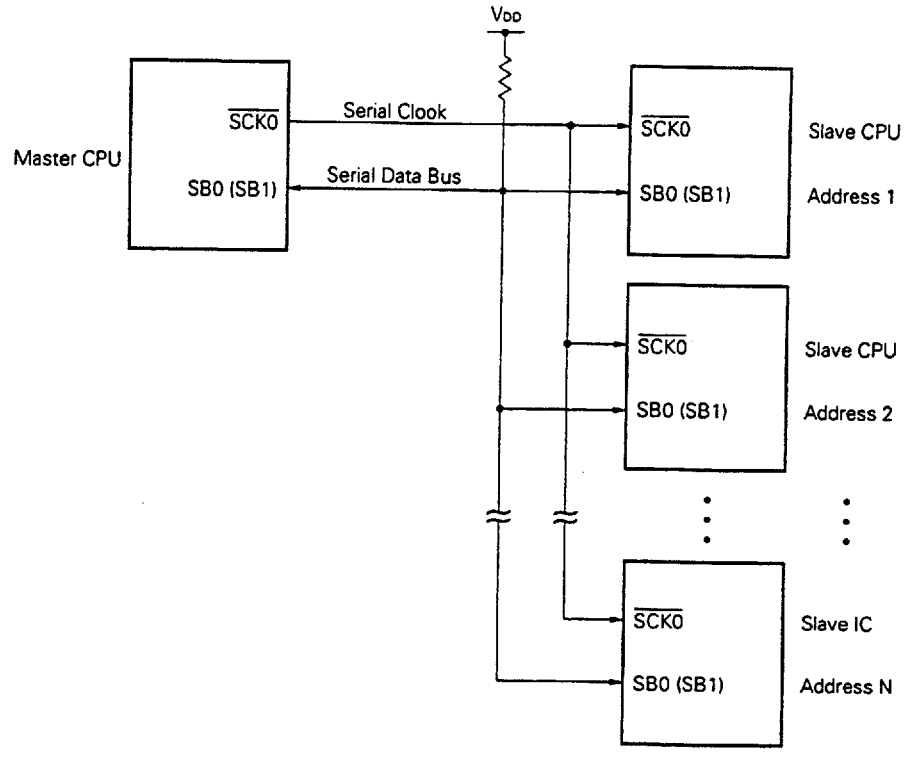
The SBI function is incorporated into various devices including 75X-series devices and 78K-series 8-bit and 16-bit single-chip microcomputers.

Fig. 13-10 shows a serial bus configuration example when a CPU having a serial interface compliant with SBI and peripheral ICs are used.

In SBI, the SB0 (SB1) serial data bus pin serves for open-drain output and so the serial data bus line is in wired-OR state. A pull-up resistor is necessary for the serial data bus line.

When using the SBI mode, refer to **(d)** under **(10) SBI mode precautions** later in the text.

Fig. 13-10 Example of Serial Bus Configuration with SBI



Caution When replacing the master CPU/slave CPU, a pull-up resistor is necessary for the serial clock line ($\overline{SCK0}$) as well because serial clock line ($\overline{SCK0}$) input/output switching is carried out asynchronously between the master and slave CPUs.

(1) SBI functions

In the conventional serial I/O method, when a serial bus is constructed by connecting two or more devices, many ports and wiring are necessary to distinguish chip select signals and command/data and to judge the busy state because only the data transfer function is available. If these operations are to be controlled by software, the software must be heavily loaded.

In SBI, a serial bus can be constructed with two signal conductors of serial clock $\overline{SCK0}$ and serial data bus SB0 (SB1). Thus, SBI is effective to decrease the number of microcomputer ports and that of wirings and routings on the board.

The SBI functions are described below.

(a) Address/command/data identify function

Serial data is distinguished into addresses, commands and data.

(b) Chip select function by address transmission

The master executes slave chip selection by address transmission.

(c) Wake-up function

The slave can easily judge address reception (chip select judgment) with the wake-up function (which can be set/reset by software).

When the wake-up function is set, the interrupt request signal (CSIIF0) is generated upon reception of a match address. Thus, when communication is executed with two or more devices, the CPU except the selected slave devices can operate regardless of serial communication.

(d) Acknowledge ($\overline{ACK}$) control function

The acknowledge signal to check serial data reception is controlled.

(e) Busy signal ($\overline{BUSY}$) control function

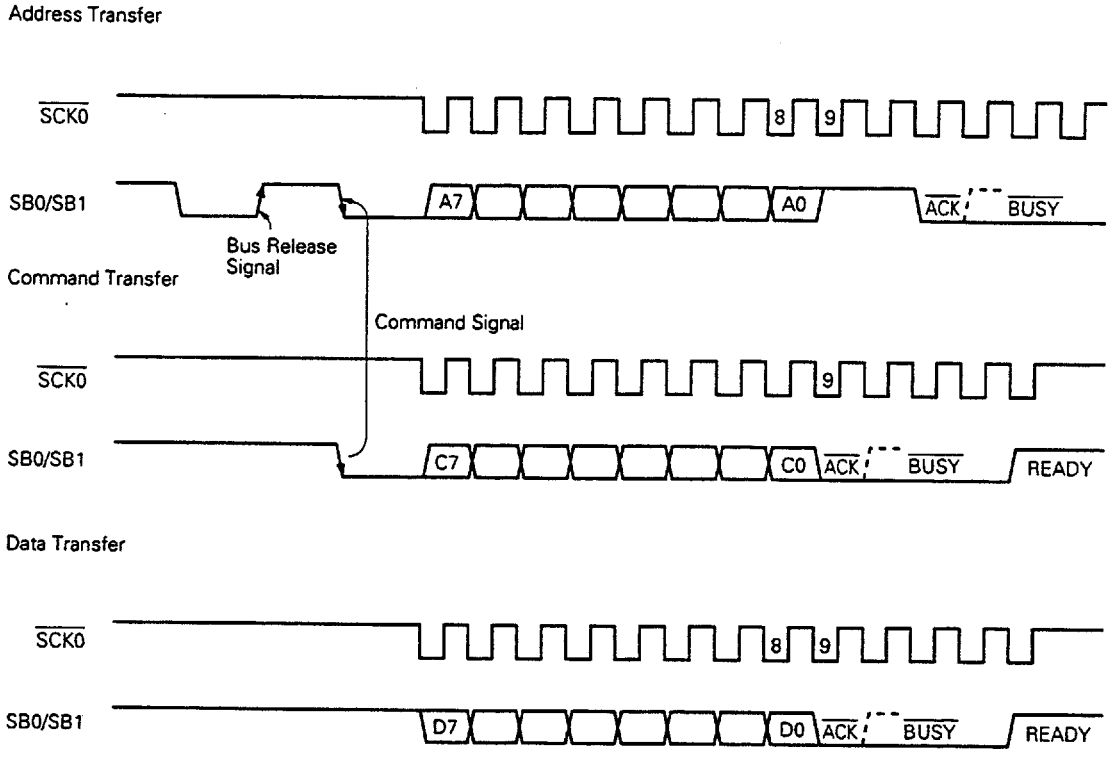
The busy signal to report the slave busy state is controlled.

(2) SBI definition

The SBI serial data format and the signals to be used are defined as follows.

Serial data to be transferred with SBI is distinguished into three types, "address", "command" and "data". Fig. 13-11 shows the address, command and data transfer timings.

Fig. 13-11 SBI Transfer Timings



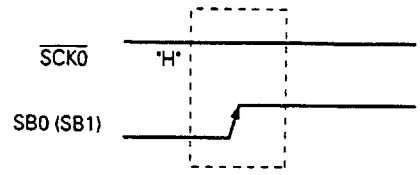
The bus release signal and the command signal are output by the master device. $\overline{BUSY}$ is output by the slave signal. $\overline{ACK}$ can be output by either the masteor slave device (normally, the 8-bit data receiver outputs).

Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{BUSY}$ reset.

(a) Bus release signal (REL)

The bus release signal is a signal with the SB0 (SB1) line which has changed from the low level to the high level when the $\overline{SCK0}$ line is at the high level (without serial clock output).
This signal is output by the master device.

Fig. 13-12 Bus Release Signal

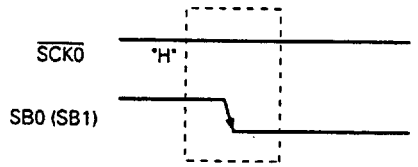


The bus release signal indicates that the master device is going to transmit an address to the slave device.
The slave device incorporates hardware to detect the bus release signal.

(b) Command signal (CMD)

The command signal is a signal with the SB0 (SB1) line which has changed from the high level to the low level when the $\overline{SCK0}$ line is at the high level (without serial clock output).
This signal is output by the master device.

Fig. 13-13 Command Signal

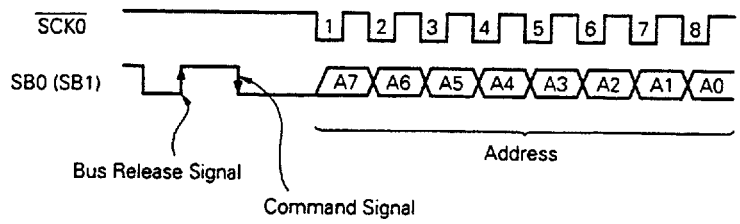


The slave device incorporates hardware to detect the command signal.

(c) Address

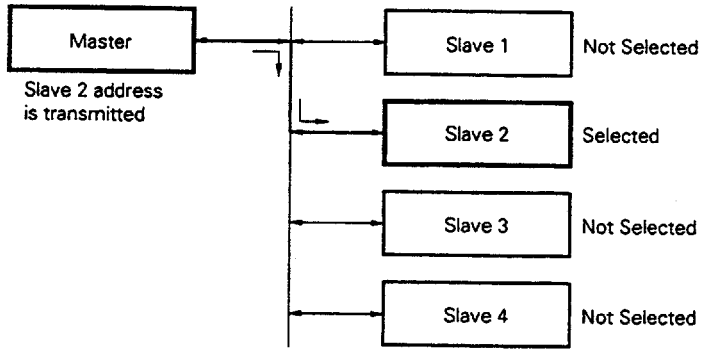
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Fig. 13-14 Addresses



8-bit data following bus release and command signals is defined as an "address".
In the slave device, this condition is detected by hardware and whether or not 8-bit data matches the own specification number (slave address) is checked by hardware. If the 8-bit data matches the slave address, the slave device has been selected. After that, communication with the master device continues until a release instruction is received from the master device.

Fig. 13-15 Slave Selection with Address



(d) Command and data

The master device transmits commands to and transmits and receives data to and from the slave device selected by address transmission.

Fig. 13-16 Commands

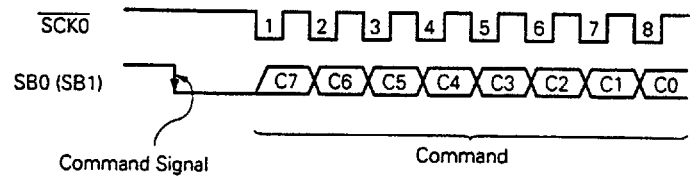
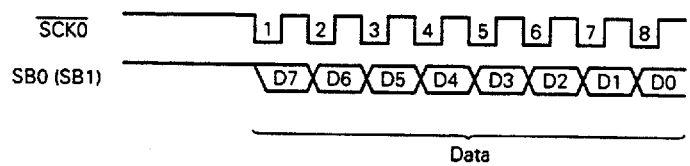


Fig. 13-17 Data



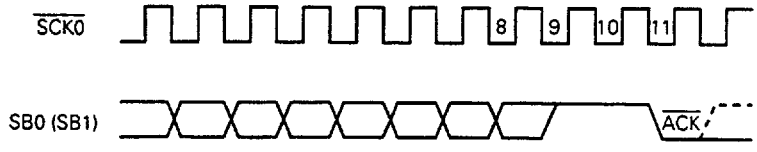
8-bit data following a command signal is defined as a "command". 8-bit data without command signal is defined as "data". Command and data operation procedures can be determined arbitrarily according to communication specifications.

(e) Acknowledge signal ($\overline{ACK}$)

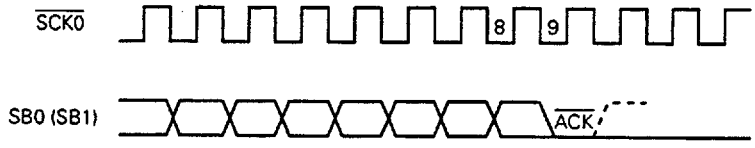
The acknowledge signal is used to check serial data reception between transmitter and receiver.

Fig. 13-18 Acknowledge Signal

[When output in synchronization with 11th clock $\overline{SCK0}$]



[When output in synchronization with 9th clock $\overline{SCK0}$]



The acknowledge signal is one-shot pulse to be generated at the falling edge of $\overline{SCK0}$ after 8-bit data transfer. It can be positioned anywhere and can be synchronized with any clock $\overline{SCK0}$.

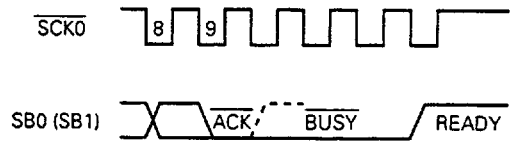
After 8-bit data transmission, the transmitter checks whether the receiver has returned the acknowledge signal. If the acknowledge signal is not returned for the preset period of time after data transmission, it can be judged that data reception has not been carried out correctly.

(f) Busy signal ($\overline{BUSY}$) and ready signal (READY)

The BUSY signal is intended to report to the master device that the slave device is preparing for data transmission/reception.

The READY signal is intended to report to the master device that the slave device is ready for data transmission/reception.

Fig. 13-19 BUSY and READY Signals



In SBI, the slave device notifies the master device of the busy state by setting SB0 (SB1) to the low level. The BUSY signal is output in the wake of the acknowledge signal output from the master or slave signal. It is set/reset at the falling edge of $\overline{SCK0}$. When the BUSY signal is reset, the master device automatically terminates the output of $\overline{SCK0}$ serial clock.

When the BUSY signal is reset and the READY signal is set, the master device can start the next transfer.

(3) Register set

The SBI mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC) and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{RESET}$ input sets CSIM0 to 00H.

The shaded area is used in the SBI mode.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W Note 1

R/W

CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
0	×	Clock externally input to $\overline{SCK0}$ pin
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/P25 Pin Function	SO0/P26 Pin Function	$\overline{SCK0}$/P27 Pin Function
0	×	3-wire serial I/O mode (Refer to **13.4.2 3-wire serial I/O mode operation**.)											
1	0	0	Note 2 ×	Note 2 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1/P26 (N-ch open-drain I/O)	$\overline{SCK0}$ (CMOS I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	
1	1	2-wire serial I/O mode (Refer to **13.4.4 2-wire serial I/O mode operation**.)											

R/W

WUP	Wake-up Function Control
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register data in SBI mode

R

COI	Slave Address Compare Result Flag Note 3
0	Slave address register not equal to serial I/O shift register 0 data
1	Slave address register equal to serial I/O shift register 0 data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation disabled
1	Operation enabled

Notes **1.** Bit 6 (COI) is a Read-Only bit.

2. Can be used freely as port function.

3. When CSIEO = 0, COI becomes 0.

Remark ×: Don't care

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulate instruction.

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

The shaded area is used in the SBI mode.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W[Note]

R/W

RELT	Used for bus release signal output. When RELT = 1, SO latch is set to 1. After SO latch setting, this bit is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W

CMDT	Used for command signal output. When CMDT = 1, SO latch is set to (1). After SO latch setting, automatically cleared to (0). Also cleared to 0 when CSIE0 = 0.

R

RELD	Bus Release Detection	
Clear Conditions (RELD = 0)		Set Conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When bus release signal (REL) is detected

R

CMDD	Command Detection	
Clear Conditions (CMDD = 0)		Set Conditions (CMDD = 1)
• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When command signal (CMD) is detected

R/W

ACKT	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction to be set to (1) and, after acknowledge signal output, automatically cleared to (0). Also cleared to (0) upon start of serial interface transfer or when CSIE0 = 0.

Note Bits 2, 3, and 6 (RELD, CMDD, ACKD) are Read-Only bits.

R/W

ACKE	Acknowledge Signal Output Control	
0	Acknowledge signal automatic output disable (output with ACKT enable)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with the falling edge of 9th $\overline{SCK0}$ clock (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{SCK0}$ clock just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

R

ACKD	Acknowledge Detection	
Clear Conditions (ACKD = 0)		Set Conditions (ACKD = 1)
• When $\overline{SCK0}$ falls after busy mode has been released after execution of transfer start instruction • When CSIE0 = 0 • When $\overline{RESET}$ input is applied		• When acknowledge signal ($\overline{ACK}$) is detected at the rising edge of $\overline{SCK0}$ clock after completion of transfer

R/W

Note BSYE	Synchronizing Busy Signal Output Control
0	Busy signal output is disabled at the falling edge of $\overline{SCK0}$ clock just after execution of the instruction to be cleared to 0.
1	Busy signal is output at the falling edge of $\overline{SCK0}$ clock following the acknowledge signal.

Note The serial interface transfer start or the address signal reception release a busy mode. However, the BSYE flag is not cleared to 0.

(c) Interrupt timing specify register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulate instruction.

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

The shaded area is used in the SBI mode.

Symbol	7	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W[Note 1]

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Source Selection[Note 2]
0	CSIIF0 is set upon termination of serial interface channel 0 transfer.
1	CSIIF0 is set upon termination of serial interface channel 0 transfer or upon bus release detection.

R

CLD	$\overline{\text{SCK0}}$ Pin Level[Note 3]
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is a Read-Only bit.

2. When using wake-up function, set 0 in SIC.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register

(4) Various signals

Figs. 13-20 to 13-25 show various signals and flag operations in SBI. Table 13-3 lists various signals in SBI.

Fig. 13-20 RELT, CMDT, RELD and CMDD Operations (Master)

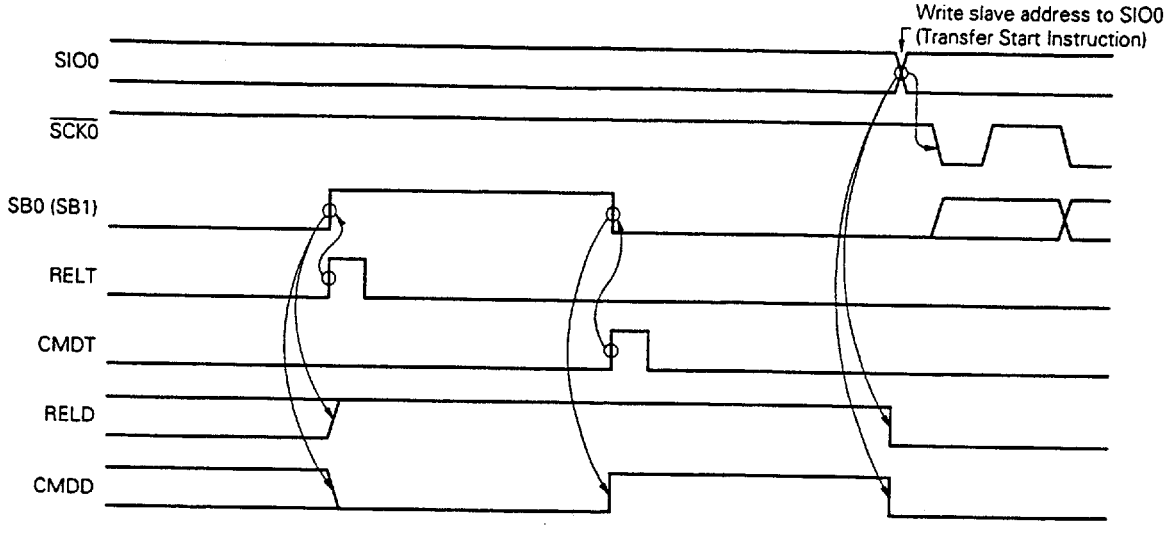


Fig. 13-21 RELD and CMDD Operations (Slave) ★

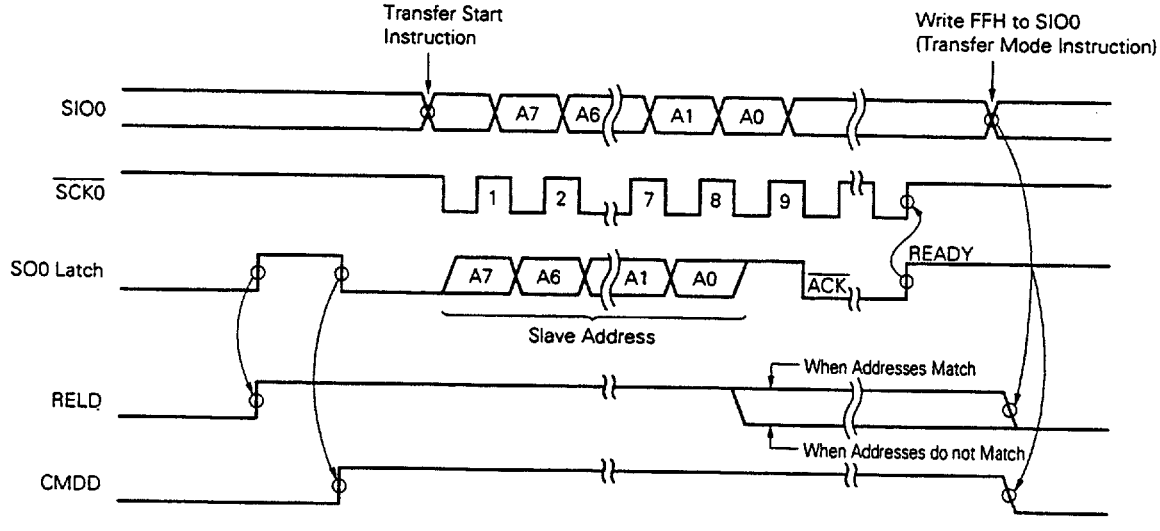
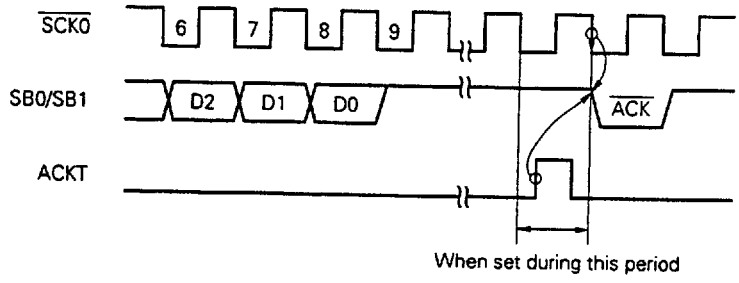


Fig. 13-22 ACKT Operation

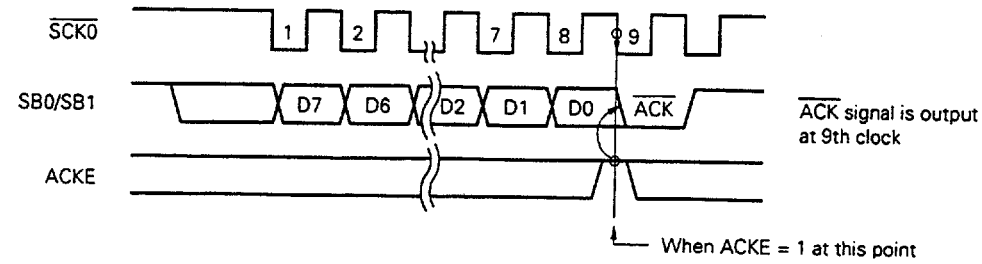


ACK signal is output for a period of one clock just after setting.

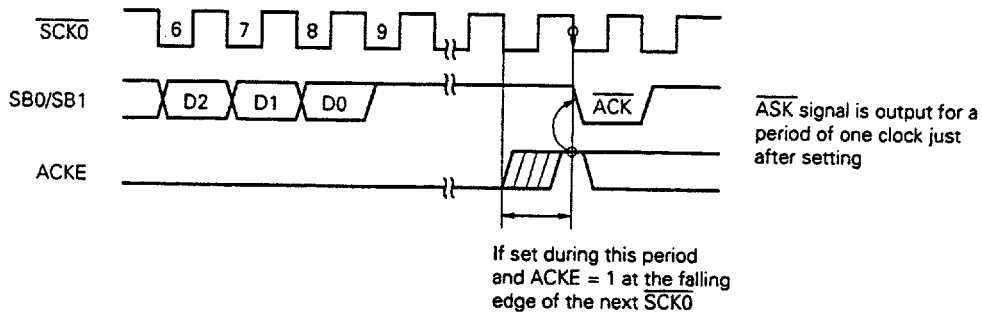
Caution Do not set ACKT before termination of transfer.

Fig. 13-23 ACKE Operations

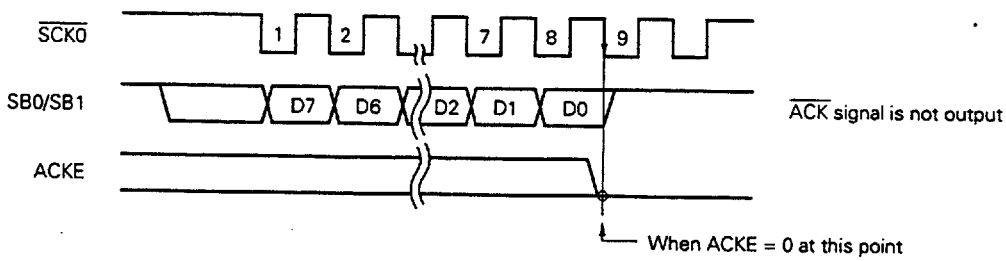
(a) When ACKE = 1 before completion of transfer



(b) When set after completion of transfer



(c) When ACKE = 0 upon completion of transfer



(d) When ACKE = 1 period is short

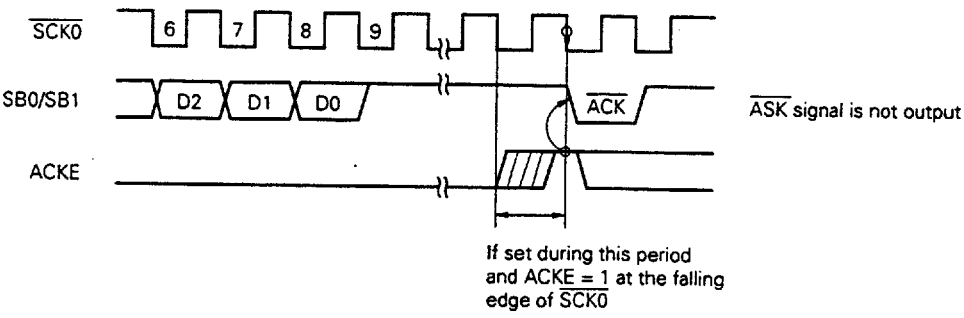
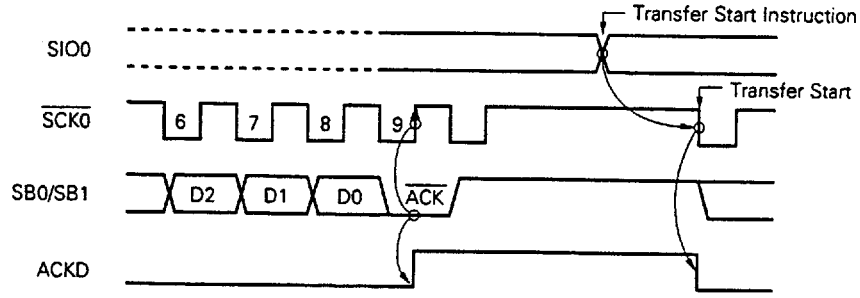
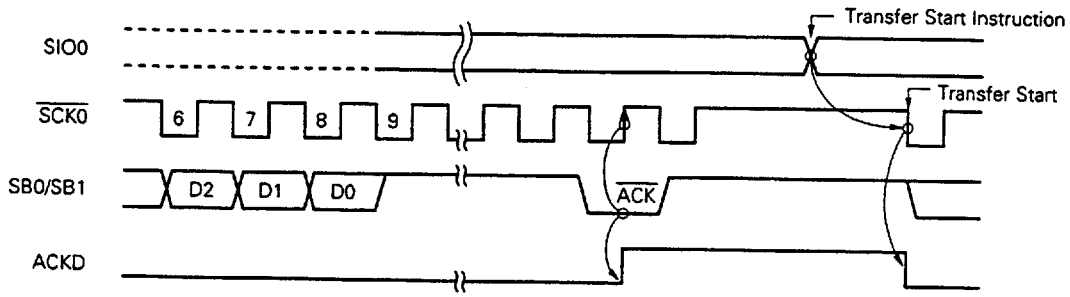


Fig. 13-24 ACKD Operations

(a) When $\overline{ACK}$ signal is output at 9th clock of $\overline{SCK0}$



(b) When $\overline{ACK}$ signal is output after 9th clock of $\overline{SCK0}$



(c) Clear timing when transfer is instructed in BUSY

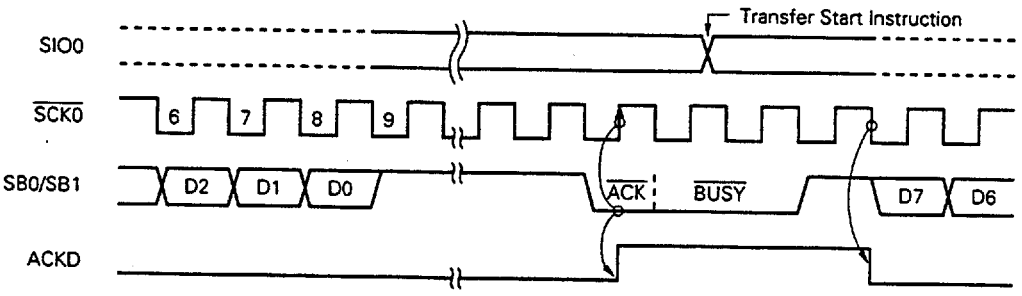


Fig. 13-25 BSYE Operation

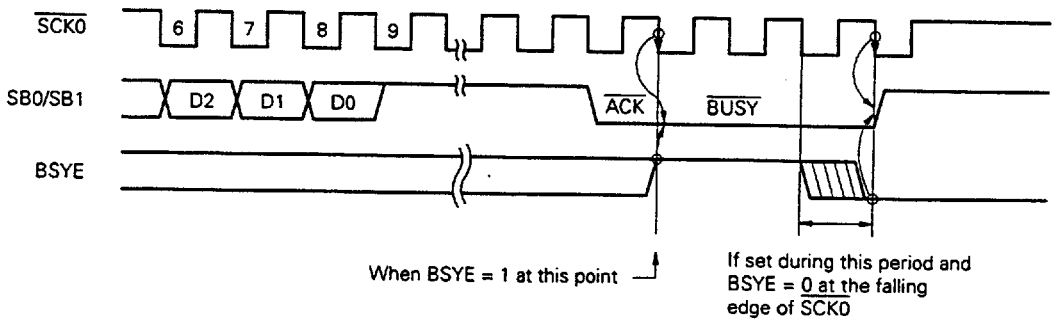
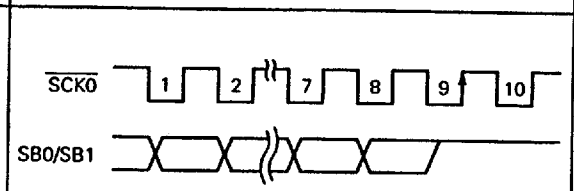
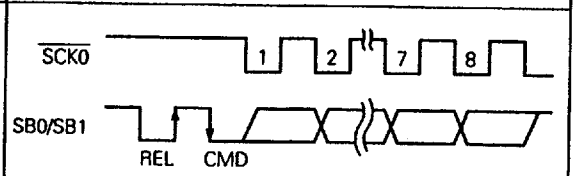
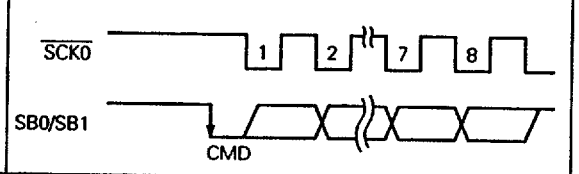
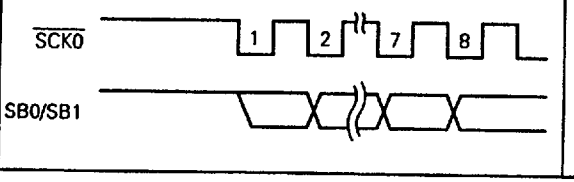


Table 13-3 Various Signals in SBI Mode (1/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0/SB1 rising edge when $\overline{SCK0}$ = 1	$\overline{SCK0}$ "H" SB0/SB1	• RELT set	• Sets RELD • Cleared CMDD	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0/SB1 falling edge when $\overline{SCK0}$ = 1	$\overline{SCK0}$ "H" SB0/SB1	• Set CMDT	• Sets CMDD	i) Transmit data is an address after REL signal output. ii) REL signal is not output and transmit data is an address.
Acknowledge signal ($\overline{ACK}$)	Master/ slave	Low-level signal to be output to SB0/SB1 during one-clock period of $\overline{SCK0}$ after completion of serial reception	**[Synchronous BUSY output]** $\overline{SCK0}$ 9 $\overline{ACK}$ $\overline{BUSY}$ SB0/SB1 D0 READY SB0/SB1 $\overline{ACK}$ $\overline{BUSY}$ D0 READY	① ACKE = 1 ② ACKT set	• Sets ACKD	Completion of reception
Busy signal ($\overline{BUSY}$)	Slave	[Synchronous BUSY signal] Low-level signal to be output to SB0/SB1 following Acknowledge signal		• BSYE = 1	—	Serial reception disable because of processing
Ready signal (READY)	Slave	High-level signal to be output to SB0/SB1 before serial transfer start and after completion of serial transfer		① BSYE = 0 ② Execution of instruction for data write to SIO0 (transfer start instruction ③ Address signal reception	—	Serial reception enabled

Table 13-3 Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Serial clock ($\overline{SCK0}$)	Master	Synchronous clock to output address/command/ data, $\overline{ACK}$ signal, synchro- nous $\overline{BUSY}$ signal, etc. Address/command/data are transferred with the first eight synchronous clocks.		When CSIE0=1, execution of instruction for data write to SIO0 (serial transfer start instruction) Note 2	Sets CSIIF0 (rising edge of 9th clock of $\overline{SCK0}$) Note 1	Timing of signal output to serial data bus
Address (A7-A0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK0}$ after output of REL and CMD signals				Address value of slave device on the serial bus
Command (C7-C0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK0}$ after output of only CMD signal without REL signal output				Instructions and messages to the slave device
Data (D7-D0)	Master/ slave	8-bit data to be transferred in synchronization with $\overline{SCK0}$ without output of REL and CMD signals				Numeric values to be processed with slave or master device

Notes 1. When WUP = 0, CSIIF0 is set at the rising edge of the 9th clock of $\overline{SCK0}$.
When WUP = 1, an address is received. Only when the address matches the slave address register (SVA) value, CSIIF0 is set.

2. In $\overline{BUSY}$ state, transfer start after the READY state is set.

(5) Pin configuration

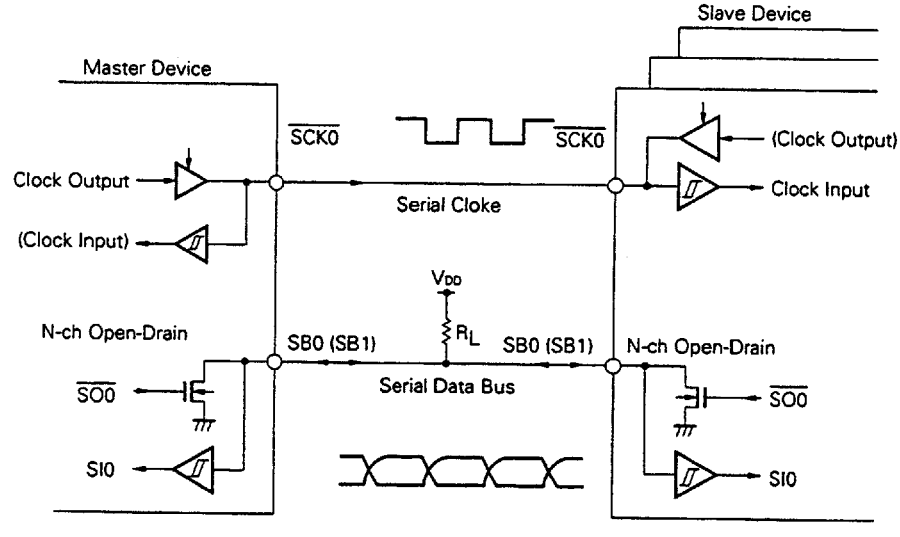
The serial clock pin ($\overline{SCK0}$) and serial data bus pin SB0 (SB1) have the following configurations.

(a) $\overline{SCK0}$: Serial clock input/output pin
  ① Master: CMOS and push-pull output
  ② Slave: Schmitt input

(b) SB0 (SB1) : Serial data input/output dual- function pin
  Both master and slave devices have an N-ch open-drain output and a Schmitt input.

Because the serial data bus line has an N-ch open-drain output, an external pull-up resistor is necessary.

Fig. 13-26 Pin Configuration



Caution Because the N-ch open-drain must be turned OFF for data reception, write FFH to SIO0 in advance. The N-ch open-drain can be turned OFF during transfer. However, when wake-up function specify bit (WUP) = 1, the N-ch transistor is always turned OFF. Thus, it is not necessary to write FFH to SIO0.

(6) Address match detection method

In the SBI mode, a particular slave device is selected by address communication from the master device and communication is started.

Address match detection is executed by hardware. With the slave address register (SVA), CSIIF0 is set in the wake-up state (WUP = 1) only when the address transmitted from the master device matches the value set to SVA.

Cautions 1. Slave selection/non-selection is detected by match detection of the slave address received after bus release (RELD = 1).
For this match detection, match interrupt (INTCSI0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

2. When detecting selection/non-selection without the use of interrupt with WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address match detection method.

(7) Error detection

In the SBI mode, the serial bus SB0/SB1 status being transmitted is fetched into the destination device, that is, the serial I/O shift register 0 (SIO0). Thus, transmit errors can be detected in the following way.

(a) Method of comparing SIO0 data before transmission to that after transmission.

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Method of using the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If "1", normal transmission is judged to have been carried out. If "0", a transmit error is judged to have occurred.

(8) Communication operation

In the SBI mode, the master device selects normally one slave device as communication target from among two or more devices by outputting an "address" to the serial bus.

After the communication target device has been determined, commands and data are transmitted/received and serial communication is realized between the master and slave devices.

Figs. 13-27 to 13-30 show data communication timing charts.

Shift operation of the shift register is carried out at the falling edge of serial clock ($\overline{SCK0}$). Transmit data is latched into the SO0 latch and is output with MSB set as the first bit from the SB0/P25 or SB1/P26 pin.

Receive data input to the SB0 (or SB1) input at the rising edge of $\overline{SCK0}$ is latched into the shift register.

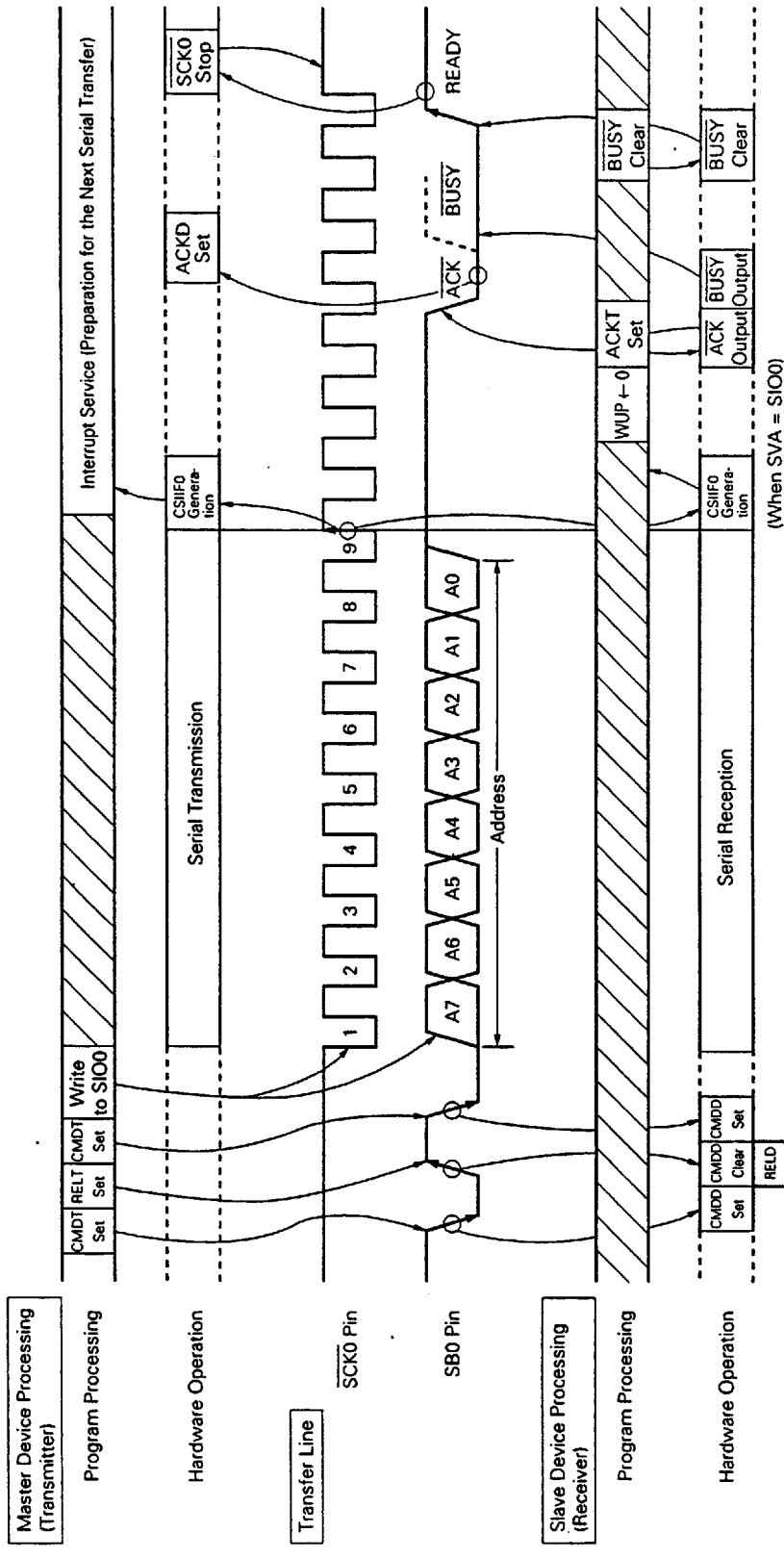


Fig. 13-27 Address Transmission from Master Device to Slave Device (WUP = 1)

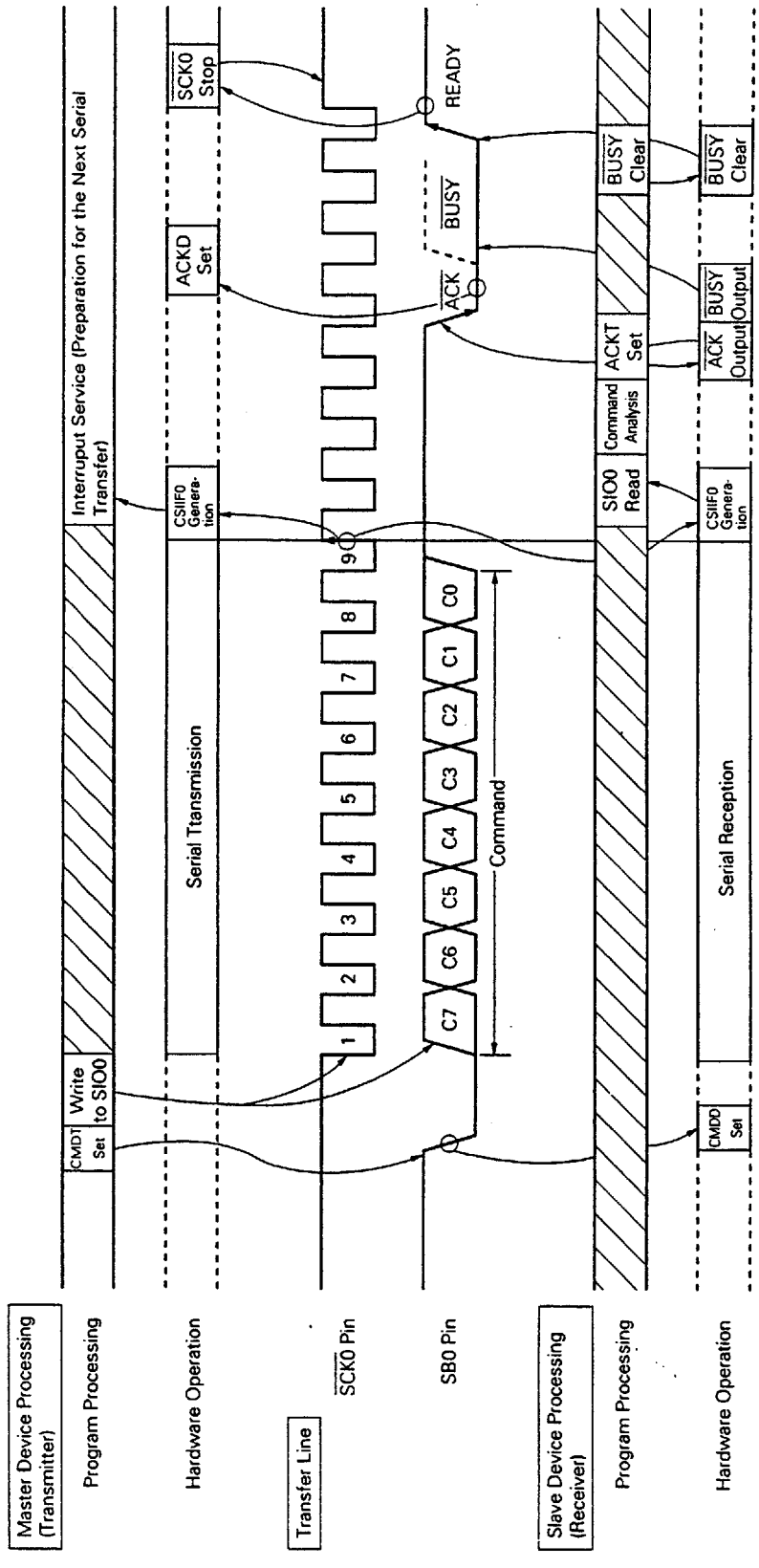


Fig. 13-28 Command Transmission from Master Device to Slave Device

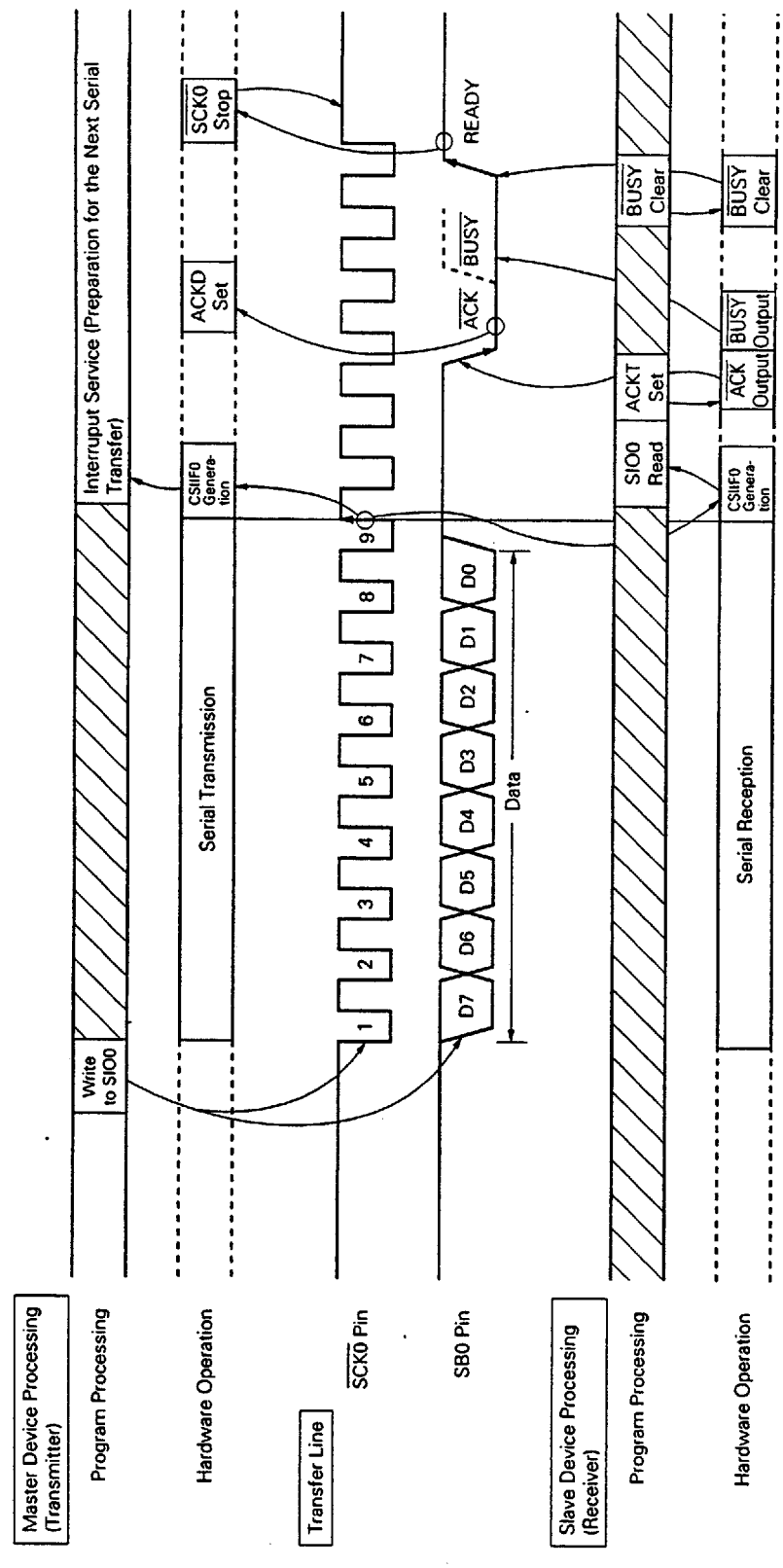


Fig. 13-29 Data Transmission from Master Device to Slave Device

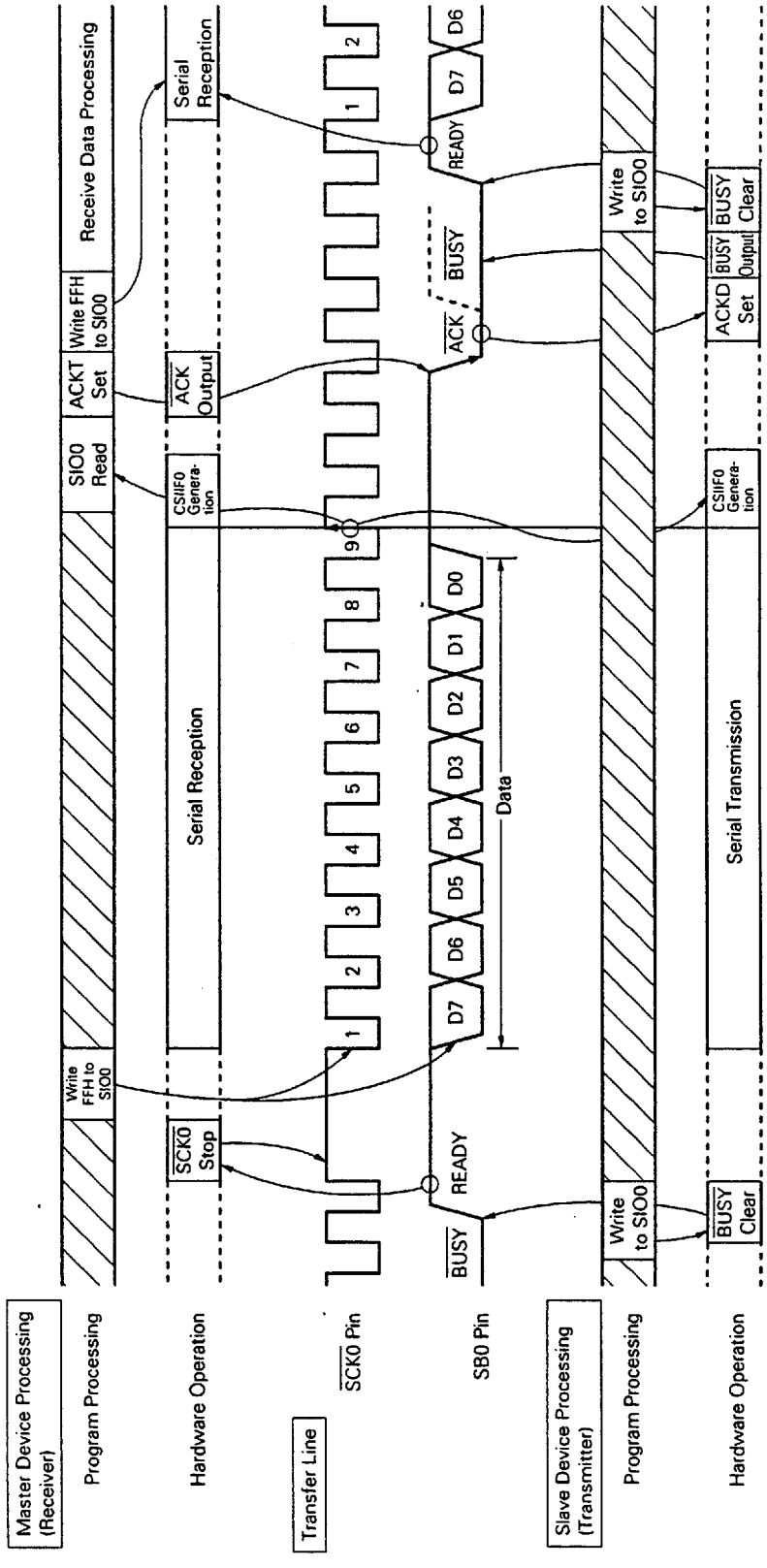


Fig. 13-30 Data Transmission from Slave Device to Master Device

(9) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{SCK0}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain must be turned OFF for data reception, write FFH to SIO0 in advance. However, when make-up function specify bit (WUP) = 1, the N-ch transistor is always turned OFF. Thus, it is not necessary to write FFH to SIO0.

3. If data is written to SIO0 when the slave is busy, the data is not lost. When the busy state is cleared and SB0 (or SB1) input is set to the high level (READY) state, transfer starts.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

(10) SBI mode precautions

(a) Slave selection/non-selection is detected by match detection of the slave address received after bus release (RELD = 1).
For this match detection, match interrupt (CSIIF0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

(b) When detecting selection/non-selection without the use of interrupt with WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address match detection method.

(c) If WUP is set to 1 during $\overline{BUSY}$ signal output, $\overline{BUSY}$ is not cleared. In SBI, the $\overline{BUSY}$ signal continues to be output after $\overline{BUSY}$ clear instruction generation to the falling edge of the next serial clock ($\overline{SCK0}$). Before setting WUP to 1, be sure to clear $\overline{BUSY}$ and then check that the SB0 (SB1) has become high-level.

(d) For pins which are to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after $\overline{\text{RESET}}$ input.

① Set the P25 and P26 output latches to 1.
② Set bit 0 (RELT) of the serial bus control register to 1.
③ Reset the P25 and P26 output latches from 1 to 0.

(e) When a device is set in the master mode, whether or not a slave is in the state of busy should be determined by the following procedure.

① An acknowledge signal ($\overline{\text{ACK}}$) or an interrupt signal generation is detected.
② Set port mode register PM25 (or PM26) of SB0/P25 (or SB1/P26) pin to the input mode.
③ Read the pin status (when a high level, the pin status is ready).

After detection of the ready status, set 0 in the port mode register and return to the output mode.

13.4.4 2-wire serial I/O mode operation

The 2-wire serial I/O mode can cope with any communication format by program.
Communication is basically carried out with two lines of serial clock ($\overline{\text{SCK0}}$) and serial data input/output (SB0 or SB1).

(1) Register set

The 2-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC) and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulate instruction.
$\overline{\text{RESET}}$ input sets CSIM0 to 00H.
The shaded area is used in the 2-wire serial I/O mode.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W[Note 1]

R/W

CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
0	×	Clock externally input to $\overline{SCK0}$ pin
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start bit	SI0/P25 Pin Function	SO0/P26 Pin Function	$\overline{SCK0}$/P27 Pin Function
0	×	3-wire serial I/O mode (Refer to **13.4.2 3-wire serial I/O mode operation**.)											
1	0	SBI mode (Refer to **13.4.3 SBI mode operation**.)											
1	1	0	×[Note 2]	×[Note 2]	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS I/O)	SB1/P26 (N-ch open-drain I/O)	$\overline{SCK0}$ (N-ch open-drain I/O)
		1	0	0	×[Note 2]	×[Note 2]	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

R/W

WUP	Wake-up Function Control
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register data in SBI mode

R

COI	Slave Address Compare Result Flag[Note 3]
0	Slave address register not equal to serial I/O shift register 0 data
1	Slave address register equal to serial I/O shift register 0 data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation disable
1	Operation enable

Notes 1. Bit 6 (COI) is a Read-Only bit.

2. Can be used freely as port function.

3. When CSIE0 = 0, COI becomes 0.

Remark × : Don't care

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulate instruction.

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

The shaded area is used in the 2-wire serial I/O mode.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W		
R/W	RELT	When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W		
R/W	CMDT	When CMDT = 1, SO latch is set to 0. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

(c) Interrupt timing specify register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulate instruction.

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

The shaded area is used in the 2-wire serial I/O mode.

Symbol	7	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W[Note 1]

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Source Selection
0	CSIIF0 is set upon termination of serial interface channel 0 transfer.
1	CSIIF0 is set upon bus release detection.

R

CLD	$\overline{\text{SCK0}}$ pin Level[Note 2]
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is a Read-Only bit.

2. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register

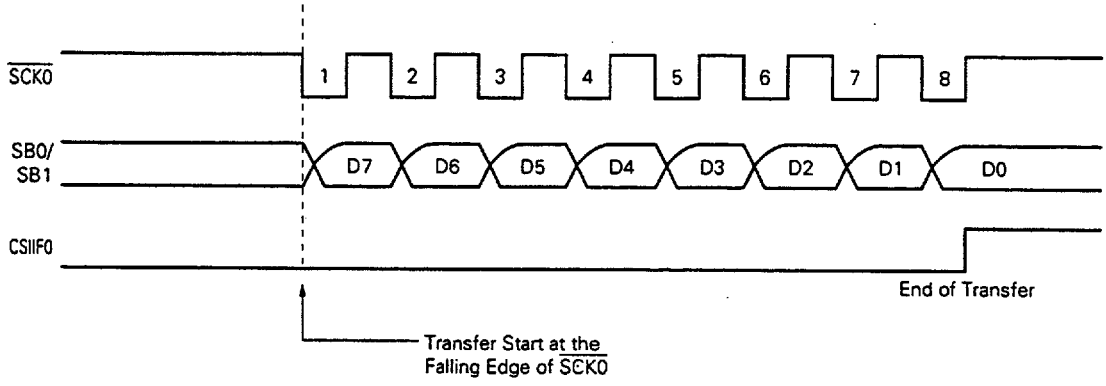
(2) Communication operation

The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of the serial clock ($\overline{SCK0}$). The transmit data is held in the SO0 latch and is output from the SB0/P25 (or SB1/P26) pin with MSB set at start. The receive data input from the SB0 (or SB1) pin is latched into the shift register at the rising edge of $\overline{SCK0}$.

Upon termination of 8-bit transfer, the shift register operation stops automatically and the interrupt request flag (CSIIF0) is set.

Fig. 13-31 2-Wire Serial I/O Mode Timings



The SB0 (or SB1) pin specified for the serial data bus serves for N-ch open-drain input/output and thus it must be externally pulled up. Because it is necessary to turn OFF the N-ch transistor, write FFH to SIO0 in advance.

The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting the RELT and CMDT bits.

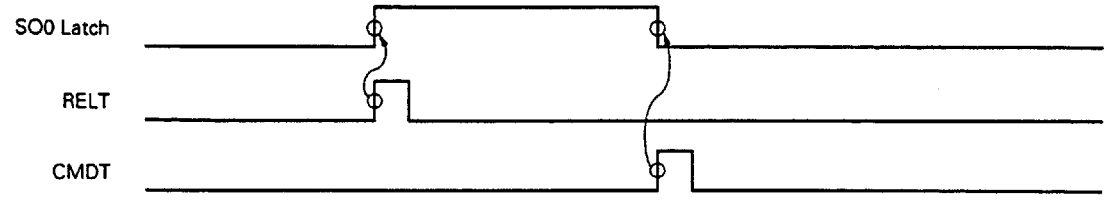
However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **13.4.5 $\overline{SCK0}$ pin output manipulation**).

(3) Various signals

Fig. 13-32 shows RELT and CMDT operations.

Fig. 13-32 RELT and CMDT Operations



(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{SCK0}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain must be turned OFF for data reception, write FFH to SIO0 in advance.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0/SB1 status being transmitted is fetched into the destination device, that is, SIO0. Thus, transmit error can be detected in the following way.

(a) Method of comparing SIO0 data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Method of using the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If "1", normal transmission is judged to have been carried out. If "0", a transmit error is judged to have occurred.

13.4.5 $\overline{SCK0}$ pin output manipulation

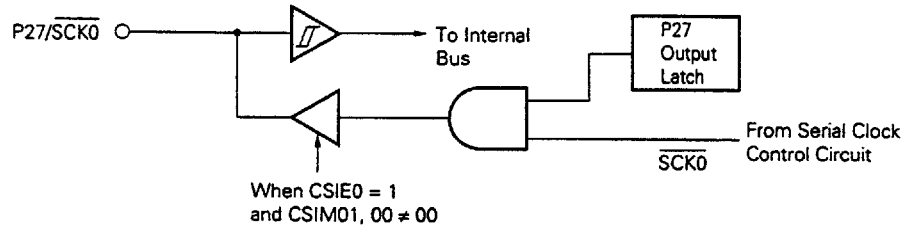
Because the SCK0/P27 pin incorporates an output latch, static output is also possible by software in addition to normal serial clock output.

P27 output latch manipulation enables any number of $\overline{SCK0}$ to be set by software (SI0/SB0 and SO0/SB1 pin to be controlled with the RELT and CMDT bits of SBIC).

$\overline{SCK0}$/P27 pin output manipulating procedure is described below.

① Set the serial operating mode register 0 (CSIM0) ($\overline{SCK0}$ pin enabled for serial operation in the output mode). $\overline{SCK0}$ = 1 with serial transfer suspended.
② Manipulate the P27 output latch with a bit manipulate instruction.

Fig. 13-33 $\overline{SCK0}$/P27 Pin Configuration



CHAPTER 14 SERIAL INTERFACE CHANNEL 1

14.1 Serial Interface Channel 1 Functions

Serial interface channel 1 employs the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not carried out. Power consumption can be reduced.

(2) 3-wire serial I/O mode

This mode is used for 8-bit data transfer using three lines, each for serial clock ($\overline{SCK1}$), serial output (SO1) and serial input (SI1).

The 3-wire serial I/O mode enables simultaneous transmission/reception and so decreases the data transfer processing time.

Since the start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB, connection is enabled with either start bit device.

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface as is the case with the 75X, 78K and 17K series.

14.2 Serial Interface Channel 1 Configuration

Serial interface channel 1 consists of the following hardware.

Table 14-1 Serial Interface Channel 1 Configuration

Item	Configuration
Register	Serial I/O shift register 1 (SIO1)
Control register	Timer clock select register 3 (TCL3) Serial operation mode register 1 (CSIM1) Port mode register 2 (PM2)

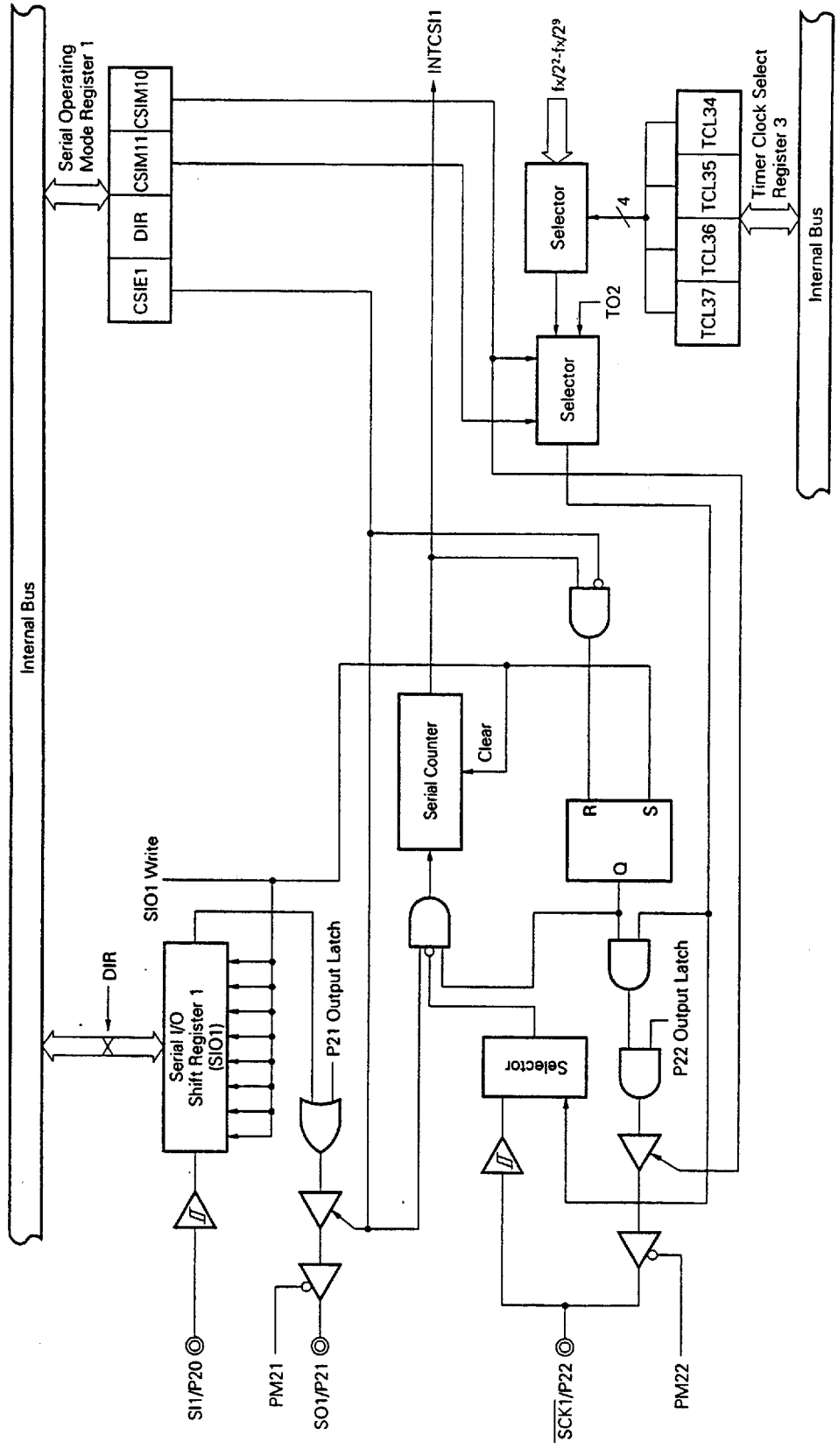


Fig. 14-1 Serial Interface Channel 1 Block Diagram

(1) Serial I/O shift register 1 (SIO1)

This is an 8-bit register to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO1 is set with an 8-bit memory manipulate instruction.

When bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is 1, writing data to SIO1 status serial operation. In transmission, data written to SIO1 is output to the serial output (SO1). In reception, data is read from the serial input (SI1) to SIO1.

$\overline{\text{RESET}}$ input sets SIO1 undefined.

(2) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

14.3 Serial Interface Channel 1 Control Registers

The following four types of registers are used to control serial interface channel 1.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 1 (CSIM1)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 1.

TCL3 is set with an 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Remark Besides setting the serial clock of serial interface channel 1, TCL3 sets the serial clock of serial interface channel 0.

Fig. 14-2 Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection
0	1	1	0	$f_x/2^2$ (1.05 MHz)
0	1	1	1	$f_x/2^3$ (524 kHz)
1	0	0	0	$f_x/2^4$ (262 kHz)
1	0	0	1	$f_x/2^5$ (131 kHz)
1	0	1	0	$f_x/2^6$ (65.5 kHz)
1	0	1	1	$f_x/2^7$ (32.7 kHz)
1	1	0	0	$f_x/2^8$ (16.4 kHz)
1	1	0	1	$f_x/2^9$ (8.2 kHz)
Other than the above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection
0	1	1	0	$f_x/2^2$ (1.05 MHz)
0	1	1	1	$f_x/2^3$ (524 kHz)
1	0	0	0	$f_x/2^4$ (262 kHz)
1	0	0	1	$f_x/2^5$ (131 kHz)
1	0	1	0	$f_x/2^6$ (65.5 kHz)
1	0	1	1	$f_x/2^7$ (32.7 kHz)
1	1	0	0	$f_x/2^8$ (16.4 kHz)
1	1	0	1	$f_x/2^9$ (8.2 kHz)
Other than the above				Setting prohibited

★ **Caution Before rewriting the data of TCL3, stop the timer once.**

Remarks 1. fx: Main system clock oscillation frequency

★ **2.** Values in parentheses when operated at fx = 5.0 MHz

(2) Serial operating mode register 1 (CSIM1)

This register sets serial interface channel 1 serial clock and operation enable/stopped.

CSIM1 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets CSIM1 to 00H.

Fig. 14-3 Serial Operation Mode Register 1 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	Note1 0	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial Interface Channel 1 Clock Selection
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/20 (Input)	SOI (CMOS output)
1	LSB		

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Operation of Shift Register 1	Controls Operation of Serial Clock Counter	SI1/P20 pin Function	SO1/P21 pin Function	$\overline{\text{SCK1}}$/P22 Pin Function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stops	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enabled	Count operation	SI1 Note 3 (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. Be sure to set 0 in bit 5.

2. Can be used freely as port function.

3. Can be used as P20 when only transmitter is used.

Remark ×: Don't care

14.4 Serial Interface Channel 1 Operations

The following two operating modes are available to the serial interface channel 1.

- Operation stop mode
- 3-wire serial I/O mode

14.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 1 (SIO1) does not carry out shift operation either and thus it can be used as ordinary 8-bit register.

In the operation stop mode, the P20/SI1, P21/SO1 and P22/$\overline{SCK1}$ pins can be used as ordinary input/output ports.

(1) Register set

The operation stop mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{RESET}$ input sets CSIM1 to 00H.

The shaded area is used in the operation stop mode.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	Note1 0	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	$\overline{SCK1}$/P22 Pin Function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stops	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enabled	Count operation	SI1 Note 3 (input)	SO1 (CMOS output)	$\overline{SCK1}$ (input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes 1. Be sure to set 0 in bit 5.

2. Can be used freely as port function.

3. Can be used as P20 (CMOS input/output) when using transmission only.

Remark ×: Don't care

14.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface as is the case with the 75X, 78K and 17K series.

Communication is carried out with three lines of serial clock ($\overline{SCK1}$), serial output (SO1) and serial input (SI1).

(1) Register set

The 3-wire serial I/O mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulate instruction.

$\overline{RESET}$ input sets CSIM1 to 00H.

The shaded area is used in the 3-wire serial I/O mode

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	0 (Note 1)	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial Interface Channel 1 Clock Selection
0	×	Clock externally input to $\overline{SCK1}$ pin (Note 1)
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/20 (input)	SOI (CMOS output)
1	LSB		

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	$\overline{SCK1}$/P22 Pin Function
0	×	× (Note 2)	× (Note 2)	× (Note 2)	× (Note 2)	× (Note 2)	× (Note 2)	Operation stops	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	1 (Note 3)	× (Note 3)	0	0	1	×	Operation enabled	Count operation	SI1 (Note 3) (input)	SO1 (CMOS output)	$\overline{SCK1}$ (input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes 1. Be sure to set 0 in bit 5.

2. Can be used freely as port function.

3. Can be used as P20 when only transmitter is used.

Remark ×: Don't care

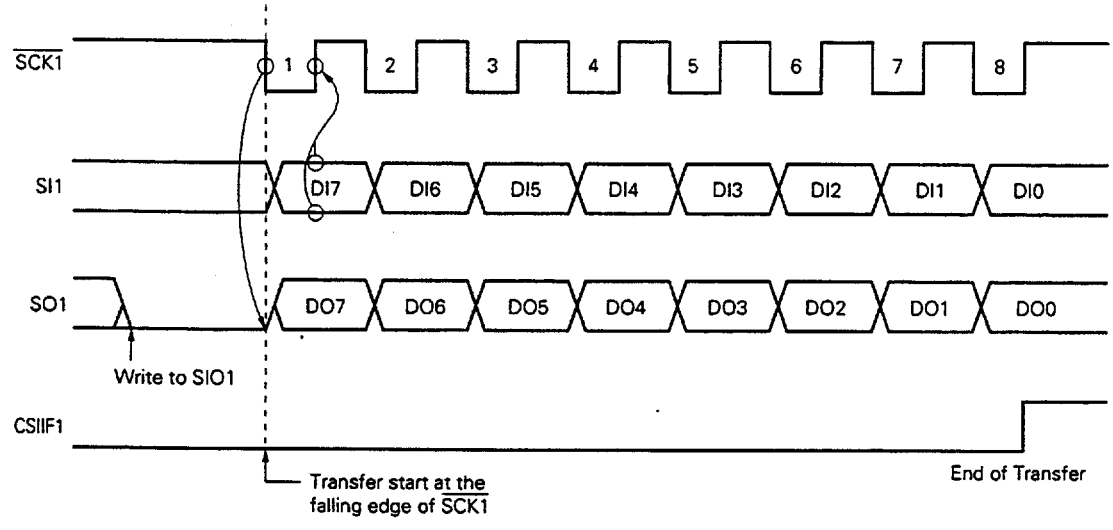
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 1 (SIO1) is carried out at the falling edge of the serial clock ($\overline{SCK1}$). The transmit data is held in the SO1 latch and is output from the SO1 pin. The receive data input to the SI1 pin is latched into SIO1 at the rising edge of $\overline{SCK1}$.

Upon termination of 8-bit transfer, the SIO1 operation stops automatically and the serial transfer end flag (CSIIF1) is set.

Fig. 14-4 3-Wire Serial I/O Mode Timings



Caution Do not set CSIE1 to 0 during transfer. If CSIE is set to 1, an undefined value will be output.

CHAPTER 15 FIP CONTROLLER/DRIVER

15.1 Fip Controller/Driver Functions

The functions of the FIP controller/driver incorporated in the μPD78024 subseries are shown below.

(1) Segment signal output (DMA operation) and automatic digit signal output can be performed by automatic reading of display data.

(2) 9 to 16 segment and 2 to 16 digit FIP (fluorescent indicator panel) control is possible by means of display mode registers 0 and 1 (DSPM0, DSPM1).

(3) Pins which are not used for FIP display can be used as output port or input/output port pins.

(4) 8-level brightness adjustment is possible by means of display mode register 1 (DSPM1).

(5) Hardware is incorporated which takes key scanning applications into consideration.
- Generation of interrupt (INTKS) indicates key scan timing.
- A key scan signal is output from the segment output pins by setting key scan data in port 11.
- The timing of key scan data output can be detected by means of the key scan flag (KSF).

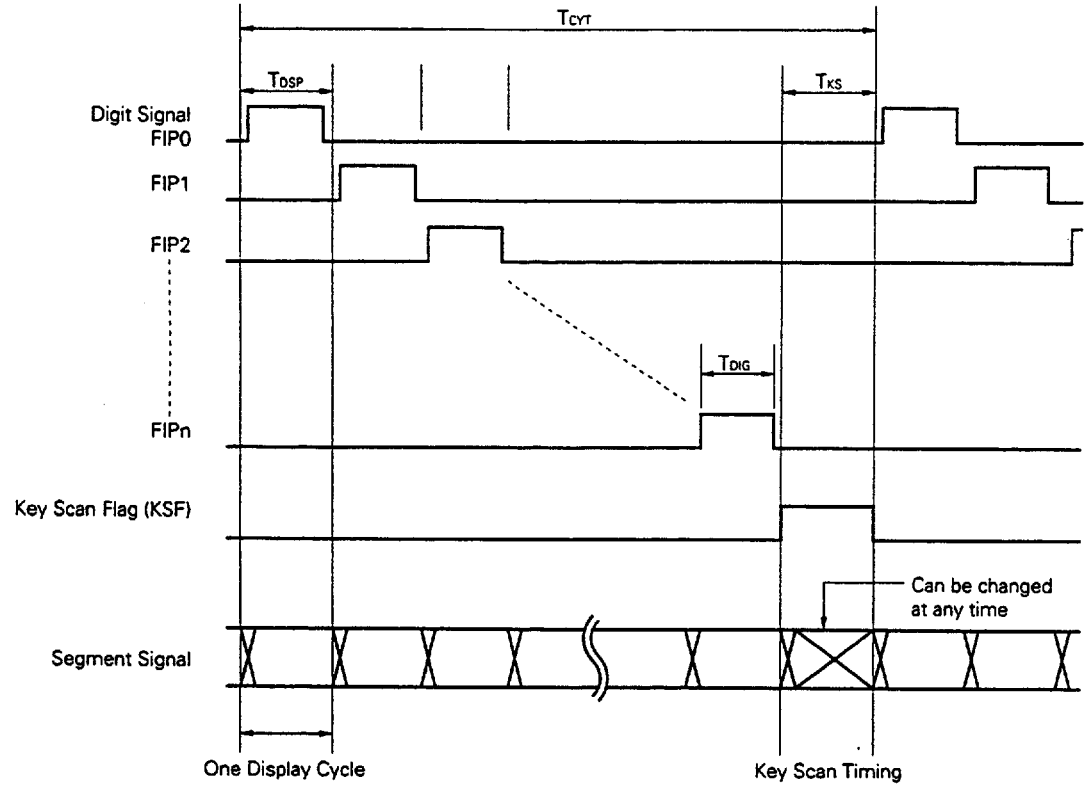
(6) A high-voltage output buffer is incorporated which allows the FIP to be driven directly.

(7) The display output pins can incorporate a pull-down resistor by means of a mask option.

Cautions 1. The FIP controller/driver is operable when selecting fx or fx/2 CPU clock. For FIP display, set CSS to 0 and PCC2 to PCC0 to 000 or 001. With other clocks (including the subsystem clock), normal display can not be performed. Before stopping the main oscillation, be sure to stop the display. (Set DIGS0 to DIGS3 to 000.)

2. When using the FIP controller/driver, the 11 and 12 port pins used as segment output should be set to the output mode (set 0 in bits corresponding to port mode registers 11 and 12).

Fig. 15-1 FIP Controller Operation Timing



n : Number of display digits −1
(2 to 16 segments selectable by means of display mode register 1 (DSPM1))

T_{DSP} : 1 display cycle
★ (1024/fx (204.8 μs: at 5.0 MHz operation) or (2048/fx (409.6 μs: at 5.0 MHz operation)

T_{KS} : Key scan timing
(T_{KS} = T_{DSP})

T_{CYT} : Display cycle
(T_{CYT} = T_{DSP} × (display digits + 1))

T_{DIG} : Digital signal pulse width
(8 widths selectable by means of display mode register 1 (DSPM1))

All 26 display output pins have a dual function as port pins. When "display stopped" is set by bits 4 to 7 (DIGS0 to DIGS3) of display mode register 1 (DSPM1), the 26 display output pins can be used as port pins.

Also, display output pins which are not used for digit signal output or segment signal output can be used as port pins even if the display enabled state is set.

Table 15-1 Correspondence between Display Output Pins and Dual-Function Pins

Display Pin	Dual-Function Pin	Input/Output
FIP0, FIP1	P80, P81	Dedicated output port
FIP2 to FIP9	P90 to P97	Dedicated output port
FIP10 to FIP17	P100 to P107	Dedicated output port
FIP18 to FIP25	P110 to P117	Input/output port

15.2 FIP Controller/Driver Configuration

The FIP controller/driver comprises the following hardware.

Table 15-2 FIP Controller/Driver Configuration

Item	Configuration
Display outputs	26 (number of segments: 9 to 16, number of digits: 2 to 16)
Control registers	Display mode register 0 (DSPM0) Display mode register 1 (DSPM1)

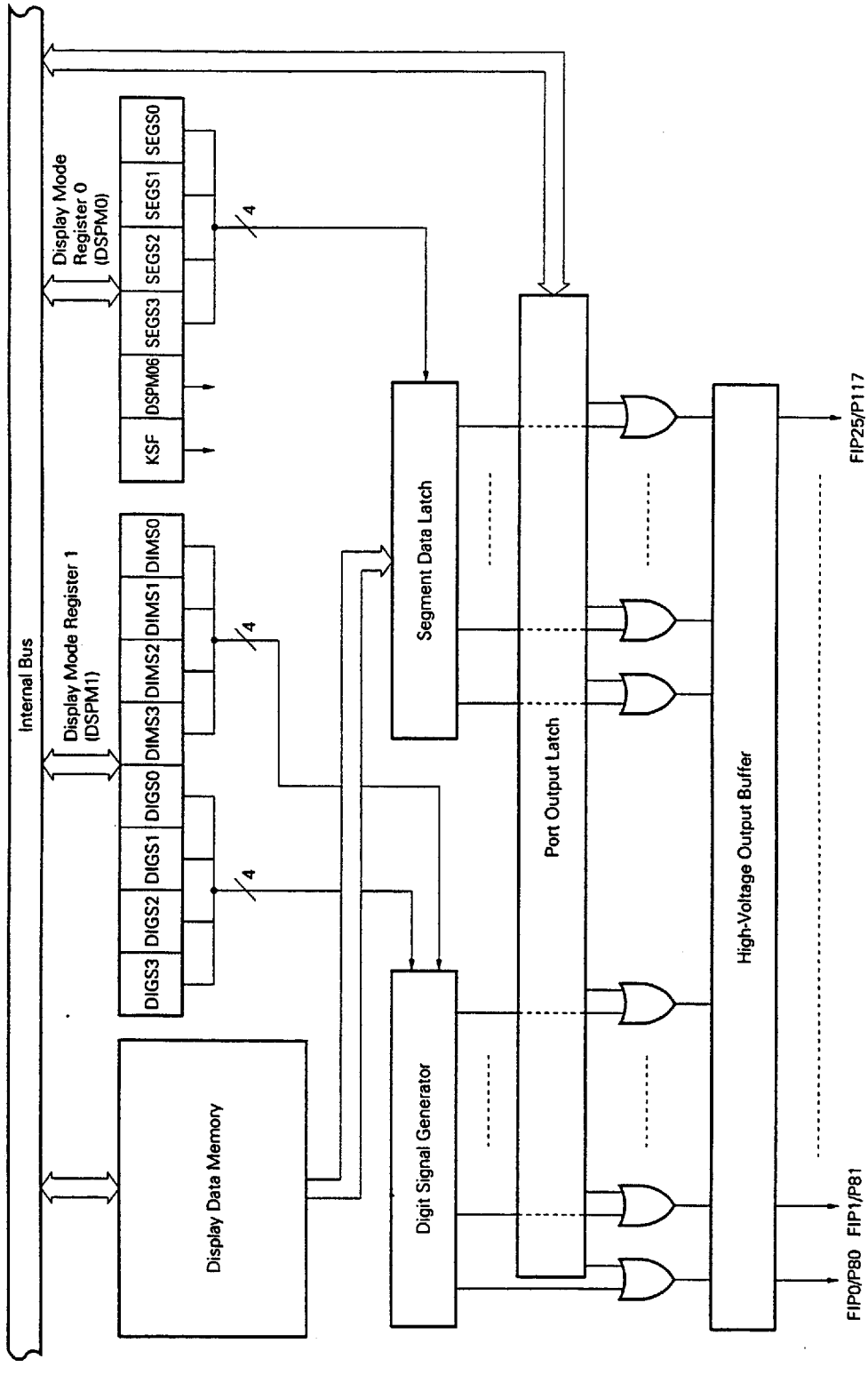


Fig. 15-2 FIP Controller/Driver Block Diagram

15.3 FIP Controller/Driver Control Registers

The FIP controller/driver is controlled by the following two registers.

- Display mode register 0 (DSPM0)
- Display mode register 1 (DSPM1)

(1) Display mode register 0 (DSPM0)(see Fig. 15-3)

This register shows the number of display segments, the subsystem clock noise elimination circuit mode setting, and the key scan timing.

DSPM0 is set by an 8-bit memory manipulation instruction. Bit 7 (KSF) only can be read by a 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DSPM0 to 00H.

(2) Display mode register 1 (DSPM1)(see Fig. 15-4)

This register selects the number of display digits and digit signal cut width setting, and the display cycle (T_{DSP}). When bit 0 (DIMS0) is set to 1, selecting 2048/fx (409.6 μs: 5.0 MHz operation) as the display cycle, leakage emission is small. However, as the number of display digits increases, the display cycle approaches the commercial power supply frequency, causing flickering of the display, and therefore 1024/fx (204.8 μs: 5.0 MHz operation) should be selected. If there is leakage emission in this case, the digit signal cut width should be adjusted by means of bits 1 to 3 (DIMS1 to DIMS3). ★ ★

DSPM1 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DSPM1 to 00H.

Fig. 15-3 Display Mode Register 0 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM0	KSF	DSPM06	0	0	SEGS3	SEGS2	SEGS1	SEGS0	FFA0H	00H	R/W Note 1

SEGS3	SEGS2	SEGS1	SEGS0	Number of Display Segments
0	0	0	0	9 Segments
0	0	0	1	10 Segments
0	0	1	0	11 Segments
0	0	1	1	12 Segments
0	1	0	0	13 Segments
0	1	0	1	14 Segments
0	1	1	0	15 Segments
0	1	1	1	16 Segments
Other than the above				Setting prohibited

DSPM06	Subsystem Clock Noise Elimination Circuit Mode Setting Note 2
0	2.5 MHz<fx≤5.0 MHz
1	1.25 MHz≤fx≤2.5 MHz

KSF	Timing Status
0	Display timing
1	Key scan timing

Notes 1. Bit 7 (KSF) is Read Only.

2. A value should be set to match the main system clock frequency (fx) used. The noise elimination circuit is effective during an FIP display operation.

Caution When using both a main system clock of 1.25 MHz or less and the FIP controller/driver, be sure to use the watch timer count on the main system clock (TCL24 = 0).

Fig. 15-4 Display Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM0	DIGS3	DIGS2	DIGS1	DIGS0	DIMS3	DIMS2	DIMS1	DIMS0	FFA1H	00H	R/W

DIMS0	Display Cycle Selection
0	1024/fx used as display cycle (1 display cycle = 204.8 μs: 5.0MHz operation)
1	2048/fx used as display cycle (1 display cycle = 409.6 μs: 5.0MHz operation)

DIMS3	DIMS2	DIMS1	Digit Signal Cut Width
0	0	0	1/16
0	0	1	2/16
0	1	0	4/16
0	1	1	6/16
1	0	0	8/16
1	0	1	10/16
1	1	0	12/16
1	1	1	14/16

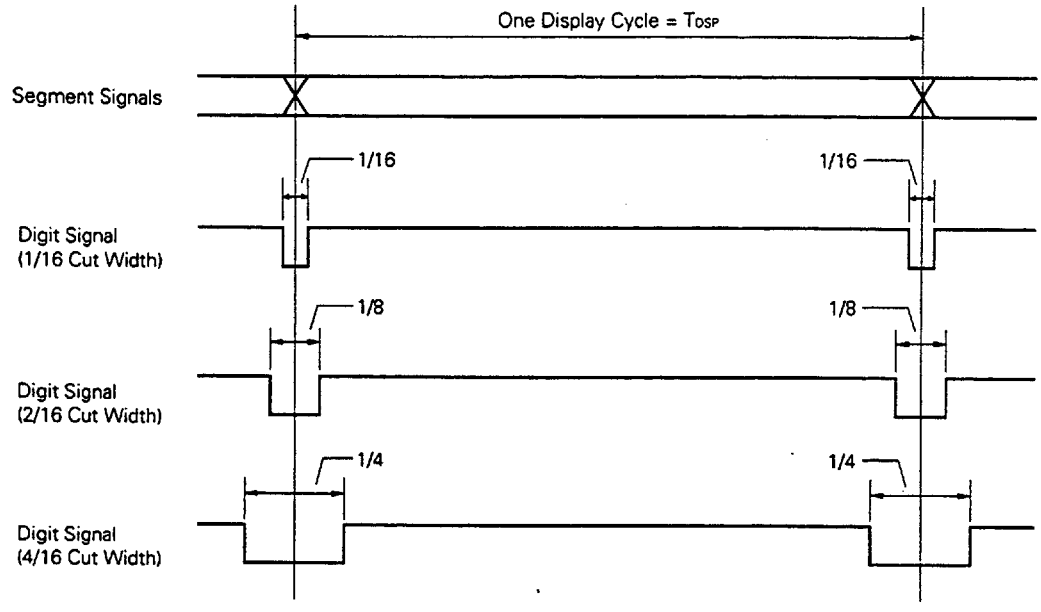
DIGS3	DIGS2	DIGS1	DIGS0	Number of Display Digits
0	0	0	0	Display stopped (Static display)Note
0	0	0	1	2 digits
0	0	1	0	3 digits
0	0	1	1	4 digits
0	1	0	0	5 digits
0	1	0	1	6 digits
0	1	1	0	7 digits
0	1	1	1	8 digits
1	0	0	0	9 digits
1	0	0	1	10 digits
1	0	1	0	11 digits
1	0	1	1	12 digits
1	1	0	0	13 digits
1	1	0	1	14 digits
1	1	1	0	15 digits
1	1	1	1	16 digits

Note Display stopped (static display): If the display halt state has been set, static display is possible by operating the port output latch.

Remark fx : Main system clock oscillation frequency

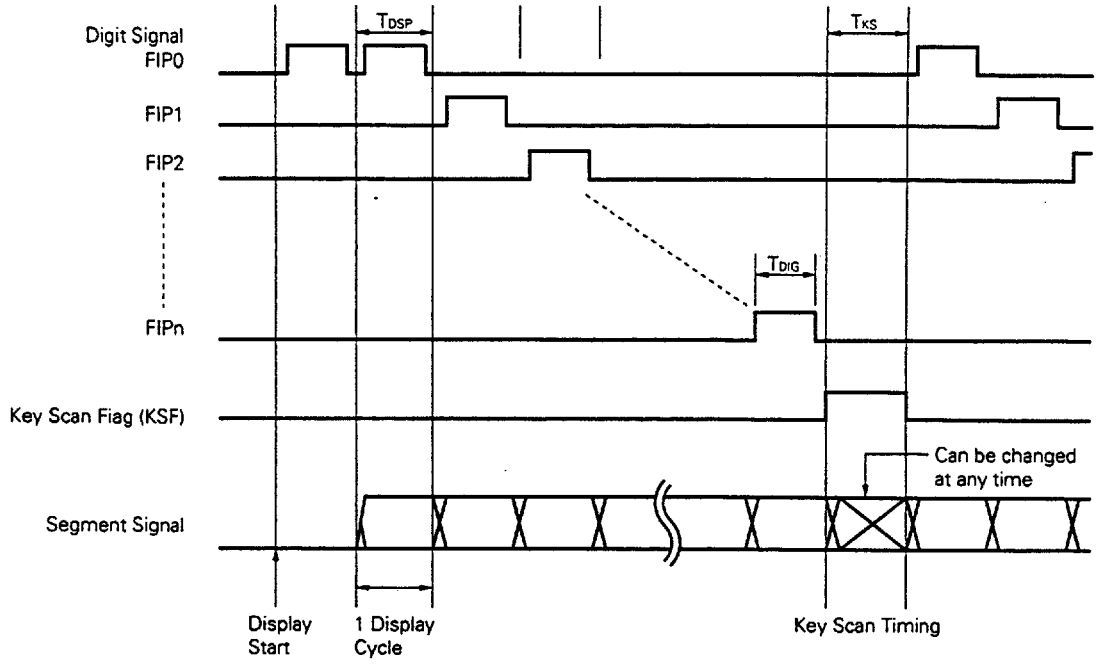
The digit signal is cut by half at the beginning and end of the dispaly period, if so specified by DIMS1-DIMS3 of DSPM1.

Fig. 15-5 Digit Signal Cut Width



When display is started from the display halt state, FIP0 is output twice.

Fig. 15-6 FIP Controller Display Start Timing



n : Number of display digits –1
2 to 16 digits can be selected by display mode register 1 (DSPM).

T_{DSP} : One Display Cycle
(1024/fx (204.8 μs: at 5.0 MHz operation) or (2048/fx (409.6 μs: at 5.0 MHz operation) ★

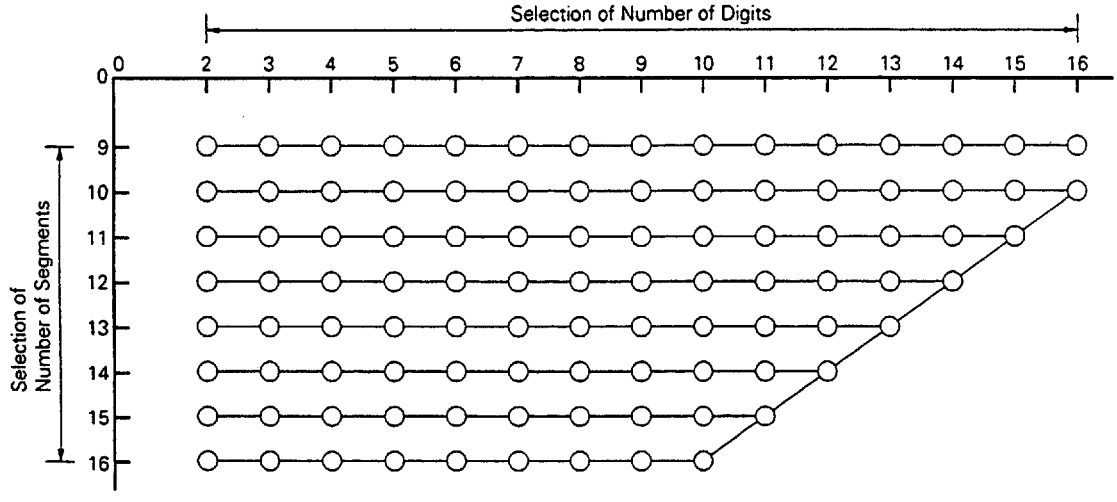
T_{KS} : Key scan timing
(T_{KS} = T_{DSP})

T_{DIG} : Digit signal pulse width
(8 widths can be selected by display mode register 1 (DSPM1))

15.4 Display Mode Selection

The number of segments and digits that can be displayed by the FIP controller/driver is determined by the display mode.

Fig. 15-7 Display Mode Selection Diagram



Caution When setting a total of digits and segments to more than 26, specification of digits takes priority over that of segments.

15.5 Display Mode and Dispaly Outputs

The on-chip FIP controller assigns pins FIP0/P80 to FIP25/P117 to digit signals and segment signals in that order up to the number set by display mode registers 0 and 1 (DISPM0, DISPM1), and assigns the remaining pins as general-purpose port pins.

The pin layout for a 14-segment display is shown below as an example.

Fig. 15-8 Pin Layout for 14-Segment Display ★

Pin Name	Selection of Number of Dispaly Digits								
	Display Stopped	2	3	.....	8	9	10	11	12
FIP0/P80	P80	T0VP80	T0VP80		T0VP80	T0VP80	T0VP80	T0VP80	T0VP80
FIP1/P81	P81	T1VP81	T1VP81		T1VP81	T1VP81	T1VP81	T1VP81	T1VP81
FIP2/P90	P90	P90	T2VP90		T2VP90	T2VP90	T2VP90	T2VP90	T2VP90
FIP3/P91	P91	P91	P91		T3VP91	T3VP91	T3VP91	T3VP91	T3VP91
FIP4/P92	P92	P92	P92		T4VP92	T4VP92	T4VP92	T4VP92	T4VP92
FIP5/P93	P93	P93	P93		T5VP93	T5VP93	T5VP93	T5VP93	T5VP93
FIP6/P94	P94	P94	P94		T6VP94	T6VP94	T6VP94	T6VP94	T6VP94
FIP7/P95	P95	P96	P95		T7VP95	T7VP95	T7VP95	T7VP95	T7VP95
FIP8/P96	P96	P96	P96		P96	T8VP96	T8VP96	T8VP96	T8VP96
FIP9/P97	P97	P97	P97		P97	P97	T9VP97	T9VP97	T9VP97
FIP10/P100	P100	S0VP100	S0VP100		S0VP100	S0VP100	S0VP100	T10VP100	T10VP100
FIP11/P101	P101	S1VP101	S1VP101		S1VP101	S1VP101	S1VP101	S0VP101	T11VP101
FIP12/P102	P102	S2VP102	S2VP102	.....	S2VP102	S2VP102	S2VP102	S1VP102	S0VP102
FIP13/P103	P103	S3VP103	S3VP103		S3VP103	S3VP103	S3VP103	S2VP103	S1VP103
FIP14/P104	P104	S4VP104	S4VP104		S4VP104	S4VP104	S4VP104	S3VP104	S2VP104
FIP15/P105	P105	S5VP105	S5VP105		S5VP105	S5VP105	S5VP105	S4VP105	S3VP105
FIP16/P106	P106	S6VP106	S6VP106		S6VP106	S6VP106	S6VP106	S5VP106	S4VP106
FIP17/P107	P107	S7VP107	S7VP107		S7VP107	S7VP107	S7VP107	S6VP107	S5VP107
FIP18/P110	P110	S8VP110	S8VP110		S8VP110	S8VP110	S8VP110	S7VP110	S6VP110
FIP19/P111	P111	S9VP111	S9VP111		S9VP111	S9VP111	S9VP111	S8VP111	S7VP111
FIP20/P112	P112	S10VP112	S10VP112		S10VP112	S10VP112	S10VP112	S9VP112	S8VP112
FIP21/P113	P113	S11VP113	S11VP113		S11VP113	S11VP113	S11VP113	S10VP113	S9VP113
FIP122/P114	P114	S12VP114	S12VP114		S12VP114	S12VP114	S12VP114	S11VP114	S10VP114
FIP23/P115	P115	S13VP115	S13VP115		S13VP115	S13VP115	S13VP115	S12VP115	S11VP115
FIP24/P116	P116	P116	P116		P116	P116	P116	S13VP116	S12VP116
FIP25/P117	P117	P117	P117		P117	P117	P117	P117	S13VP117

Remark V : Logical sum (OR)
T0-T11 : Digit pin
S0-S13 : Segment pin

15.6 Display Data Memory

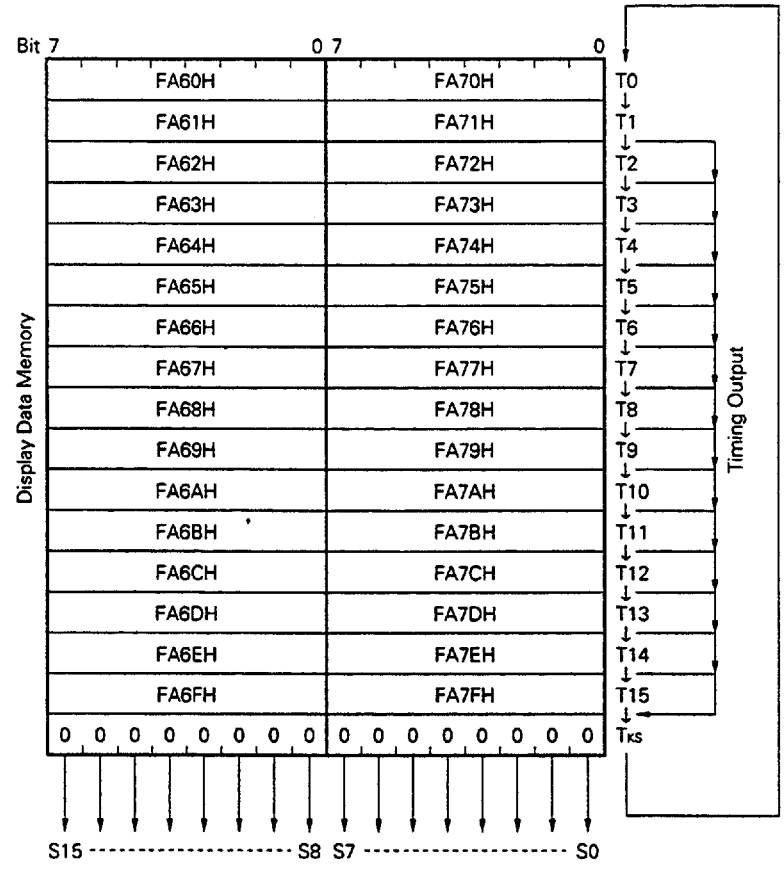
The display data memory is an area which stores the segment data to be displayed.

The display data memory is mapped onto addresses FA60H to FA7FH. The FIP controller reads the data stored in the display data memory for the FIP display without regard to the operation of the CPU (DMA operation).

Any area which is not used for display purposes can be used as ordinary RAM.

At the key scan timing (T_{KS}), all the segment outputs are set to "0" outputs and the output latch data of port 11 is output to FIP18/P110 to FIP25/P117.

Fig. 15-9 Relation between Display Data Memory Contents and Segment Output



15.7 Key Scan Flag and Key Scan Data

15.7.1 Key scan flag

The key scan flag (KSF) is set to 1 during key scanning and reset to 0 automatically during display.

KSF is mapped onto bit 7 of display mode register 0 (DSPM0), and can be tested as a bit unit. It cannot be written to.

By testing KSF it is possible to determine whether or not key scanning is in progress and whether key-input data is correct.

15.7.2 Key scan data

The data stored in port 11 is output to pins FIP18 through FIP25 during key scanning.

Changing the port 11 output data enables key scanning to be performed using pins FIP18 through FIP25.

Caution The output latch of port 11 must be set to 0 when the key scan timing is complete.

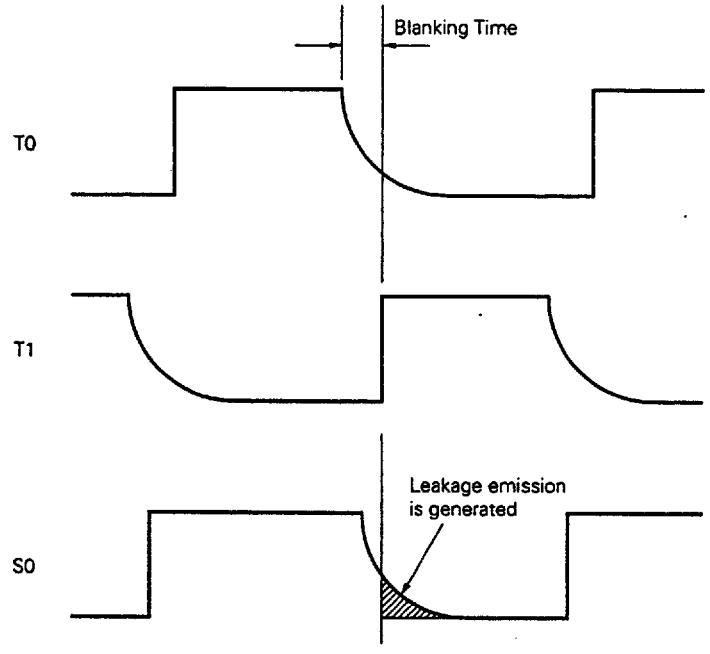
15.8 Fluorescent Indicator Panel Leakage Emission

When a fluorescent indicator panel is driven using a μPD78024 subseries device, leakage emission may occur. There are two probable causes of this as described below.

(1) Leakage emission due to short blanking time

Fig. 15-10 shows the signal waveform when the first digit of a 2-digit display is lit and the second is not lit. If the blanking time is short as shown in this figure, the T1 signal rises before the segment signals have disappeared, resulting in leakage emission. In general, a blanking time of around 20 μs is required, and therefore the value set in display mode register 1 (DSPM1) must be considered.

Fig. 15-10 Schematic Diagram of Leakage Emission Due to Short Blanking Time



(2) Leakage emission due to capacitance between fluorescent indicator panel segments and *grid*

Leakage emission may still occur even if the blanking time is adequate, as shown in Fig. 15-12. This is because there is capacitance between the fluorescent indicator panel grid and segments as indicated as C_{SG} in Fig. 15-11, and when a segment signal is turned on, the timing signal pin is raised via C_{SG}. If this voltage reaches or exceeds the level of the cut-off voltage (E_K) as shown in Fig. 15-12, leakage emission occurs.

This spike voltage varies according to the size of C_{SG} and the value of the internal pull-down resistor (R_L). The larger the value of C_{SG} and the larger the value of R_L, the higher the voltage, and the more likely the occurrence of leakage emission.

The value of C_{SG} depends on the display area of the fluorescent indicator panel: the larger the area, the larger the value of C_{SG}. Therefore, the value of the pull-down resistor required to prevent the occurrence of leakage emission depends on the size of the fluorescent indicator panel.

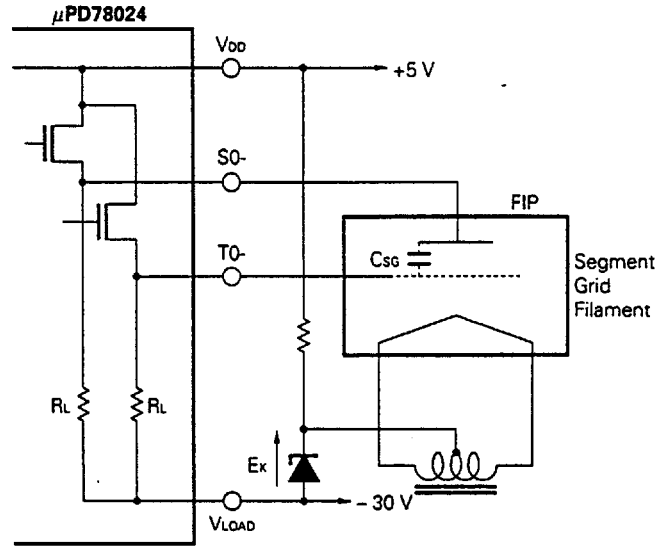
On the other hand, since the pull-down resistor value which can be incorporated by a mask option is comparatively large, it may not be possible to suppress leakage emission with this internal pull-down resistor alone.

If adequate display quality cannot be obtained, various countermeasures should be considered such as deepening the back bias (raising E_K), adding a filter to the fluorescent indicator panel, or externally connecting a pull-down resistor of several 10kΩ to the timing signal pin.

The likelihood of occurrence of leakage emission due to C_{SG} depends on the duty cycle of the spike voltage included in the overall display cycle, and thus the likelihood of occurrence is greater the smaller the number of display digits.

This is still effective even if the brightness of the display is reduced.

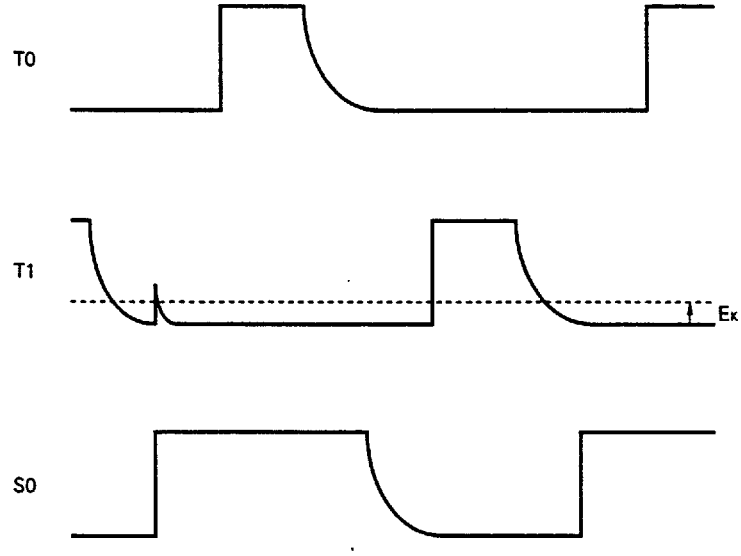
Fig. 15-11 Explanatory Diagram of Leakage Emission Due to C_{SG}



Remarks 1. E_K: Cut-off voltage

2. R_L: Internal pull-down resistor

Fig. 15-12 Schematic Diagram of Leakage Emission Due to C_{SG}



CHAPTER 16 INTERRUPT FUNCTIONS AND TEST FUNCTIONS

16.1 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 interrupt 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 register (PR).
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 16-1**).
A standby release signal is generated.
Four external interrupts and nine 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 state. The software interrupt does not undergo interrupt priority control.

16.2 Interrupt Sources and Configuration

A total of 15 non-maskable, maskable and software interrupts are incorporated in the interrupt sources (see **Table 16-1**).

Table 16-1 Interrupt Source List

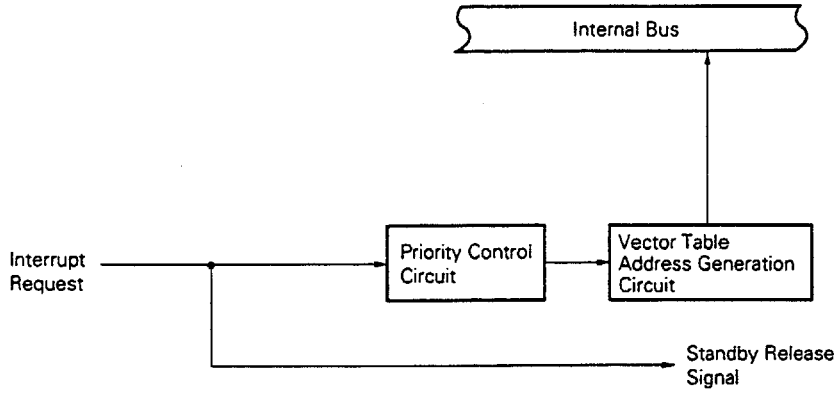
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	A
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			B
	1	INTP0	Pin input edge detection	External	0006H	C
	2	INTP1			0008H	D
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTCSI0	End of serial interface channel 0 transfer	Internal	000EH	B
	6	INTCSI1	End of serial interface channel 1 transfer		0010H	
	7	INTTM3	Reference time interval signal from watch timer		0012H	
	8	INTTM0	Generation of 16-bit timer/event counter match signal		0014H	
	9	INTTM1	Generation of 8-bit timer/counter 1 match signal		0016H	
	10	INTTM2	Generation of 8-bit timer/counter 2 match signal		0018H	
	11	INTAD	End of A/D converter conversion		001AH	
	12	INTKS	Key scan timings from FIP controller/ driver		001CH	
Software	—	BRK	BRK instruction execution	Internal	003EH	E

Notes 1. Default priorities are intended for two or more simultaneously generated non-maskable interrupts. 0 is the highest priority and 12 is the lowest priority.

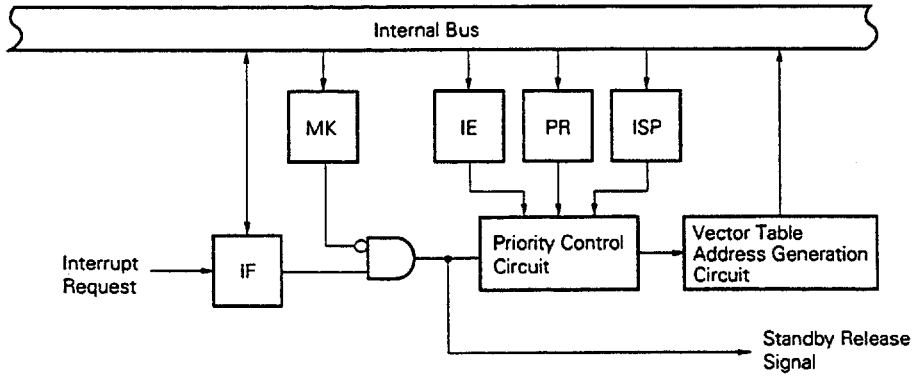
2. Basic configuration types A to E correspond to A to E in Fig. 16-1.

Fig. 16-1 Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

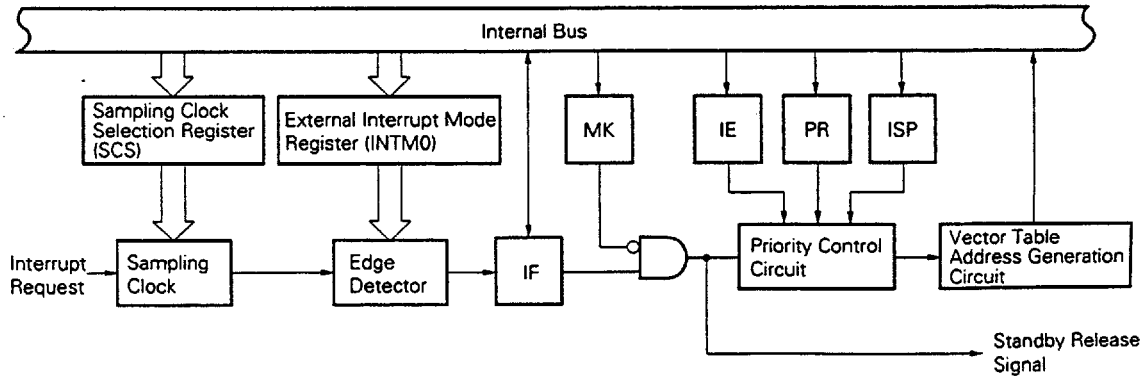
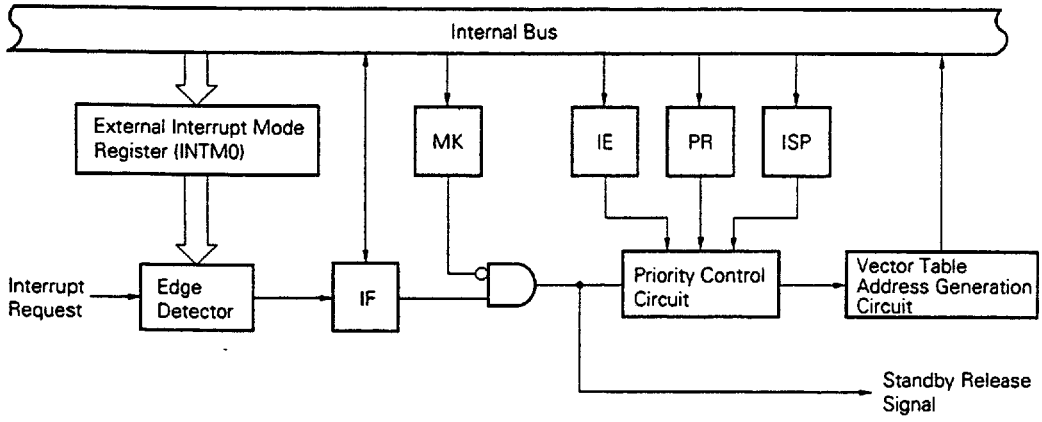
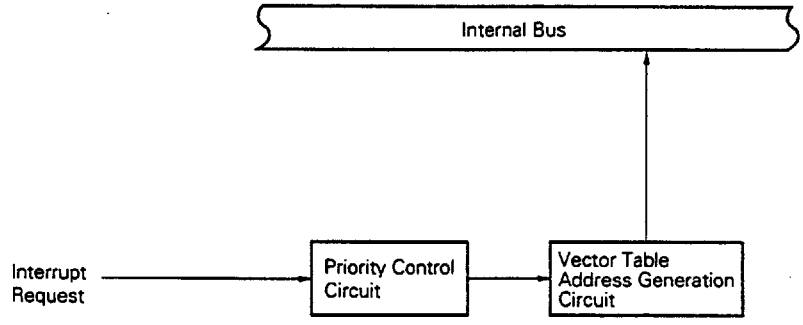


Fig. 16-1 Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



IF : interrupt request flag
IE : interrupt enable flag
ISP : in-service priority flag
MK : interrupt mask flag
PR : priority specification flag

16.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0)
- Interrupt mask flag register (MK0)
- Priority specify flag register (PR0)
- External interrupt mode register (INTM0)
- Sampling clock selection register (SCS)
- Program status word (PSW)

Table 16-2 gives a listing of interrupt request flags, interrupt mask flags and priority specify flag names corresponding to interrupt request sources.

Table 16-2 Various Flags Corresponding to Interrupt Request Sources

Interrupt Request Signal Name	Interrupt Request Flag	Interrupt Mask Flag	Priority Specify Flag
INTP0	PIF0	PMK0	PPR0
INTP1	PIF1	PMK1	PPR1
INTP2	PIF2	PMK2	PPR2
INTP3	PIF3	PMK3	PPR3
INTTM0	TMIF0	TMMK0	TMPR0
INTTM1	TMIF1	TMMK1	TMPR1
INTTM2	TMIF2	TMMK2	TMPR2
INTTM3	TMIF3	TMMK3	TMPR3
INTWDT	TMIF4	TMMK4	TMPR4
INTCSI0	CSIIF0	CSIMK0	CSIPR0
INTCSI1	CSIIF1	CSIMK1	CSIPR1
INTAD	ADIF	ADMK	ADPR
INTKS	KSIF	KSMK	KSPR

(1) Interrupt request flag register (IF0)

The interrupt request is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0 is set with a 1-bit, 8-bit or 16-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets IF0 to 00H.

Fig. 16-2 Interrupt Request Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
IF0L	TMIF3	CSIIF1	CSIIF0	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W
	7	6	⑤	④	③	②	①	⓪			
IF0H	0	0	WTIF	KSIF	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

xxIF	Interrupt Request Flag
0	Interrupt request signal is not generated.
1	Interrupt request signal is generated, interrupt request state is entered into.

Cautions 1. TMIF4 flag is R/W enabled only when a watchdog timer is used as interval timer. If a watchdog timer is used as non-maskable interrupt, set TMIF4 flag to 0.

2. As port 0 has a dual function as an 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.

(2) Interrupt mask flag register (MK0)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0 is set with a 1-bit, 8-bit or 16-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets MK0 to FFH.

Fig. 16-3 Format of Interrupt Mask Flag Register

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
MK0L	TMMK3	CSIMK1	CSIMK0	PMK3	PMK2	PMK1	PMK0	TMMK4	FFF4H	FFH	R/W
	7	6	⑤	④	③	②	①	⓪			
MK0H	1	1	WTMK	KSMK	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

xxMK	Interrupt Serving and Standby Mode Control
0	Interrupt serving enabled and standby mode release enabled
1	Interrupt serving disabled and standby mode release disabled

Cautions 1. If TMMK4 flag is read when a watchdog timer is used as non-maskable interrupt, MK0 becomes undefined.

2. As port 0 has a dual function as an 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.

(3) Priority specify flag register (PR0)

The priority specify flag is used to set the corresponding maskable interrupt priority orders.

PR0 is set with a 1-bit, 8-bit or 16-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets PR0 to FFH.

Fig. 16-4 Priority Specify Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
PR0L	TMPR3	CSIPR1	CSIPR0	PPR3	PPR2	PPR1	PPR0	TMPR4	FFF8H	FFH	R/W
	7	6	5	④	③	②	①	⓪			
PR0H	1	1	1	KSPR	ADPR	TMPR2	TMPR1	TMPR0	FFE9H	FFH	R/W

xxPR	Priority Level Selection
0	High priority level
1	Low priority level

Caution When a watchdog timer is used as non-maskable interrupt, set TMPR4 flag to 1.

(4) External interrupt mode register (INTM0)

This register is used to set the valid edges of INTP0 to INTP2.

INTM0 is set with an 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets INTM0 to 00H.

Remarks 1. INTP0 pin also serves as TI0/P00.

2. INTP3 is fixed to the falling edge.

Fig. 16-5 External Interrupt Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM0	ES31	ES30	ES21	ES20	ES11	ES10	0	0	FFFCH	00H	R/W

ES11	ES10	INTP0n Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES21	ES20	INTP1 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	INTP2 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

(5) Sampling clock select register (SCS)

This register is used to set the valid edge clock sampling clock to be input to INTP0. When remote controlled data reception is carried out using INTP0, digital noise is removed with sampling clocks.

SCS is set with an 8-bit memory manipulate instruction.

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

Fig. 16-6 Sampling Clock Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	INTP0 Sampling Clock Selection
0	0	$fx/2^{N+1}$
0	1	Setting prohibited
1	0	$fx/2^6$ (78.1 kHz)
1	1	$fx/2^7$ (39.1 kHz)

Caution $fx/2^{N+1}$ is a clock to be supplied to the CPU and $fx/2^6$ and $fx/2^7$ are clocks to be supplied to the peripheral hardware. $fx/2^{N+1}$ stops in the HALT mode.

Remark 1. N : Value (N = 0 to 4) at bits 0 to 2 (PCC0 to PCC2) of processor clock control register

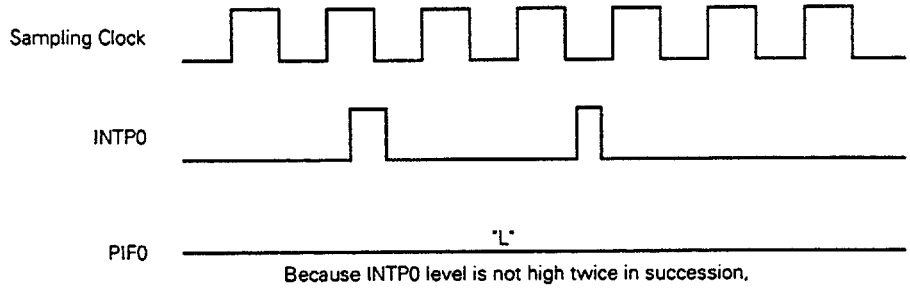
2. fx : Main system clock oscillation frequency

★ **3.** Values in parentheses when operated at fx = 5.0 MHz

When the INTP0 input level is high twice in succession, a noise elimination circuit sets the PIF0 flag to 1.

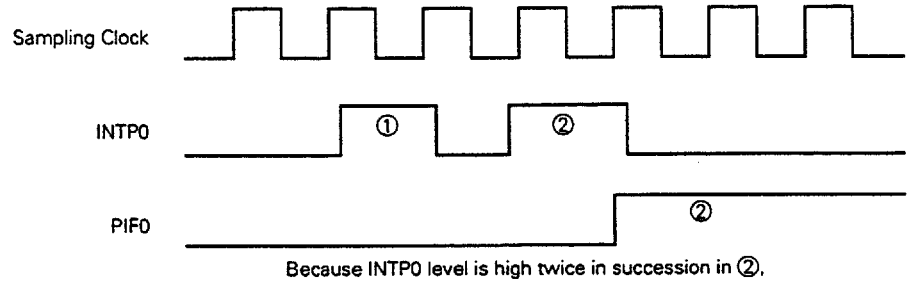
Fig. 16-7 Noise Elimination Circuit Input/Output Timing (When Rising Edge is Detected)

(a) When input is less than the sampling cycle (t_{SMP})



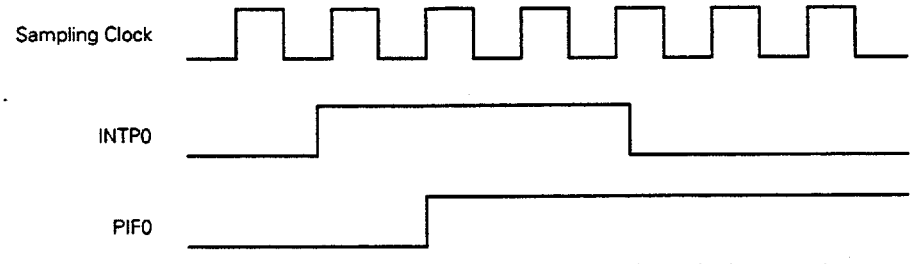
Because INTP0 level is not high twice in succession, PIF0 output remains at low level.

(b) When input is equal to or twice the sampling cycle (t_{SMP})



Because INTP0 level is high twice in succession in ②, the PIF0 flag is set to 1.

(c) When input is twice or more than the sampling cycle (t_{SMP})



Because INTP0 level is high more than twice in succession, the PIF0 flag is set to 1.

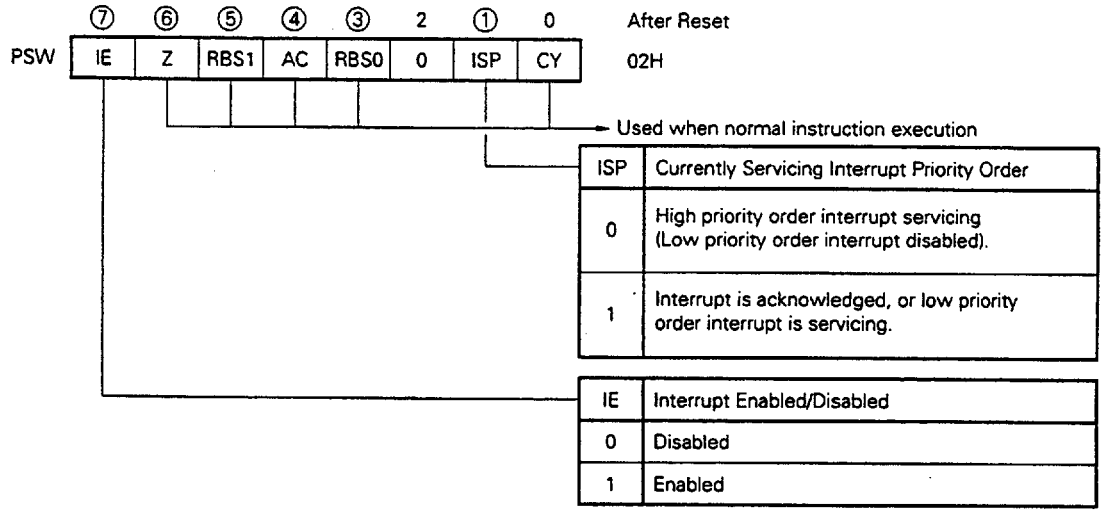
(6) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control multiple processing are mapped.

Besides 8-bit unit read/write, this register can carry out operations with a bit manipulate instruction and dedicated instructions (EI and DI). When a vectored interrupt is acknowledged, if the BRK instruction is executed, the interrupt is saved into a stack and the IE flag is reset to 0. If a maskable interrupt is acknowledged, the contents of the priority specify flag of the acknowledged interrupt are transferred to the ISP flag. The acknowledged interrupt is also saved into the stack with the PUSH PSW instruction. It is reset from the stack with the RETI, RETB and POP PSW instructions.

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

Fig. 16-8 Program Status Word Format



16.4 Interrupt Servicing Operations

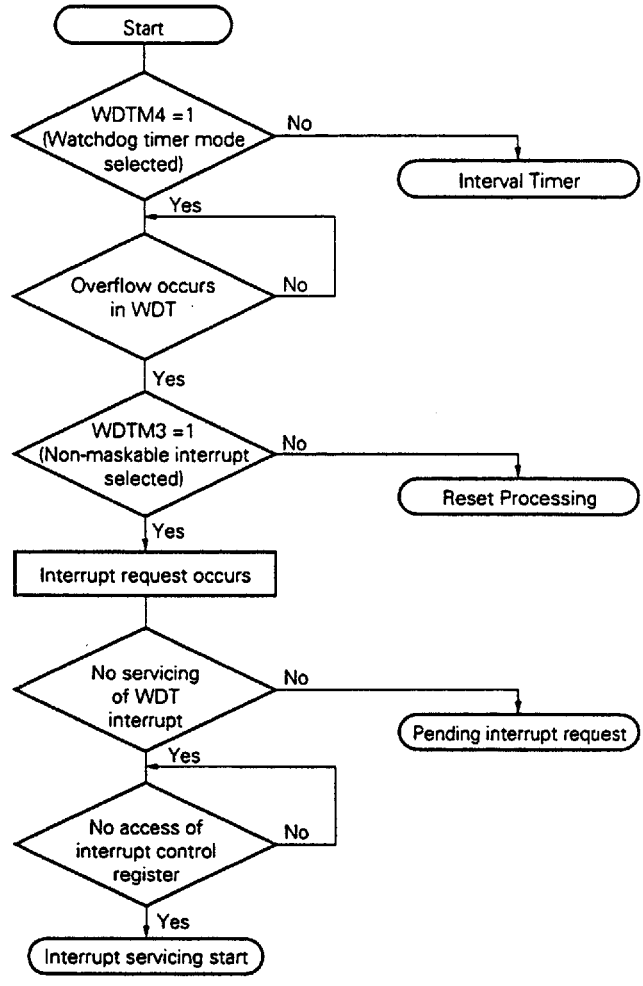
16.4.1 Non-maskable interrupt acknowledge operation

A non-maskable interrupt is unconditionally acknowledged if in an interrupt 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 acknowledged interrupt is saved in the stacks, PSW and PC, in that order, the IE and ISP flags are reset to 0, and the vector table contents 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. If a new non-maskable interrupt request is generated twice or more during non- maskable interrupt service program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt service program execution.

Fig. 16-9 Non-Maskable Interrupt Acknowledge Flowchart



Remarks 1. WDTM : Watchdog timer mode register
2. WDT : Watchdog timer

Fig. 16-10 Non-Maskable Interrupt Acknowledge Timing

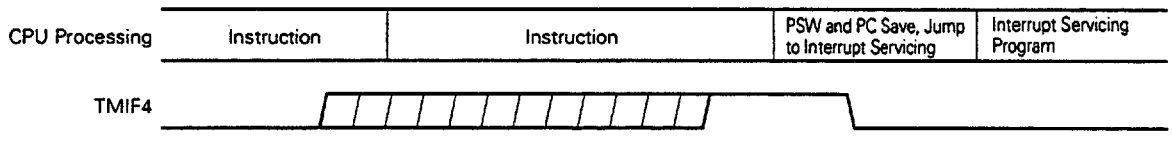
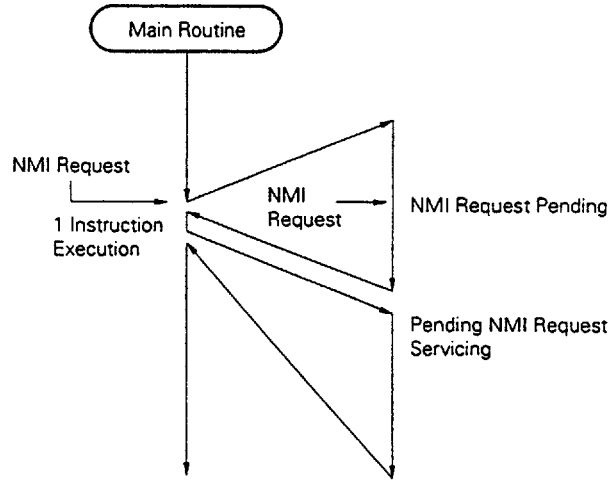
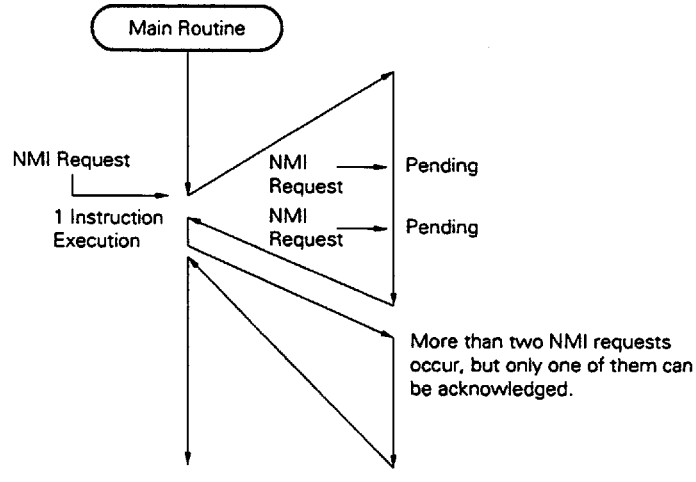


Fig. 16-11 Non-Maskable Interrupt Request Acknowledge Operation

(a) If a new 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



16.4.2 Maskable interrupt acknowledge operation

A maskable interrupt becomes acknowledgeable when an interrupt request is set to 1 and the interrupt MK flag is cleared to 0. A vectored interrupt is acknowledged in an interrupt enable state (with IE flag set to 1). However, a low-priority interrupt is not acknowledged during high-priority interrupt service (with ISP flag reset to 0).

Wait times from maskable interrupt request generation to interrupt servicing are as follows.

Table 16-3 Times from Maskable Interrupt Request Generation to Interrupt Service

	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 is maximized.

Remark 1 clock : $\frac{1}{f_{CPU}}$

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specify flag is acknowledged first. Two or more requests specified for the same priority, the default priorities apply.

Any pending interrupts are acknowledged when they become acknowledgeable.

Fig. 16-12 shows interrupt acknowledge algorithms.

If a maskable interrupt request is acknowledged, the acknowledged interrupt is saved in the stacks, PSW and PC, in that order, the IE flag is reset to 0, and the acknowledged interrupt priority specify flag contents are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into a branched to PC.

Return from the interrupt is possible with the RETI instruction.

Fig. 16-12 Interrupt Acknowledge Processing Algorithm

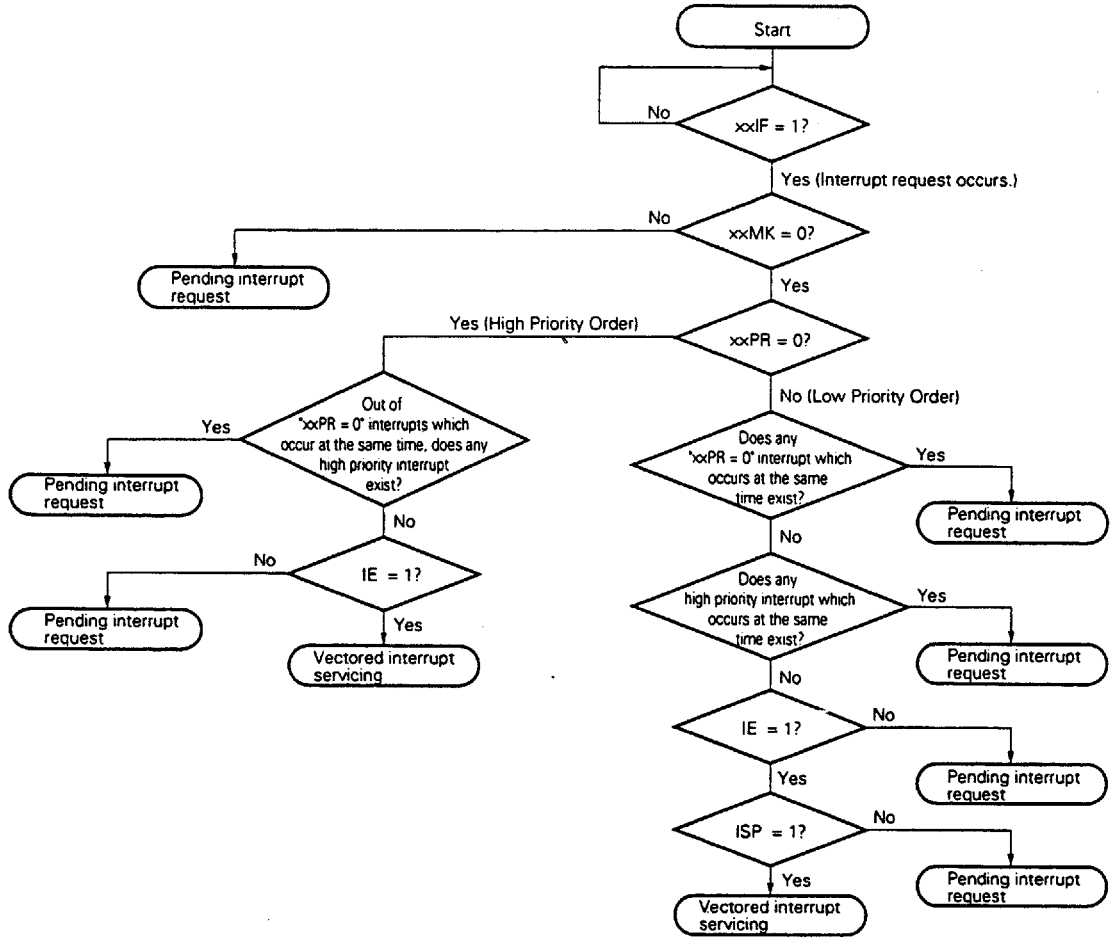


Fig. 16-13 Interrupt Acknowledge Timing (Minimum Time)

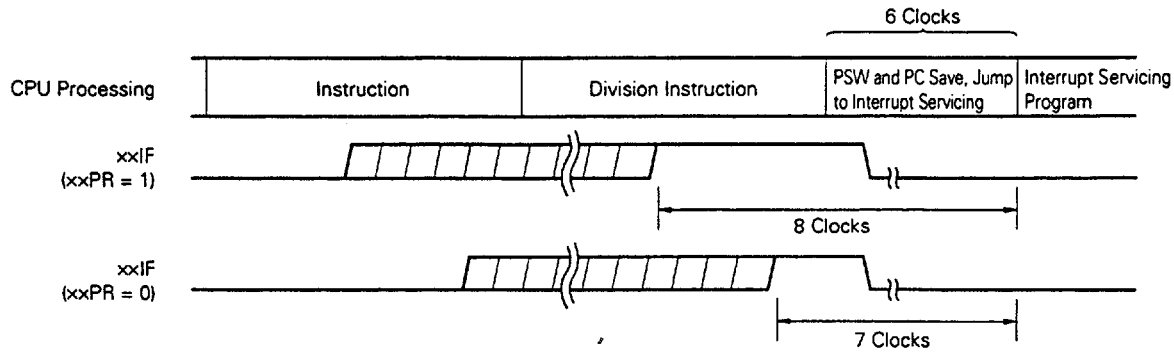
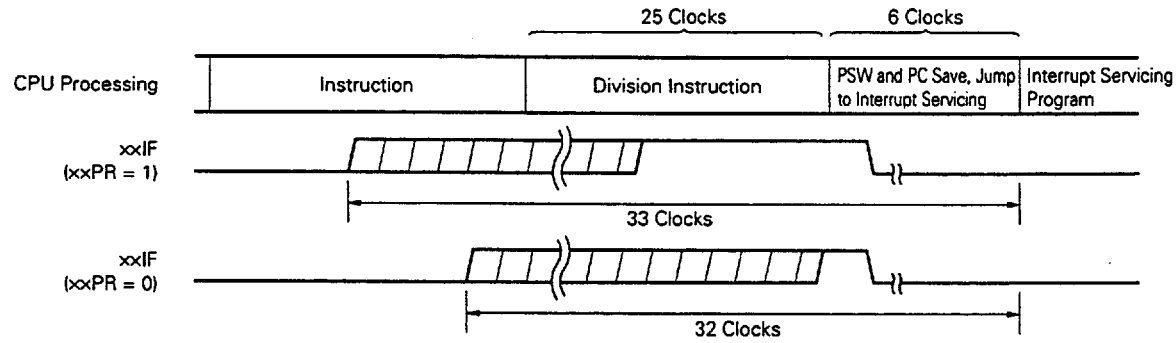


Fig. 16-14 Interrupt Acknowledge Timing (Maximum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$

16.4.3 Software interrupt acknowledge operation

A software interrupt is acknowledged by BRK instruction execution. It cannot be disabled.

If a software interrupt is acknowledged, it is saved in the stacks, PSW and PC, in that order, the IE flag is reset to 0 and the contents of the vector tables (003EH and 003FH) are loaded into PC and branched.

Reset from the software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for resetting from the software interrupt.

16.4.4 Multiple interrupt service

Multiple interrupt, in which another interrupt is acknowledged during execution of an interrupt, can be controlled by priorities.

Two types of priority control are available; control in the order of default priority and programmable priority control by setting the priority specify flag register (PR0). In the former, if two or more interrupts are generated simultaneously, interrupt servicing is carried out in accordance with the priority (default priority) preassigned to each interrupt request (see **Table 16-1**). In the later, interrupt requests are divided into a high-priority group and a low-priority group by setting the bits corresponding to PR0. The following are the interrupt requests enabled for multiple interrupts.

Table 16-4 Interrupt Requests Enabled for Multiple Interrupt during Interrupt Service

Multiple Interrupt Request / Interrupt being Acknowledged		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			xxPR = 0		xxPR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt processing		×	×	×	×	×
Maskable interrupt servicing	ISP = 0	O	O	×	×	×
	ISP = 1	O	O	×	O	×
Software interrupt servicing		O	O	×	O	×

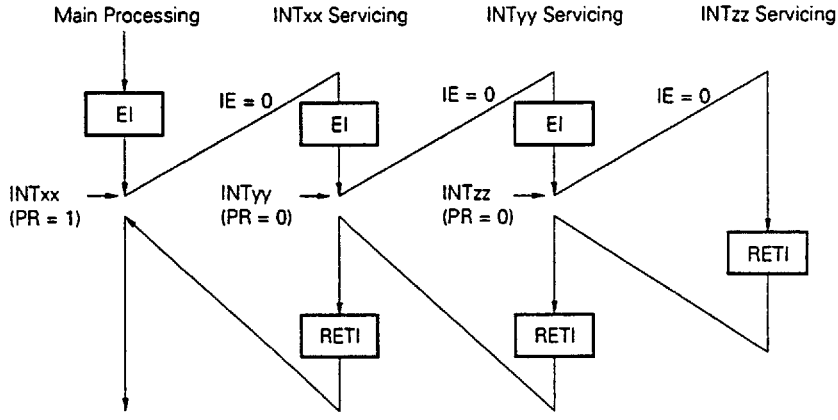
Remarks 1. O : Multiple interrupt enable
× : Multiple interrupt disable

2. ISP and IE are flags of PSW. ★
ISP = 0 : Interrupt with higher priority is processed.
ISP = 1 : No interrupt is accepted, or interrupt with low priority is processed.
IE = 0 : Interrupt disabled
IE = 1 : Interrupt enabled

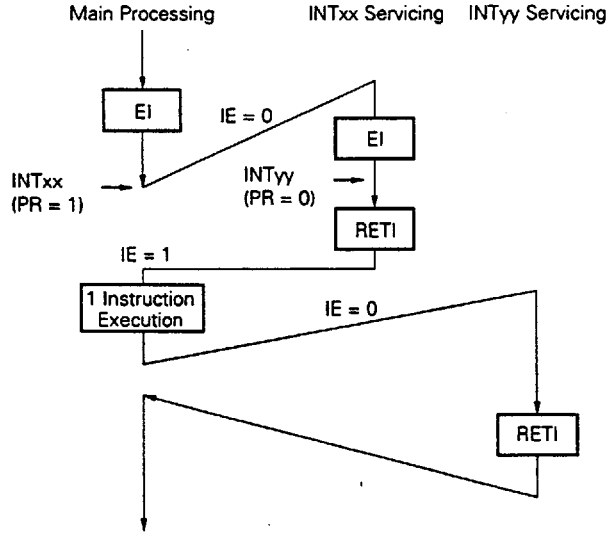
3. xxPR is a flag of PR0. ★
xxPR = 0 : High priority level
xxPR = 1 : Low priority level

Fig. 16-15 Multiple Interrupt Example

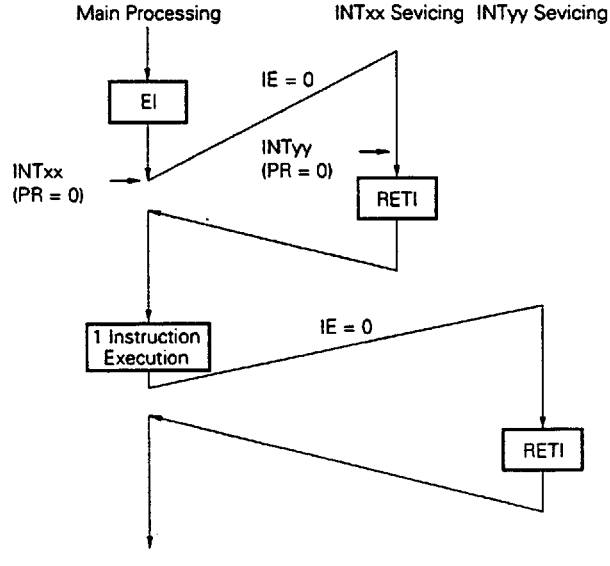
Example 1



Example 2



Example 3



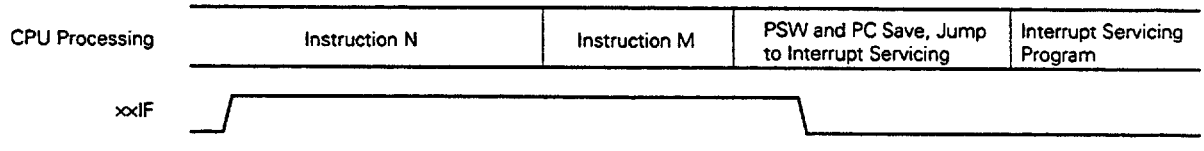
16.4.5 Interrupt pending

Interrupt acknowledge is temporarily made pending between one of the following instructions and the next instruction to be executed.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1/AND1/OR1/XOR1 CY, PSW. bit
- SET1/CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT/BF/BTCLR PSW. bit, $addr16
- EI
- DI
- Manipulate instructions for IF0, MK0, PR0 and INTM0 registers

Caution The IE flag is cleared to 0 with a software interrupt (by BRK instruction execution). Thus, if a maskable interrupt request is generated during BRK instruction execution, only non-maskable interrupt requests are acknowledged.

Fig. 16-16 Pending Interrupt Request



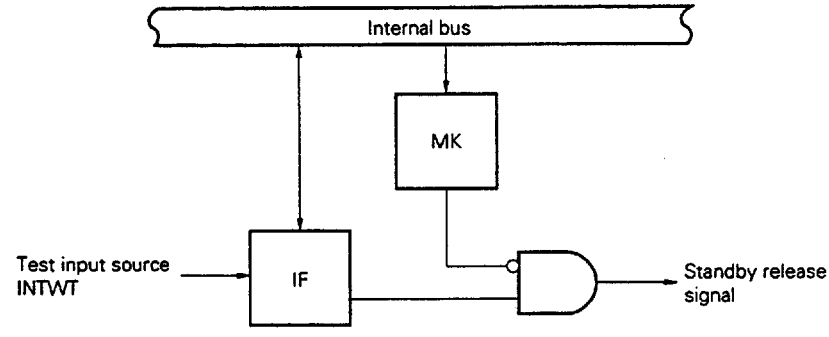
Remarks 1. Instruction N : Interrupt request reserve instruction
2. Instruction M : Instruction except interrupt reserve instructions
3. xxIF operation is not affected by xxPR values.

★ 16.5 Test Functions

The test input flag is set to 1 without vector processing performed.
A standby release signal is generated.
The following one test input source is provided. The basic configuration is as shown in Fig. 16-17.

Test Input Source		Internal/External
Name	Trigger	
INTWT	Overflow of watch timer	Internal

Fig. 16-17 Basic Configuration of Test Function



IF : Test input flag
MK : Test mask flag

16.5.1 Registers controlling test function

The following two registers control the test function:

- Interrupt request flag register 0H (IF0H)
- Interrupt mask flag register 0H (MK0H)

The names of the test input flag and test mask flag corresponding to the test input signal are as follows:

Test Input Signal Name	Test Input Flag	Test Mask Flag
INTWT	WTIF	WTMK

(1) Interrupt request flag register (IF0H)

This register indicates whether an overflow of the watch timer is detected.

IF0H is set by using a 1-, 8-, or 16-bit memory manipulation instruction.

The contents of this register are cleared to 00H when the $\overline{\text{RESET}}$ signal has been input.

Fig. 16-18 Format of Interrupt Request Flag Register 0H

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IF0H	0	0	WTIF	KSIF	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

WTIF	Watch timer overflow detection flag
0	Not detected
1	Detected

(2) Interrupt mask flag register (MK0H)

This register enables or disables releasing the standby mode by the watch timer.

MK0H is set by using a 1-, 8-, or 16-bit memory manipulation instruction.

The contents of this register are set to FFH when the $\overline{\text{RESET}}$ signal has been input.

Fig. 16-19 Format of Interrupt Mask Flag Register 0H

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
MK0H	1	1	WTMK	KSMK	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

WTMK	Controls standby mode by watch timer
0	Enables releasing of standby mode
1	Disables releasing of standby mode

16.5.2 Accepting test input signal

The INTWT flag is set when an overflow occurs in the watch timer. By checking the INTWT flag in a cycle shorter than the overflow cycle of the watch timer, a watch function can be realized.

[MEMO]

PAGE(S) INTENTIONALLY BLANK

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is intended 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. In this mode, current consumption cannot be decreased as in the STOP mode. The HALT mode is valid to restart immediately upon interrupt request and to carry out intermittent operations like clock operations.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops and the whole system stops. CPU current consumption can be considerably decreased.

Data memory low-voltage hold (down to V_{DD} = 2 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 necessary 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.

However, because a wait time is necessary 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 any mode, all the contents of the register, flag and data memory just before standby mode setting 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 proceeding to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

17.1.2 Standby function control register

A wait time after the STOP mode is cleared upon interrupt request till the oscillation stabilizes is controlled with the oscillation stabilization time select register (OSTS).

OSTS is set with an 8-bit memory manipulate instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H. Thus, it takes 2^{17}/fx till to clear the STOP mode by $\overline{\text{RESET}}$ input.

Fig. 17-1 Oscillation Stabilization Time Select Register Format

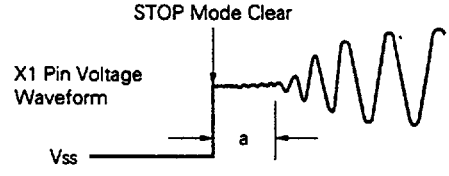
Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Oscillation Stabilization Time Selection
0	0	0	2^{12}/fx (819.2 μs)
0	0	1	2^{14}/fx (3.28 ms)
0	1	0	2^{15}/fx (6.55 ms)
0	1	1	2^{16}/fx (13.1 ms)
1	0	0	2^{17}/fx (26.2 ms)
Other than the above			Setting prohibited

Remarks 1. fx : Main system clock oscillation frequency

★ **2.** Values in parentheses when operated at fx = 5.0 MHz

Caution The wait time after STOP mode clear does not include a time (see a below) from STOP mode clear to clock oscillation start whether by $\overline{\text{RESET}}$ input or interrupt generation.



17.2 Standby Function Operations

17.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating status in the HALT mode is described below.

Table 17-1 Operation Status in HALT Mode ★

Setting of HALT Mode / Item		Execution of HALT Instruction during Main System Clock Operation		Execution of HALT Instruction during Subsystem Clock Operation	
		When subsystem clock is not used	When subsystem clock is used	When main system clock continues oscillation	When main system clock stops oscillation
Clock generator circuit		Both main system clock and subsystem clock can oscillate. Clock supply to CPU is stopped.			
CPU		Stops operation			
Port (output latch)		Retains status immediately before execution of HALT instruction			
16-bit timer/event counter		Can operate			Stops operation
8-bit timers/event counters					Can operate when TI1 and TI2 are selected as count clocks
Watchdog timer					Stops operation
A/D converter					Stops operation
Watch timer		Can operate when $f_X/2^7$ are selected as as count clocks	Can operate		Can operate when f_{XT} are selected as count clocks
FIP controller/driver		Operation disabled			
Serial interface		Can operate			Can operate when external $\overline{SCK}$ is used
External interrupt	INTP0	Can operate when clock to peripheral hardware ($f_X/2^6$, $f_X/2^7$) is selected as sampling clock			Stops operation
	INTP1-INTP3	Can operate			

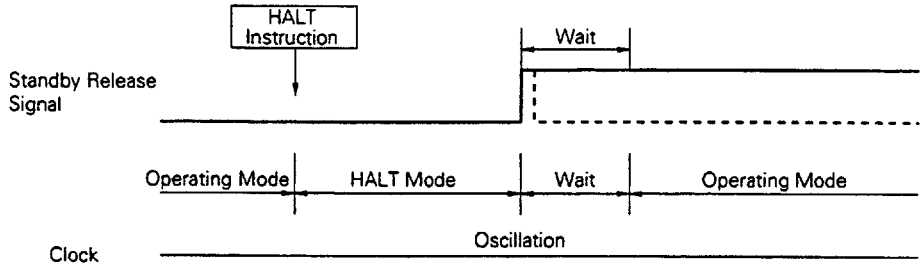
(2) HALT mode clear

The HALT mode can be cleared with the following four 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 disabled, the next address instruction is executed.

Fig. 17-2 HALT Mode Clear upon Interrupt Generation



Remarks 1. The broken line indicates the case when the request which has cleared the standby status is acknowledged.

2. Wait time is described as follows

- Branch to vector : 8 to 9 clocks
- No branch to vector : 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.

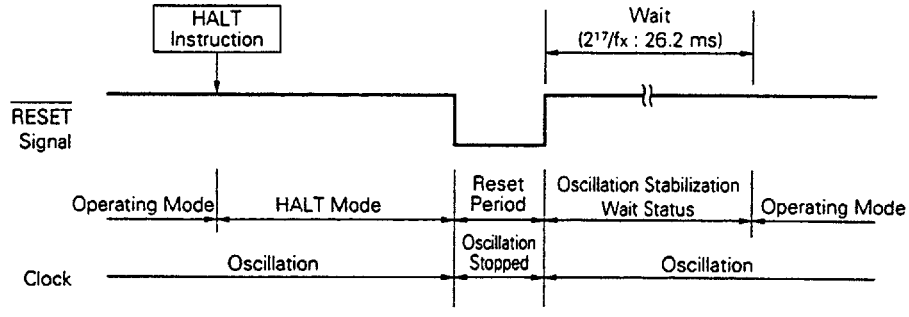
(c) Clear by unmasked test input

The HALT mode is cleared by unmasked test input and the next address instruction of the HALT instruction is executed.

(d) Clear by $\overline{\text{RESET}}$ input

As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Fig. 17-3 Clear by HALT Mode $\overline{\text{RESET}}$ Input



Remark fx = 5.0 MHz

Table 17-2 Operation after HALT Mode Clear

Clear 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
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	HALT mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark ×: Don't care

17.2.2 STOP mode

(1) STOP mode set and operating state

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, X1 input is internally short-circuited to Vss (ground potential) to prevent the crystal resonator from leaking. Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

2. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operating mode is set.

The operating status in the STOP mode is described below.

★ **Table 17-3 Operating Status in STOP Mode**

Item \ Setting of STOP Mode		When Subsystem Clock Is Used	When Subsystem Clock Is Not Used
Clock Generator		Only main system clock stops oscillation	
CPU		Operation stop	
Output port (output latch)		Status immediately before STOP instruction execution is held	
16-bit timer/event counter		Operation stop	
8-bit timers/event counters		Operable only when TI1 and TI2 are selected for the count clock	
Watchdog timer		Operation stop	
A/D converter			
Watch timer		Operate only when f_{XT} is selected for the count clock	Operation stop
FIP controller/driver		Operation disabled	
Serial interface		Operable only when an external input clock is selected for serial clock	
External interrupt	INTP0	Not operable	
	INTP1-INTP3	Operable	

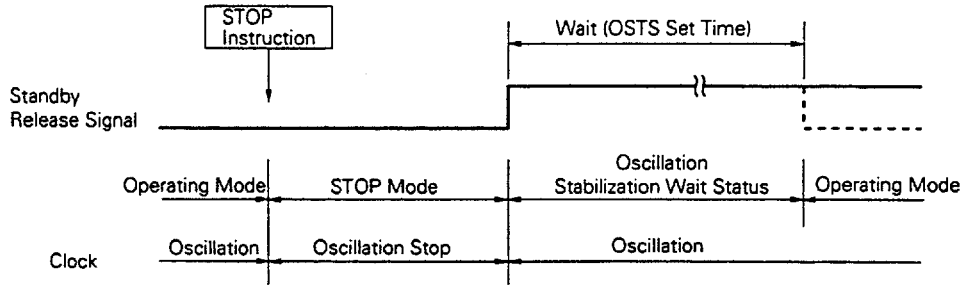
(2) STOP mode clear

The STOP 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 STOP mode. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Fig. 17-4 STOP Mode Clear upon Interrupt Generation



Remark The broken line indicates the case when the request which has cleared the standby status is acknowledged.

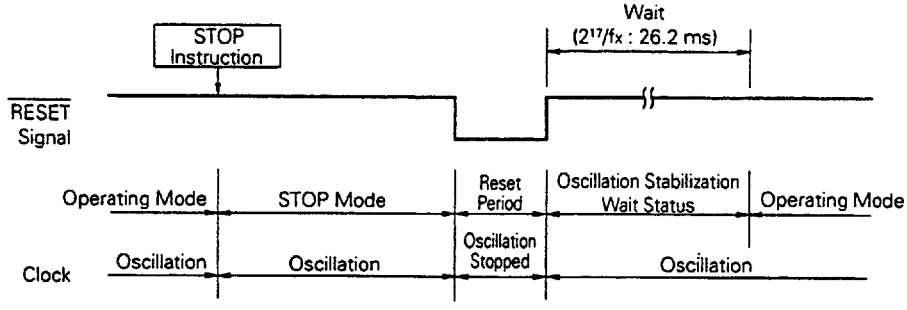
(b) Clear by unmasked test input

The STOP mode is cleared by unmasked test input. After the lapse of oscillation stabilization time, the next address instruction of the STOP instruction is executed.

(c) Clear by $\overline{\text{RESET}}$ input

The STOP mode is cleared and after the lapse of oscillation stabilization time, reset operation is carried out.

★ **Fig. 17-5 Clear by STOP Mode $\overline{\text{RESET}}$ Input**



Remark fx=5.0 MHz

Table 17-4 Operation after STOP Mode Clear

Clear 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
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark ×: Don't care

CHAPTER 18 RESET FUNCTION

18.1 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 as shown in Table 18-1. 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 $\overline{\text{RESET}}$ input, 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 **Figs. 18-2** to **18-4**).

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.

Fig. 18-1 Block Diagram of Reset Function

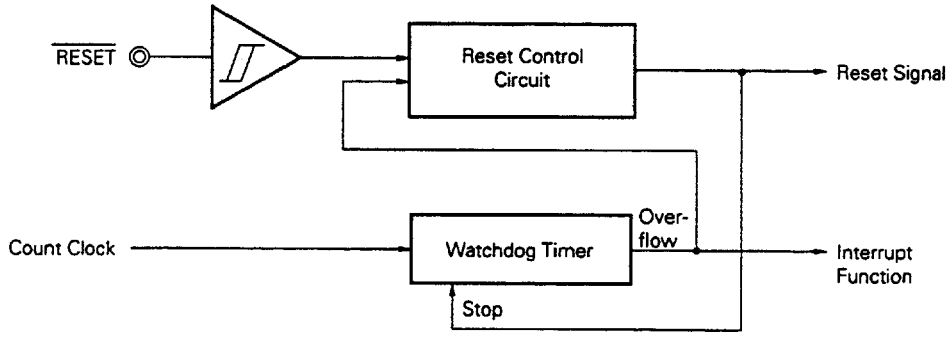


Fig. 18-2 Timing of Reset by $\overline{\text{RESET}}$ Input

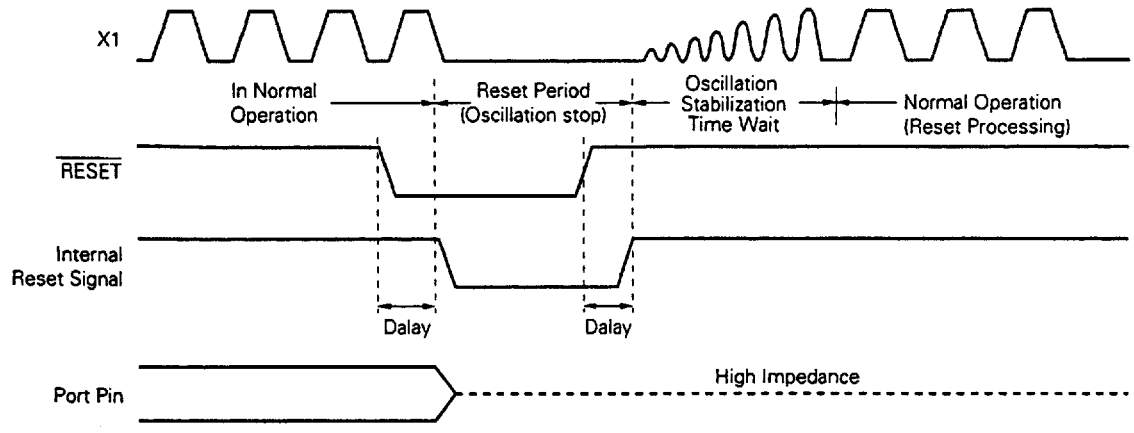


Fig. 18-3 Timing of Reset due to Watchdog Timer Overflow

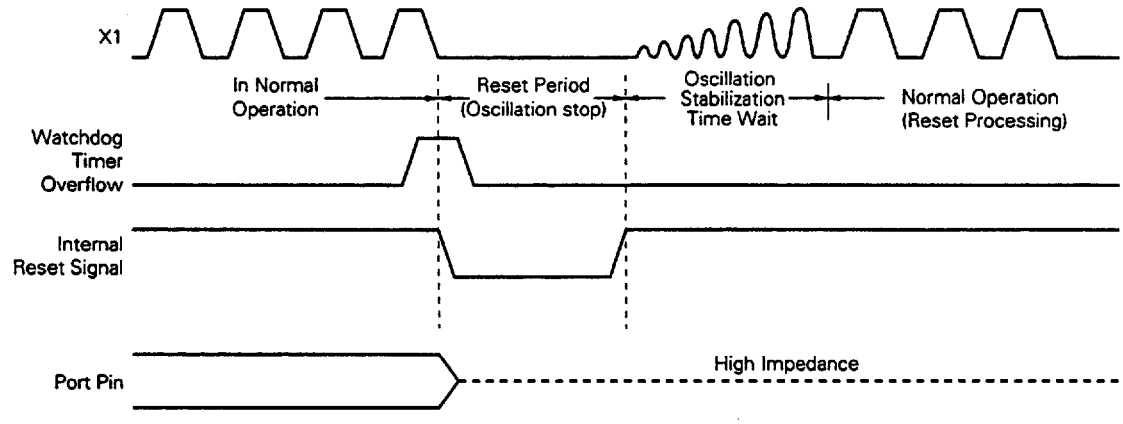


Fig. 18-4 Timing of Reset Input in STOP Mode by $\overline{\text{RESET}}$ Input

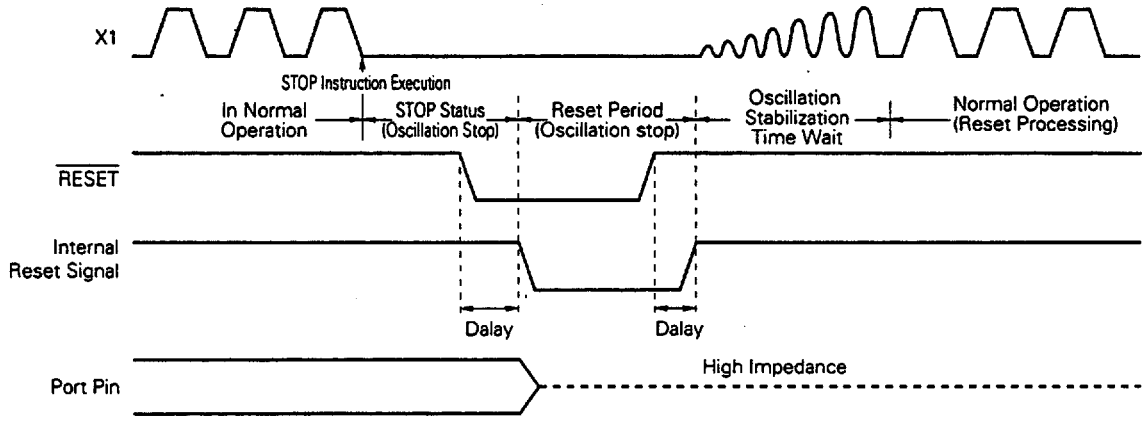


Table 18-1 Hardware Status after Reset (1/2)

Hardware		Status after Reset
Program counter (PC)[Note 1]		The contents of reset vector tables (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)	Ports 0 to 3, ports 8 to 11 (P0 to P3, P8 to P11)	00H
Port mode register	(PM0)	1FH
	(PM1, PM2, PM3, PM11)	FFH
Pull-up resistor option register (PUO)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)[Note 3]		48H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	00H
	Compare register (CR00)	Undefined
	Capture register (CR01)	Undefined
	Clock select register (TCL0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
8-bit timers/event counters	Timer registers (TM1, TM2)	00H
	Compare registers (CR10, CR20)	Undefined
	Clock select register (TCL1)	00H
	Mode control registers (TMC1, TMC2)	00H
	Output control register (TOC1)	00H
Watch timer	Clock select register (TCL2)	00H
Watchdog timer		
	Mode register (WDTM)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remains unchanged after reset.

2. The post-reset status is held in the standby mode.

3. Only the μPD78P024 incorporates IMS.

Table 18-1 Hardware Status after Reset (2/2)

Hardware		Status after Reset
Serial interface	Clock select register (TCL3)	88H
	Shift registers (SIO0 and SIO1)	Undefined
	Mode registers (CSIM0 and CSIM1)	00H
	Serial bus interface control register (SBIC)	00H
	Slave address register (SVA)	Undefined
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
FIP controller/driver	Display mode register 0 (DSPM0)	00H
	Display mode register 1 (DSPM1)	00H
Interrupt	Request flag register (IF0)	00H
	Mask flag register (MK0)	FFH
	Priority specification flag register (PR0)	FFH
	External interrupt mode register (INTM0)	00H
	Sampling clock select register (SCS)	00H

CHAPTER 19 μPD78P024

The μPD78P024 is a product which incorporates a one-time PROM enabled for read only in one time or EPROM enabled for program write, erase and rewrite. Table 19-1 lists differences between μPD78P024 and mask ROM product.

Table 19-1 Difference between μPD78P044 and Mask ROM Product

Item	μPD78P024	Mask ROM Product
IC pin	None	Available
V_{PP} pin	Available	None
Memory size switching register	Available	None
Mask option for FIP0 to FIP15	Pull-down resistor is incorporated.	Pull-down resistor can be incorporated by mask option.
Mask option for FIP18 to FIP25	Pull-down resistor is not incorporated.	
Mask option for port 3	Pull-down resistor is not incorporated.	

19.1 Memory Size Switching Register

The μPD78P024 can select the internal memory with the memory size switching register (IMS). The same memory mapping as that of a mask ROM product with a different internal memory is possible.

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

$\overline{\text{RESET}}$ input sets IMS to 48H.

Cautions 1. Only the μPD78P024 incorporates IMS. Mask ROM products do not incorporate it.

2. When using a mask ROM model (μPD78023, 78024), do not set a value other than that shown in Table 19-2 to IMS. ★

★ **Fig. 19-1 Memory Size Switching Register Format**

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	48H	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
0	1	1	0	24 K bytes
1	0	0	0	32 K bytes
Other than the above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
0	1	0	512 K bytes
Other than the above			Setting prohibited

The IMS set values which set the same memory map as the mask ROM product are shown in Table 19-2.

Table 19-2 Memory Size Switching Register Set Value

Target Mask ROM Product	IMS Set Value
μPD78023	46H
μPD78024	48H

19.2 PROM Programming

The μPD78P024 incorporates a 32K-byte PROM as program memory. When programming, set the PROM programming mode with the V_{PP} and $\overline{RESET}$ pins. Refer to **1.5 Pin Configuration (2) PROM programming mode** for details of the recommended conditions for unused pins.

Caution Write program in an address range of 000H to 7FFFH (specify the last address 7FFFFH). Program cannot be written to the PROM with a PROM programmer that cannot specify a write address. ★

19.2.1 Operating mode

When a voltage of +5 V or +12.5 V is applied to the V_{PP} pin and a low level is input to the $\overline{RESET}$ pin, the PROM programming mode is set. When the $\overline{CE}$, $\overline{OE}$ and $\overline{PGM}$ pins are set, the PROM programming mode is set to the operating mode shown in Table 19-3.

The PROM contents can be read by setting the read mode.

Table 19-3 Operating Modes for PROM Programming

Operating mode \ Pin	$\overline{RESET}$	V_{PP}	V_{DD}	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	D0-D7
Page data latch	L	+12.5 V	+6.5 V	H	L	H	Data input
Page write				H	H	L	High impedance
Byte write				L	H	L	Data input
Program verify				L	L	H	Data output
Program inhibit				×	H	H	High impedance
				×	L	L	
Read		+5 V	+5 V	L	L	H	Data output
Output disable				L	H	×	High impedance
Standby				H	×	×	High impedance

Remark ×: L or H

(1) Read mode

The read mode is entered by setting $\overline{CE}$ = L, $\overline{OE}$ = L.

(2) Output disable mode

Data output becomes high impedance and the output disable mode is entered by setting $\overline{OE}$ = H.

Therefore, when two or more μPD78P024s are connected to the data bus, controlling the $\overline{OE}$ pin allows data to be read from one arbitrary device.

(3) Standby mode

The standby mode is entered by setting $\overline{CE}$ = H.

In this mode, data output becomes high imped ance irrespective of the $\overline{OE}$ state.

(4) Page data latch mode

The page data latch mode is entered by setting $\overline{CE}$ = H, $\overline{PGM}$ = H and $\overline{OE}$ = L early in the page write mode.

In this mode, data of 4 bytes/page is latched in the address/data latch circuit

(5) Page write mode

After address and data of 4 byte/page are latched in the page data latch mode, applying 0.1 ms program pulse (active low) to the $\overline{PGM}$ pin with $\overline{CE}$ = H and $\overline{OE}$ = H executes a page write. Then program verify can be performed by setting $\overline{CE}$ = L and $\overline{OE}$ = L.

If programming is not possible by one program pulse, write and verify should be repeated X times ($X \leq 10$).

(6) Byte write mode

Applying 0.1 ms program pulse (active low) to the $\overline{PGM}$ pin with $\overline{CE}$ = L and $\overline{OE}$ = H executes a byte write. Then program verify can be performed by setting $\overline{OE}$ = L.

If programming is not possible by one program pulse, write and verify should be repeated X times ($X \leq 10$).

(7) Program verify mode

The program verify mode is entered by setting $\overline{CE}$ = L, $\overline{PGM}$ = H and $\overline{OE}$ = L. After the write, whether or not the write has been performed correctly should be checked in this mode.

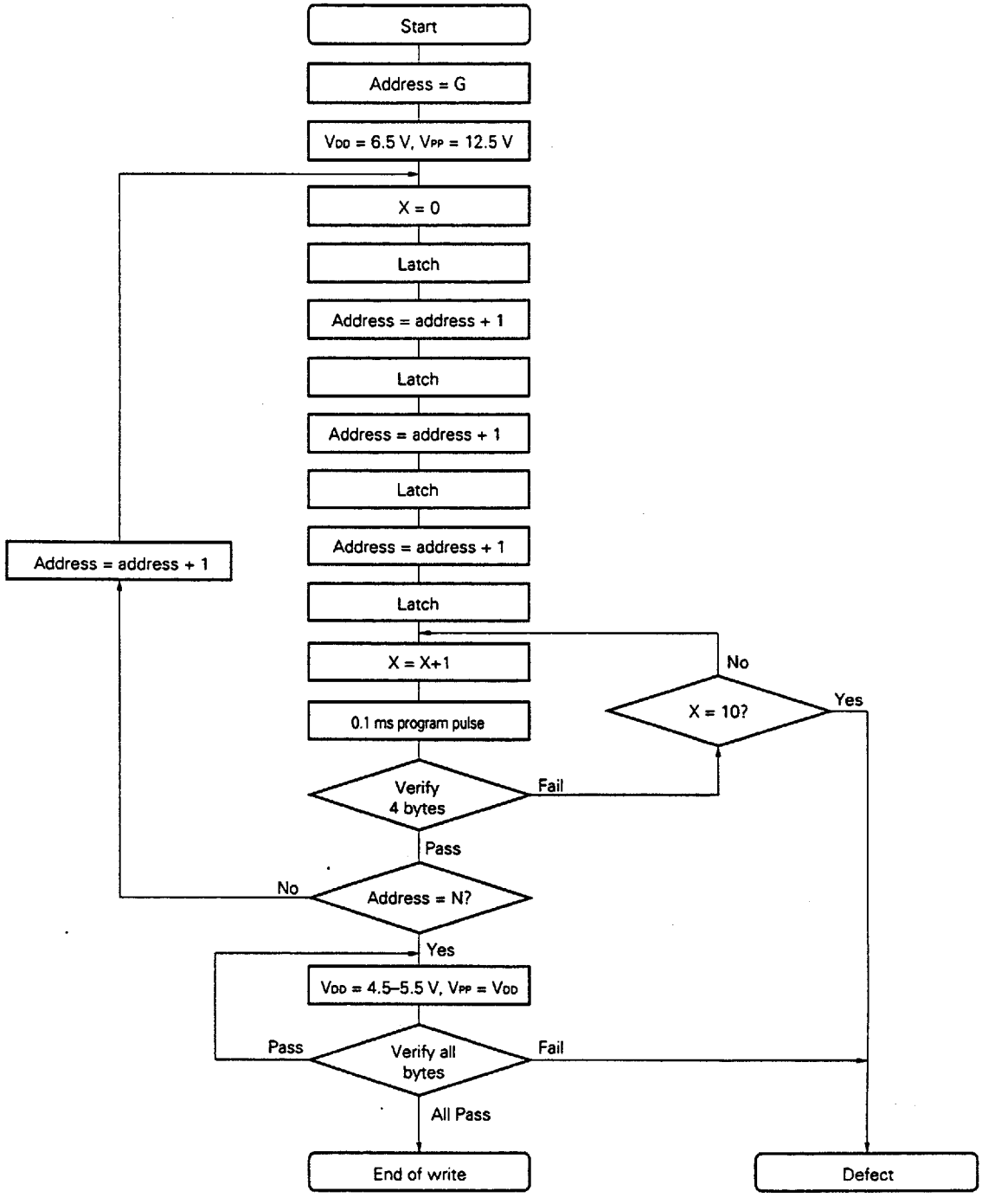
(8) Program inhibit mode

The program inhibit mode is used when pins $\overline{OE}$, V_{PP} and D0 to D7 of two or more μPD78P024s are connected in parallel and a write is performed to one of those devices.

When performing a write, the above page write mode or byte write mode is used. At this time, no write is performed to a device in which the $\overline{PGM}$ pin has been driven high.

19.2.2 PROM programming procedure

Fig. 19-2 Page Program Mode Flowchart



Remarks 1. G = Start address

2. N = Last address of program

Fig. 19-3 Page Program Mode Timing

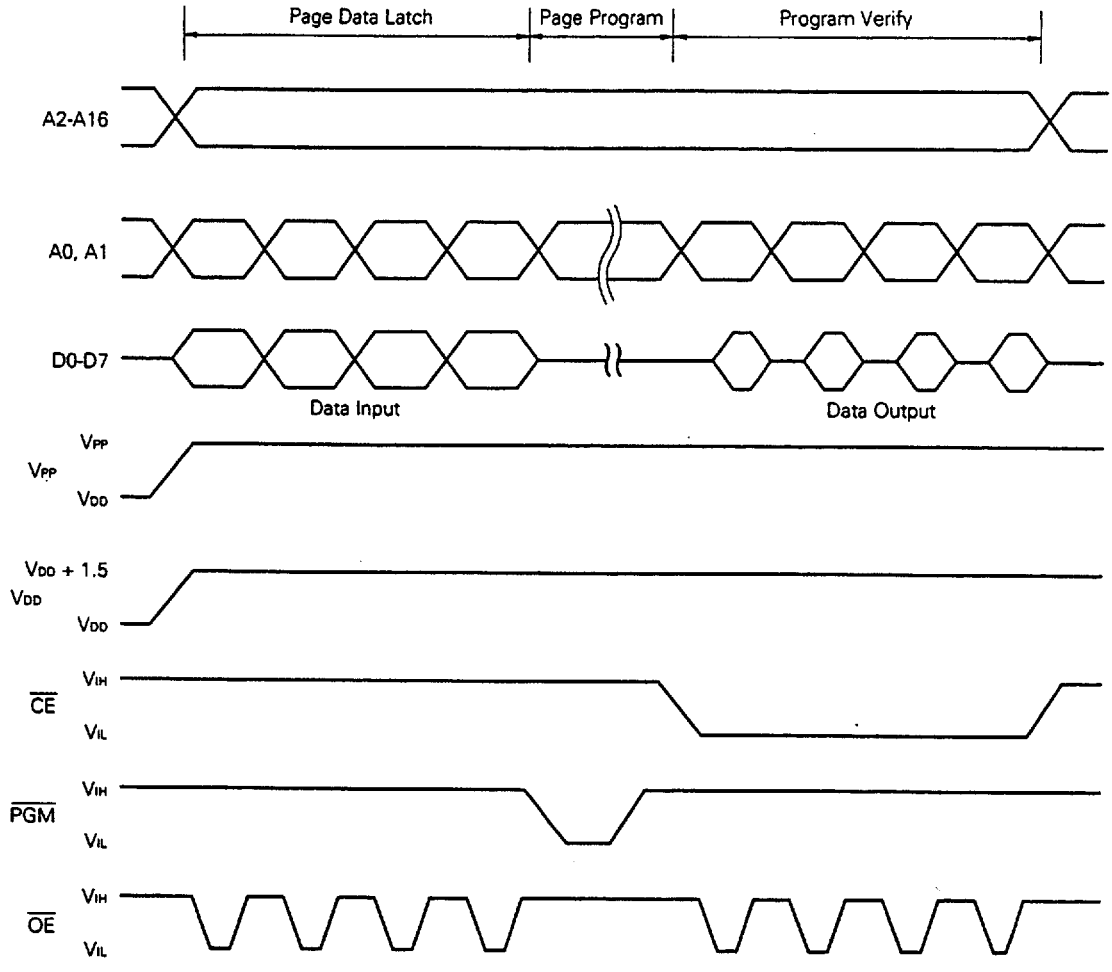
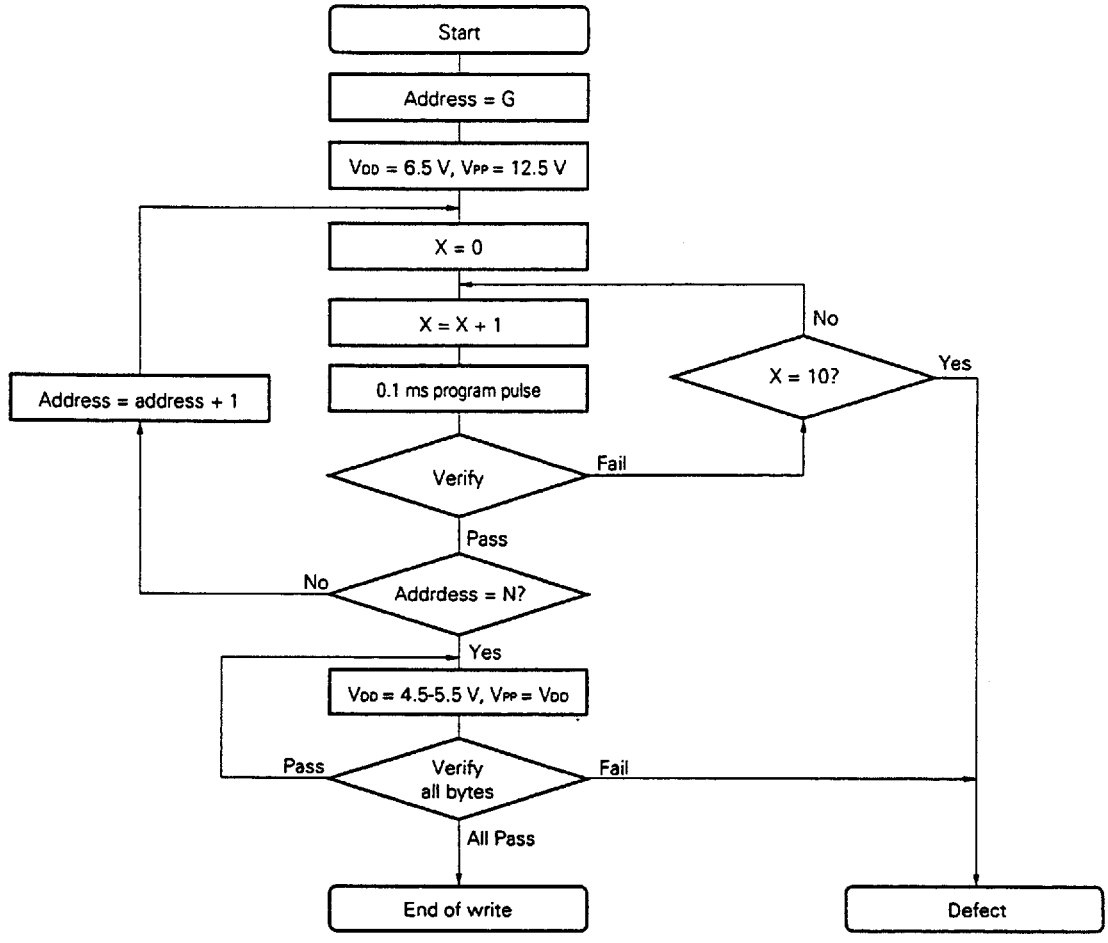


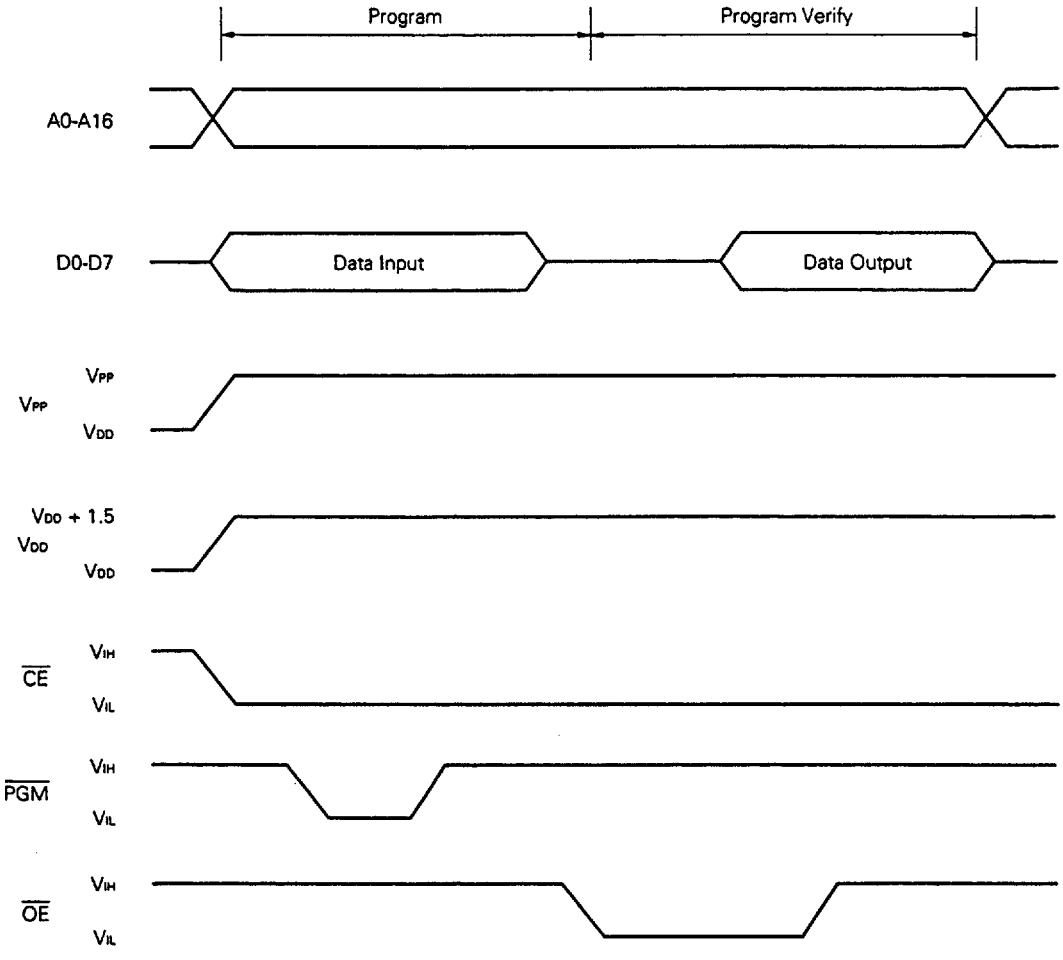
Fig. 19-4 Byte Program Mode Flowchart



Remarks 1. G = Start address

2. N = Last address of program

Fig. 19-5 Byte Program Mode Timing



Cautions 1. V_{DD} should be applied before V_{PP} and cut after V_{PP}.

2. Ensure that V_{PP}, including overshoot, does not exceed +13.5 V.

3. Removing and reinserting while +12.5 V is being applied to V_{PP} may cause an adverse effect on reliability.

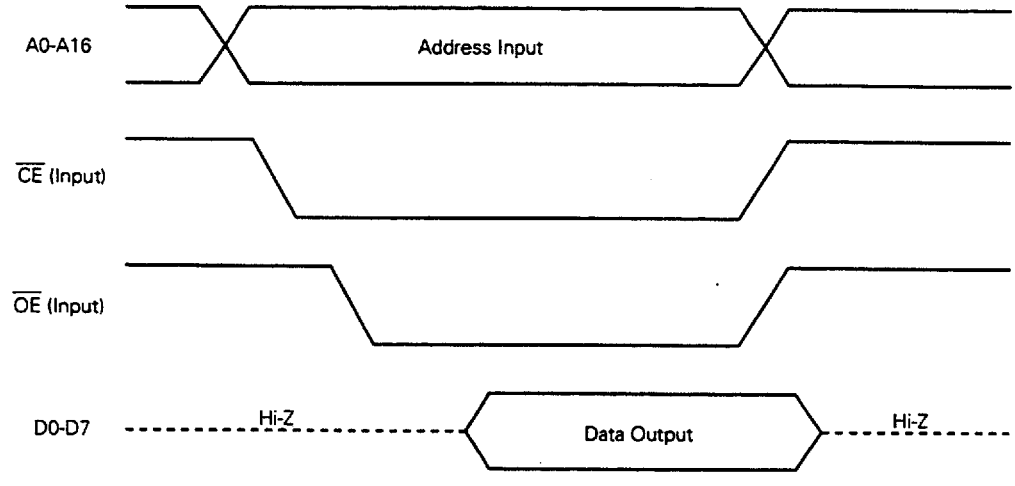
19.2.3 PROM reading procedure

The PROM contents can be read to the external data bus (D0 to D7) using the following procedure.

(1) Fix the $\overline{\text{RESET}}$ pin low, supply +5 V to the V_{PP} pin and handle the other unused pins as shown in **1.5 Pin Configuration (Top View) (2) PROM programming mode**.
(2) Supply +5 V to the V_{DD} and V_{PP} pins.
(3) Input the address of the data to be read to pins A0 to A16.
(4) Read mode.
(5) Output data to pins D0 to D7.

The timings of the above steps (2) to (5) are as shown in Fig. 19-6.

Fig. 19-6 PROM Read Timing



19.3 Erasure Method (μPD78P024DW Only)

In the μPD78P024DW, the data contents written in the program memory can be erased (FFH) and rewritten.

The data contents can be erased by exposure to rays with wavelength shorter than approximately 400 nm via the erasure window. Normally, ultra-violet rays with wave length of 254 nm are irradiated. The irradiation amount required for complete erasure is as follows.

- UV intensity x erasure time: 15W•s/cm^2 or more
- Erasure time: 15 to 20 minutes (when a 12,000 μW/cm^2 UV lamp is used. However, the erasure time may be increased due to deterioration of the UV lamp, dirt or stains on the package window surface).

When performing erasure, ensure that the UV lamp should be placed 2.5 cm or more from the erasure window. If a filter is provided with the UV lamp, remove it before irradiation.

19.4 Erasure Window Seal (μPD78P024DW Only)

To protect from miserasure by rays other than that of the lamp for erasing EPROM contents, or to protect internal circuit other than EPROM from misoperating by rays, stick a protection seal on the erasure window when EPROM contents erasure is not performed.

19.5 One-Time PROM Products Screening

The one-time PROM product (μPD78P024CW, 78P024GF-3BE) cannot be tested completely by NEC before it is shipped, because of its structure.

It is recommended to perform screening to verify PROM after writing necessary data and performing high-temperature storage under the condition below.

Storage Temperature	Storage Time
125°C	24 hours

Under the trademark QTOP™ Microcomputer, NEC will write in stamp, screen, and verify your One-Time PROM on a fee-service basis. We are currently developing this service for the μPD78P024. For further information, please contact your dealer.

CHAPTER 20 INSTRUCTION SET OVERVIEW

This chapter presents a list of the instructions of the μPD78024 subseries. For the detailed operation and machine language code (op code) of each instruction, refer to **78K/0 SERIES USER'S MANUAL - INSTRUCTION (IEU-1372)**.

20.1 Legend

20.1.1 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 are described as they are. Each symbol has the following meaning.

- # : Immediate data specification
- ! : Absolute address specification
- $: Relative address specification
- [ ] : Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers, r and rp, either 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 20-1 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 symbols (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 address 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-RB3

Note FFD0H-FFDFH are not addressable.

Remark Refer to **Table 3-2 List of Special Function Registers** for symbols of special function registers.

20.1.2 Description of "operation" column

A : A register; 8-bit accumulator
X : X register
B : B register
C : C register
D : D register
E : E register
H : H register
L : L register
AX : AX register pair; 16-bit accumulator
BC : BC register pair
DE : DE register pair
HL : HL register pair
PC : Program counter
SP : Stack pointer
PSW : Program status word
CY : Carry flag
AC : Auxiliary carry flag
Z : Zero flag
RBS : Register bank select flag
IE : Interrupt request enable flag
NMIS : Flag indicating non-maskable interrupt servicing in progress
( ) : Memory contents indicated by address or register contents in parentheses
x_H, x_L : Higher 8 bits or lower 8 bits of 16-bit register
$\wedge$: Logical product (AND)
$\vee$: Logical sum (OR)
$\veebar$: Exclusive logical sum (exclusive OR)
$\overline{\quad}$: Inverted data
addr16 : 16-bit immediate data or label
jdisp8 : Signed 8-bit data (displacement value)

20.1.3 Description of "flag operation" column

(Blank) : Unchanged
0 : Cleared to 0
1 : Set to 1
× : Set/cleared according to the result
R : Previously saved value is restored

20.2 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	A ← (addr16)	
		!addr16, A	3	8	9	(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	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL + byte]	2	8	9	A ← (HL + byte)	
		[HL + byte], A	2	8	9	(HL + byte) ← A	
		A, [HL + B]	1	6	7	A ← (HL + B)	
		[HL + B], A	1	6	7	(HL + B) ← A	
		A, [HL + C]	1	6	7	A ← (HL + C)	
		[HL + C], A	1	6	7	(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	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10	A ↔ (HL + B)	
		A, [HL + C]	2	8	10	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or in the instruction with no data access
2. When an area except the internal high-speed RAM area is accessed
3. Except r = A

Remark 1 instruction clock cycle is 1 CPU clock cycle (f_{CPU}) selected by PCC.

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	AX ← (addr16)			
		!addr16, AX	3	10	12	(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	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9	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	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	×	×	×

Notes **1.** When the internal high-speed RAM area is accessed or in the 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

Remark 1 instruction clock cycle is 1 CPU clock (f_{CPU}) selected by PCC.

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	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	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	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	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	A ← A∧(addr16)	×		
		A, [HL]	1	4	5	A ← A∧[HL]	×		
		A, [HL + byte]	2	8	9	A ← A∧[HL + byte]	×		
		A, [HL + B]	2	8	9	A ← A∧[HL + B]	×		
		A, [HL + C]	2	8	9	A ← A∧[HL + C]	×		

Notes 1. When the internal high-speed RAM area is accessed or in the instruction with no data access
2. When an area except the internal high-speed RAM area is accessed
3. Except r = A

Remark 1 instruction clock cycle is 1 CPU clock (f_{CPU}) selected by PCC.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	4	–	A ← A∨byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨byte	×		
		A, r Note 3	2	4	–	A ← A∨r	×		
		r, A	2	4	–	r ← r∨A	×		
		A, saddr	2	4	5	A ← A∨(saddr)	×		
		A, !addr16	3	8	9	A ← A∨(addr16)	×		
		A, [HL]	1	4	5	A ← A∨(HL)	×		
		A, [HL + byte]	2	8	9	A ← A∨(HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A∨(HL + B)	×		
		A, [HL + C]	2	8	9	A ← A∨(HL + C)	×		
	XOR	A, #byte	2	4	–	A ← A⊻byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)⊻byte	×		
		A, r Note 3	2	4	–	A ← A⊻r	×		
		r, A	2	4	–	r ← r⊻A	×		
		A, saddr	2	4	5	A ← A⊻(saddr)	×		
		A, !addr16	3	8	9	A ← A⊻(addr16)	×		
		A, [HL]	1	4	5	A ← A⊻(HL)	×		
		A, [HL + byte]	2	8	9	A ← A⊻(HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A⊻(HL + B)	×		
		A, [HL + C]	2	8	9	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	A – (addr16)	×	×	×
		A, [HL]	1	4	5	A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A – (HL + C)	×	×	×

Notes **1.** When the internal high-speed RAM area is accessed or in the instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except r = A

Remark 1 instruction clock cycle is 1 CPU clock (f_{CPU}) selected by PCC.

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	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjust-ment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipula-tion	**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	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	(HL) . bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or in the instruction with no data access
2. When an area except the internal high-speed RAM area is accessed

Remark 1 instruction clock cycle is 1 CPU clock (f_{CPU}) selected by PCC.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulation	**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	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	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	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	(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	(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 in the instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark 1 instruction clock cycle is 1 CPU clock (f_{CPU}) selected by PCC.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$			
	CALLF	!addr11	2	5	–	$(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP-2$			
	CALLT	[addr5]	1	6	–	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$			
	BRK		1	6	–	$(SP-1) \leftarrow$ PSW, $(SP-2) \leftarrow (PC+1)_H$, $(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (003FH), $PC_L \leftarrow$ (003EH), $SP \leftarrow SP-3$, IE ← 0			
	RET		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	RETI		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, PSW ← (SP + 2), $SP \leftarrow SP+3$, NMIS ← 0	R	R	R
	RETB		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, PSW ← (SP + 2), $SP \leftarrow SP+3$	R	R	R
Stack manipulation	**PUSH**	PSW	1	2	–	$(SP-1) \leftarrow$ PSW, $SP \leftarrow SP-1$			
		rp	1	4	–	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$			
	POP	PSW	1	2	–	PSW ← (SP), $SP \leftarrow SP+1$	R	R	R
		rp	1	4	–	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
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 in the instruction with no data access
2. When an area except the internal high-speed RAM area is accessed

Remark 1 instruction clock cycle is 1 CPU clock (f_{CPU}) selected by PCC.

Instruction Group	Mnemonic	Operand	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	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	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	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 in the instruction with no data access
2. When an area except the internal high-speed RAM area is accessed

Remark 1 instruction clock cycle is 1 CPU clock (f_{CPU}) selected by PCC.

20.3 Instruction List 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

1st Operand \ 2nd 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
r1											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

2nd Operand / 1st Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, or HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

2nd Operand / 1st Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A. bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr. bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr. bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW. bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL]. bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instruction/branch instruction

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

1st Operand \ 2nd 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

This following development tools are available for the development of systems which employ the μPD78024 subseries.

Language Processing Software

RA78K/0 Relocatable Assembler	This is a program to convert a program 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.
	Order code: μSxxxxRA78K0
CC78K/0 C Compiler	This is a program to convert a program written in C language into an object code executable with a microcomputer.
	Order code: μSxxxxCC78K0
CC78K/0-L C Library Source File	This is a source program of functions comprising the object library which is contained in the CC78K/0 C compiler package.
	Order code: μSxxxxCC78K0-L

Remarks 1. When CC78K/0 compiler package, the RA78K/0 assembler package (separately sold) is required.

2. Purchase the CC78K/0-L C compiler library source file if you change the object library included in the C compiler package according to your specifications.

3. Ordering code xxxx varies according to the host machine to be used. Refer to the following table.

Host Machine	OS	Supply Medium	xxxx in Ordering Code
PC-9800 series	MS-DOS™ (Ver.3.30 to Ver.5.00A[Note])	3.5-inch 2HD	5A13
		5-inch 2HD	5A10
★ IBM PC/AT™	PC DOS™ (Ver.3.3 to Ver.5.0)	3.5-inch 2HC	7B13
		5-inch 2HC	7B10
HP9000 series 300™	HP-UX™ (rel.7.05B)	Cartridge tape (QIC-24)	3H15
★ HP9000 series 700™			3P15
SPARCstation™	SunOS™ (rel.4.1.1)		3K15
EWS-4800 series (RISC)	EWS-UX/V (rel.4.0)		3M15

★ **Note** The task swap function, which is provided with MS-DOS Ver. 5.00/5.00A and PC DOS Ver 5.0, is not available with the software.

PROM Programming Tools

<table>
<tr><td>Hardware</td><td>PG-1500</td><td colspan="4">This is a PROM programmer capable of programming the single-chip microcomputer built in the PROM by manipulating from the stand-alone or host machine through connection of an accessory board and program adapter separately purchasable.
It can also program representative PROMs ranging from 256K bits to 4M bits.</td></tr>
<tr><td></td><td>PA-78P024CW
PA-78P024GF^{Note1}</td><td colspan="4">This is a PROM programmer adapter for the μPD78P024. It is connected to the PG-1500 for use.
PA-78P024CW : For 64-pin plastic shrink DIP (750 mil)
PA-78P024GF : For 64-pin plastic QFP (14 × 20 mm)</td></tr>
<tr><td>Software</td><td>PG-1500 controller</td><td colspan="4">The PG-1500 is controlled in the host machine through connection with the host machine via serial and parallel interfaces.</td></tr>
<tr><td></td><td></td><td>Host Machine</td><td>OS</td><td>Supply Medium</td><td>Ordering Code (Product Name)</td></tr>
<tr><td></td><td></td><td>PC-9800 series</td><td>MS-DOS (Ver.3.30 to Ver.5.00A^{Note2})</td><td>3.5-inch 2HD</td><td>μS5A13PG1500</td></tr>
<tr><td></td><td></td><td></td><td></td><td>5-inch 2HD</td><td>μS5A10PG1500</td></tr>
<tr><td></td><td></td><td>IBM PC/AT</td><td>PC DOS (Ver.3.3 to Ver.5.0^{Note2})</td><td>3.5-inch 2HC</td><td>μS7B13PG1500 ★</td></tr>
<tr><td></td><td></td><td></td><td></td><td>5-inch 2HC</td><td>μS7B10PG1500</td></tr>
</table>

Notes 1. Under development

2. The task swap function, which is provided with MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0, is not available with this software. ★

Debugging Tool

A circuit emulator (IE-78000-R) is available as μPD78024 subseries debugging tool. Its system configuration is as follows.

Debugging Tool (1/2)

Hardware	IE-78000-R	The IE-78000-R is an in-circuit emulator for use with the 78K/0 series. For development of the μPD78024 subseries, use the IE-78000-R in combination with an emulation probe. Debugging can be executed efficiently through connection with the host machine and the PROM programmer.
	IE-78044-R-EM	The IE-78044-R-EM is an emulation board common to the μPD78044 subseries.
	EP-78024CW-R	Emulation probe common to the μPD78024 subseries. This is for 64-pin plastic shrink DIP (750 mil).
	EP-78024GF-R	Emulation probe for μPD78024 subseries. This is for 64-pin plastic QFP (14 × 20 mm). The EP-78024GFR is provided with one 64-pin conversion socket EV-9200G-64 that makes it easier to develop user systems.
	EV-9200G-64	A conversion socket mounted on the user system board created for 64-pin plastic QFP (14 × 20 mm) use. Used together with the EP-78024GF-R.

Remark The above tools can be used for any of the mPD78024 series products.

Debugging Tool (2/2)

Software	SD78K0	Screen debugger for the 78K/0 series. Connects the IE-78000-R and host machine via a serial or parallel interface, and controls the IE-78000-R on the host machine.			
		Host Machine	OS	Supply Medium	Ordering code (Product Name)
		PC-9800 series	MS-DOS (Ver.3.30 to Ver.5.00A[Note])	3.5-inch 2HD	μS5A13SD78K0
				5-inch 2HD	μS5A10SD78K0
		IBM PC/AT	PC DOS (Ver.3.3 to Ver.5.0[Note])	3.5-inch 2HC	μS7B13SD78K0 ★
				5-inch 2HC	μS7B10SD78K0
	DF78024	Device file for the μPD78024 subseries. Used in conjunction with SD78K0.			
		Host Machine	OS	Supply Medium	Ordering Code (Product Name)
		PC-9800 series	MS-DOS (Ver.3.30 to Ver.5.00A[Note])	3.5-inch 2HD	μS5A13DF78024
				5-inch 2HD	μS5A10DF78024
		IBM PC/AT	PC DOS (Ver.3.3 to Ver.5.0[Note])	3.5-inch 2HC	μS7B10DF78024 ★
				5-inch 2HC	μS7B10DF78024

Note The task swap function, which is provided with Ver. 5.00/5.00A, is not available with this software.

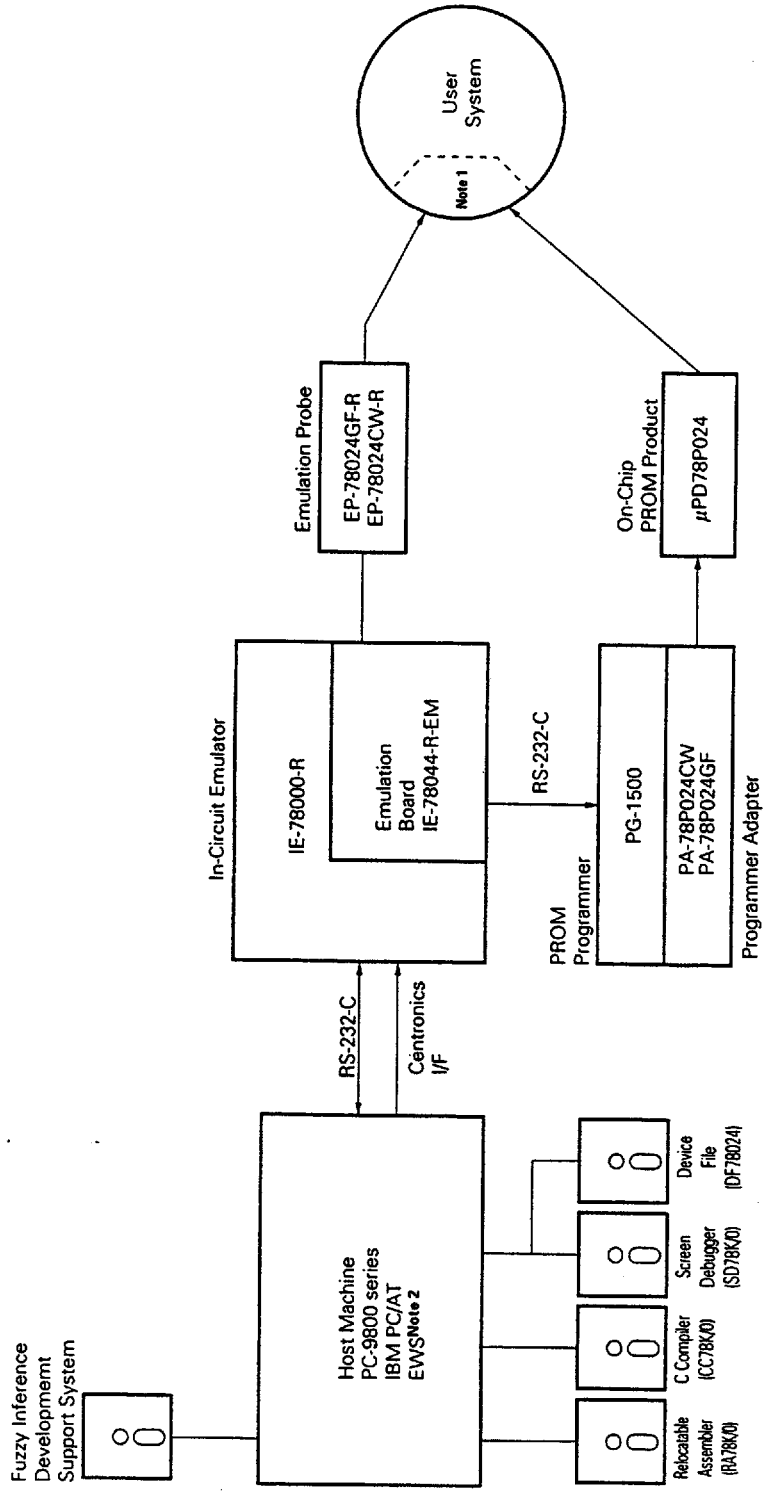
Remark The above tools can be used for any of the μPD78024 subseries products.

System upgrade from other in-circuit emulator to IE-78000-R

If you have 78K series or 75X series in-circuit emulator, you can use it as the equivalent of a 78K/0 in-circuit emulator by replacing the break board in the unit with the IE-78000-R-BK.

Series Name	Your In-Circuit Emulator	Board to be Purchased
75X Series	IE-75000-R, IE-75001-R	IE-78000-R-BK
78K/I Series	IE-78130-R, IE-78140-R	
78K/II Series	IE-78230-R, IE-78230-R-A IE-78240-R, IE-78240-R-A	
78K/III Series	IE-78320-R, IE-78327-R IE-78330-R, IE-78350-R	
78K/VI Series	IE-78600-R	

Development Tool Configuration

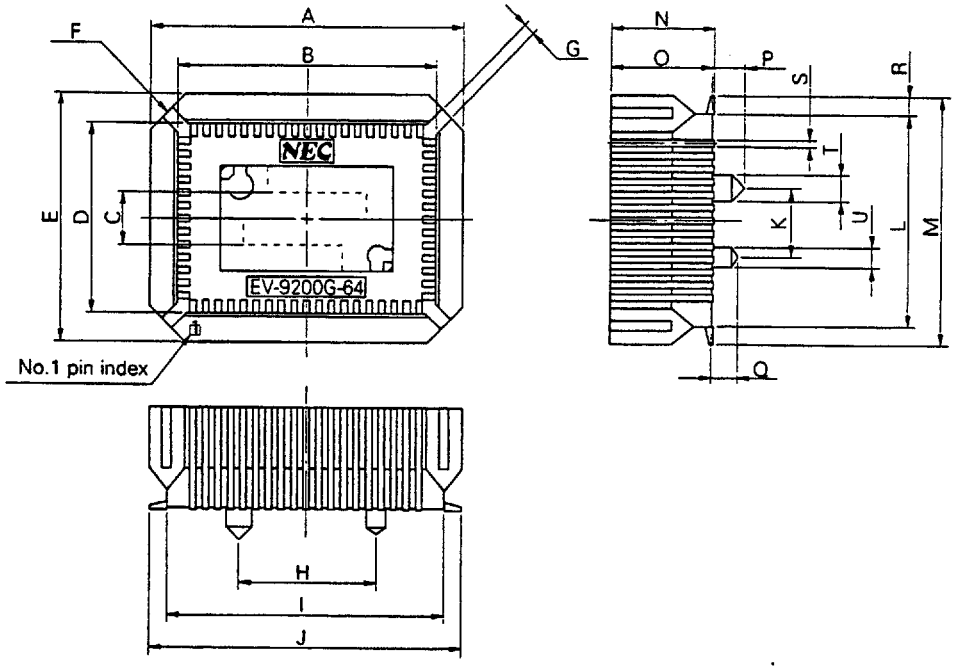


Notes 1. EV-9200G-64 (When the 64-Pin Plastic QFP (14 x 20mm) is developed.)

2. Only the language processing software operates in the EWS (HP9000 series 300, HP9000 series 700, SPARCstation, EWS-4800 series).

Conversion Socket (EV-9200G-64) Package Drawings and Recommended Pattern of Circuit Installation

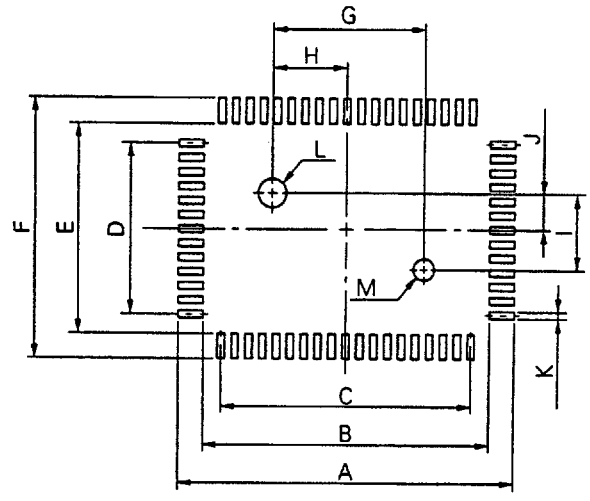
Fig. A-1 EV-9200G-64 External View (Reference)



EV-9200G-64-G0

ITEM	MILLIMETERS	INCHES
A	25.0	0.984
B	20.30	0.799
C	4.0	0.157
D	14.45	0.569
E	19.0	0.748
F	4-C 2.8	4-C 0.11
G	0.8	0.031
H	11.0	0.433
I	22.0	0.866
J	24.7	0.972
K	5.0	0.197
L	16.2	0.638
M	18.9	0.744
O	8.0	0.315
N	7.8	0.307
P	2.5	0.098
Q	2.0	0.079
R	1.35	0.053
S	0.35±0.1	$0.014^{+0.004}_{-0.005}$
T	φ2.3	φ0.091
U	φ1.5	φ0.059

Fig. A-2 EV-9200G-64 Recommended Pattern of Circuit Board Installation (Reference)



EV-9200G-64-P0

ITEM	MILLIMETERS	INCHES
A	25.7	1.012
B	21.0	0.827
C	1.0±0.02 × 18=18.0±0.05	$0.039^{+0.002}_{-0.001} \times 0.709 = 0.709^{+0.002}_{-0.003}$
D	1.0±0.02 × 12=12.0±0.05	$0.039^{+0.002}_{-0.001} \times 0.472 = 0.472^{+0.003}_{-0.002}$
E	15.2	0.598
F	19.9	0.783
G	11.00±0.08	$0.433^{+0.004}_{-0.003}$
H	5.50±0.03	$0.217^{+0.001}_{-0.002}$
I	5.00±0.08	$0.197^{+0.003}_{-0.004}$
J	2.50±0.03	$0.098^{+0.002}_{-0.001}$
K	0.6±0.02	$0.024^{+0.001}_{-0.002}$
L	φ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
M	φ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (IEI-1207).

[MEMO]

PAGE(S) INTENTIONALLY BLANK

Fuzzy Inference Development Support System ★

FE9000/FE9200 Fuzzy Knowledge Data Creation Tool	This is a program to support fuzzy knowledge data (fuzzy rule and membership function) input/edit and evaluation (simulation).
	Ordering code: μSxxxxFE9000 (PC-9800 series) μSxxxxFE9200 (IBM PC/AT)[Note1]
FT9080/FT9085 Translator	This is a program to convert fuzzy knowledge data obtained by the fuzzy knowledge data creation tool into an RA78K/0 assembler source program.
	Ordering code: μSxxxxFT9080 (PC-9800 series) μSxxxxFT9085 (IBM PC/AT)
FI78K0 Fuzzy Inference Module	This is a program to execute fuzzy inference. This program executes fuzzy inference by linking with the fuzzy knowledge data translated by a translator.
	Ordering code: μSxxxxFI78K0 (PC-9800 series, IBM PC/AT)

Remark xxxx in the ordering codes differ depending on the host machine to be used. See the tables below.

μSxxxxFE9000
μSxxxxFT9080
μSxxxxFI78K0

xxxx	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (Ver.3.30 to Ver.5.00A[Note2])	3.5-inch 2HD
5A10			5-inch 2HD

μSxxxxFE9200[Note 1]
μSxxxxFT9085
μSxxxxFI78K0

xxxx	Host Machine	OS	Supply Medium
7B13	IBM PC/AT	PC DOS (Ver.3.3 to Ver.5.0[Note2])	3.5-inch 2HC
7B10			5-inch 2HC

Notes 1. Windows™ (Ver.3.0 to Ver.3.1) is needed to operate the FE2000.
2. The task swap function, which is provided with Ver. 5.00/5.00A is not available with this software.