To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

RENESAS

USER'S MANUAL

μPD78098 SUB-SERIES

8-BIT SINGLE-CHIP MICROCOMPUTER

μPD78094
μPD78095
μPD78096
μPD78098A
μPD78P098A

Document No. IEU-1381A
(O. D. No. IEU-854A)
Date Published March 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.

Major Revisions in This Edition (1/2)

Page	Contents
Throughout	μPD78P098 was erased from Corresponding Products μPD78098A, 78P098A were added to Corresponding Products μPD78094, 78095, 78096: Under development → Development completed
P. 24	The recommended connecting method of unused pins P130/ANO0 and P131/ANO1 was changed
P. 21, 79	Cautions for use of pins P130/ANO0 and P131/ ANO1 were added
P. 80	**4.3 Port Function Control Registers** was revised
P. 81	**Table 4-3 Port Mode Register and Output Latch Settings when Using Dual Function** was added
P. 92	**Fig. 5-3 Subsystem Clock Feedback Resistor** was added
P. 94	Cautions were added to **Fig. 5-5 Oscillation Mode Selection Register Format**
P. 105	Cautions were added to **5.6.2 System clock and CPU clock switching procedure**
P. 116	Cautions were added to **6.3 (2) 16-bit timer mode control register**
P. 134	**6.4.4 (3) Pulse width measurement with free-running counter and two capture registers** was revised and cautions were added
P. 136	**6.4.4 (4) Pulse width measurement by means of restart** was revised and cautions were added
P. 141	**6.4.7 (1) One-shot pulse output using software trigger** was revised
P. 143	**Fig. 6-32 Timing of One-Shot Pulse Output Operation Using Software Trigger** was changed
P. 145	**Fig. 6-34 Timing of One-Shot Pulse Output Operation Using External Trigger (with Rising Edge Specified)** was changed
P. 164	**Table 7-7 8-Bit Timer/Event Counter 2 Interval Time** was added
P. 170	**Table 7-10 Square-Wave Output Ranges when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter** was added
P. 196	**Fig. 12-1 A/D Converter Block Diagram** was changed
P. 199	**Fig. 12-2 A/D Converter Mode Register Format** was changed
P. 213	Cautions were added to **13.1 D/A Converter Functions**
P. 214	**Fig. 13-1 D/A Converter Block Diagram** was changed
P. 226	**Fig. 14-4 Serial Operating Mode Register 0 Format** was changed

Major Revisions in This Edition (2/2)

Page	Contents
P. 250	**Fig. 14-21 RELT, CMDT, RELD and CMDD Operations (Slave)** was revised
P. 273	**Fig. 15-1 Serial Interface Channel 1 Block Diagram** was revised
P. 277	**Fig. 15-3 Serial Operation Mode Register 1 Format** was changed and Notes were erased
P. 279	**Fig. 15-5 Automatic Data Transmit/Receive Interval Specify Register Format** was revised and Notes were erased
P. 308	**15.4.3 (3) (h) Automatic data transmit/receive interval** was revised
P. 313	**Fig. 16-1 Serial Interface Channel 2 Block Diagram** was revised
P. 319	**Table 16-2 Serial Interface Channel 2 Operating Mode Settings** was revised
P. 347	**CHAPTER 18 INTERRUPT FUNCTIONS AND TEST FUNCTIONS** was changed
P. 401	**Fig. 20-13 IEBus Controller Mode Register Format** was revised
P. 414	**Fig. 20-23 Formats of Multiaddress Calling Destination Address Registers 1 and 2** was changed
P. 435	**23.2 Internal Expansion RAM Size Switching Register** was added
P. 436	Cautions were added to **23.3 PROM Programming**
–	**24.3 Instruction Codes** and **24.4 Operation List** were erased
P. 461, 462	**APPENDIX A DEVELOPMENT TOOLS** was revised IE-78098-R-EM, DF78098: Under development → Development completed
P. 467, 468	**APPENDIX B EMBEDDED SOFTWARE** was revised
–	**APPENDIX C INSTRUCTION INDEX (In Alphabetical Order)** was erased
P. 475	**APPENDIX D REVISION HISTORY** was added

The mark ★ shows revised points.

PREFACE

Readers This manual has been prepared for user engineers who want to understand the functions of the μPD78098 sub-series and design and develop its application systems and programs.

The μPD78P098AKK-T in the μPD78098 sub-series does not provide reliability that satisfies mass production of your application system. Use this model for experiment or function evaluation only.

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

Organization The μPD78098 sub-series manuals are separated into two manuals; this manual and Instruction manual (common to 78K/0 series). ★

μPD78098 sub-series User's manual	78K/0 series User's manual Instruction
● Pin functions ● Internal block function ● Interrupt ● Other on-chip peripheral functions	● CPU function ● Instruction set ● Instruction description

How to Read This Manual

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

☐ When you want to understand the functions in general:
→ Read this manual in the order of the contents.

☐ How to interpret the register format:
→ For the circled bit number, the bit name is defined as a reserved word in RA78K/0, and in CC78K/0, already defined in the header file named sfrbit.h.

☐ When you know the names of registers and want to check the details of these registers: ★
→ Refer to **APPENDIX C REGISTER INDEX**.

☐ When you want to know the details of the μPD78098 sub-series instruction functions: ★
→ Refer to the separate volume **78K/0 SERIES USER'S MANUAL-INSTRUCTION (IEU-1372)**.

Legend

Data representation weight	:	High digits on the left and low digits on the right
Active low representations	:	$\overline{\text{xxx}}$ (top bar over pin or signal name)
Note	:	Description of "Note" in the text.
Caution	:	Information requiring particular attention
Remark	:	Additional explanatory material
Numeral representations	:	Binary .....xxxx or xxxxB
		Decimal xxxx
		Hexadecimal ....xxxxH

RELATED DOCUMENTS

● Documents related to devices

Part number / Document name	μPD78094	μPD78095	μPD78096	μPD78098A	μPD78P098A
Preliminary Product Information	–				To be published soon
Data Sheet	Planned				Planned
User's Manual	This manual				
User's Manual Instruction	IEU-1372				

● Documents related to development tools

Document name		Document number
RA78K Series Assembler Package User's Manual	Operation	EEU-1399
	Language	EEU-1404
RA78K Series Structured Assembler Preprocessor User's Manual		EEU-1402
CC78K Series C Compiler User's Manual	Operation	EEU-1280
	Language	EEU-1284
CC78K Series Library Source File User's Manual		–
PG-1500 PROM Programmer User's Manual		EEU-1335
PG-1500 Controller User's Manual		EEU-1291
IE-78000-R User's Manual		EEU-1398
IE-78000-R-BK User's Manual		EEU-1427
IE-78098-R-EM User's Manual		To be published soon
SD78K/0 Screen Debugger User's Manual	Introduction	EEU-1414
	Reference	EEU-1413

● **Documents related to embedded software**

Document name		Document number
78K/0 Series Real-Time OS User's Manual	Basic	–
	Install	–
	Debugger	–
	Technical	–
Fuzzy Inference Development Support System Brochure		–
Fuzzy Knowledge Data Creation Tool User's Manual		EEU-1438
78K/0, 78K/II, 87AD Series Fuzzy Inference Development Support System Translator User's Manual		EEU-1444
78K/0 Series Fuzzy Inference Development Support System Fuzzy Inference Module User's Manual		–
78K/0 Series Fuzzy Inference Development Support System Fuzzy Inference Debugger User's Manual		EEU-1458

● **Other documents**

Document name	Document number
Package Manual	IEI-1213
Semiconductor Device Mounting Technology Manual	IEI-1207
Quality Grade of NEC Semiconductor Devices	IEI-1209
Reliability and Quality Control of NEC Semiconductor Devices	IEI-1203
Electrostatic Discharge (ESD) Test	IEI-1201
Guide to Quality Assurance of Semiconductor Devices	MEI-1202
Guide to Microcomputer-Related Products - Other Manufacturers	–

Caution: The contents of the above documents are subject to change without notice. Be sure to use the latest edition for designing.

CONTENTS

LIST OF FIGURES (1/7)

LIST OF FIGURES (2/7)

LIST OF FIGURES (3/7)

LIST OF FIGURES (4/7)

LIST OF FIGURES (5/7)

LIST OF FIGURES (6/7)

LIST OF FIGURES (7/7)

LIST OF TABLE (1/3)

LIST OF TABLE (2/3)

LIST OF TABLE (3/3)

CHAPTER 1 OUTLINE

1.1 Features

○ Internal high-capacity ROM and RAM

Item / Part number	Program memory (ROM)	Data memory		
		Internal high-speed RAM	Buffer RAM	Internal expansion RAM
μPD78094	32K bytes	1024 bytes	32 bytes	None
μPD78095	40K bytes			
μPD78096	48K bytes			
μPD78098A	60K bytes			2048 bytes ★
μPD78P098A	60K bytes[Note1]			2048 bytes[Note2] ★

Notes:

(1) The capacity of internal PROM can be changed by means of the memory size switching register.

(2) 0 or 2048 bytes can be selected for the RAM size by means of the internal expansion RAM size switching register.

○ External Memory Expansion Space: 64K bytes

○ Instruction execution time changeable from high speed (0.5 μs: In main system clock 6.0 MHz operation) to ultra-low speed (122 μs: In subsystem clock 32.768 kHz operation)

○ Instruction set suited to system control
- Bit manipulation possible in all address spaces
- On-chip multiply and divide instructions

○ 69 I/O ports (4 N-ch open drain)

○ IEBus™ controller
- Effective transfer rate: 3.9 kbps/17 kbps/26 kbps
- Operating power supply voltage range: 4.5 to 6.0 V (when 2/3 divider in clock generator circuit is used)

○ 8-bit resolution A/D converter: 8 channels

○ 8-bit resolution D/A converter: 2 channels

○ Serial interface: 3 channels
- 3-wire/SBI/2-wire mode: 1 channel
- 3-wire mode
  (Automatic transmit/receive function): 1 channel
- 3-wire UART 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

○ 23 vectored interrupts

○ 2 test inputs

○ 2 types of on-chip clock oscillator circuits (main system clock and subsystem clock)

○ Operating power supply voltage range: 2.7 to 6.0 V

1.2 Applications

Automobile audio system, CD changer, etc.

★ 1.3 Ordering Information

Part Number	Package	Internal ROM
μPD78094GC-xxx-3B9	80-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78095GC-xxx-3B9	80-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78096GC-xxx-3B9	80-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78098AGC-xxx-3B9[Note1]	80-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78P098AGC-3B9[Note1]	80-pin plastic QFP (14 × 14 mm)	One-time PROM
μPD78P098AKK-T[Note2]	80-pin ceramic WQFN (14 × 14 mm)	EPROM

Notes:
(1) Under development
(2) Planned

Remark: xxx indicates ROM code suffix.

★ 1.4 Quality Grade

Part Number	Package	Quality grade
μPD78094GC-xxx-3B9	80-pin plastic QFP (14 × 14 mm)	Standard
μPD78095GC-xxx-3B9	80-pin plastic QFP (14 × 14 mm)	Standard
μPD78096GC-xxx-3B9	80-pin plastic QFP (14 × 14 mm)	Standard
μPD78098AGC-xxx-3B9[Note1]	80-pin plastic QFP (14 × 14 mm)	Standard
μPD78P098AGC-3B9[Note1]	80-pin plastic QFP (14 × 14 mm)	Standard
μPD78P098AKK-T[Note2]	80-pin ceramic WQFN (14 × 14 mm)	None (for functional evaluation)

Notes:
(1) Under development
(2) Planned

Remark: xxx indicates ROM code suffix.

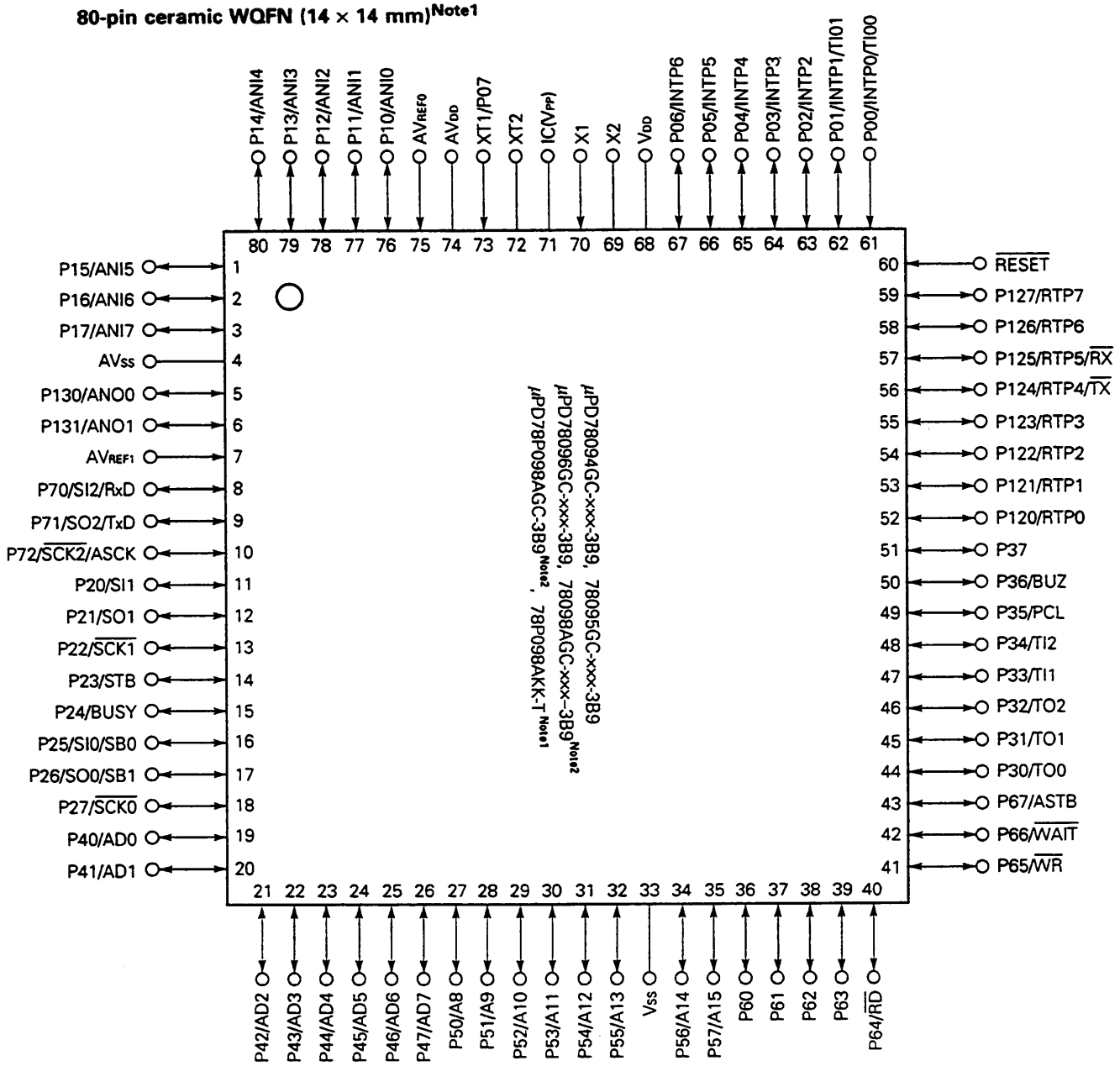
The μPD78P098AKK-T does not provide reliability that satisfies mass production of your application system. Use this model for experiment or function evaluation only.

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 operating mode

80-pin plastic QFP (14 × 14 mm)
80-pin ceramic WQFN (14 × 14 mm)[Note1]



Notes:

(1) Planned

(2) Under development

Cautions:

(1) Be sure to connect IC (Internally Connected) pin to Vss directly.

(2) Connect AVDD pin to VDD.

(3) Connect AVss pin to Vss.

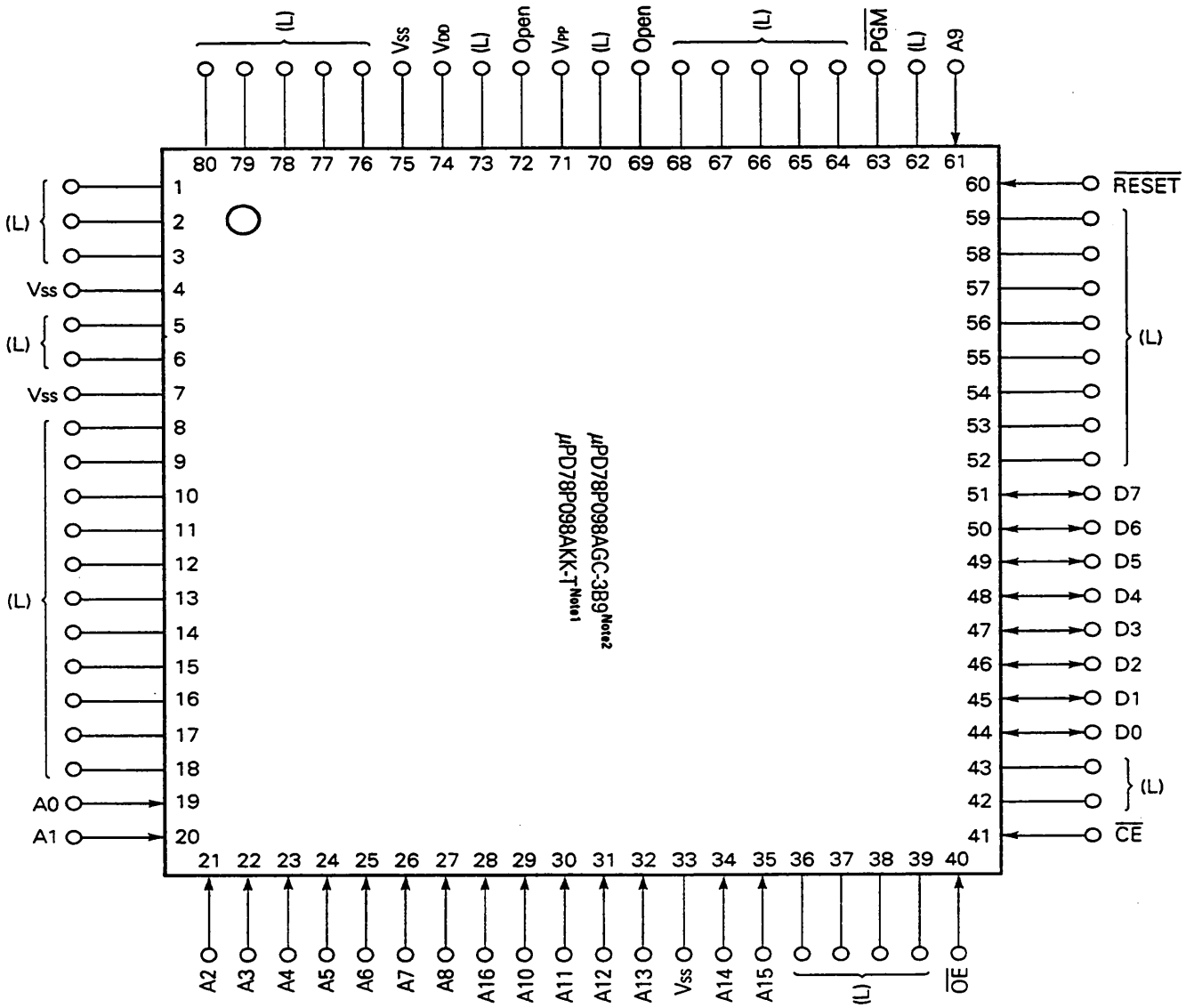
Remark: Pin connection in parentheses is intended for μPD78P098A.

P00 to P07 : Port0
P10 to P17 : Port1
P20 to P27 : Port2
P30 to P37 : Port3
P40 to P47 : Port4
P50 to P57 : Port5
P60 to P67 : Port6
P70 to P72 : Port7
P120 to P127 : Port12
P130, P131 : Port13
RTP0 to RTP7 : Real-Time Output Port
INTP0 to INTP6 : Interrupt From Peripherals
TI00, TI01 : Timer Input
TI1, TI2 : Timer Input
TO0 to TO2 : Timer Output
SB0, SB1 : Serial Bus
SI0 to SI2 : Serial Input
SO0 to SO2 : Serial Output
$\overline{SCK0}$ to $\overline{SCK2}$: Serial Clock
RxD : Receive Data (UART)
TxD : Transmit Data (UART)
ASCK : Asynchronous Serial Clock
$\overline{RX}$: Receive Data (IEBus Controller)
$\overline{TX}$: Transmit Data (IEBus Controller)

PCL : Programmable Clock
BUZ : Buzzer Clock
STB : Strobe
BUSY : Busy
AD0 to AD7 : Address/Data Bus
A8 to A15 : Address Bus
$\overline{RD}$: Read Strobe
$\overline{WR}$: Write Strobe
$\overline{WAIT}$: Wait
ASTB : Address Strobe
X1, X2 : Crystal (Main System Clock)
XT1, XT2 : Crystal (Subsystem Clock)
$\overline{RESET}$: Reset
ANI0 to ANI7 : Analog Input
ANO0 to ANO1 : Analog Output
AV_{DD} : Analog Power Supply
AV_{SS} : Analog Ground
$AV_{REF0, 1}$: Analog Reference Voltage
V_{DD} : Power Supply
V_{PP} : Programming Power Supply
V_{SS} : Ground
IC : Internally Connected

(2) PROM programming mode

80-pin plastic QFP (14 × 14 mm)
80-pin ceramic WQFN (14 × 14 mm)[Note1]



Notes:

(1) Planned

(2) Under development

Cautions:

(1) (L) : Connect individually to Vss via a pull-down resistor.

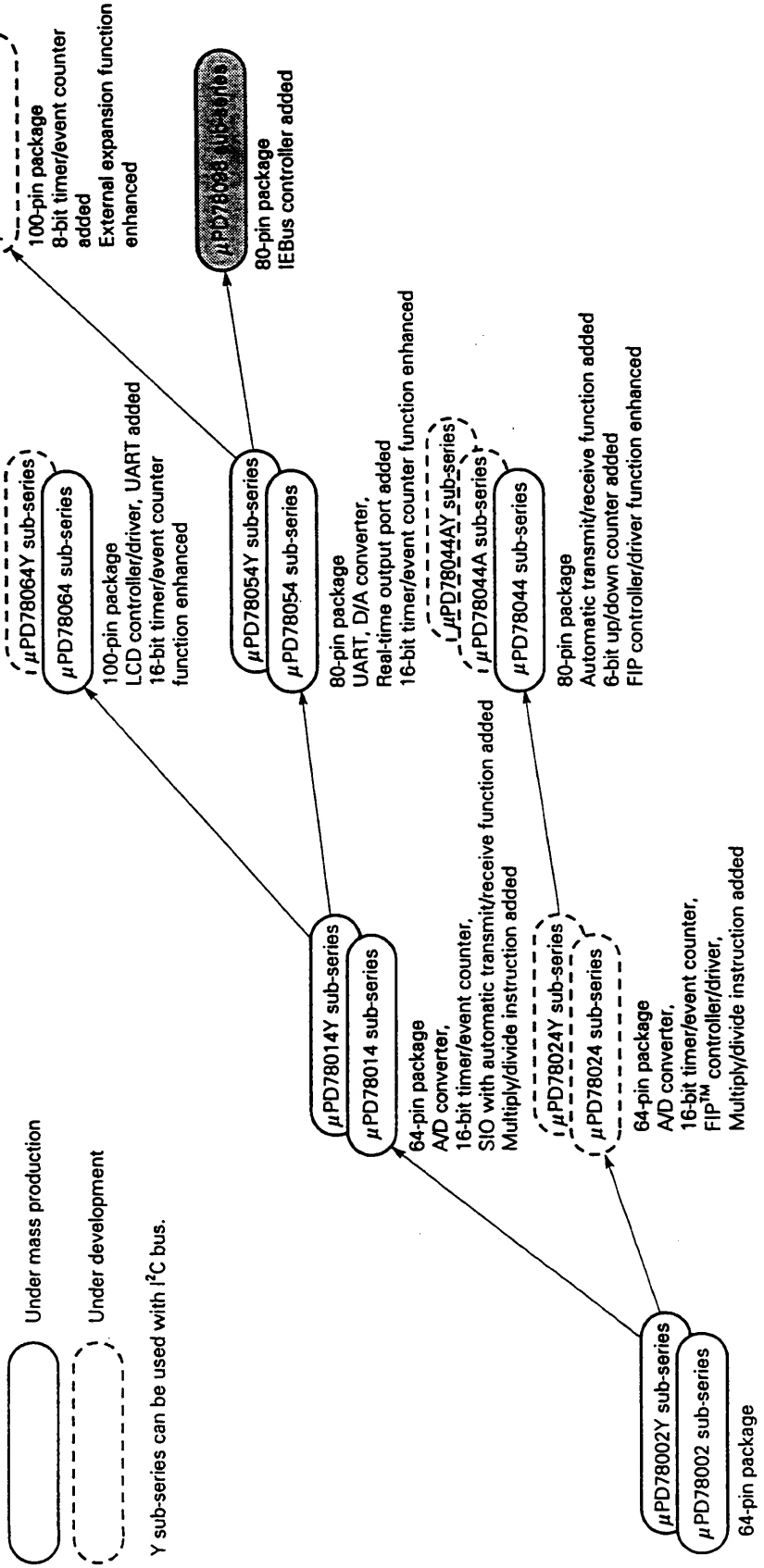
(2) Vss : Connect to the ground.

(3) $\overline{\text{RESET}}$: Set to the low level.

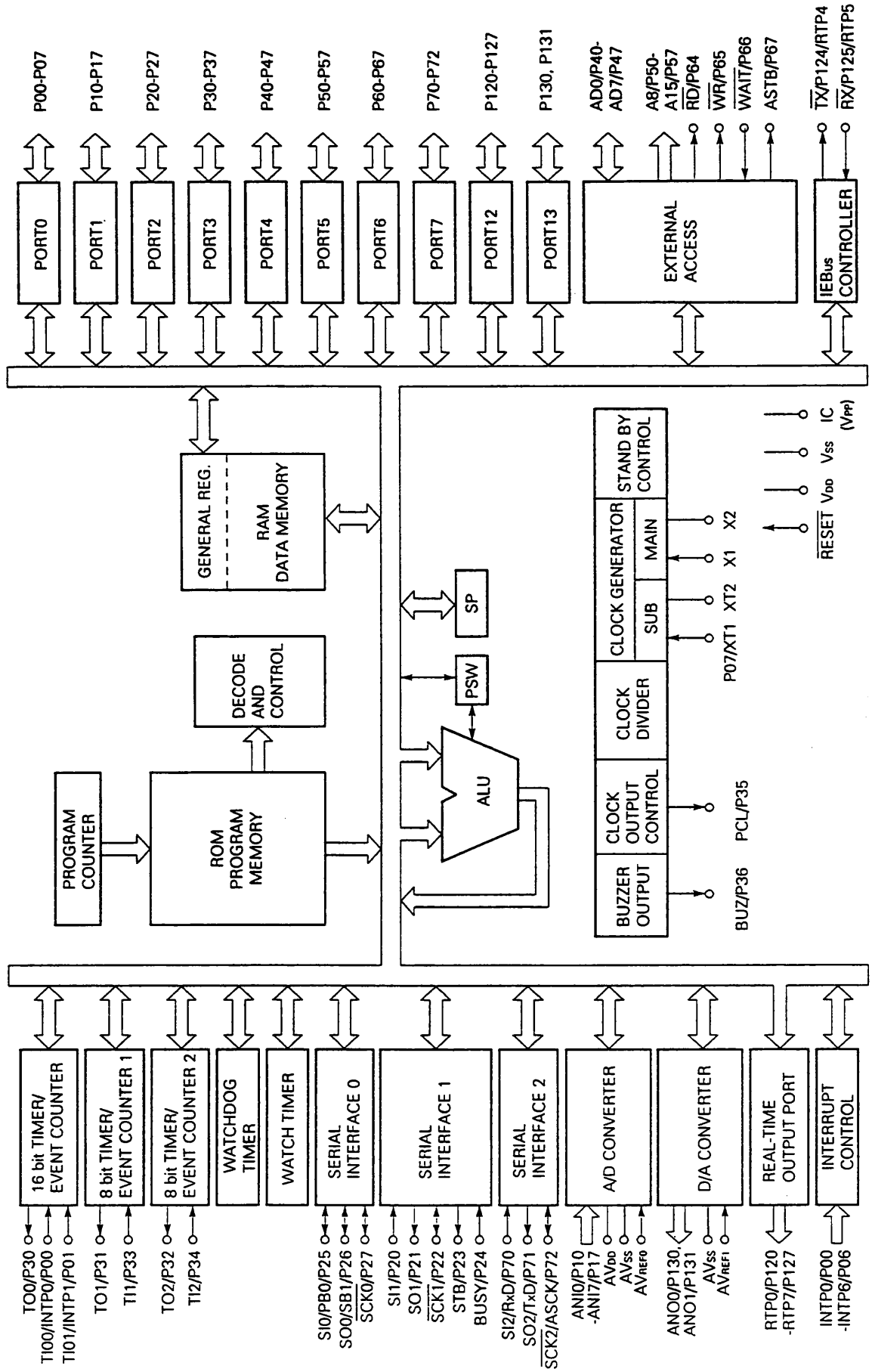
(4) Open : Do not connect anything.

A0 to A16	: Address Bus	$\overline{\text{OE}}$	: Output Enable	V_{DD}	: Power Supply
D0 to D7	: Data Bus	$\overline{\text{PGM}}$	: Program	V_{PP}	: Programming Power Supply
$\overline{\text{CE}}$	: Chip Enable	$\overline{\text{RESET}}$	: Reset	Vss	: Ground

★ 1.6 78K/0 Series Development



1.7 Block Diagram



Cautions:

(1) The internal ROM capacity depends on the product.

(2) Pin connection in parentheses is intended for the μPD78P098A.

1.8 Outline of Function

Item \ Part Number			μPD78094	μPD78095	μPD78096	μPD78098A[Note 1]	μPD78P098A[Notes 1, 2]
Internal memory	ROM	Configuration	Mask ROM				PROM
		Capacity	32K bytes	40K bytes	48K bytes	60K bytes	60K bytes[Note 3]
	Internal high-speed RAM	Capacity	1024 bytes				
	Buffer RAM	Capacity	32 bytes				
	Internal expansion RAM	Capacity	None			2048 bytes	2048 bytes[Note 4]
Memory space			64K bytes				
General register			8 bits × 8 × 4 banks				
Instruction cycle	With main system clock selected		0.5 μs/1.0 μs/2.0 μs/4.0 μs/8.0 μs/16.0 μs (when operated at 6.0 MHz)				
	With subsystem clock selected		122 μs (when operated at 32.768 kHz)				
Instruction set			• 16 bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, and Boolean operation) • BCD adjust and other related operations				
I/O port			• Total : 69 I/O ports • CMOS input : 2 inputs • CMOS input/output : 63 inputs/outputs • N-ch open drain I/O : 4 inputs/outputs				
IEBus controller			Effective transfer rate: 3.9 kbps/17 kbps/26 kbps				
A/D converter			• 8-bit resolution × 8 channels				
D/A converter			• 8-bit resolution × 2 channels				
Serial interface			• 3-wire/SBI/2-wire mode selection possible : 1 channel • 3-wire mode (Maximum 32-byte on-chip automatic transmit/receive function) : 1 channel • 3-wire/UART mode selectable : 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)				
Clock output			15.6 kHz, 31.3 kHz, 62.5 kHz, 125 kHz, 250 kHz, 500 kHz, 1.0 MHz, 2.0 MHz, 4.0 MHz (when operated at 6.0 MHz with main system clock) 32.768 kHz (when operated at 32.768 kHz with subsystem clock)				

Notes:

(1) Under development

(2) The μPD78P098A is a PROM version corresponding to the μPD78094, 78095, 78096, and 78098A.

(3) The capacity of internal PROM can be changed by means of the memory size switching register.

(4) 0 or 2048 bytes can be selected for the RAM size by means of the internal expansion RAM size switching register.

Item \ Part Number		μPD78094	μPD78095	μPD78096	μPD78098A[Note 1]	μPD78P098A[Notes 1, 2]
Buzzer output		977 Hz, 1.95 kHz, 3.9 kHz, 7.8 kHz (when operated at 6.0 MHz with main system clock)				
Vectored interrupt	Maskable interrupt	Internal : 14 External : 7				
	Non-maskable interrupt	Internal : 1				
	Software interrupt	Internal : 1				
Test input		Internal : 1 External : 1				
Operating power supply voltage range		V_{DD} = 2.7 to 6.0 V				
Package		• 80-pin plastic QFP (14 × 14 mm) • 80-pin ceramic WQFN (14 × 14 mm)[Note 3] (μPD78P098A only)				

Notes:

(1) Under development

(2) The μPD78P098A is a PROM version corresponding to the μPD78094, 78095, 78096 and 78098A.

(3) Planned

[MEMO]

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

2.1.1 Normal operating mode pins

(1) Port pins (1/3)

Pin Name	Input/ Output	Function		When Reset	Dual-Function Pin
P00	Input	Port 0. 8-bit input/output port.	Input only	Input	INTP0/TI00
P01	Input/ output		Input/output specifiable bit-wise. If used as an input port, a pull-up resistor can be connected by software.	Input	INTP1/TI01
P02					INTP2
P03					INTP3
P04					INTP4
P05					INTP5
P06					INTP6
P07[Note 1]	Input		Input only	Input	XT1
P10 to P17	Input/ output	Port 1. 8-bit input/output port. Input/output specifiable bit-wise. If used as 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					$\overline{\text{SCK1}}$
P23					STB
P24					BUSY
P25					SI0/SB0
P26					SO0/SB1
P27					$\overline{\text{SCK0}}$

Notes:

(1) When the P07/XT1 pin is used as an input port, set the bit 6 (FRC) of the processor clock control register to 1 (do not use the feedback resistor internal to the subsystem clock oscillator).

(2) When pins P10/ANI0 to P17/ANI7 are used as an analog input of the A/D converter, a pull-up resistor is automatically not used.

(1) Port pins (2/3)

Pin Name	Input/ Output	Function		When Reset	Dual-Function Pin
P30	Input/ output	Port 3. 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	TO0
P31					TO1
P32					TO2
P33					TI1
P34					TI2
P35					PCL
P36					BUZ
P37					—
P40 to P47	Input/ output	Port 4. 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. Test input flag (KRIF) is set to 1 by falling edge detection.		Input	AD0 to AD7
P50 to P57	Input/ output	Port 5. 8-bit input/output port. LED can be driven directly. Input/output specifiable bit-wise. If used as an input port, a pull-up resistor can be connected by software.		Input	A8 to A15
P60	Input/ output	Port 6. 8-bit input/output port. Input/output specifiable bit-wise.	N-ch open drain input/ output port. On-chip pull-up resistor specifiable by mask option. (Mask ROM version only). LEDs can be driven directly.	Input	—
P61					
P62					
P63					
P64			If used as an input port, a pull-up resistor can be connected by software.		$\overline{RD}$
P65					$\overline{WR}$
P66					$\overline{WAIT}$
P67					ASTB
P70	Input/ output	Port 7. 3-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	SI2/RxD
P71					SO2/TxD
P72					$\overline{SCK2}$/ASCK

(1) Port pins (3/3)

Pin Name	Input/ Output	Function	When Reset	Dual-Function Pin
P120 to P123	Input/ output	Port 12. 8-bit input/output port. Input/output specifiable bit-wise. If used as an input port, pull-up resistor can be connected by software.	Input	RTP0 to RTP3
P124				RTP4/$\overline{TX}$
P125				RTP5/$\overline{RX}$
P126, P127				RTP6, RTP7
P130, P131	Input/ output	Port 13. 2-bit input/output port. Input/output specifiable bit-wise. If used as an input port, pull-up resistor can be connected by software.	Input	ANO0, ANO1

(2) Non-Port Pins (1/2)

Pin Name	Input/ Output	Function	When Reset	Dual-Function Pin
INTP0	Input	External interrupt inputs with specifiable valid edges (rising edge, falling edge, both rising and falling edges).	Input	P00/TI00
INTP1				P01/TI01
INTP2				P02
INTP3				P03
INTP4				P04
INTP5				P05
INTP6				P06
SI0	Input	Serial interface serial data input	Input	P25/SB0
SI1				P20
SI2				P70/RxD
SO0	Output	Serial interface serial data output	Input	P26/SB1
SO1				P21
SO2				P71/TxD
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
$\overline{SCK2}$				P72/ASCK
STB	Output	Serial interface automatic transmit/receive strobe output	Input	P23
BUSY	Input	Serial interface automatic transmit/receive busy input	Input	P24
RxD	Input	Asynchronous serial interface serial data input	Input	P70/SI2
TxD	Output	Asynchronous serial interface serial data output	Input	P71/SO2
ASCK	Input	Asynchronous serial interface serial clock input	Input	P72/$\overline{SCK2}$
TI00	Input	Input of external count clock to 16-bit timer (TM0)	Input	P00/INTP0
TI01		Input of capture trigger signal to capture register (CR00)		P01/INTP1
TI1		Input of external count clock to 8-bit timer (TM1)		P33
TI2		Input of external count clock to 8-bit timer (TM2)		P34
TO0	Output	16-bit timer output (dual function as14-bit PWM output)	Input	P30
TO1		8-bit timer output		P31
TO2				P32
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P35
BUZ	Output	Buzzer output	Input	P36
RTP0 to RTP7	Output	Real-time output port outputting data in synchronization with trigger	Input	P120 to P127

(2) Non-Port Pins (2/2)

Pin Name	Input/ Output	Function	When Reset	Dual-Function Pin
$\overline{TX}$	Output	IEBus controller data output	Input	P124/RTP4
$\overline{RX}$	Input	IEBus controller data input	Input	P125/RTP5
AD0 to AD7	Input/ Output	Low-order address/data bus when expanding external memory	Input	P40 to P47
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
$\overline{RD}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{WR}$		Strobe signal output for write operation to external memory		P65
$\overline{WAIT}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to port 4 and 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ANO0, ANO1	Output	D/A converter analog output	Input	P130, P131
AV_{REF0}	Input	A/D converter reference voltage input	—	—
AV_{REF1}	Input	D/A converter reference voltage input	—	—
AV_{DD}	—	A/D converter analog power supply. Connected to V_{DD}.	—	—
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	P07
XT2	—		—	—
V_{DD}	—	Positive power supply	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect to Vss in normal operating mode.	—	—
Vss	—	Ground potential	—	—
IC	—	Internally connected. Directly connected to Vss.	—	—

2.1.2 PROM programming mode pins (μPD78P098A only)

Pin Name	Input/ Output	Function
$\overline{\text{RESET}}$	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 $\overline{\text{RESET}}$ 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
$\overline{\text{CE}}$	Input	PROM enable input/program pulse input
$\overline{\text{OE}}$	Input	Read strobe input to PROM
$\overline{\text{PGM}}$	Input	Program/program inhibit input in PROM programming mode
V_{DD}	—	Positive power supply
Vss	—	Ground potential

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt input, an 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 P07 function as input-only ports and P01 to P06 function as input/output ports.

P01 to P06 can be specified for input or output ports bit-wise with a 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 L.

(2) Control mode

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

(a) INTP0 to INTP6

INTP0 to INTP6 are external interrupt input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges). INTP0 or INTP1 becomes a 16-bit timer/event counter capture trigger signal input pin with a valid edge input.

(b) TI00

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

(c) TI01

Pin for capture trigger signal input to capture register (CR00) of 16-bit timer/event counter

(d) XT1

Crystal connect pin for subsystem clock oscillation

2.2.2 P10 to P17 (Port 1)

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

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with a port mode register 1. If used as an input port, pull-up resistor can be connected to these ports with a pull-up resistor option register L.

(2) Control mode

These ports function as A/D converter analog input pins. The pull-up resistor is automatically unused when the pins specified for analog input.

2.2.3 P20 to P27 (Port 2)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as data input/output to/from the serial interface, clock input/output, automatic trsansmit/receive busy input, and strobe output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with a 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 L.

(2) Control mode

These ports function as serial interface data input/output, clock input/output, automatic transmit/receive busy input, and strobe output functions.

(a) SI0, SI1, SO0, 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

(d) BUSY

Serial interface automatic transmit/receive busy input pins

(e) STB

Serial interface automatic transmit/receive strobe output pins

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

2.2.4 P30 to P37 (Port 3)

These are 8-bit input/output ports. Beside serving as input/output ports, they function as timer input/output, clock output and buzzer 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 bit-wise as input or output ports with port mode register 3. When they are used as input ports, a pull-up resistor can be connected with a pull-up resistor option register L.

(2) Control mode

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

(a) TI1 and TI2

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

(b) TO0 to TO2

Timer output pins

(c) PCL

Clock output pin

(d) BUZ

Buzzer output pin

2.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address/data bus.
The test input flag (KRIF) can be set to 1 by detecting a falling edge.
The following operating mode can be specified in 8-bit units.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified in 8-bit units for input or output ports by using the memory expansion mode register. When they are used as input ports, a pull-up resistor can be connected with a pull-up resistor option register L.

(2) Control mode

These ports function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode. When pins are used as an address/data bus, the pull-up resistor is automatically not used.

2.2.6 P50 to P57 (Port 5)

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

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5. When they are used as input ports, a pull-up resistor can be connected with a pull-up resistor option register L.

(2) Control mode

These ports function as high-order address bus pins (A8 to A15) in external memory expansion mode. When pins are used as an address bus, the pull-up resistor is automatically not used.

2.2.7 P60 to P67 (Port 6)

These are 8-bit input/output ports. Besides serving as input/output ports, they are used for control in external memory expansion mode.
P60 to P63 can drive LEDs directly.
The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports.
They can be specified bit-wise as input or output ports with port mode register 6.
P60 to P63 are N-ch open drain outputs. Mask ROM version can contain pull-up resistors with the mask option.
When P64 to P67 are used as input ports, a pull-up resistor can be connected with a pull-up resistor option register L.

(2) Control mode

These ports function as control signal output pins ($\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB) in external memory expansion mode. When a pin is used as a control signal output, the pull-up resistor is automatically not used.

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

2.2.8 P70 to P72 (Port 7)

This is a 3-bit input/output port. In addition to its use as an input/output port, it also has serial interface data input/output and clock input/output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 7 functions as a 3-bit input/output port. Bit-wise specification as an input port or output port is possible by means of port mode register 7. When used as an input port, a pull-up resistor can be connected by means of pull-up resistor option register L.

(2) Control mode

Port 7 functions as serial interface data input/output and clock input/output.

(a) SI2, SO2

Serial interface serial data input/output pins

(b) $\overline{SCK2}$

Serial interface serial clock input/output pin

(c) RxD, TxD

Asynchronous serial interface serial data input/output pins

(d) ASCK

Asynchronous serial interface serial clock input pin

2.2.9 P120 to P127 (Port 12)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as a real-time output port. P124 and P125 are also provided with an IEBus controller data input/output function.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 12. When they are used as input ports, a pull-up resistor can be connected with pull-up resistor option register H.

(2) Control mode

(a) Real-time output port function (RTP0-RTP7)

These ports functions as real-time output ports outputting data in synchronization with a trigger.

(b) IEBus controller function (P124, P125 only: $\overline{TX}$, $\overline{RX}$)

P124 and P125 function as the data input/output port of the IEBus controller. When these pins are used in this mode, the pull-up resistor is automatically not used.

2.2.10 P130 and P131 (Port 13)

These are 2-bit input/output ports. Besides serving as input/output ports, they are used for D/A converter analog output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 2-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 13. When they are used as input ports, a pull-up resistor can be connected with a pull-up resistor option register H.

(2) Control mode

These ports allow D/A converter analog output (ANO0 and ANO1).

Caution: When $AV_{REF} < V_{DD}$ and the D/A converter is used for only one channel, perform the one of the following procedures for pins not used as analog output. ★

- **Set the port mode register (PM13x) to 1 (input mode) and connect to Vss.**
- **Set the port mode register (PM13x) to 0 (output mode), set the output latch to 0, and output low level.**

2.2.11 AV_{REF0}

A/D converter reference voltage input pin.

When A/D converter is not used, connect this pin to Vss.

2.2.12 AV_{REF1}

D/A converter reference voltage input pin.

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

2.2.13 AV_{DD}

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

2.2.14 AVss

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

2.2.15 $\overline{\text{RESET}}$

This is a low level active system reset input pin.

2.2.16 X1 and X2

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

2.2.17 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

2.2.18 V_{DD}

Positive power supply pin

2.2.19 Vss

Ground potential pin

2.2.20 V_{PP} (μPD78P098A only)

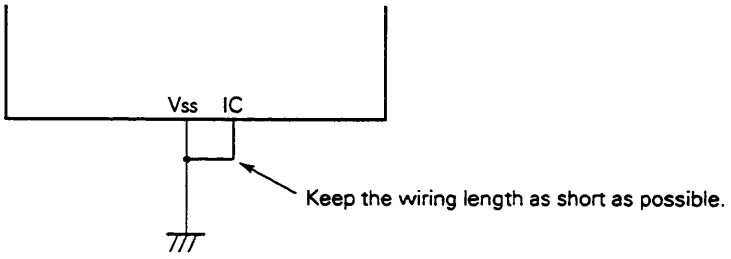
High-voltage apply pin for PROM programming mode setting and program write/verify. Connect to Vss in normal operating mode.

2.2.21 IC (Mask ROM version only)

The IC (Internally Connected) pin sets a test mode in which the μPD78098 sub-series is tested before shipment. Usually, connect the IC pin directly to Vss with as short a wiring length as possible.

If there is a potential difference between the IC and Vss pins because the wiring length between the IC and Vss pin is too long, or external noise is superimposed on the IC pin, your program may not run correctly.

○ Directly connect the IC pin to the Vss.



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 conditions for 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/TI00	2	Input	Connect to Vss.
P01/INTP1/TI01	8-A	Input/output	Input : Connect to Vss. Output: Open
P02/INTP2			
P03/INTP3			
P04/INTP4			
P05/INTP5			
P06/INTP6			
P07/XT1	16	Input	Connect to Vss.
P10/ANI0 to P17/ANI7	11	Input/output	Input : Connect to VDD or Vss. Output: Open
P20/SI1	8-A	Input/output	Input : Connect to VDD or Vss. Output: Open
P21/SO1	5-A		
P22/$\overline{SCK1}$	8-A		
P23/STB	5-A		
P24/BUSY	8-A		
P25/SI0/SB0	10-A		
P26/SO0/SB1			
P27/$\overline{SCK0}$			
P30/TO0	5-A	Input/output	Input : Connect to VDD or Vss. Output: Open
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			
P37			

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P40/AD0 to P47/AD7	5-E	Input/output	Input : Connect to Vss. Output: Open
P50/A8 to P57/A15	5-A	Input/output	Input : Connect to V_{DD} or Vss. Output: Open
P60 to P63 (Mask ROM version)	13-B		
P60 to P63 (μPD78P098A)	13-D		
P64/$\overline{RD}$	5-A		
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
P70/SI2/RxD	8-A	Input/output	Input : Connect to V_{DD} or Vss. Output: Open
P71/SO2/TxD	5-A		
P72/$\overline{SCK2}$/ASCK	8-A		
P120/RTP0 to P123/RTP3	5-A	Input/output	Input : Connect to V_{DD} or Vss. Output: Open
P124/RTP4/$\overline{TX}$			
P125/RTP5/$\overline{RX}$			
P126/RTP6, P127/RTP7			
P130/ANO0, P131/ANO1	12-A	Input/output	Input : Connect to Vss. Output: Open
$\overline{RESET}$	2	Input	—
XT2	16	—	Open
AV_{REF0}	—		Connect to Vss.
AV_{REF1}			Connect to V_{DD}.
AV_{DD}			
AVss			Connect to Vss.
IC (Mask ROM version)			Connect directly to Vss.
V_{PP} (μPD78P098A)			Connect to Vss.

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

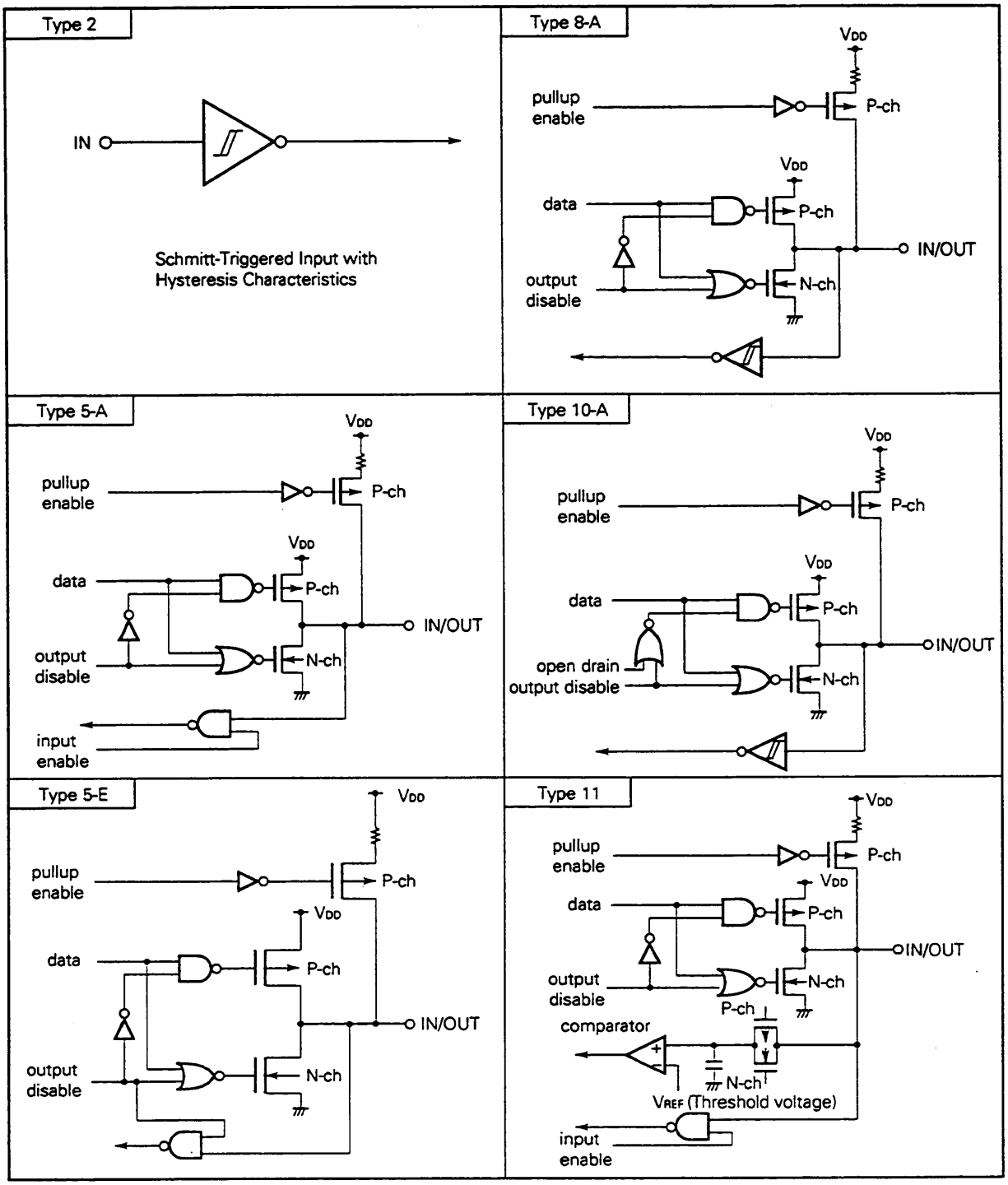
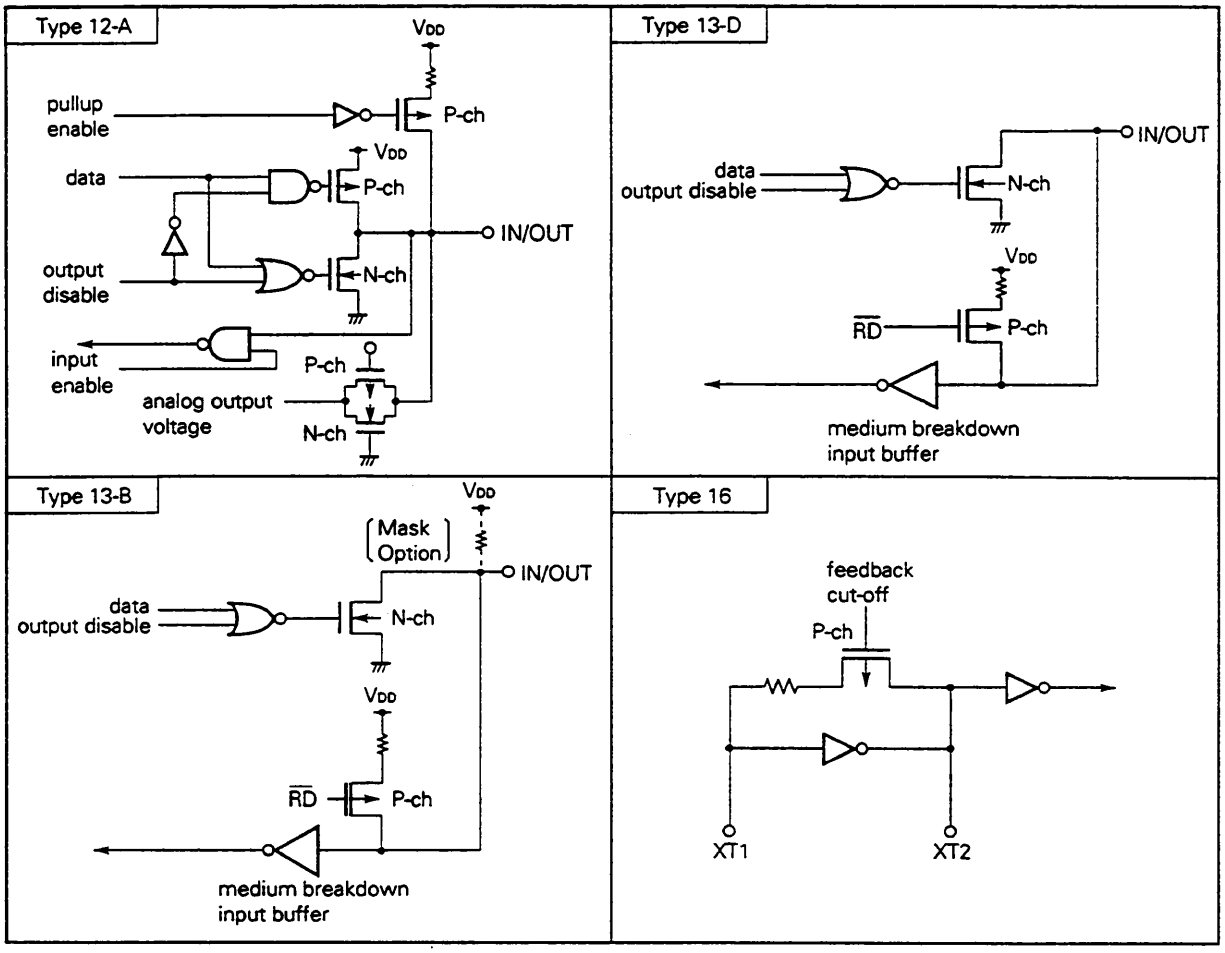


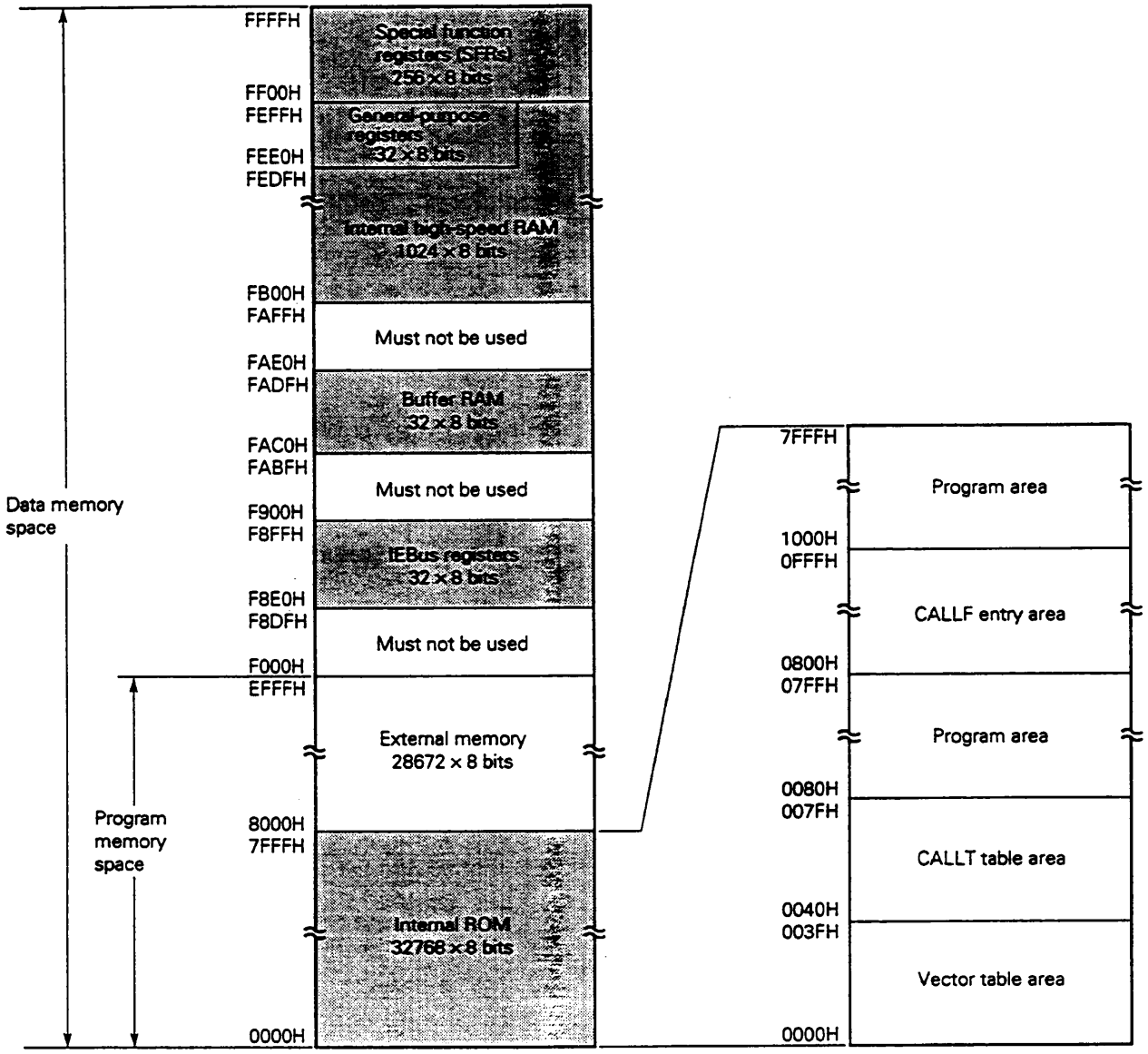
Fig. 2-1 Pin Input/Output Circuit List (2/2)



3.1 Memory Spaces

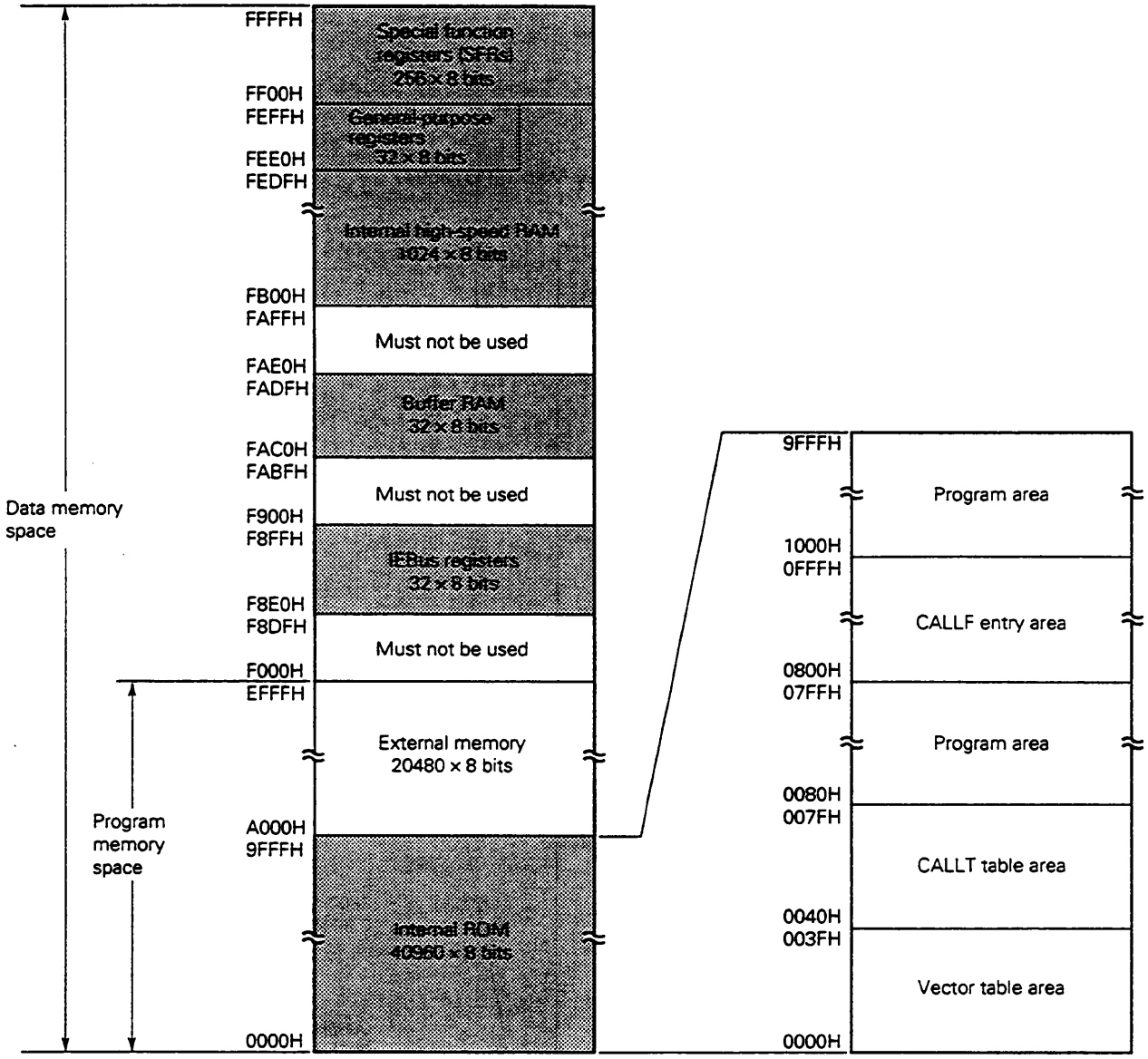
Figs. 3-1 to 3-5 show memory maps.

Fig. 3-1 Memory Map (μPD78094)



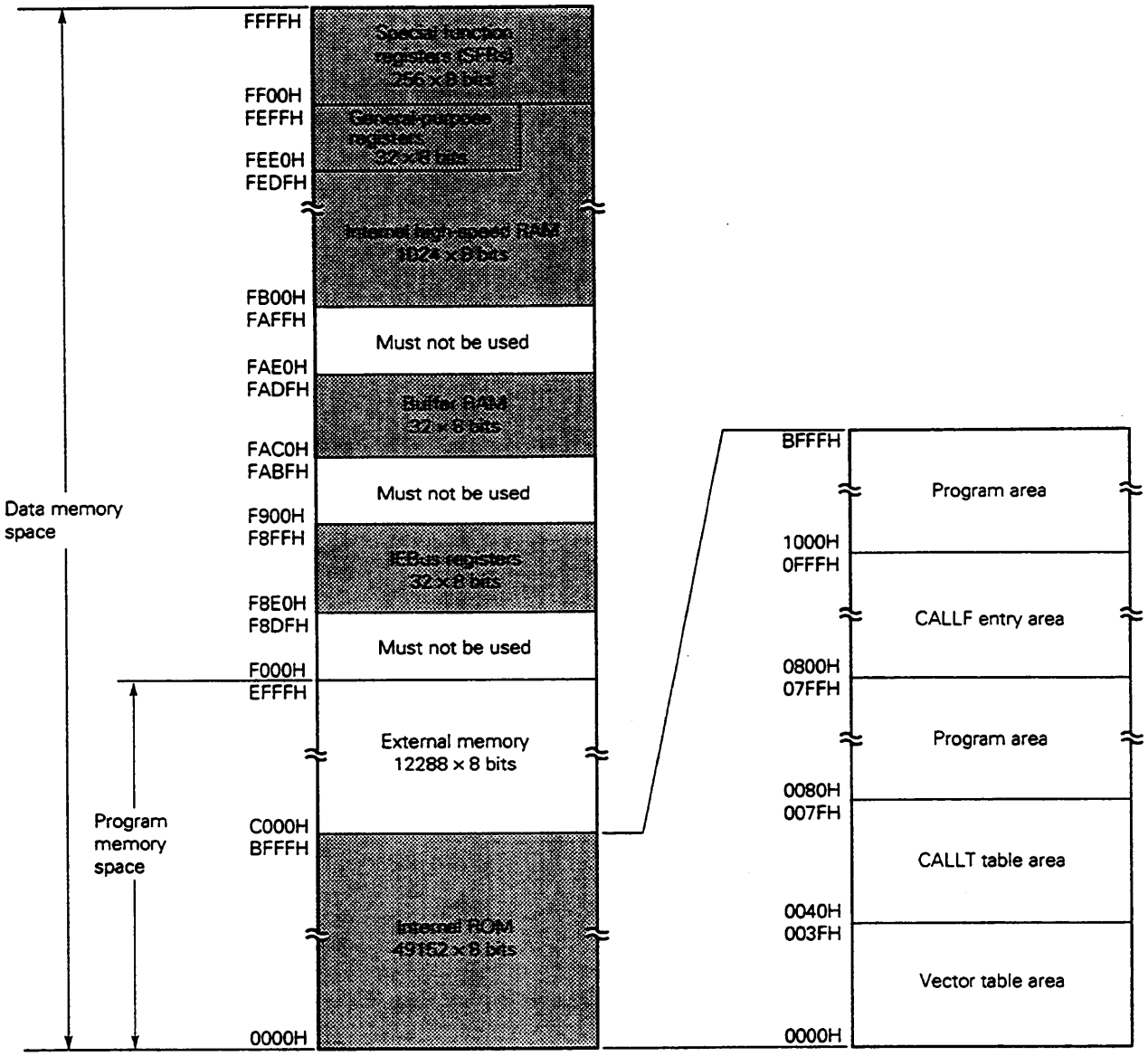
Remark: ▬ : internal memory

Fig. 3-2 Memory Map (μPD78095)



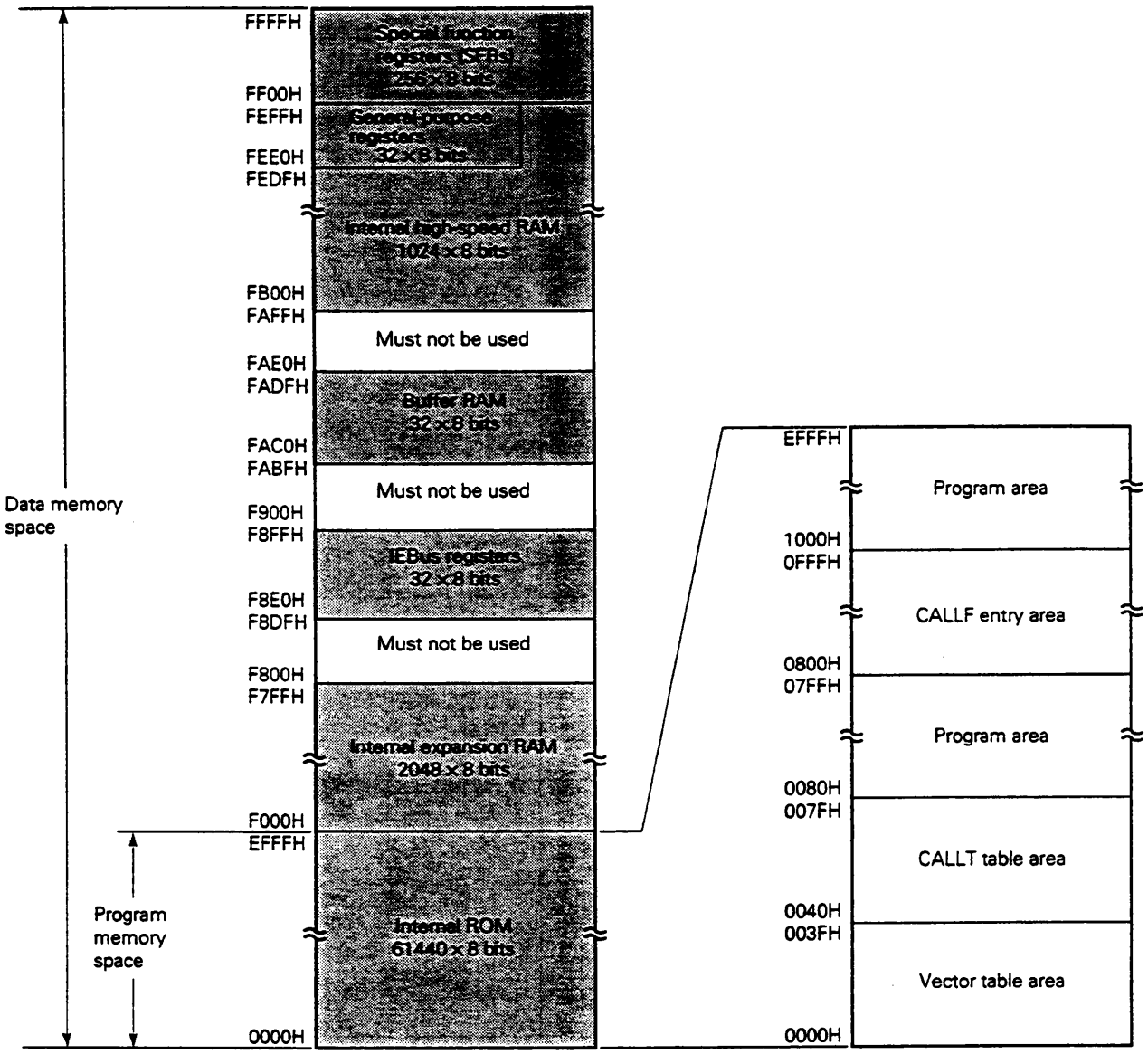
Remark: ▒▒ : internal memory

Fig. 3-3 Memory Map (μPD78096)



Remark: ▒▒ : internal memory

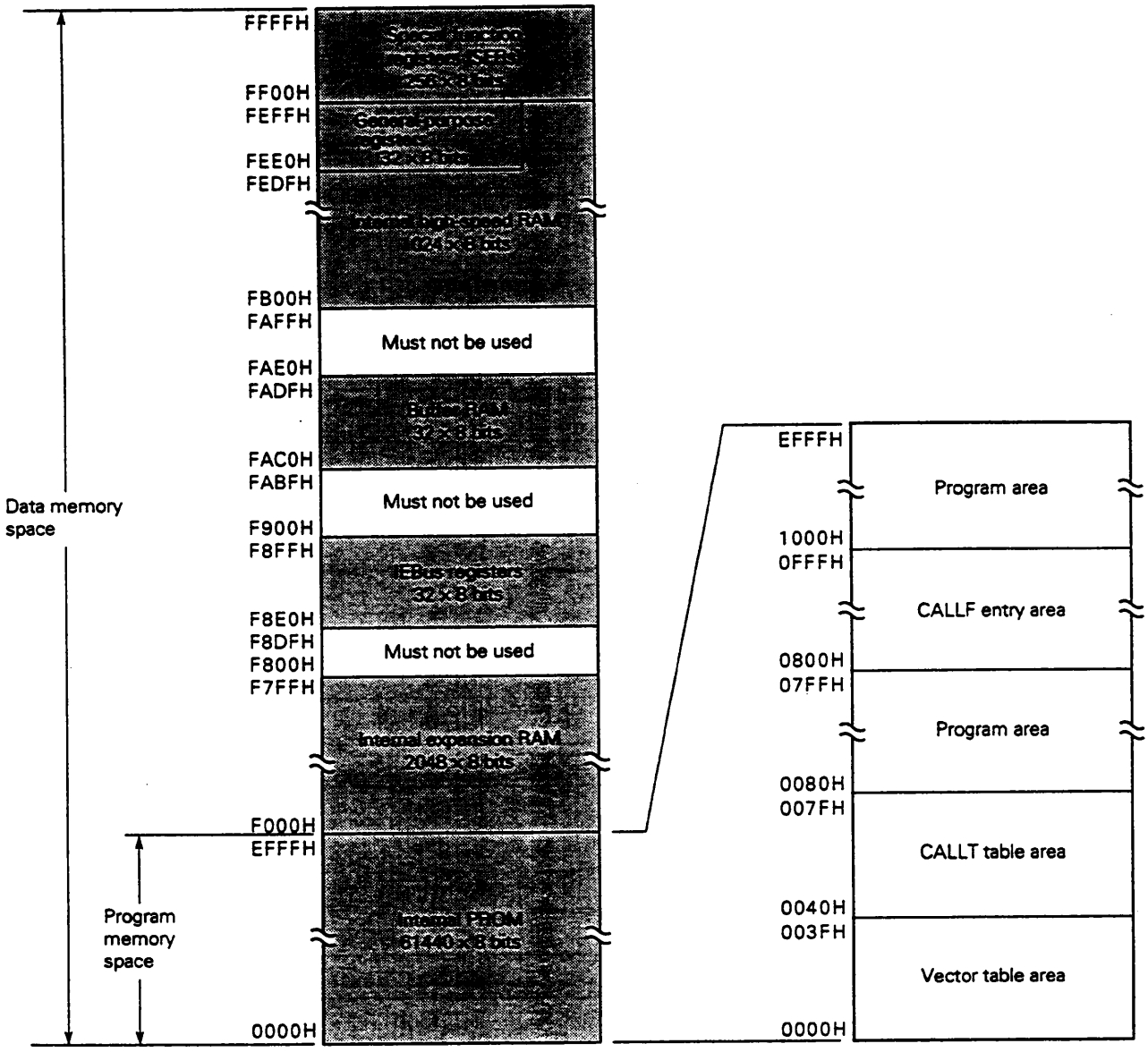
Fig. 3-4 Memory Map (μPD78098A)



Remark: ■ : internal memory

Caution: When the internal ROM is 60K bytes, the external device expansion function cannot be used. By setting the internal ROM to below 56K bytes using the memory size switching register, the area from the last address to address EFFFH of the internal ROM can be used as the external memory.

Fig. 3-5 Memory Map (μPD78P098A)



Remark: ▭ :internal memory

Caution: **When the internal PROM is 60K bytes, the external device expansion function cannot be used. By setting the internal PROM to below 56K bytes using the memory size switching register, the area from the last address to address EFFFH of the internal PROM can be used as the external memory.** ★

3.1.1 Internal program memory space

The μPD78094, μPD78095, μPD78096 and μPD78098A have mask ROMs with configurations of 32768 × 8 bits, 40960 × 8 bits, 49152 × 8 bits and 61440 × 8 bits respectively. The μPD78P098A has a PROM with a configuration of 61440 × 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
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTP5
0012H	INTP6
0014H	INTCSI0
0016H	INTCSI1
0018H	INTSER
001AH	INTSR/INTCSI2
001CH	INTST
001EH	INTTM3
0020H	INTTM00
0022H	INTTM01
0024H	INTTM1
0026H	INTTM2
0028H	INTAD
002AH	INTIE
003EH	BRK

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The μPD78098 sub-series units incorporate the following RAMs.

(1) Internal high-speed RAM

Addresses FB00H-FEFFH are of 1024 × 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 can also be used as a stack.

(2) Buffer RAM

Buffer RAM is allocated to the 32-byte area from FAC0H to FADFH. Buffer RAM can also be used as normal RAM.

(3) Internal expansion RAM ★

The internal expansion RAM can be allocated to the 2048-byte area from F000H to F7FFH only for the μPD78098A and 78P098A.

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 External memory space

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

3.1.5 IEBus register space

IEBus registers are assigned to an area of F8E0H to F8FFH.

3.1.6 Data memory addressing

The μPD78098 sub-series is provided with a variety of addressing modes which take account of memory manipulability, etc. In particular, specific types of addressing can be used which match the functions of the special function registers (SFRs), general registers, etc., in the area in which the data memory is incorporated FB00H to FFFFH. Data memory addressing is shown in Figs. 3-6 to 3-10.

Fig. 3-6 Data Memory Addressing (μPD78094)

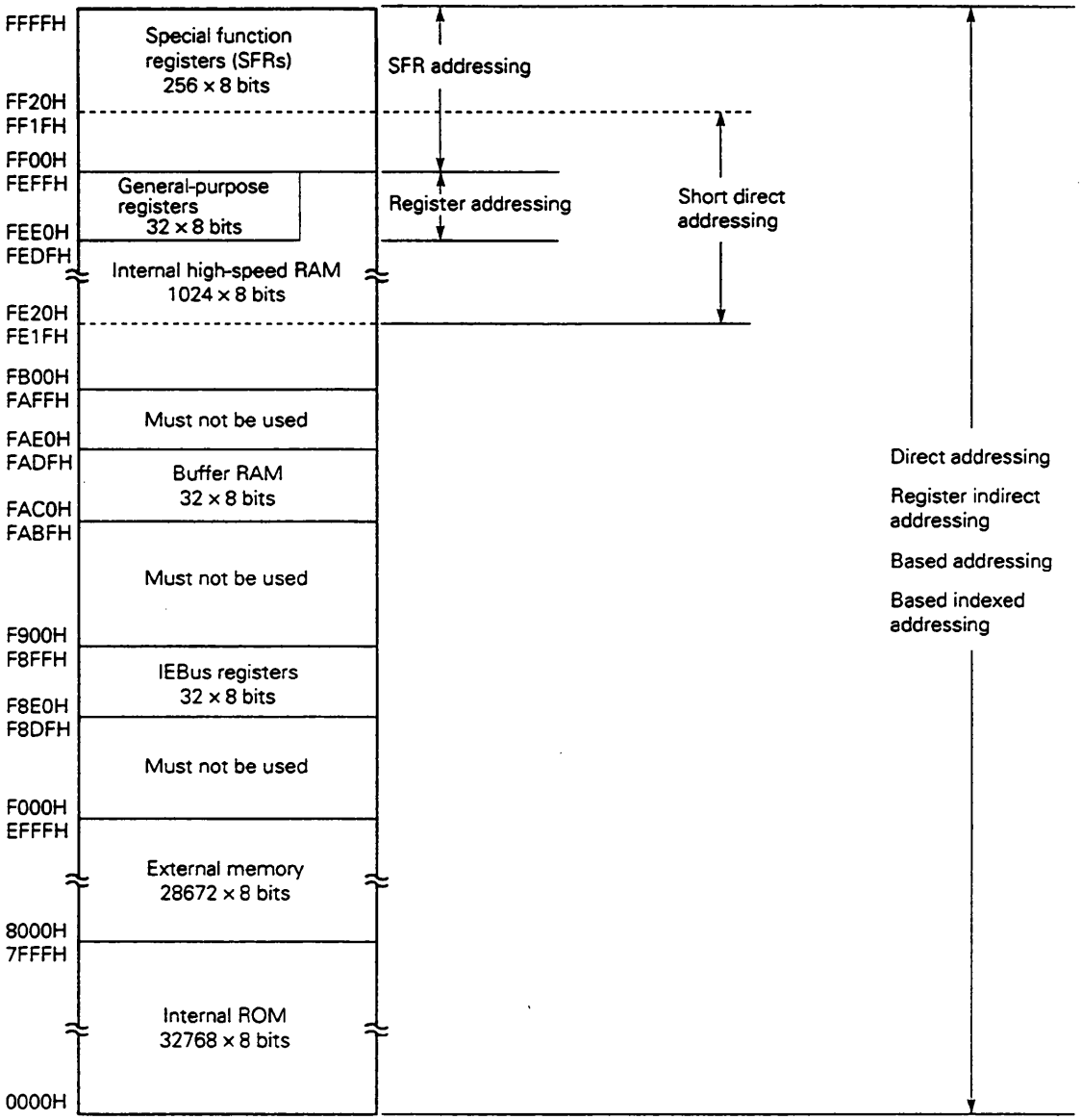


Fig. 3-7 Data Memory Addressing (μPD78095)

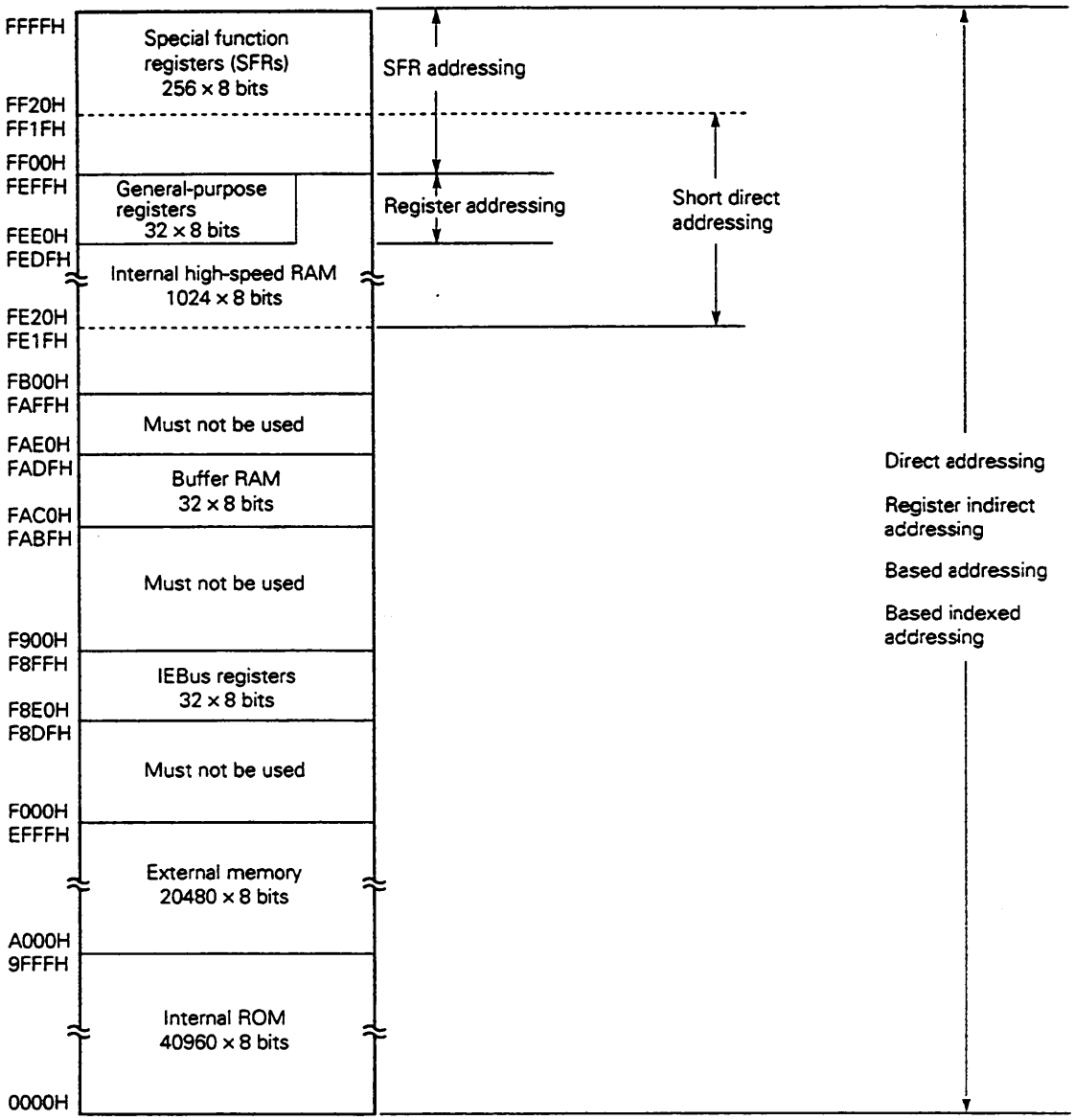


Fig. 3-8 Data Memory Addressing (μPD78096)

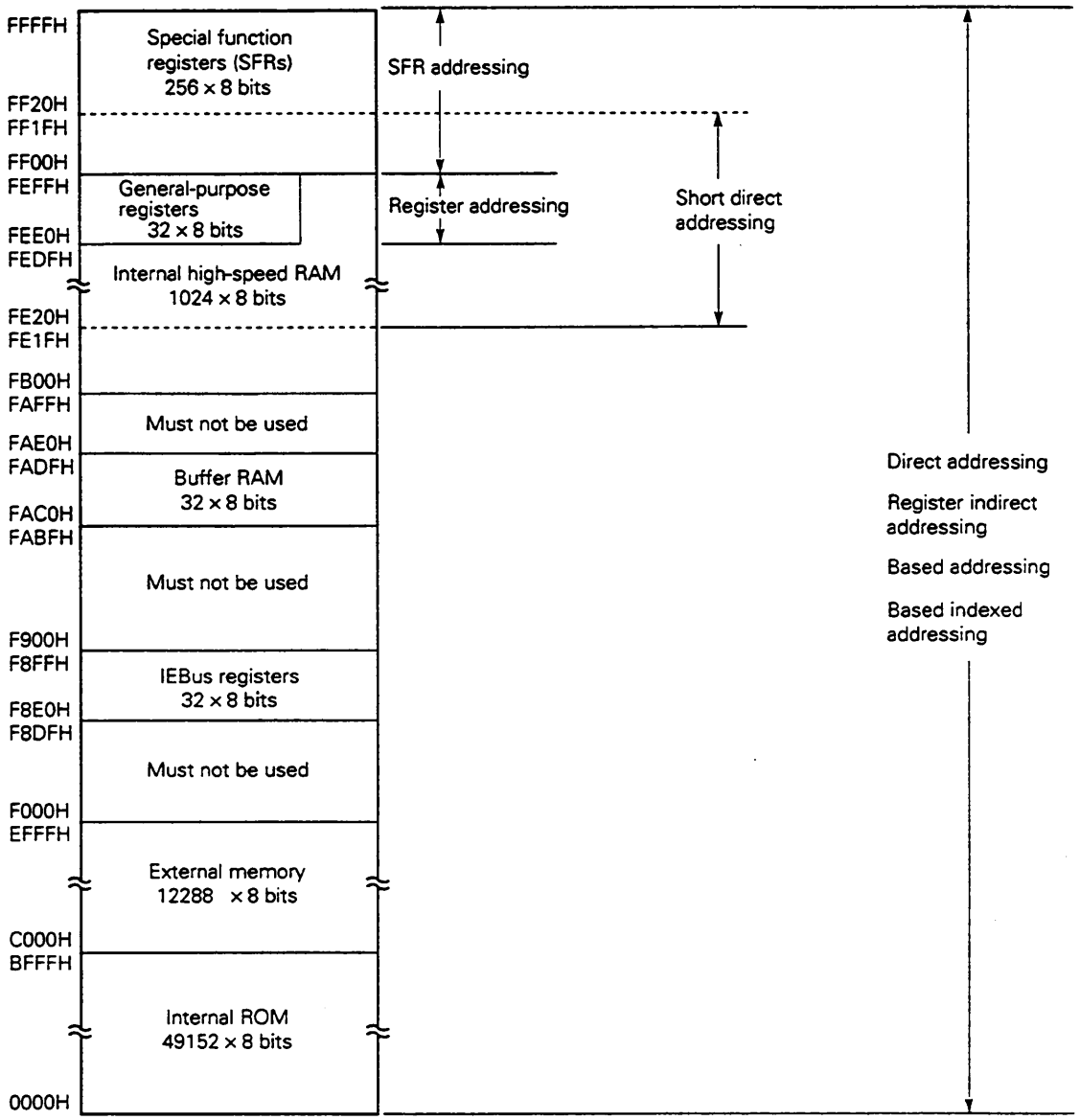


Fig. 3-9 Data Memory Addressing (μPD78098A)

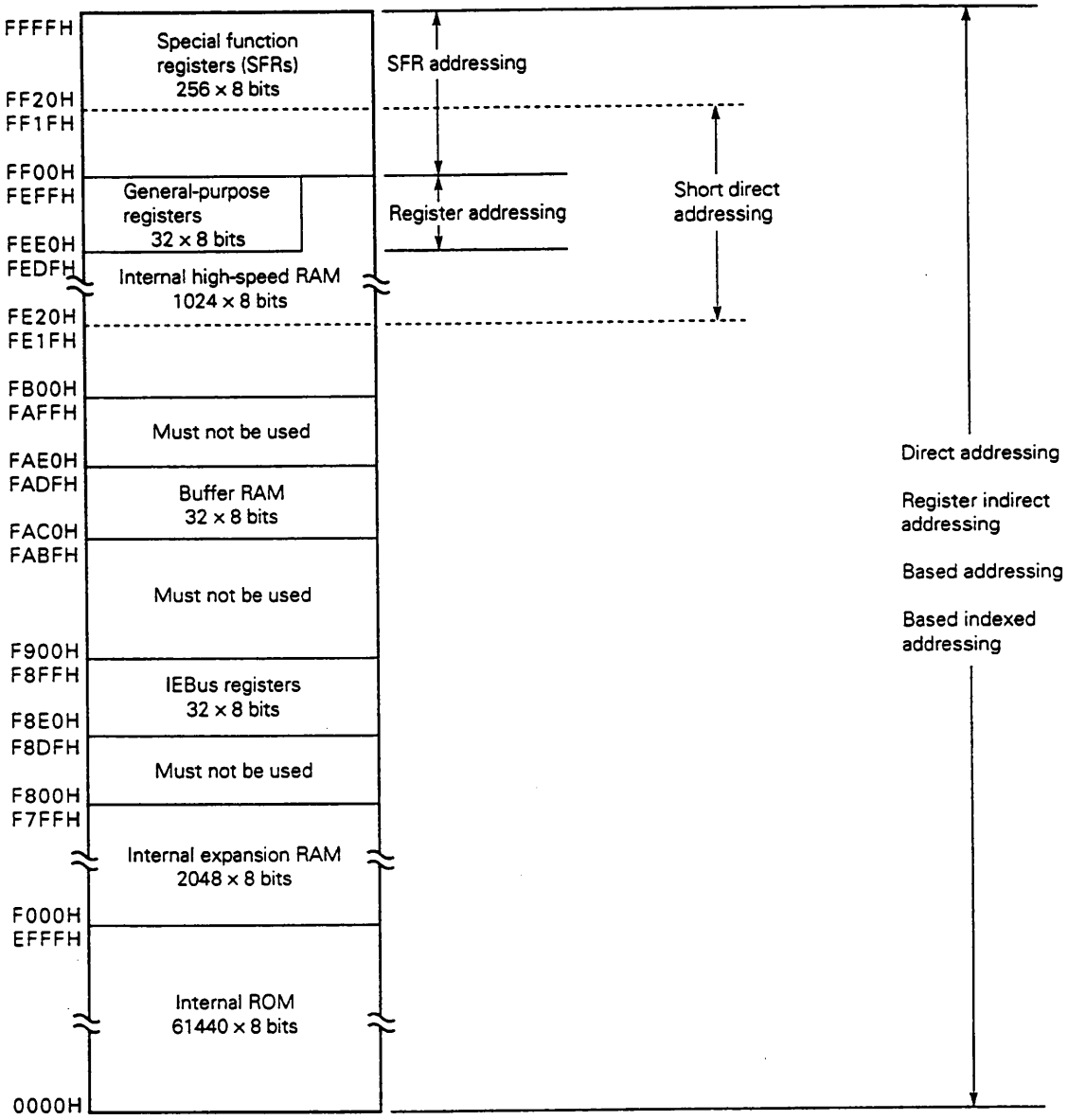
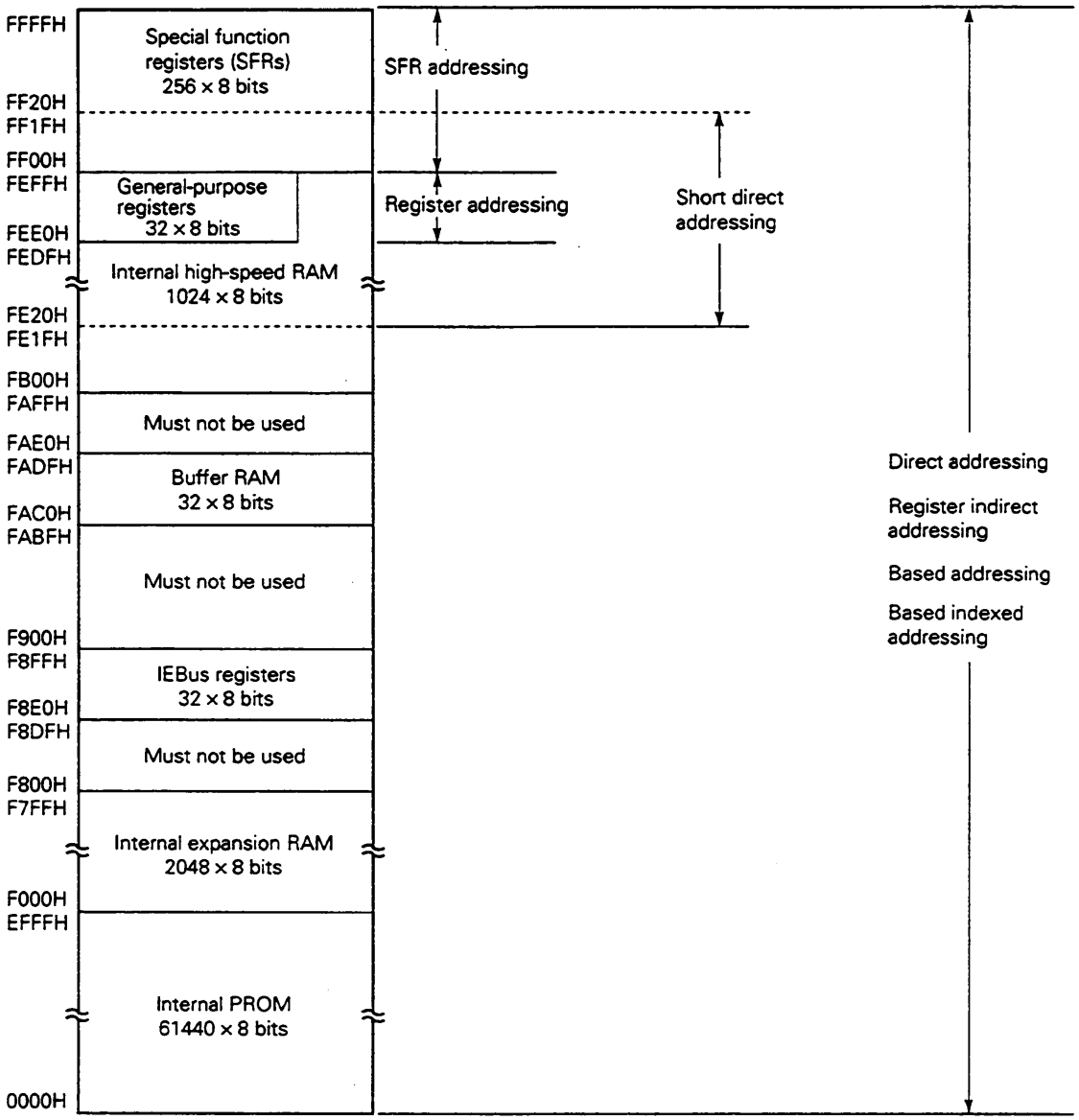


Fig. 3-10 Data Memory Addressing (μPD78P098A)



3.2 Processor Registers

The μPD78098 sub-series units incorporate the following processor registers.

3.2.1 Control registers

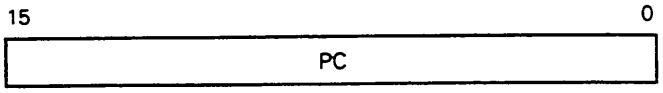
The control registers control the program sequence, statuses and stack memory. A program counter, a program status word and a stack pointer are control registers.

(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. 3-11 Program Counter Configuration



(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags to be set/reset by instruction execution. Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically reset upon execution of the RETB, RETI and POP PSW instructions.

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

Fig. 3-12 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 inservice priority flag (ISP), an interrupt mask flag for various interrupt sources and a priority specification flag.

The IE is reset to (0) upon DI instruction execution or interrupt 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 and RBS1)

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts.

When this flag is 0, low-level vectored interrupts specified with a priority specify flag register (PR) are disabled for acknowledgment. When it is 1, all interrupts are acknowledgeable. Actual acknowledgment is controlled with interrupt enable flag (IE).

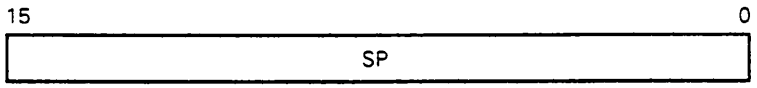
(f) Carry flag (CY)

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 (FB00H to FEFFH) can be set as the stack area.

Fig. 3-13 Stack Pointer Configuration



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

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

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

Fig. 3-14 Data to be Saved to Stack Memory

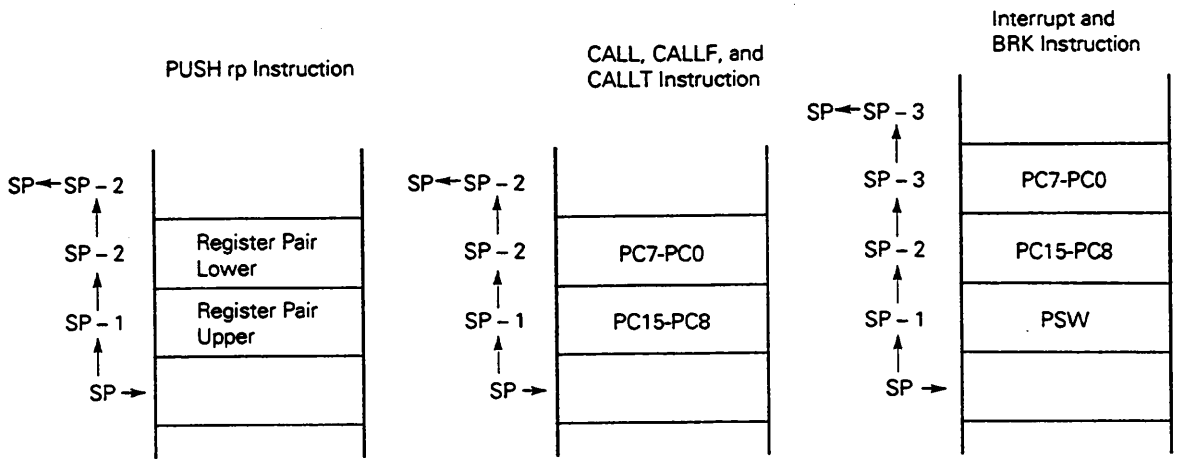
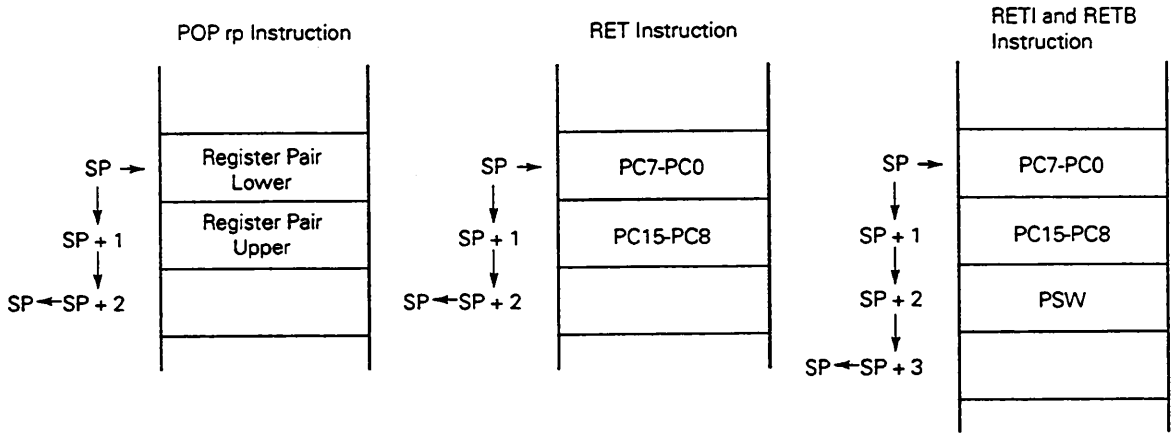


Fig. 3-15 Data to be Reset from Stack Memory



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).

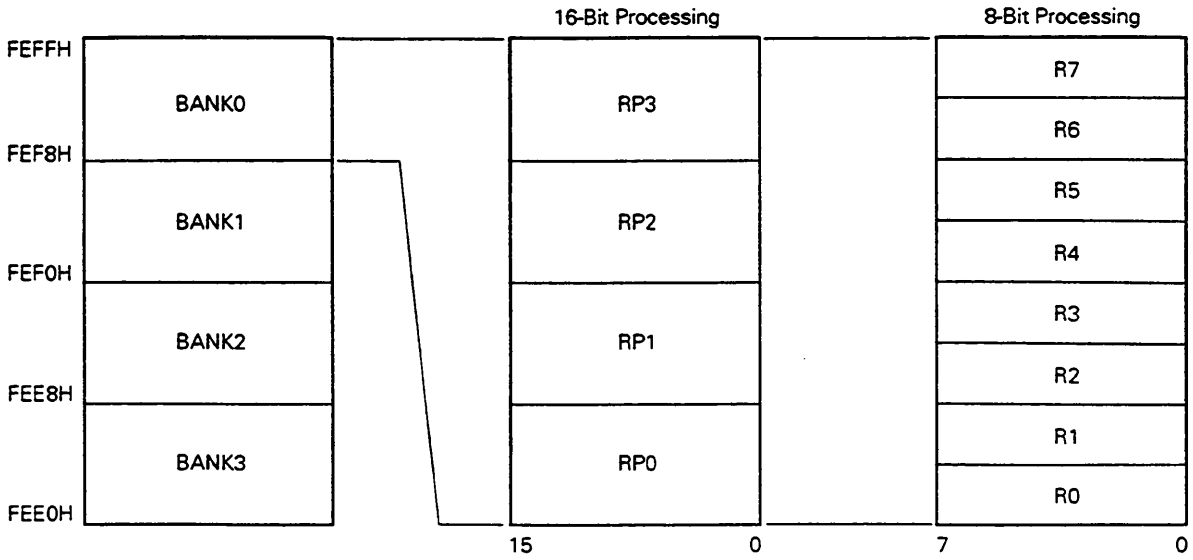
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE and HL).

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

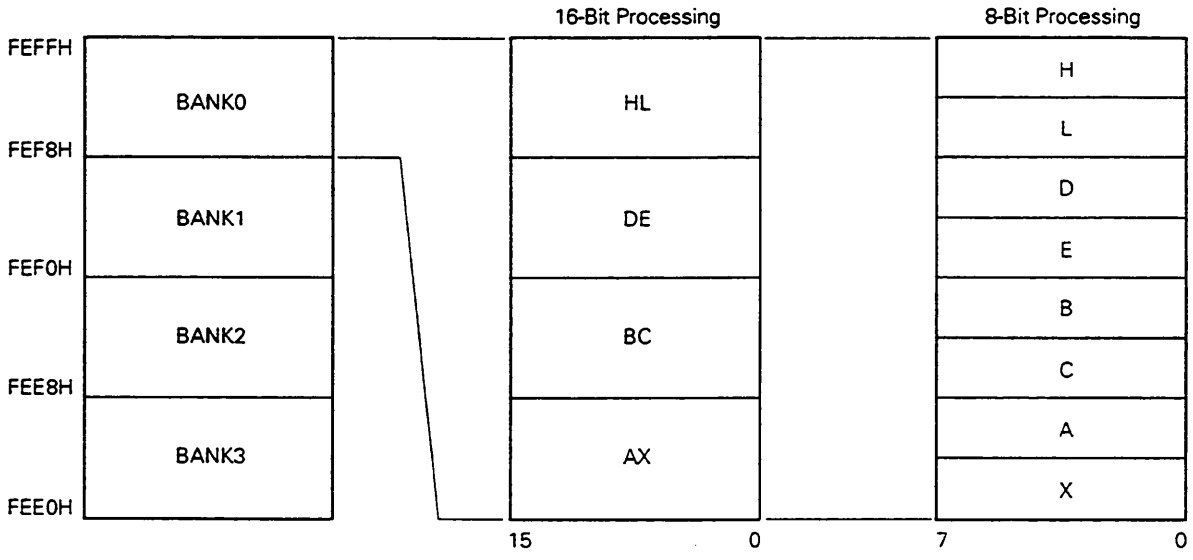
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interruption for each bank.

Fig. 3-16 General Register Configuration

(a) Absolute Name



(b) Function Name



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

Table 3-2 gives a list of special function registers. The meaning of items in the table is as follows.

- Symbol
  This is a symbol used in assembler (RA78K/0) to indicate an address of the built-in special function register. It is describable 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.
- When reset
  Indicates each register status upon $\overline{\text{RESET}}$ input.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			When Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	○	○	–	00H
FF01H	Port 1	P1		○	○	–	
FF02H	Port 2	P2		○	○	–	
FF03H	Port 3	P3		○	○	–	
FF04H	Port 4	P4		○	○	–	Undefined
FF05H	Port 5	P5		○	○	–	
FF06H	Port 6	P6		○	○	–	
FF07H	Port 7	P7		○	○	–	00H
FF0CH	Port 12	P12		○Note	○	–	
FF0DH	Port 13	P13		○	○	–	
FF10H FF11H	Capture/compare register 00	CR00		–	–	○	Undefined
FF12H FF13H	Capture/compare register 01	CR01		–	–	○	
FF14H FF15H	16-bit timer register	TM0	R	–	–	○	00H
FF16H	Compare register 10	CR10	R/W	–	○	–	Undefined
FF17H	Compare register 20	CR20		–	○	–	
FF18H	8-bit timer register 1	TMS TM1	R	–	○	○	00H
FF19H	8-bit timer register 2	TMS TM2		–	○		
FF1AH	Serial I/O shift register 0	SIO0	R/W	–	○	–	Undefined
FF1BH	Serial I/O shift register 1	SIO1		–	○	–	
FF1FH	A/D conversion result register	ADCR	R	–	○	–	
FF20H	Port mode register 0	PM0	R/W	○	○	–	FFH
FF21H	Port mode register 1	PM1		○	○	–	
FF22H	Port mode register 2	PM2		○	○	–	
FF23H	Port mode register 3	PM3		○	○	–	
FF25H	Port mode register 5	PM5		○	○	–	
FF26H	Port mode register 6	PM6		○	○	–	
FF27H	Port mode register 7	PM7		○	○	–	
FF2CH	Port mode register 12	PM12		○	○	–	
FF2DH	Port mode register 13	PM13		○	○	–	
FF30H	Real-time output buffer register L	RTBL		–	○	–	00H
FF31H	Real-time output buffer register H	RTBH		–	○	–	
FF34H	Real-time output port mode register	RTPM		○	○	–	
FF36H	Real-time output port control register	RTPC		○	○	–	

Note: Do not perform 1-bit manipulation when the IEBus controller is used.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			When Reset
				1 Bit	8 Bits	16 Bits	
FF40H	Timer clock select register 0	TCL0	R/W	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	–	
FF4CH	Capture/compare control register 0	CRC0		O	O	–	04H
FF4EH	16-bit timer output control register	TOC0		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 0	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	–	
FF69H	Automatic data transmit/receive control register	ADTC		O	O	–	
FF6AH	Automatic data transmit/receive address pointer	ADTP		–	O	–	
FF6BH	Automatic data transmit/receive interval specify register	ADTI		O	O	–	
FF70H	Asynchronous serial interface mode register	ASIM		O	O	–	
FF71H	Asynchronous serial interface status register	ASIS	R	O	O	–	
FF72H	Serial operating mode register 2	CSIM2	R/W	O	O	–	
FF73H	Baud rate generator control register	BRGC		–	O	–	
FF74H	Transmit shift register / Receive buffer register	TXS / RXB SIO2	W / R	–	O	–	FFH
FF80H	A/D converter mode register	ADM	R/W	O	O	–	01H
FF84H	A/D converter input select register	ADIS		–	O	–	00H
FF90H	D/A conversion value set register 0	DACS0		–	O	–	
FF91H	D/A conversion value set register 1	DACS1		–	O	–	
FF98H	D/A converter mode register	DAM		O	O	–	

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			When Reset
					1 Bit	8 Bits	16 Bits	
FFD0H to FFDFH	External access area[Note 1]			R/W	○	○	–	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		○	○	○	00H
FFE1H	Interrupt request flag register 0H		IF0H		○	○		
FFE2H	Interrupt request flag register 1L	IF1L			○	○	–	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		○	○	○	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		○	○		
FFE6H	Interrupt mask flag register 1L	MK1L			○	○	–	
FFE8H	Priority order specify flag register 0L	PR0	PR0L		○	○	○	
FFE9H	Priority order specify flag register 0H		PR0H		○	○		
FFEAH	Priority order specify flag register 1L	PR1L			○	○	–	
FFECH	External interrupt mode register 0	INTM0			–	○	–	00H
FFEDH	External interrupt mode register 1	INTM1			–	○	–	
FFF0H	Memory size switching register	IMS		W	–	○	–	Note 2
FFF2H	Oscillation mode selection register	OSMS			–	○	–	00H
FFF3H	Pull-up resistor option register H	PUOH		R/W	○	○	–	
★ FFF4H	Internal expansion RAM size switching register	IXS[Note3]		W	–	○	–	08H[Note4]
FFF6H	Key return mode register	KRM		R/W	○	○	–	00H
FFF7H	Pull-up resistor option register L	PUOL			○	○	–	10H
FFF8H	Memory expansion mode register	MM			○	○	–	00H
FFF9H	Watchdog timer mode register	WDTM			○	○	–	04H
FFFAH	Oscillation stabilization time select register	OSTS			–	○	–	
FFFBH	Processor clock control register	PCC			○	○	–	

Notes:

(1) The external access area cannot be accessed in SFR addressing. Access the area with an instruction that can specify addresses of 16 bits.

(2) The value when reset depends on products.
μPD78094 : C8H, μPD78095 : CAH, μPD78096 : CCH, μPD78098A : CFH,
μPD78P098A : CFH
When using a mask ROM version, do not set values other than those in reset to IMS (excluding when using the external device expansion function for μPD78098A).

(3) μPD78098A and μPD78P098A only.

(4) When using a mask ROM version, do not set values other than those in reset to IXS.

3.2.4 IEBus registers

The IEBus registers are mainly used to control the IEBus controller, and is assigned to an area of F8E0H to F8FFH. These registers can be manipulated by operation instructions and transfer instructions.

- 1-bit manipulation

  The IEBus registers cannot be read or written in 1-bit units.

  To write 1 bit of these registers, use an 8-bit transfer instruction with information for the other 7 bits.

  To reference (read) 1 bit, read an 8-bit value to the A register, and then execute a bit manipulation instruction on the A register.

- 8-bit manipulation

  To manipulate the IEBus registers in 8-bit units, describe a symbol reserved by assembler as the operand of an 8-bit manipulation instruction (!addr16).

  The 8 bits to be manipulated can also be specified by an address.

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

- Symbol

  These symbols are used by the assembler (RA78K/0) that indicates the address of the internal IEBus register and can be used as operand of instructions.

- R/W

  This column indicates whether the corresponding IEBus register can be read (Read) or written (Write).

  R/W : Read/write

  R : Read only

  W : Write only

- Bit units for manipulation

  The IEBus registers can be manipulated only in 8-bit units.

- When reset

  Indicates the status of each register when the $\overline{\text{RESET}}$ signal is input.

Table 3-3 IEBus Register List

Address	Register Name	Symbol	R/W	Manipulatable Bit Unit			When Reset
				1 Bit	8 Bits	16 Bits	
F8E0H	Clock select register 1	IECL1	R/W	–	O	–	00H
F8E1H	Clock select register 2	IECL2		–	O	–	
F8E2H	A/D current cutting select register	IEAD		–	O	–	
F8E3H	IEBus controller mode register	IECM		–	O	–	
F8F0H	Control register	CTR	W	–	O	–	xxx00x01B
	Status register 1	STR1	R				01011xxxB
F8F1H	Command register	CMR	W	–	O	–	00H
	Status register 2	STR2	R				02H
F8F2H	Unit address register 1	UAR1	W	–	O	–	Undefined
	Receive data number register 1	RDR1	R				00H
F8F3H	Unit address register 2	UAR2	W	–	O	–	Undefined
	Receive data number register 2	RDR2	R				00H
F8F4H	Slave address register 1	SAR1	W	–	O	–	Undefined
	Lock address register 1	LOR1	R				
F8F5H	Slave address register 2	SAR2	W	–	O	–	Undefined
	Lock address register 2	LOR2	R				0000xxxxB
F8F6H	Master communication register	MCR	W	–	O	–	Undefined
	Multiaddress calling address register 1	DAR1	R				
F8F7H	Multiaddress calling address register 2	DAR2	R	–	O	–	Undefined
F8F8H	Return code register	RCR	R	–	O	–	4FH
F8FEH	Transmit buffer register	TBF	W	–	O	–	Undefined
	Receive buffer register	RBF	R				

3.3 Instruction Address Addressing

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

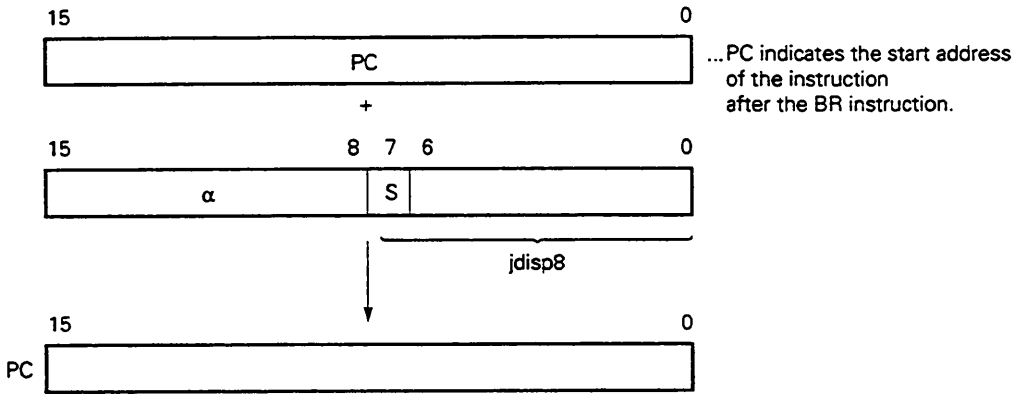
3.3.1 Relative addressing

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

3.3.2 Immediate addressing

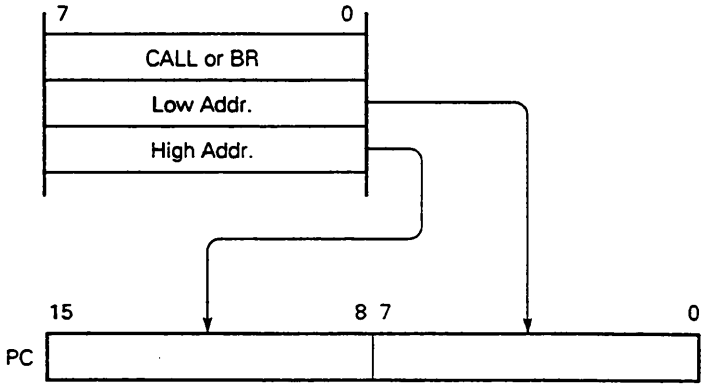
[Function]

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

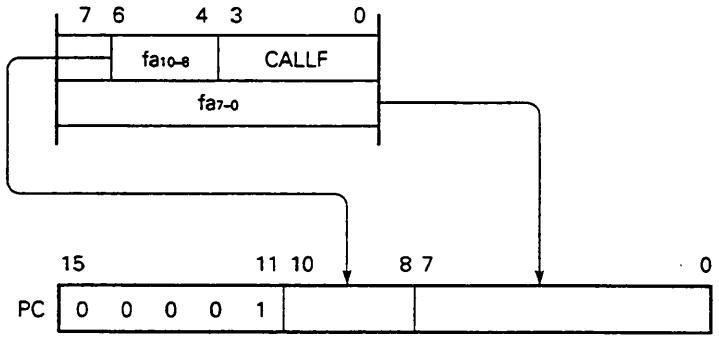
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

[Illustration]

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



In the case of CALLF !addr11 instruction



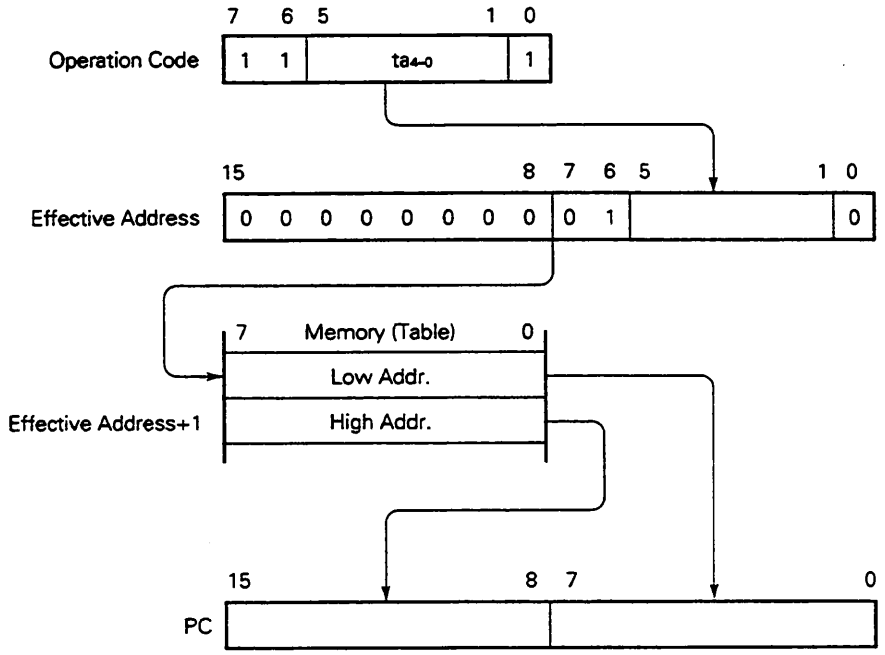
3.3.3 Table indirect addressing

[Function]

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

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

[Illustration]



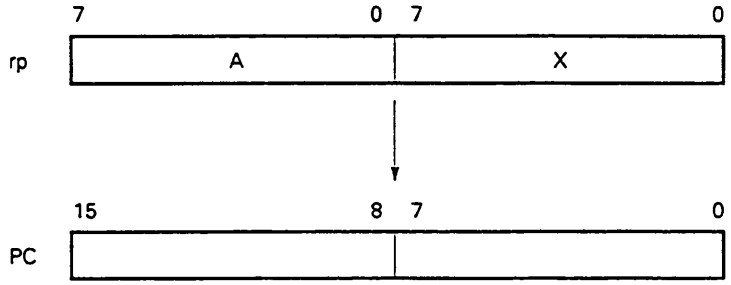
3.3.4 Register addressing

[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 μPD78098 sub-series 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 dividend and quotient storage
ADJBA/ADJBS	A register for storage of numeric values subject to decimal adjustment
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 x 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 and RBS1).

Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

[Operand format]

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

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

[Description example]

MOV A, C; when selecting C register as r

Operation code	0	1	1	0	0	0	1	0

INCW DE; when selecting DE register pair as rp

Operation code	1	0	0	0	0	1	0	0

3.4.3 Direct addressing

[Function]

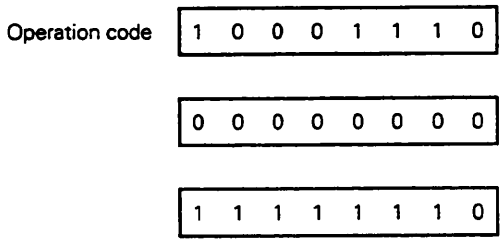
The memory to be manipulated is addressed with immediate data in an instruction word becoming an operand address.

[Operand format]

Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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



3.4.4 Short direct addressing

[Function]

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

This addressing is applied to the 256-byte space FE20H to FF1FH. An internal RAM and a special function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

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

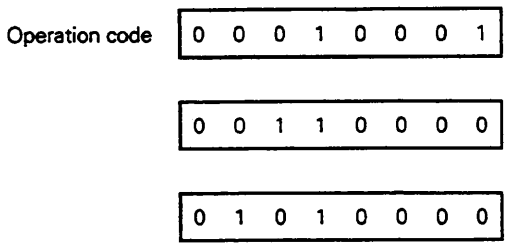
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1.

[Operand format]

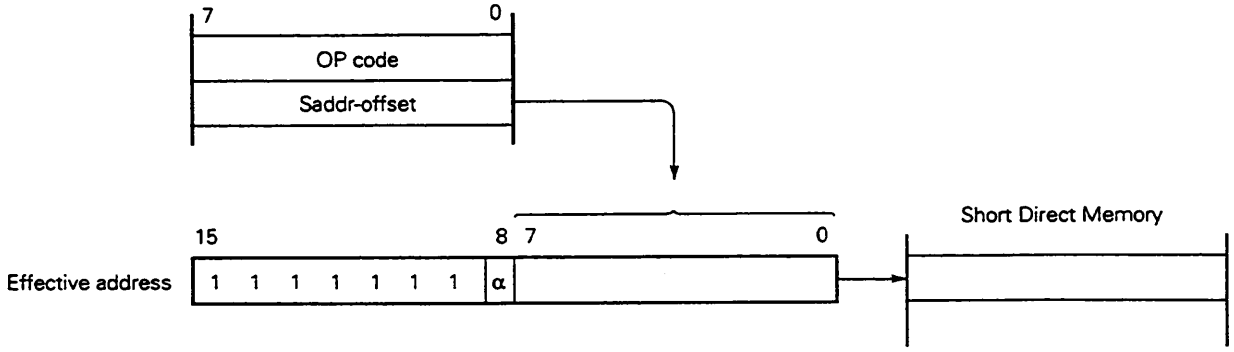
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Description example]

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



[Illustration]



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

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

3.4.5 Special function register (SFR) addressing

[Function]

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

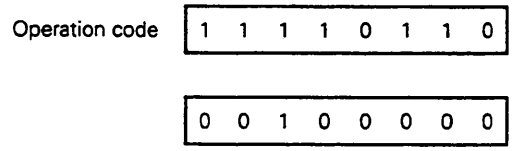
This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFR mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

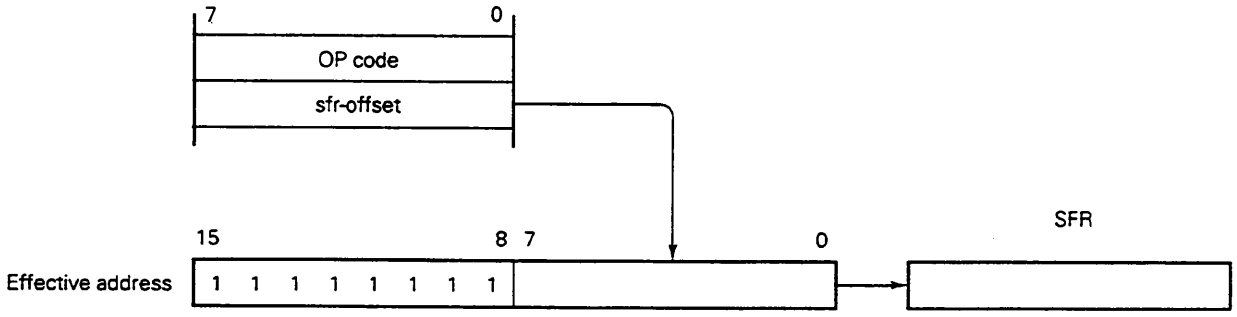
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

MOV PM0, A; when selecting PM0 as sfr



[Illustration]



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

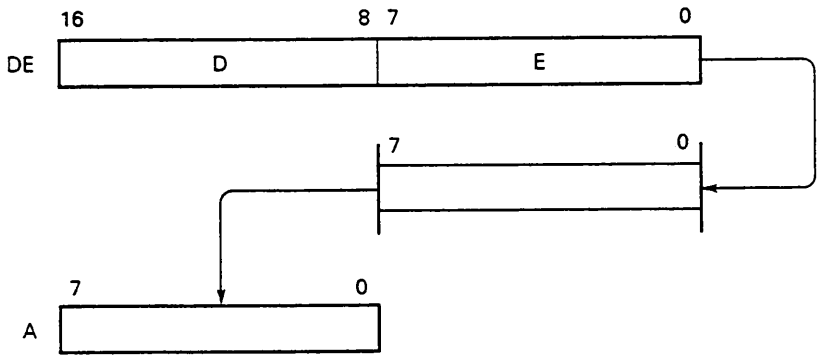
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

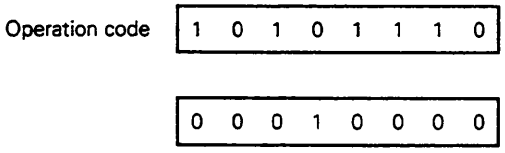
8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified with the register bank select flag (RBS0 and RBS1) and the sum is used to address the memory. Addition is performed by expanding the offset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

[Operand format]

Identifier	Description
—	[HL+byte]

[Description example]

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



3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction are added to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified with the register bank select flag (RBS0 and RBS1) and the sum is used to address the memory.

Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

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

[Description example]

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

Operation code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

[MEMO]

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD78098 sub-series units incorporate two input ports and sixty-seven 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

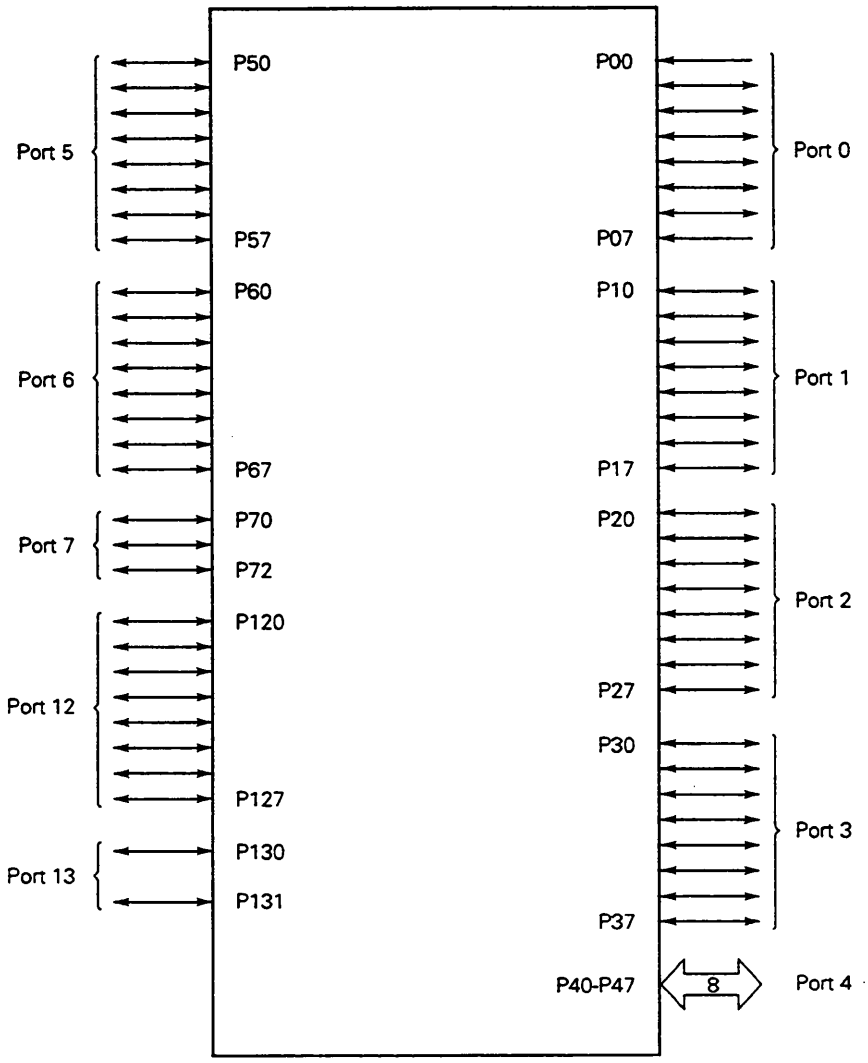


Table 4-1 Port Functions (1/2)

<table>
<tr><th>Pin Name</th><th colspan="2">Function</th><th>Dual-Function Pin</th></tr>
<tr><td>P00</td><td rowspan="8">Port 0.
8-bit input/output port.</td><td>Input only.</td><td>INTP0/TI00</td></tr>
<tr><td>P01</td><td rowspan="6">Input/output specifiable bit-wise.
If used as an input port, a pull-up resistor can be connected by software.</td><td>INTP1/TI01</td></tr>
<tr><td>P02</td><td>INTP2</td></tr>
<tr><td>P03</td><td>INTP3</td></tr>
<tr><td>P04</td><td>INTP4</td></tr>
<tr><td>P05</td><td>INTP5</td></tr>
<tr><td>P06</td><td>INTP6</td></tr>
<tr><td>P07</td><td>Input only.</td><td>XT1</td></tr>
<tr><td>P10 to P17</td><td colspan="2">Port 1.
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.</td><td>ANI0 to ANI7</td></tr>
<tr><td>P20</td><td colspan="2" rowspan="8">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.</td><td>SI1</td></tr>
<tr><td>P21</td><td>SO1</td></tr>
<tr><td>P22</td><td>$\overline{SCK1}$</td></tr>
<tr><td>P23</td><td>STB</td></tr>
<tr><td>P24</td><td>BUSY</td></tr>
<tr><td>P25</td><td>SI0/SB0</td></tr>
<tr><td>P26</td><td>SO0/SB1</td></tr>
<tr><td>P27</td><td>$\overline{SCK0}$</td></tr>
<tr><td>P30</td><td colspan="2" rowspan="8">Port 3.
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.</td><td>TO0</td></tr>
<tr><td>P31</td><td>TO1</td></tr>
<tr><td>P32</td><td>TO2</td></tr>
<tr><td>P33</td><td>TI1</td></tr>
<tr><td>P34</td><td>TI2</td></tr>
<tr><td>P35</td><td>PCL</td></tr>
<tr><td>P36</td><td>BUZ</td></tr>
<tr><td>P37</td><td>-</td></tr>
<tr><td>P40 to P47</td><td colspan="2">Port 4. 8-bit input/output port.
Input/output specifiable in 8-bit units.
If used as an input port, a pull-up resistor can be connected by software.
Test input flag (KRIF) is set to 1 by falling edge detection.</td><td>AD0 to AD7</td></tr>
<tr><td>P50 to P57</td><td colspan="2">Port 5. 8-bit input/output port.
LED can be driven directly.
Input/output specifiable bit-wise.
If used as an input port, a pull-up resistor can be connected by software.</td><td>A8 to A15</td></tr>
</table>

Table 4-1 Port Functions (2/2)

<table>
<tr><th>Pin Name</th><th colspan="2">Function</th><th>Dual-Function Pin</th></tr>
<tr><td>P60</td><td rowspan="8">Port 6.
8-bit input/output port.
Input/output specifiable bit-wise.</td><td rowspan="2">N-ch open drain input/output port.
On-chip pull-up resistor specifiable by mask option. (Mask ROM version only). LEDs can be driven directly.</td><td rowspan="2">–</td></tr>
<tr><td>P61</td></tr>
<tr><td>P62</td><td rowspan="6">If used as an input port, a pull-up resistor can be connected by software.</td><td>$\overline{RD}$</td></tr>
<tr><td>P63</td><td>$\overline{WR}$</td></tr>
<tr><td>P64</td><td>$\overline{WAIT}$</td></tr>
<tr><td>P65</td><td>ASTB</td></tr>
<tr><td>P66</td><td colspan="2" rowspan="3">Port 7.
3-bit input/output port.
Input/output specifiable bit-wise.
If used as an input port, a pull-up resistor can be connected by software.</td><td>SI2/RxD</td></tr>
<tr><td>P67</td><td>SO2/TxD</td></tr>
<tr><td>P70</td><td>$\overline{SCK2}$/ASCK</td></tr>
<tr><td>P71</td><td colspan="2" rowspan="4">Port 12.
8-bit input/output port.
Input/output specifiable bit-wise.
If used as an input port, pull-up resistor can be connected by software.</td><td>RTP0 to RTP3</td></tr>
<tr><td>P72</td><td>RTP4/$\overline{TX}$</td></tr>
<tr><td>P120 to P123</td><td>RTP5/$\overline{RX}$</td></tr>
<tr><td>P124</td><td>RTP6, RTP7</td></tr>
<tr><td>P125</td><td colspan="2">Port 13.
2-bit input/output port.
Input/output specifiable bit-wise.
If used as an input port, pull-up resistor can be connected by software.</td><td>ANO0, ANO1</td></tr>
<tr><td>P126, P127</td></tr>
<tr><td>P130, P131</td></tr>
</table>

4.2 Port Configuration

A port consists of the following hardware.

Table 4-2 Port Configuration

Item	Configuration	
Control register	Port mode register	(PMm: m = 0 to 3, 5 to 7, 12, 13)
	Pull-up resistor option register	(PUOH, PUOL)
	Memory expansion mode register	(MM)[Note]
	Key return mode register	(KRM)
Port	Total: 69 ports (2 inputs, 67 inputs/outputs)	
Pull-up resistor	• Mask ROM products Total: 67 (software control: 63 mask option control: 4) • μPD78P098A Total: 63	

Note: Memory expansion mode register specifies port4 input/output.

4.2.1 Port 0

Port 0 is a 8-bit input/output port with output latch. P01 to P06 pins can specify the input mode/output mode in 1-bit units with the port mode register 0. P00 and P07 pins are input-only ports. When P01 to P06 pins are used as input ports, a pull-up resistor can be connected to them in 6-bit units with a pull-up resistor option register L.

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 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 P07 Configurations

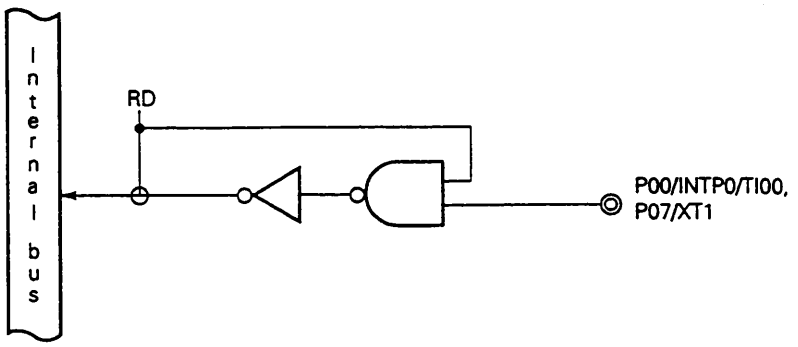
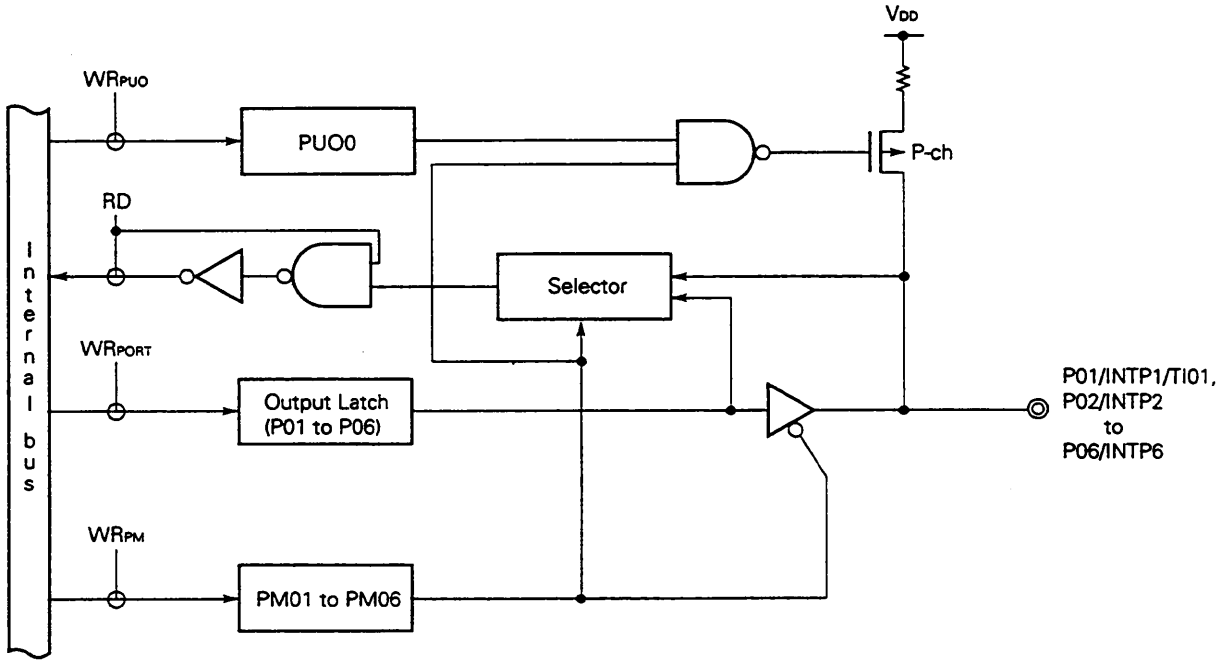


Fig. 4-3 P01 to P06 Configurations



Remarks:

(1) PUO : Pull-up resistor option register

(2) PM : Port mode register

(3) RD : Port 0 read signal

(4) WR : Port 0 write signal

4.2.2 Port 1

Port 1 is an 8-bit input/output port with output latch. It can specify the input mode/output mode in 1-bit units with a port mode register 1. When P10 to P17 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 L.

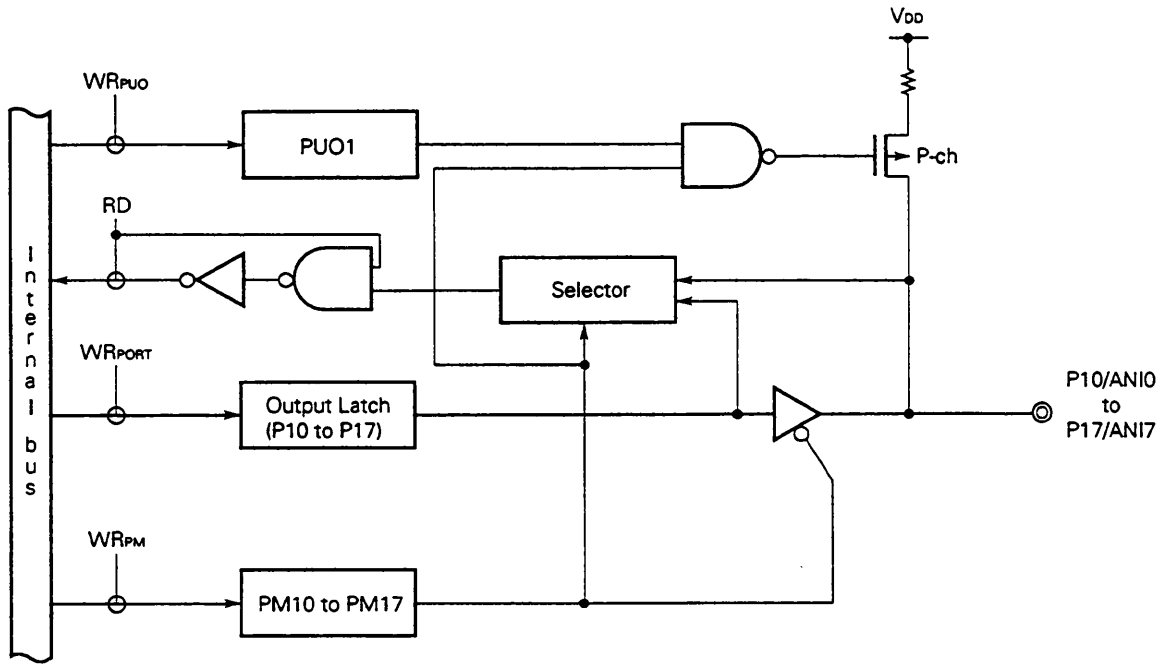
Dual-functions include an A/D converter analog input.

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

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

Caution: A pull-up resistor cannot be used for pins used as A/D converter analog input.

Fig. 4-4 P10 to P17 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 1 read signal
(4) WR : Port 1 write signal

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 the 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 L.

Dual-functions include serial interface data input/output, clock input/output, automatic transmit/receive busy input, and strobe output.

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

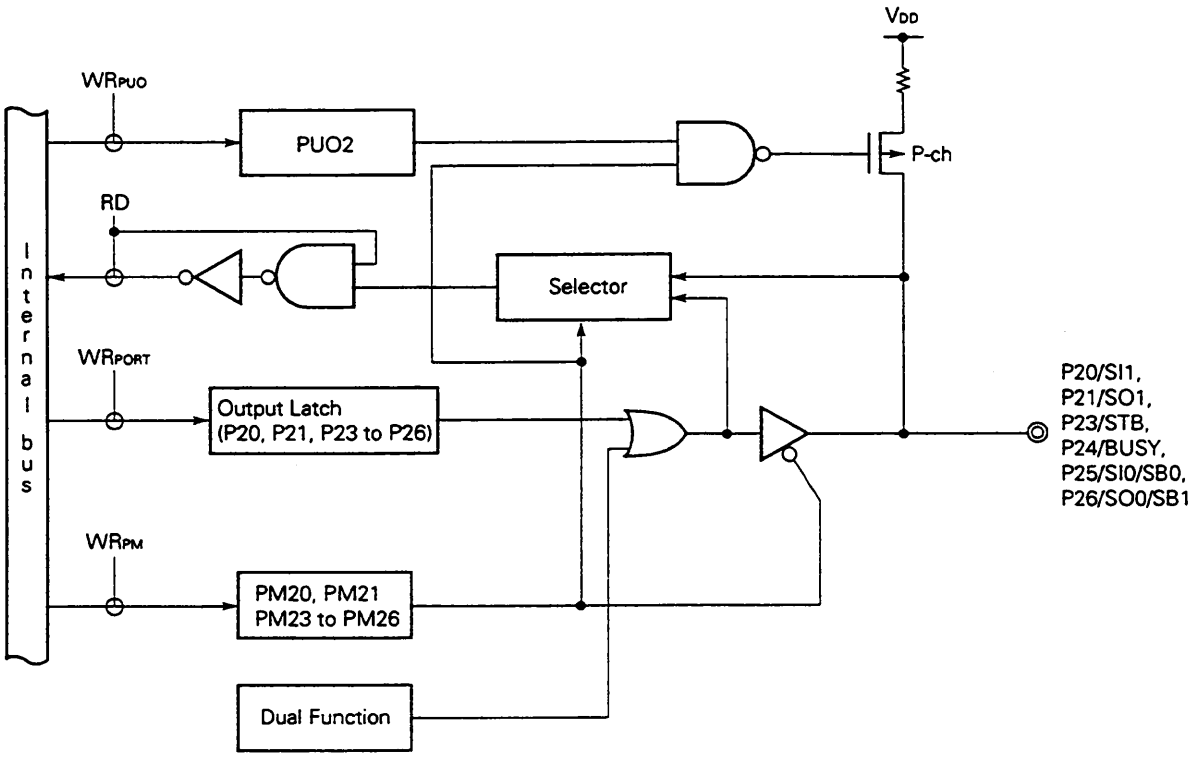
Figs. 4-5 and 4-6 show a block diagram of port 2.

Cautions:

(1) When using as a serial interface, the input/output and output latch must be set according to the functions used. For details of how to set, refer to the format of the serial operation mode register 0 in Fig. 14-4 and the format of the serial operation mode register 1 in Fig. 15-3. ★

(2) When reading the pin state in SBI mode, set PM2n to 1 (n = 5, 6) (Refer to 14.4.3 SBI mode operation (10) SBI mode precautions (e)).

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



Remarks:

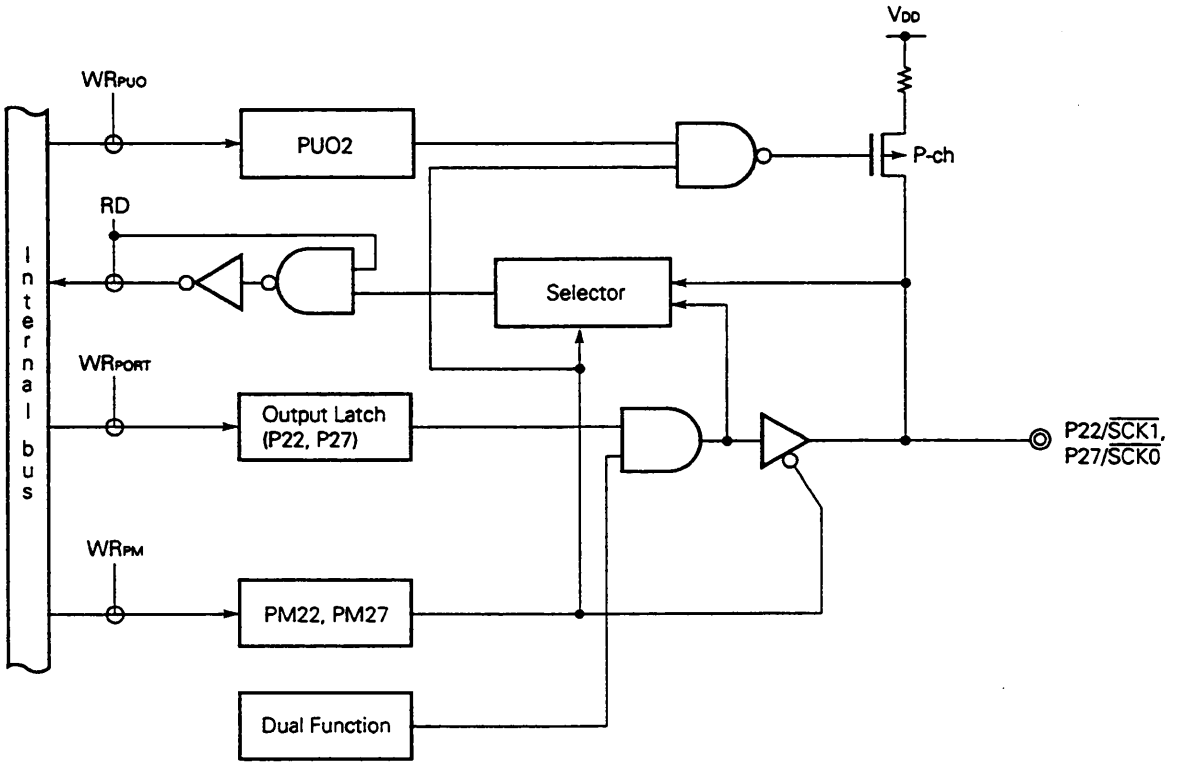
(1) PUO : Pull-up resistor option register

(2) PM : Port mode register

(3) RD : Port 2 read signal

(4) WR : Port 2 write signal

Fig. 4-6 P22 and P27 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 2 read signal
(4) WR : Port 2 write signal

4.2.4 Port 3

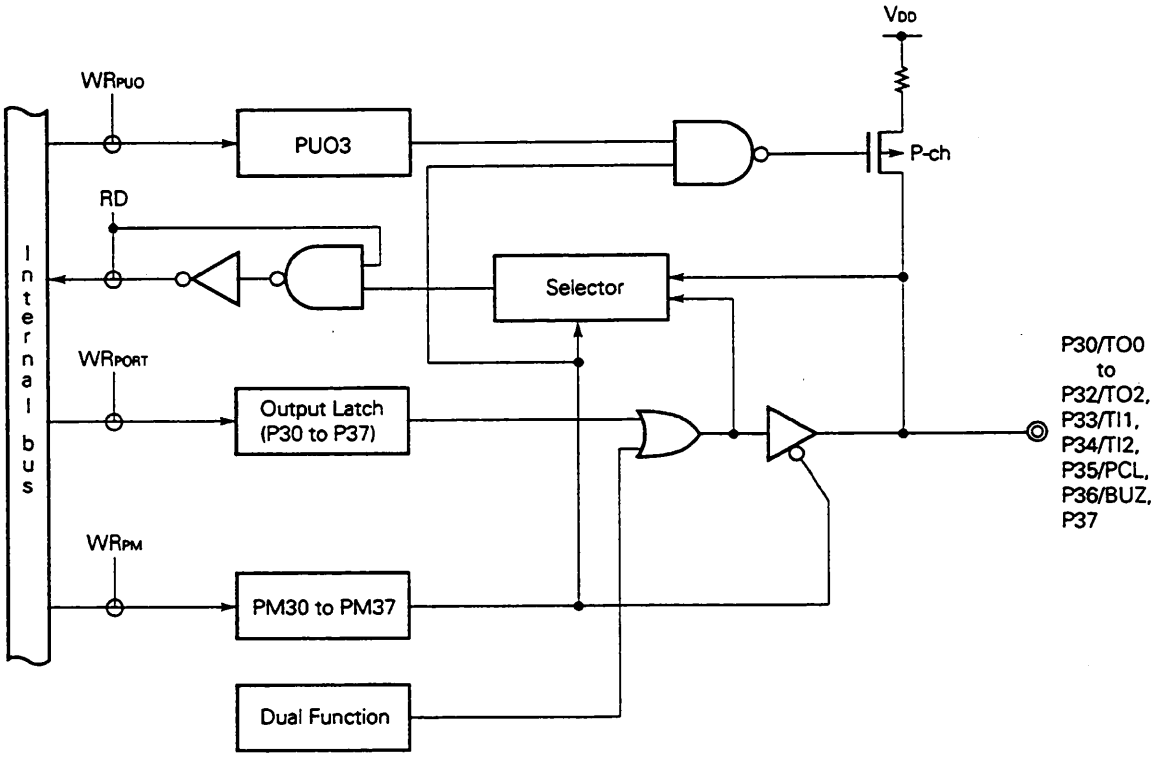
Port 3 is an 8-bit input/output port with output latch. P30 to P37 pins can specify the input mode/output mode in 1-bit units with the port mode register 3. When P30 to P37 pins are used as input ports, a pull-up resistor can be connected to them in 8-bit units with a pull-up resistor option register L.

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

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

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

Fig. 4-7 P30 to P37 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 3 read signal
(4) WR : Port 3 write signal

4.2.5 Port 4

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

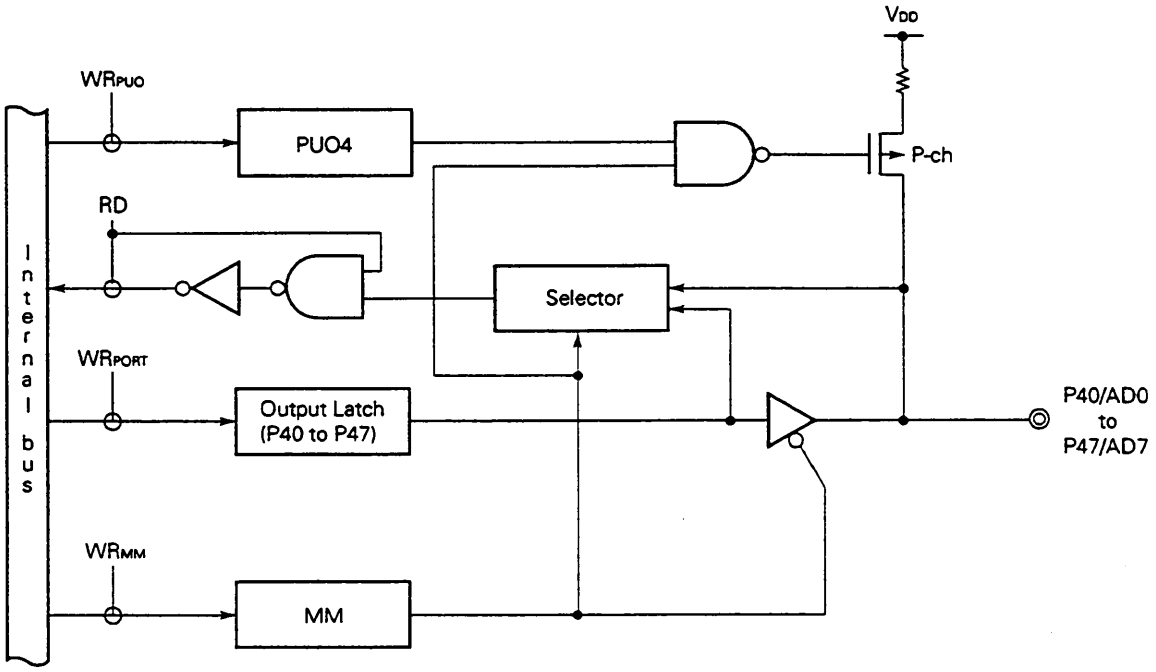
The test input flag (KRIF) can be set to 1 by detecting falling edges.

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

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

Figs. 4-8 and 4-9 show a block diagram of port 4 and block diagram of falling edge detection circuit, respectively.

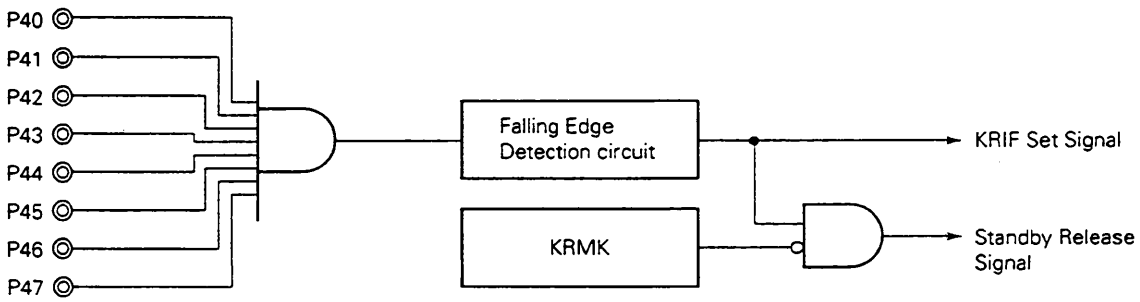
Fig. 4-8 P40 to P47 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) MM : Memory expansion mode register
(3) RD : Port 4 read signal
(4) WR : Port 4 write signal

Fig. 4-9 Block Diagram of Falling Edge Detection Circuit



4.2.6 Port 5

Port 5 is an 8-bit input/output port with output latch. P50 to P57 pins can specify the input mode/output mode in 1-bit units with the port mode register 5. When P50 to P57 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 L.

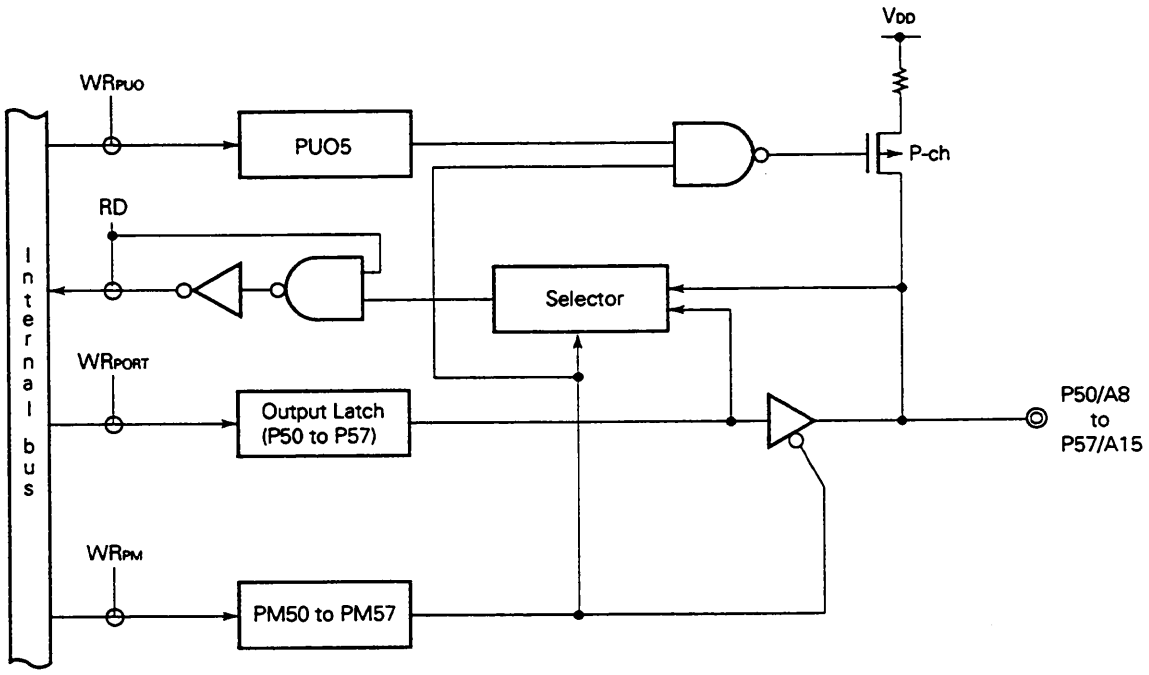
Port 5 can drive LEDs directly.

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

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

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

Fig. 4-10 P50 to P57 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 5 read signal
(4) WR : Port 5 write signal

4.2.7 Port 6

Port 6 is an 8-bit input/output port with output latch. P60 to P67 pins can specify the input mode/output mode in 1-bit units with the port mode register 6.

When pins P64 to P67 are used as input ports, a pull-up resistor can be connected to them in 4-bit units with pull-up resistor option register L. Mask ROM products can contain pull-up resistors to pins P60 to P63 by the mask option. The μPD78P098A has no mask option.

Pins P60 to P63 can drive LEDs directly.

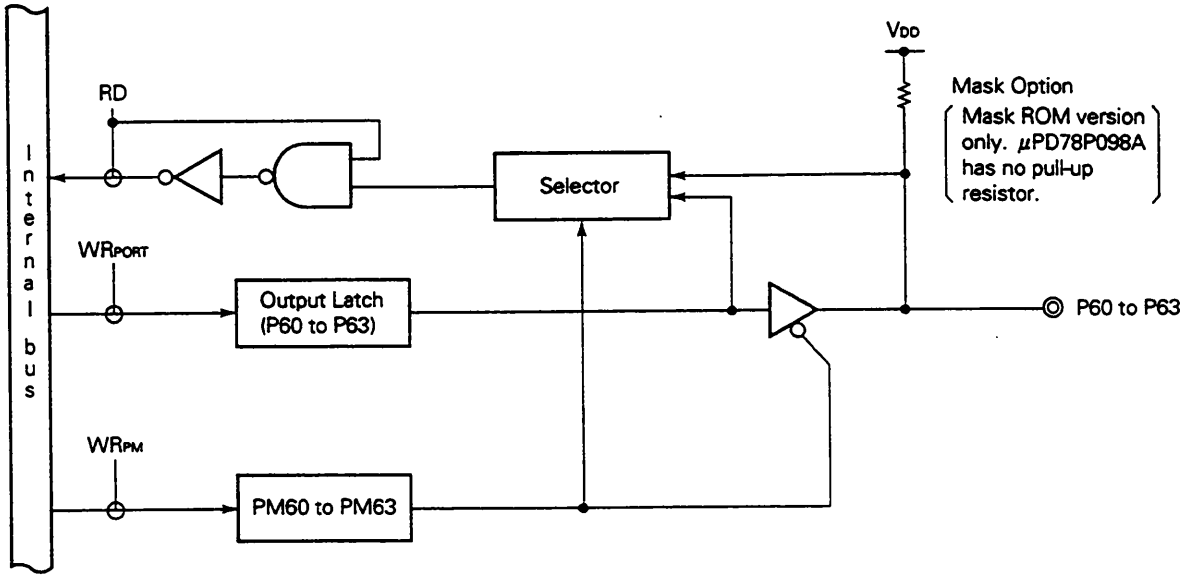
Dual-functions include the control signal output function in external memory expansion mode.

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

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

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

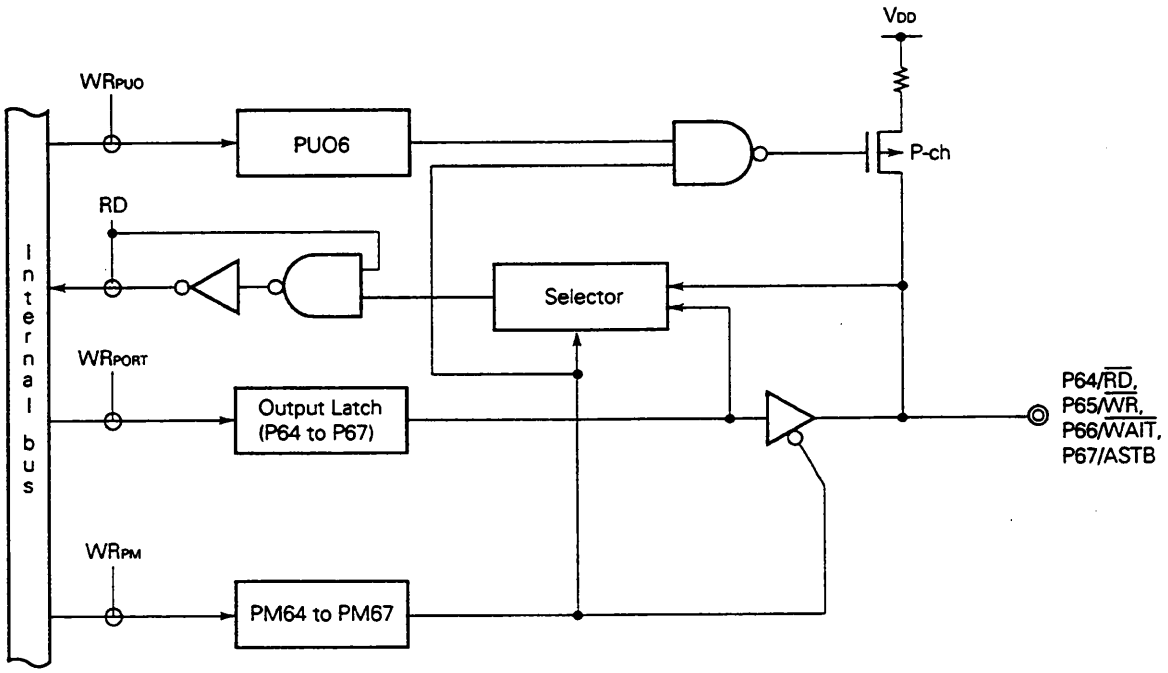
Fig. 4-11 P60 to P63 Configurations



Remarks:

(1) PM : Port mode register
(2) RD : Port 6 read signal
(3) WR : Port 6 write signal

Fig. 4-12 P64 to P67 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 6 read signal
(4) WR : Port 6 write signal

4.2.8 Port 7

This is a 3-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 7. When pins P70 to P72 are used as input port pins, a pull-up resistor can be connected as a 3-bit unit by means of pull-up resistor option register L.

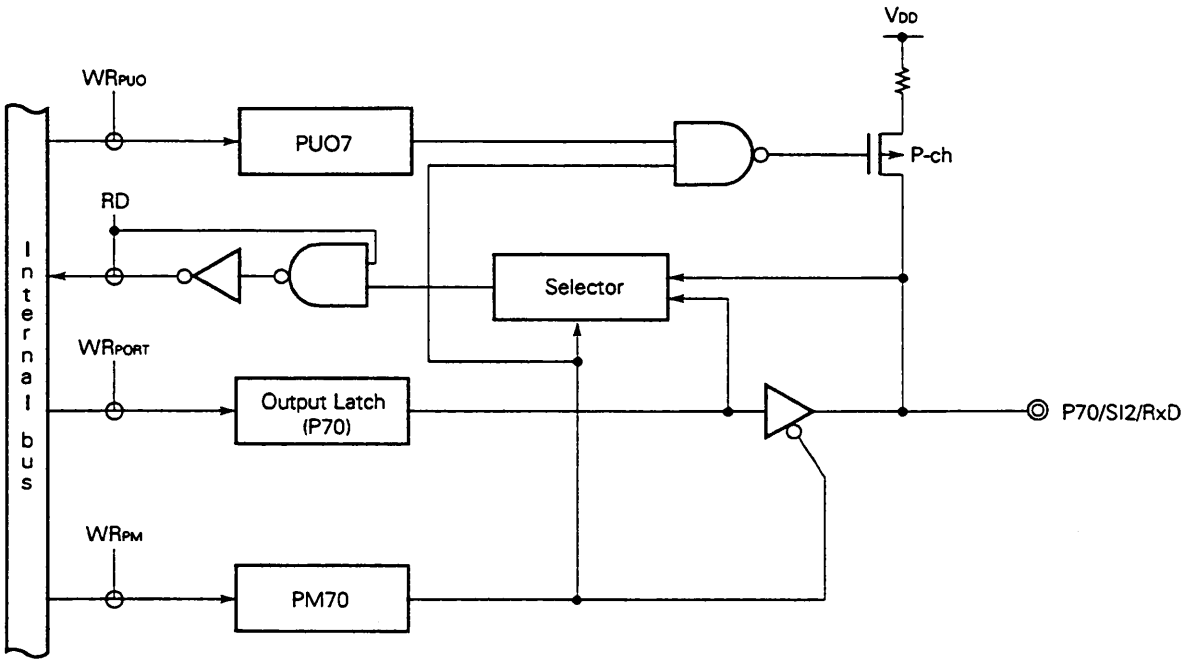
Dual-functions include serial interface channel 2 data input/output and clock input/output.

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

Port 7 block diagrams are shown in Figs. 4-13 and 4-14.

★ **Caution: When using as a serial interface, the input/output and output latch must be set according to the functions used. For details of how to set, refer to the list of serial interface channel 2 operation mode settings in Table 16-2.**

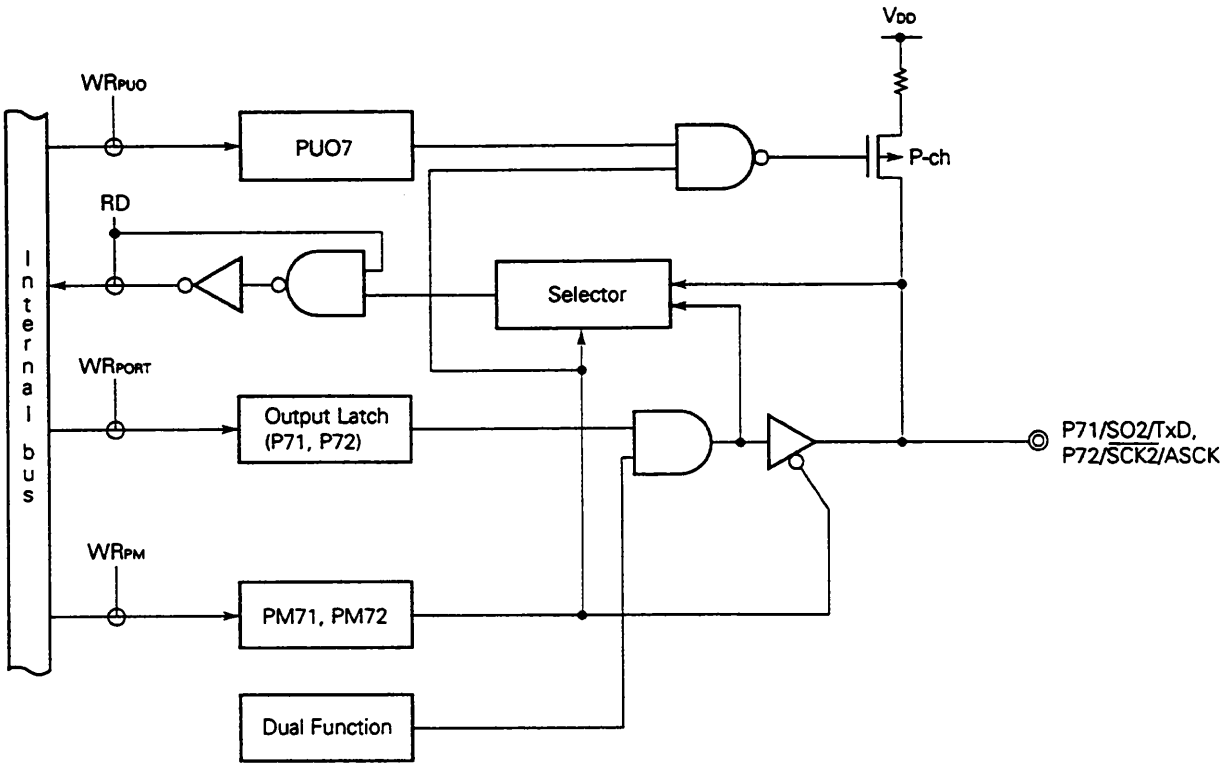
Fig. 4-13 P70 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 7 read signal
(4) WR : Port 7 write signal

Fig. 4-14 P71 and P72 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 7 read signal
(4) WR : Port 7 write signal

4.2.9 Port 12

This is an 8-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 12. When pins P120 to P127 are used as input port pins, a pull-up resistor can be connected as an 8-bit unit by means of pull-up resistor option register H.

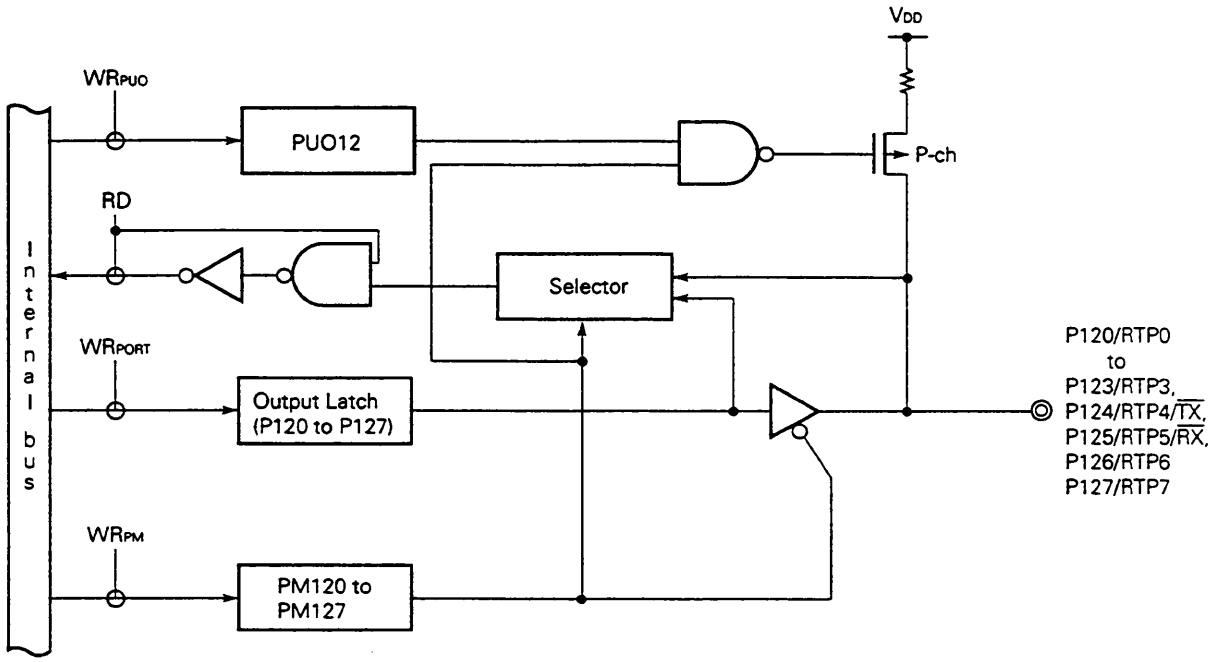
This port also provides real-time output function and IEBus controller function.

Do not execute a bit manipulation instruction on port 12 when the IEBus controller function is used.

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

The port 12 block diagram is shown in Fig. 4-15.

Fig. 4-15 P120 to P127 Configurations



Remarks:

(1) PUO : Pull-up resistor option register
(2) PM : Port mode register
(3) RD : Port 12 read signal
(4) WR : Port 12 write signal

4.2.10 Port 13

This is a 2-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 13. When pins P130 and P131 are used as input port pins, a pull-up resistor can be connected as a 2-bit unit by means of pull-up resistor option register H.

These pins are dual function pin and serve as D/A converter analog outputs.

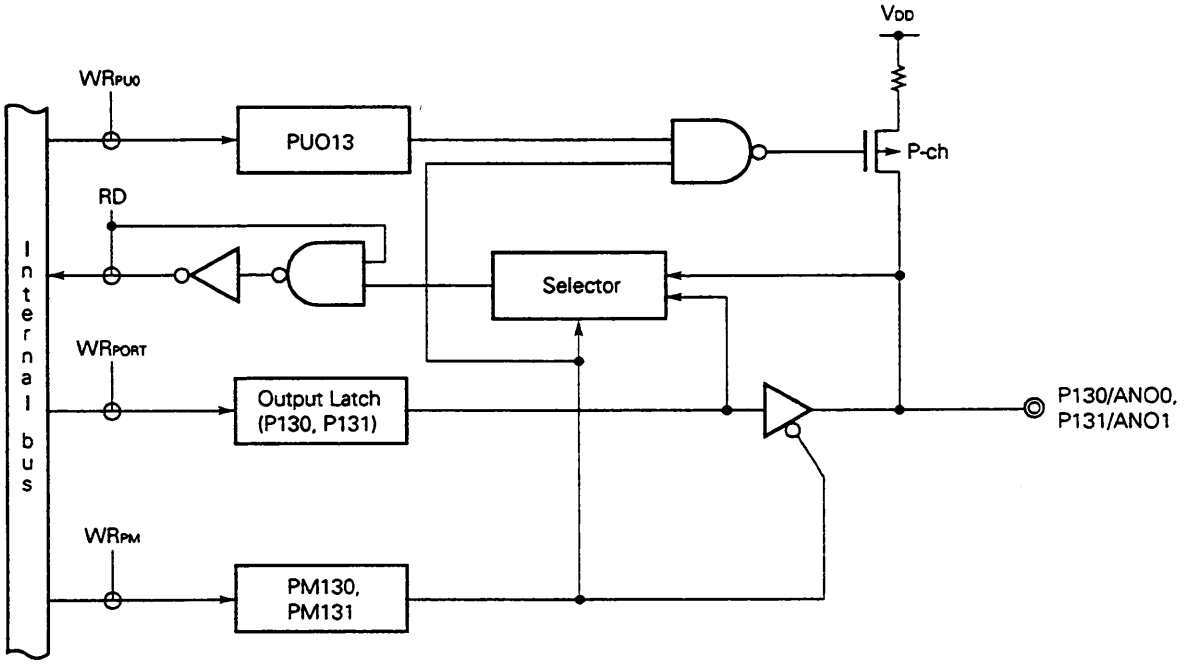
$\overline{\text{RESET}}$ input sets the input mode.

The port 13 block diagram is shown in Fig. 4-16.

Caution: When $AV_{REF} < V_{DD}$ and the D/A converter is used for only one channel, perform the one of the following procedures for pins not used as analog output. ★

- **Set the port mode register (PM13x) to 1 (input mode) and connect to Vss.**
- **Set the port mode register (PM13x) to 0 (output mode), set the output latch to 0, and output low level.**

Fig. 4-16 P130 and P131 Configurations



Remarks:

(1) PUO : Pull-up resistor option register

(2) PM : Port mode register

(3) RD : Port 13 read signal

(4) WR : Port 13 write signal

4.3 Port Function Control Registers

The following four types of registers control the ports.

- Port mode registers (PM0-PM3, PM5-PM7, PM12, PM13)
- Pull-up resistor option register (PUOH, PUOL)
- Memory expansion mode register (MM)
- Key return mode register (KRM)

★ **(1) Port mode registers (PM0 to PM3, PM5 to PM7, PM12, PM13)**

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

PM0 to PM3, PM5 to PM7, and PM13 are independently set with a 1-bit or 8-bit memory manipulation instruction.

PM12 is set with the 8-bit memory manipulation instruction.

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

When using port pins as dual function pins, set the port mode register and output latch as shown in Table 4-3.

Cautions:

(1) P00 and P07 are input-only pins.

(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) The memory expansion mode register specifies P40 to P47 as input/output pins.

(4) When the IEBus controller is used, the following read-modify-write instructions for port 12 cannot be used.

```
SET1    ; Manipulates any bits of port 12
CLR1    ; Manipulates any bits of port 12
BTCLR   ; Manipulates any bits of port 12
OR      P12, #byte
XOR     P12, #byte
AND     P12, #byte
```

Table 4-3 Port Mode Register and Output Latch Settings when Using Dual Function

Pin Name	Dual Function		PMxx	Pxx	Pin Name	Dual Function		PMxx	Pxx
	Name	I/O				Name	I/O		
P00	INTP0	I	1 (Fixed)	None	P50 to P57	A8 to A15	O	x[Note2]	
	TI00	I	1 (Fixed)	None	P64	$\overline{RD}$	O	x[Note2]	
P01	INTP1	I	1	x	P65	$\overline{WR}$	O	x[Note2]	
	TI01	I	1	x	P66	$\overline{WAIT}$	I	x[Note2]	
P02 to P06	INTP2 to INTP6	I	1	x	P67	ASTB	O	x[Note2]	
P07[Note1]	XT1	I	1 (Fixed)	None	P120 to P123	RTP0 to RTP3	O	0	Random
P10 to P17[Note1]	ANI0 to ANI7	I	1	x	P124	RTP4	O	0	Random
P30 to P32	TO0 to TO2	O	0	0		$\overline{TX}$	O	1	0
P33, P34	TI1, TI2	I	1	x	P125	RTP5	O	0	Random
P35	PCL	O	0	0		$\overline{RX}$	I	1	x
P36	BUZ	O	0	0	P126, P127	RTP6, RTP7	O	0	Random
P40 to P47	AD0 to AD7	I/O	x[Note2]		P130, P131[Note1]	ANO0, ANO1	O	1	0

Notes:

(1) Becomes unstable while the dual function is used.

(2) When using pins P40 to P47, P50 to P57, P64 to P67 as dual function pins, set the functions using the memory expansion mode register.

Remark: X : don't care (no setting required)
PMxx : Port mode register
Pxx : Port output latch

Cautions:

(1) When not using the external wait in the memory expansion mode, pin P66 can be used as an input/output port.

(2) When using port 2 and port 7 as serial interface pins, the input/output and output latch must be set according the functions used. For details of how to set, refer to the format of the serial operation mode register 0 in Fig. 14-4, the format of the serial operation mode register 1 in Fig. 15-3, and the list of serial interface channel 2 operation mode settings in Table 16-2.

Fig. 4-17 Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	1	FF20H	FFH	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	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	1	1	1	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W
PM13	1	1	1	1	1	1	PM131	PM130	FF2DH	FFH	R/W

PMmn	Pmn Pin Input/Output Mode Selection (m=0-3, 5-7, 12, 13 : n=0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(2) Pull-up resistor option register (PUOH, PUOL)

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

PUOH and PUOL are set with a 1-bit or 8-bit memory manipulation instruction.

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

Cautions:

(1) P00 and P07 pins do not incorporate a pull-up resistor.

(2) When ports 1, 4, 5, and P64 to P67 pins are used as dual function pins, a pull-up resistor cannot be used even if 1 is set in PUOm (m = 1, 4 to 6).

(3) P60 to P63 pins can be used with pull-up resistor by mask option only for mask ROM version.

Fig. 4-18 Pull-Up Resistor Option Register Format

Symbol	7	6	⑤	④	3	2	1	0	Address	When Reset	R/W
PUOH	0	0	PUO13	PUO12	0	0	0	0	FFF3H	00H	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
PUOL	PUO7	PUO6	PUO5	PUO4	PUO3	PUO2	PUO1	PUO0	FFF7H	00H	R/W

PUOm	Pm Internal Pull-up Resistor Selection (m=0-7, 12, 13)
0	Internal pull-up resistor not used
1	Internal pull-up resistor used

Caution: Bits 0 to 3, 6, and 7 of PUOH should be set to 0.

(3) Memory expansion mode register (MM)

This register is used to set input/output of port 4.

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

$\overline{\text{RESET}}$ input sets this register to 10H.

Fig. 4-19 Memory Expansion Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
MM	0	0	PW1	PW0	0	MM2	MM1	MM0	FFF8H	10H	R/W

MM2	MM1	MM0	Single-chip/ Memory Expansion Mode Selection		P40 to P47, P50 to P57, P64 to P67 Pin State				
					P40 to P47	P50 to P53	P54, P55	P56, P57	P64 to P67
0	0	0	Single-chip mode		Port mode: Input	Port mode			
0	0	1			Port mode: Output				
0	1	1	Memory expansion mode	256-byte mode	AD0 to AD7	Port mode			P64=$\overline{\text{RD}}$ P65=$\overline{\text{WR}}$ P66=$\overline{\text{WAIT}}$ P67=ASTB
1	0	0		4K-byte mode		A8 to A11	Port mode		
1	0	1		16K-byte mode			A12, A13	Port mode	
1	1	1		Full address mode[Note]				A14, A15	
Other than above			Setting prohibited						

PW1	PW0	Wait Control
0	0	No wait
0	1	Wait (one wait state insertion)
1	0	Setting prohibited
1	1	Wait control by external wait pin

Note: The full address mode allows external expansion for all areas of the 64K-byte address space, except the internal ROM, RAM, SFR, and use-prohibited areas.

Remarks:

(1) P60 to P63 pins enter the port mode in both the single-chip and memory expansion mode.

(2) Besides setting port 4 input/output, MM also sets the wait count and external expansion area.

Caution: When the 2/3 divider of the clock generator circuit (in divider 1) is used to generate the main system clock, the external device expansion function cannot be used.

(4) Key return mode register (KRM) ★

This register sets the enabling/disabling of standby function release by the key return signal (port 4 falling edge detection).

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

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

Fig. 4-20 Key Return Mode Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	When Reset	R/W
KRM	0	0	0	0	0	0	KRMK	KRIF	FFF6H	02H	R/W

KRIF	Key Return Signal Detection Flag
0	Not detected
1	Detected (Port 4 falling edge detection)

KRMK	Standby Mode Control by Key Return Signal
0	Standby mode release enabled
1	Standby mode release disabled

Caution: When falling edge detection of port4 is used, KRIF should be cleared to 0 (will not be cleared to 0 automatically).

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 mode

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

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

(2) Input mode

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

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

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

4.4.2 Reading from input/output port

(1) Output mode

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

(2) Input mode

The pin status is read by a transfer instruction.

The output latch contents do not change.

4.4.3 Operations on input/output port

(1) Output mode

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

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

(2) Input mode

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

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

4.5 Selection of Mask Option

The following mask option is provided in mask ROM version. The μPD78P098A has no mask option.

Table 4-4 Comparison between Mask ROM Version and the μPD78P098A

Pin Name	Mask ROM Version	μPD78P098A
Mask option for P60 to P63	Bitwise-selectable on-chip pull-up resistors	No on-chip pull-up resistor

[MEMO]

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 oscillator

This circuit oscillates at frequencies of 1 to 6.5 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register.

(2) Subsystem clock oscillator

The circuit oscillates at a frequency of 32.768 kHz. Oscillation cannot be stopped. If the subsystem clock oscillator is not used, the setting of not using the internal feedback resistance can be set by the processor clock control register. This enables to decrease power consumption in the STOP mode.

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) Oscillation mode selection register (OSMS) Clock select registers 1, 2 (IECL1, IECL2)
Oscillator	Main system clock oscillator Subsystem clock oscillator

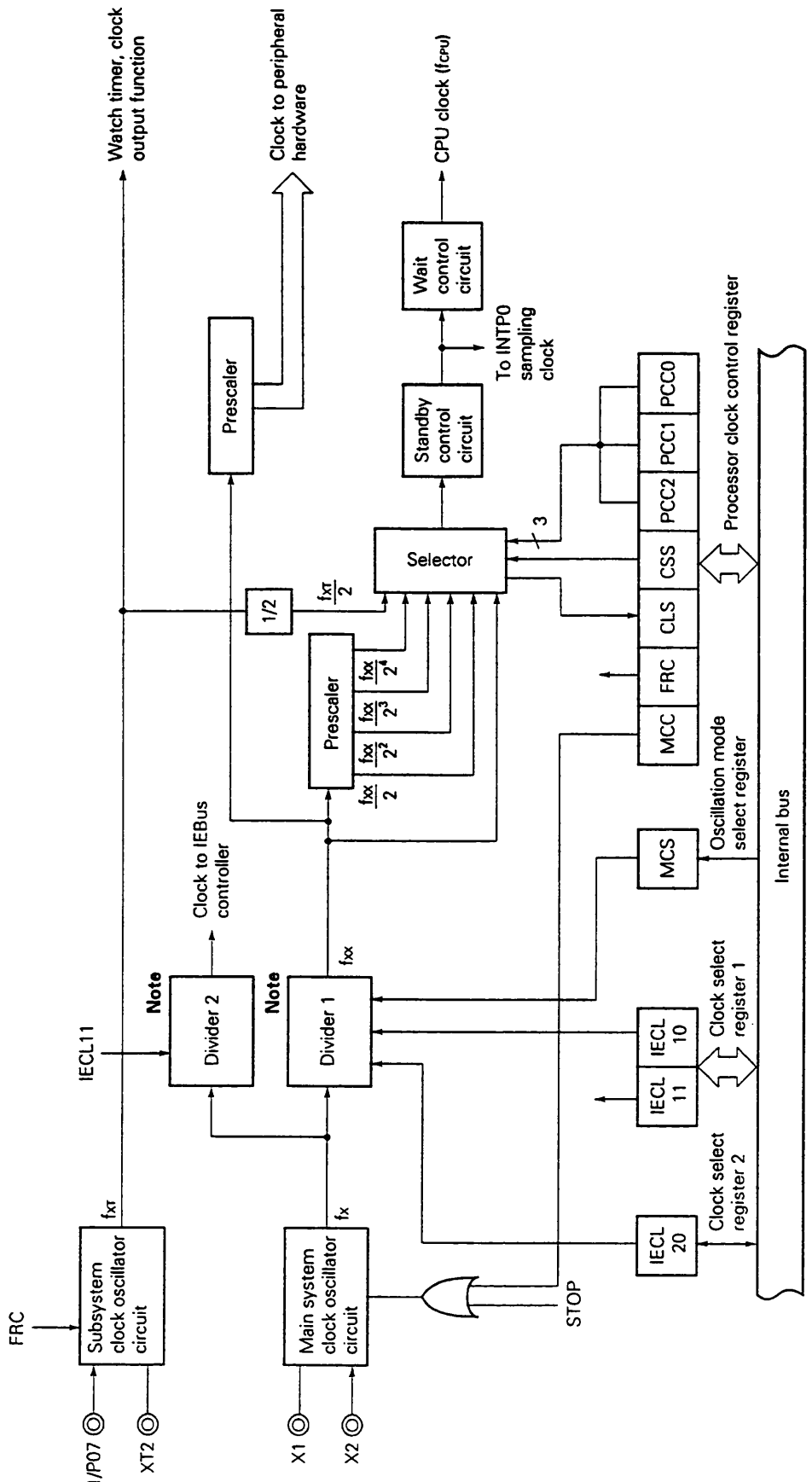


Fig. 5-1 Block Diagram of Clock Generator

Note: For the configuration of dividers 1 and 2, refer to Fig. 5-2.

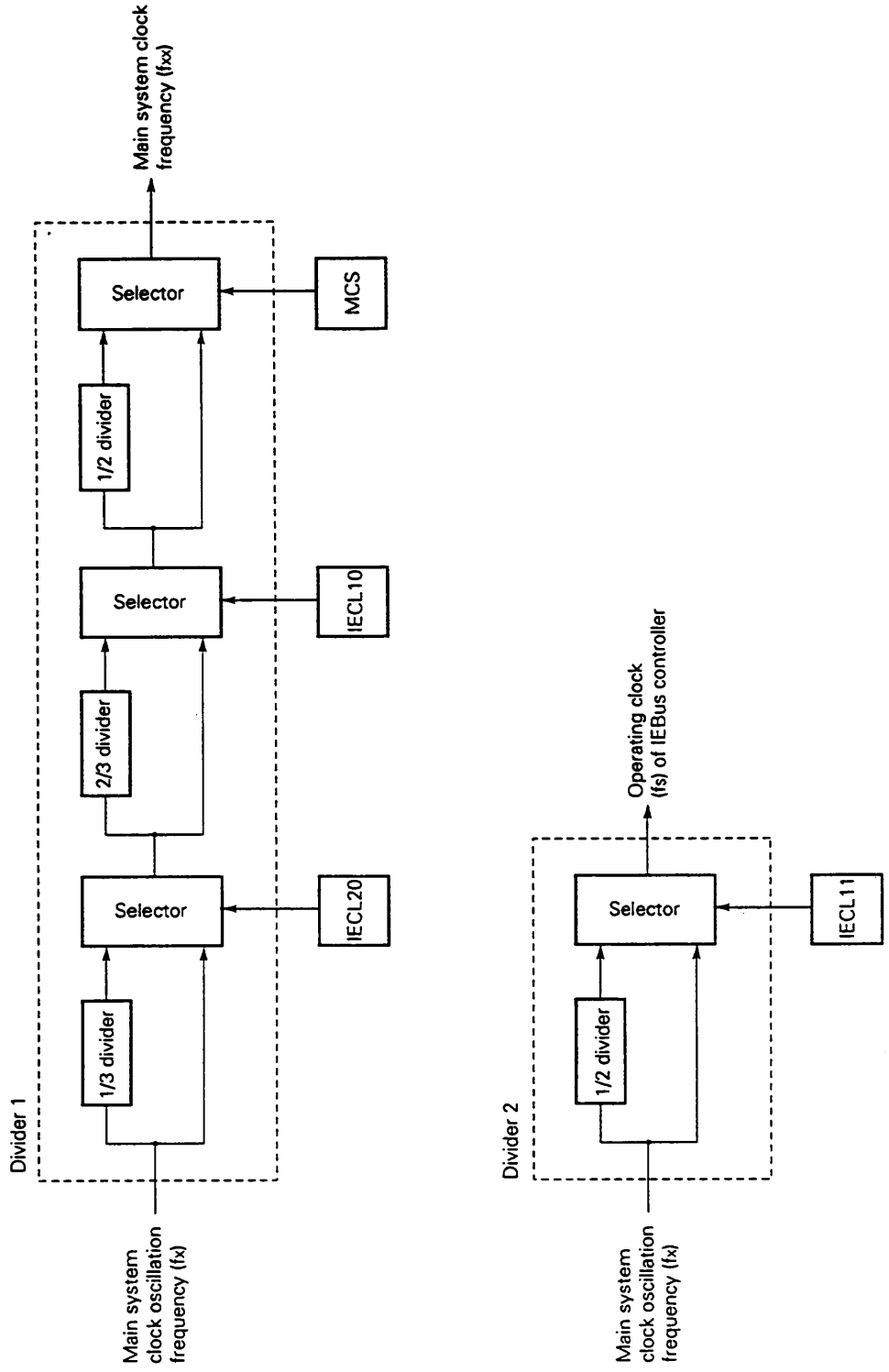


Fig. 5-2 Configuration of Dividers 1 and 2

5.3 Clock Generator Control Register

The clock generator is controlled by the following three registers:

- Processor clock control register (PCC)
- Oscillation mode selection register (OSMS)
- Clock select registers 1, 2 (IECL1, IECL2)

(1) Processor clock control register (PCC)

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

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

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

★ **Fig. 5-3 Subsystem Clock Feedback Resistor**

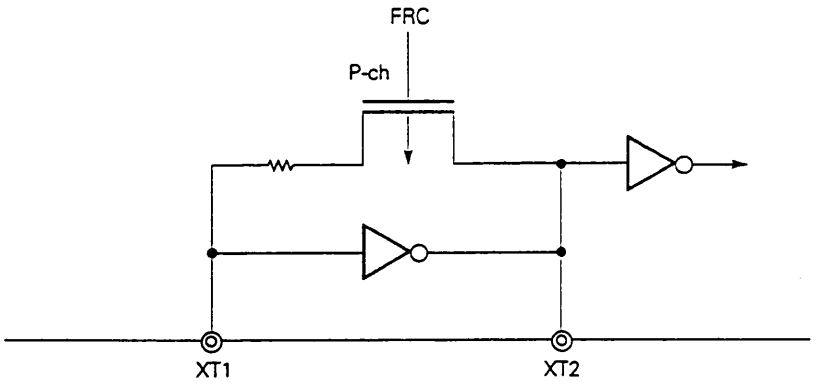


Fig. 5-4 Processor Clock Control Register Format

Symbol	⑦	⑥	⑤	④	3	2	1	0	Address	When Reset	R/W
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W Note1

R/W	CSS	PCC2	PCC1	PCC0	CPU Clock Selection (f_{CPU})
	0	0	0	0	f_{XX}
		0	0	1	$f_{XX}/2$
		0	1	0	$f_{XX}/2^2$
		0	1	1	$f_{XX}/2^3$
		1	0	0	$f_{XX}/2^4$
	1	0	0	0	$f_{XT}/2$ (122 μs)
		0	0	1	
		0	1	0	
		0	1	1	
		1	0	0	
	Other than above				Setting prohibited

R	CLS	CPU Clock Status
	0	Main system clock
	1	Subsystem clock

R/W	FRC	Subsystem Clock Feedback Resistor Selection
	0	Internal feedback resistor used
	1	Internal feedback resistor not used

R/W	MCC	Main System Clock Oscillation Control Note 2
	0	Oscillation possible
	1	Oscillation stopped

Notes:

(1) Bit 5 is Read Only.

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

Caution: Bit 3 must be set to 0.

Remarks:

(1) f_{XX} : Main system clock frequency

(2) f_{XT} : Subsystem clock oscillation frequency

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

(2) Oscillation mode selection register (OSMS)

This register specifies whether 1/2 divider in divider 1 is used. When this register is used in combination with clock select registers 1 and 2 (IECL1 and IECL2), eight types of main system clock frequencies fxx are selectable.

OSMS is set with 8-bit memory manipulation instruction.

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

Fig. 5-5 Oscillation Mode Selection Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
OSMS	0	0	0	0	0	0	0	MCS	FFF2H	00H	W

MCS	Main System Clock Divider Control (divider 1)
0	Uses 1/2 divider in divider 1
1	Does not use 1/2 divider in divider 1

★ **Cautions:**

(1) When the oscillation mode is switched, the clock supplied to the peripheral hardware (fx: main system clock) will also be switched.

(2) Be sure to clear bits 1-7 to 0.

(3) Clock select registers 1 and 2 (IECL1 and IECL2)

These registers specifies whether the 1/2 divider and 2/3 divider in divider 1 are used and whether the 1/2 divider in divider 2 is used.

When these registers are used in combination with the oscillation mode select register (OSMS), eight types of main system clock frequencies can be selected.

IECL1 and IECL2 are set with 8-bit memory manipulation instructions.

$\overline{\text{RESET}}$ input sets IECL1 and IECL2 to 00H.

Fig. 5-6 Format of Clock Select Register 1

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
IECL1	0	0	0	0	0	0	IECL11	IECL10	F8E0H	00H	R/W

IECL10	Controls divider of main system clock (divider 1)
0	Does not use 2/3 divider in divider 1
1	Uses 2/3 divider in divider 1

IECL11	Controls divider of main system clock (divider 2)
0	Uses 1/2 divider in divider 2
1	Does not use 1/2 divider in divider 2

Caution: Be sure to clear bits 2-7 to 0.

Fig. 5-7 Format of Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
IECL2	0	0	0	0	0	0	0	IECL20	F8E1H	00H	R/W

IECL20	Controls divider of main system clock (divider 1)
0	Does not use 1/3 divider in divider 1
1	Uses 1/3 divider in divider 1

Caution: Be sure to clear bits 1-7 to 0.

Table 5-2 Relationships between Setting of Divider 1 and Main System Clock Frequency

MCS	IECL20	IECL10	Main system clock frequency (fxx)
0	0	0	fx/2
	0	1	fx/3
	1	0	fx/6
	1	1	fx/9
1	0	0	fx
	0	1	2fx/3
	1	0	fx/3 (2fx/6)
	1	1	2fx/9

Table 5-3 CPU Clock (f_{CPU}) List

CSS	PCC2	PCC1	PCC0	CPU clock selection (f_{CPU})								
				MCS	0	0	0	0	1	1	1	1
				IECL20	0	0	1	1	0	0	1	1
				IECL10	0	1	0	1	0	1	0	1
0	0	0	0	f_{XX}	$f_X/2$ (0.67 μs)	$(2f_X/3)/2$ (1.00 μs)	$(f_X/3)/2$ (2.00 μs)	$(2f_X/9)/2$ (3.00 μs)	f_X (setting prohibited)	$2f_X/3$ (0.50 μs)	$f_X/3$ (1.00 μs)	$2f_X/9$ (1.50 μs)
	0	0	1	$f_{XX}/2$	$f_X/2^2$ (1.33 μs)	$(2f_X/3)/2^2$ (2.00 μs)	$(f_X/3)/2^2$ (4.00 μs)	$(2f_X/9)/2^2$ (6.00 μs)	$f_X/2$ (0.67 μs)	$(2f_X/3)/2$ (1.00 μs)	$(f_X/3)/2$ (2.00 μs)	$(2f_X/9)/2$ (3.00 μs)
	0	1	0	$f_{XX}/2^2$	$f_X/2^3$ (2.67 μs)	$(2f_X/3)/2^3$ (4.00 μs)	$(f_X/3)/2^3$ (8.00 μs)	$(2f_X/9)/2^3$ (12.0 μs)	$f_X/2^2$ (1.33 μs)	$(2f_X/3)/2^2$ (2.00 μs)	$(f_X/3)/2^2$ (4.00 μs)	$(2f_X/9)/2^2$ (6.00 μs)
	0	1	1	$f_{XX}/2^3$	$f_X/2^4$ (5.33 μs)	$(2f_X/3)/2^4$ (8.00 μs)	$(f_X/3)/2^4$ (16.0 μs)	$(2f_X/9)/2^4$ (24.0 μs)	$f_X/2^3$ (2.67 μs)	$(2f_X/3)/2^3$ (4.00 μs)	$(f_X/3)/2^3$ (8.00 μs)	$(2f_X/9)/2^3$ (12.0 μs)
	1	0	0	$f_{XX}/2^4$	$f_X/2^5$ (10.7 μs)	$(2f_X/3)/2^5$ (16.0 μs)	$(f_X/3)/2^5$ (32.0 μs)	$(2f_X/9)/2^5$ (48.0 μs)	$f_X/2^4$ (5.33 μs)	$(2f_X/3)/2^4$ (8.00 μs)	$(f_X/3)/2^4$ (16.0 μs)	$(2f_X/9)/2^4$ (24.0 μs)
1	0	0	0	$f_{XT}/2$ (122 μs)								
	0	0	1									
	0	1	0									
	0	1	1									
	1	0	0									
Other than above				Setting prohibited								

Remarks:

(1) f_X : main system clock oscillation frequency

(2) f_{XX} : main system clock frequency

(3) f_{XT} : subsystem clock oscillation frequency

(4) Figures in parentheses indicate the minimum instruction execution time when operating at f_X = 6.0 MHz or f_{XT} = 32.768 kHz: $2/f_{CPU}$.

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

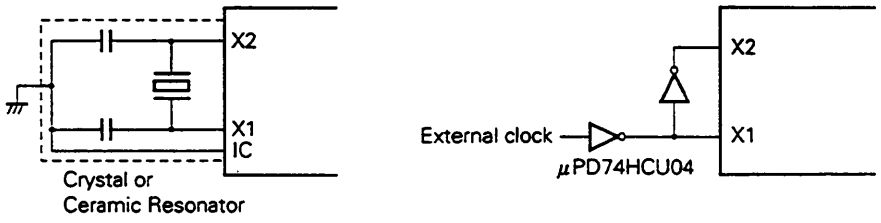
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 6.0 MHz) connected to the X1 and X2 pins.

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

Fig. 5-8 shows an external circuit of the main system clock oscillator.

Fig. 5-8 External Circuit of Main System Clock Oscillator

(a) Crystal or ceramic oscillation **(b) External clock**



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

5.4.2 Subsystem clock oscillator

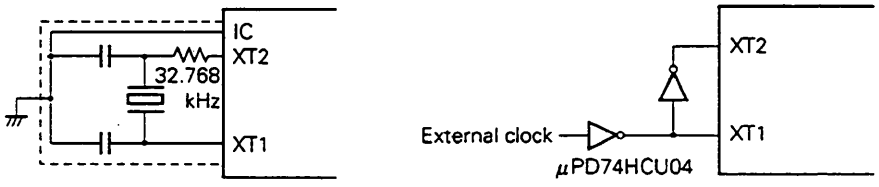
The subsystem clock oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

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

Fig. 5-9 shows an external circuit of the subsystem clock oscillator.

Fig. 5-9 External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation **(b) External clock**



Cautions:

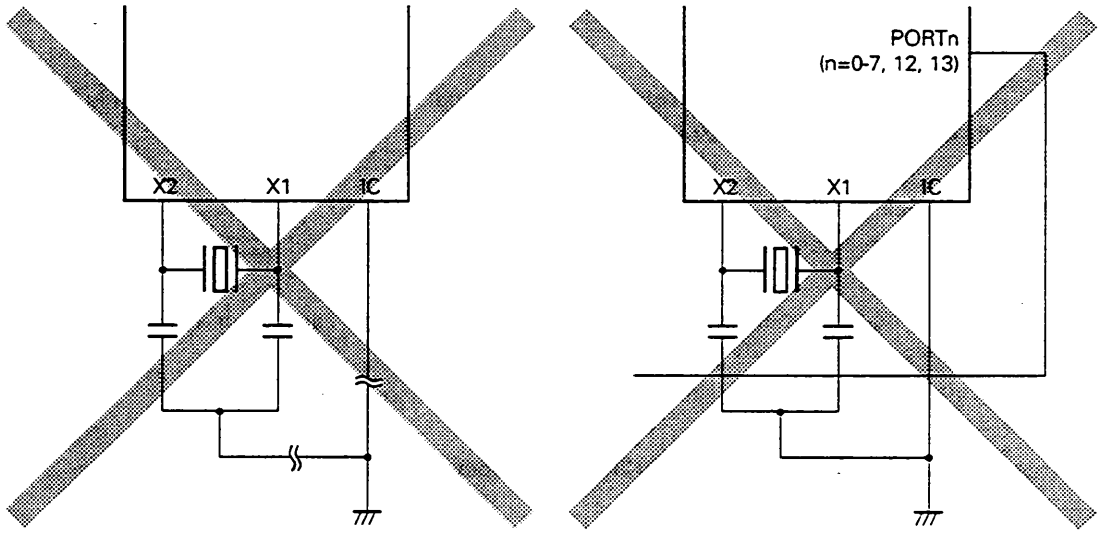
(1) When using a main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken line area in Figs. 5-8 and 5-9 as follows to prevent any effects from wiring capacities.

- **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. 5-10 shows examples of oscillator having bad connection.

Fig. 5-10 Examples of Oscillator with Bad Connection (1/2)

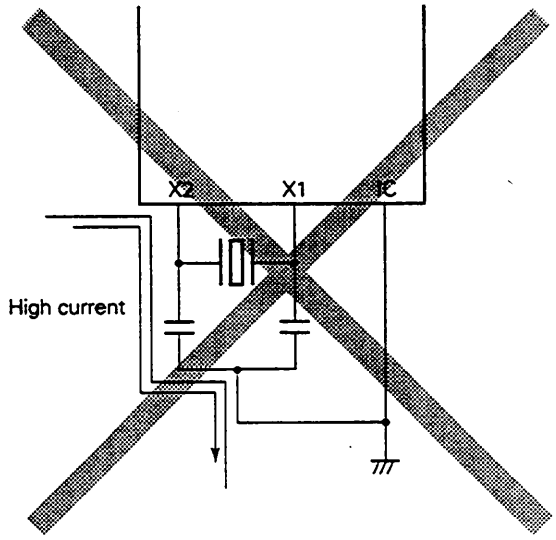
(a) Too long wiring of connected circuit **(b) Crossed signal lines**



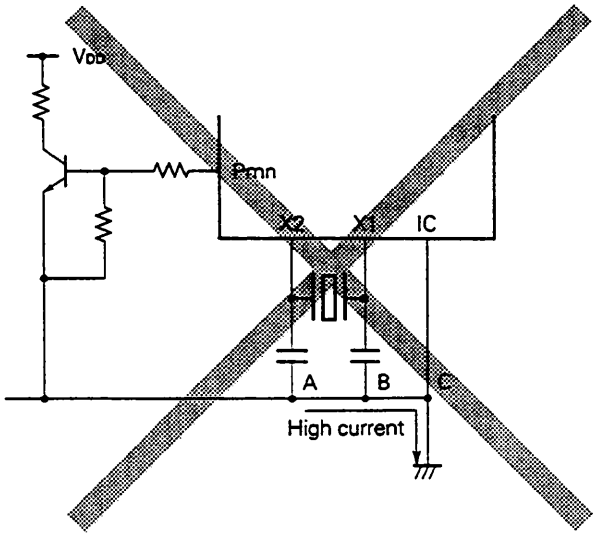
Remark: When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Fig. 5-10 Examples of Oscillator with Bad Connection (2/2)

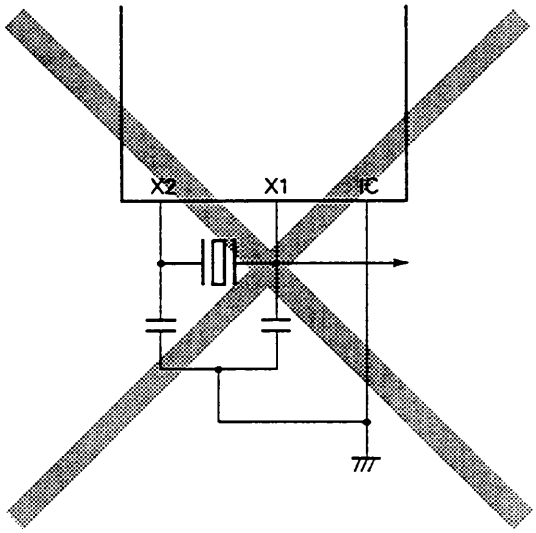
(c) High alternating current close to signal lines



(d) Current flowing through ground line of oscillator circuit (potentials at points A, B, and C change)



(e) Signal extracted



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.

Cautions

(2) If XT2 and X1 are wired in parallel, malfunctioning may occur because the cross-talk noise of X1 is superimposed on XT2.

To prevent this, do not wire XT2 and X1 in parallel, and directly connect the IC pin between XT2 and X1 to Vss.

5.4.3 Divider

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

5.4.4 When no subsystem clocks are used

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

★ XT1: Connect to Vss
XT2: Open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To prevent that from happening, the above internal feedback resistance 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 fxx
- Subsystem clock fxT
- CPU clock fCPU
- Clock to peripheral hardware

The funciton and operation of the clock generator circuit are determined by the processor clock control register (PCC), oscillation mode select register (OSMS), and clock select registers (IECL1 and IECL2) as follows:

(a) Upon generation of $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (10.7 μs when operated at 6.0 MHz) is selected (PCC = 04H, OSMS = 00H, IECL1 = 00H, IECL2 = 00H). Main system clock oscillation stops while low level is applied to $\overline{\text{RESET}}$ pin.

(b) With the main system clock selected, one of the forty (19 types) CPU clock stages can be selected by setting the PCC.

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

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

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

(f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to 16-bit timer/event counter, the watch timer, and clock output functions only. Thus, 16-bit timer/event counter (when selecting watch timer output for count clock operating with subsystem clock), the watch function, and the clock output function can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped. (Except external input clock operation)

5.5.1 Main system clock operations

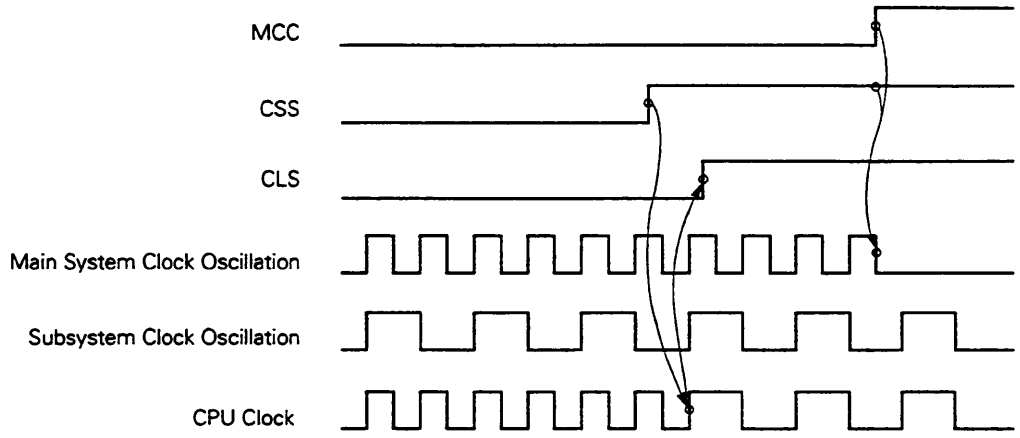
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

(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. 5-11**).

Fig. 5-11 Main System Clock Stop Function (1/2)

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



(b) Operation when MCC is set with main system clock operation

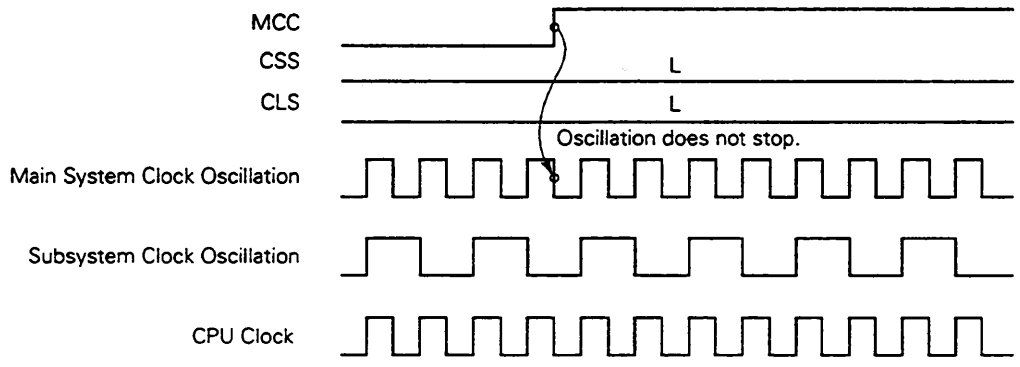
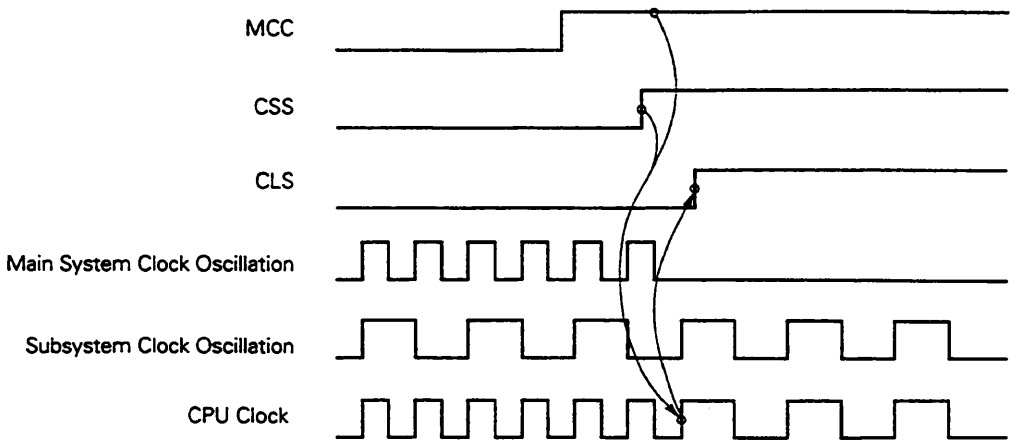


Fig. 5-11 Main System Clock Stop Function (2/2)

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



5.5.2 Subsystem clock operations

When operated with the 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 the subsystem clock is in operation.

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for swtichover between system clock and CPU clock

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

The actual switchover operation is not performed directly after writing to the PCC, but operation continues on the pre-switchover clock for several instructions (see **Table 5-4**).

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 5-4 Maximum Time Required for CPU Clock Switchover

Set Values before Switcher				Set Values after Switcher																							
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				$f_{XX}/2f_{XT}$ instructions (62 instructions)			
	0	0	1	8 instructions								8 instructions				8 instructions				8 instructions				$f_{XX}/4f_{XT}$ instructions (31 instructions)			
	0	1	0	4 instructions				4 instructions								4 instructions				4 instructions				$f_{XX}/8f_{XT}$ instructions (16 instructions)			
	0	1	1	2 instructions				2 instructions				2 instructions								2 instructions				$f_{XX}/16f_{XT}$ instructions (8 instructions)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								$f_{XX}/32f_{XT}$ instructions (4 instructions)			
1	×	×	×	1 instruction				1 instruction				1 instruction				1 instruction				1 instruction							

Remarks:

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

(2) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz and f_{XT} = 32.768 kHz.

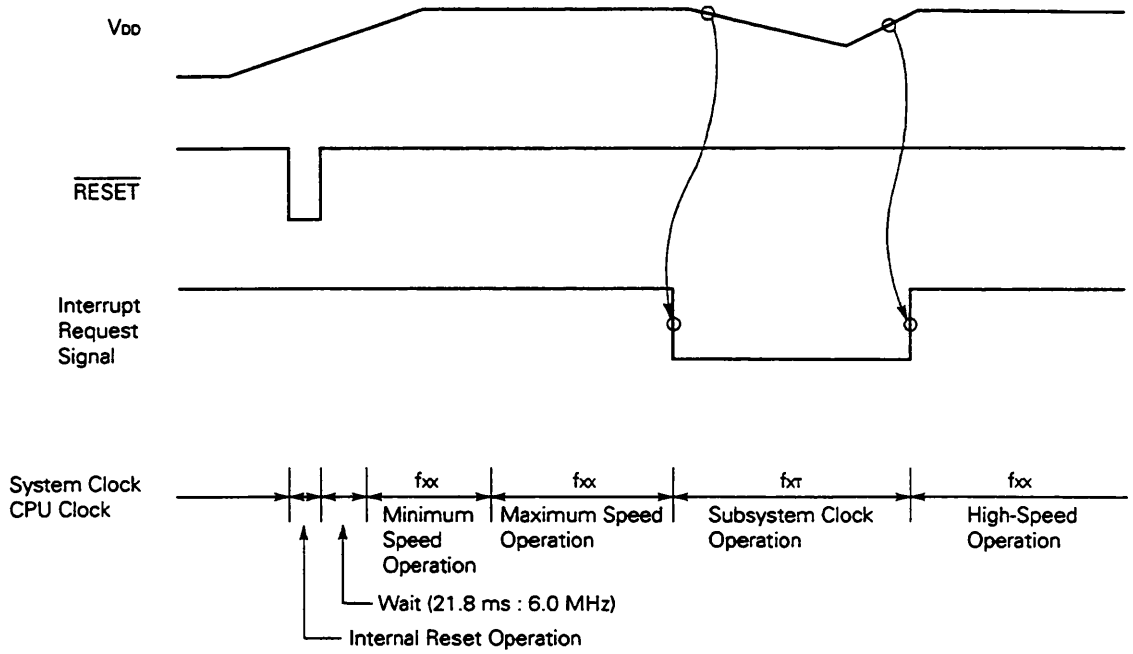
Caution: Selection of the CPU clock cycle scaling factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be performed simultaneously.

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

5.6.2 System clock and CPU clock switching procedure

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

Fig. 5-12 System Clock and CPU Clock Switching



① The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, main system clock starts oscillation. At this time, oscillation stabilization time (2^{17}/fx) is secured automatically.
After that, the CPU starts executing the instruction at the minimum speed of the main system clock (10.7 μs when operated at 6.0 MHz).

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

Caution: When stopping the main system clock, operating using the subsystem clock, and switching the main system clock again, switch after ensuring the oscillation stabilization time with the program. ★

[MEMO]

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

The timers incorporated into the μPD78098 sub-series are outlined below.

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

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

(2) 8-bit 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 perform 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	2 channels[Note 3]	2 channels	1 channel[Note 1]	1 channel[Note 2]
	External event counter	O	O	-	-
Function	Timer output	O	O	-	-
	PWM output	O	-	-	-
	Pulse width measurement	O	-	-	-
	Square-wave ouptut	O	O	-	-
	One-shot pulse output	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.

(3) When capture/compare registers (CR00, CR01) are specified as compare registers.

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
- One-shot pulse output

(1) Interval timer

TM0 generates interrupts at the preset time interval.

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

Minimum interval time	Maximum interval time	Resolution
2 × TI00 input cycle	2^{16} × TI00 input cycle	TI00 input edge cycle
1/fxx (250 ns)	2^{15} × 1/fxx (8.19 ms)	1/2fxx (125 ns)
2 × 1/fxx (500 ns)	2^{16} × 1/fxx (16.4 ms)	1/fxx (250 ns)
2^2 × 1/fxx (1.0 μs)	2^{17} × 1/fxx (32.8 ms)	2 × 1/fxx (500 ns)
2^3 × 1/fxx (2.0 μs)	2^{18} × 1/fxx (65.5 ms)	2^2 × 1/fxx (1.0 μs)
2 × watch timer output cycle	2^{16} × watch timer output cycle	Watch timer output edge cycle

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 MHz

(2) PWM output

TM0 can generate 14-bit resolution PWM output.

(3) Pulse width measurement

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

(4) External event counter

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

(5) Square-wave output

TM0 can output a square wave with any selected frequency.

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

Minimum pulse time	Maximum pulse time	Resolution
2 x TI00 input cycle	2^{16} x TI00 input cycle	TI00 input edge cycle
1/fxx (250 ns)	2^{15} x 1/fxx (8.19 ms)	1/2fxx (125 ns)
2 x 1/fxx (500 ns)	2^{16} x 1/fxx (16.4 ms)	1/fxx (250 ns)
2^2 x 1/fxx (1.0 μs)	2^{17} x 1/fxx (32.8 ms)	2 x 1/fxx (500 ns)
2^3 x 1/fxx (2.0 μs)	2^{18} x 1/fxx (65.5 ms)	2^2 x 1/fxx (1.0 μs)
2 x watch timer output cycle	2^{16} x watch timer output cycle	Watch timer output edge cycle

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 MHz

(6) One-shot pulse output

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

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	Capture/compare register: 16 bits x 2 (CR00, CR01)
Timer output	1 (TO0)
Control register	Timer clock select register 0 (TCL0) 16-bit timer mode control register (TMC0) Capture/compare control register 0 (CRC0) 16-bit timer output control register (TOC0) Port mode register 3 (PM3) External interrupt mode register 0 (INTM0) Sampling clock select register (SCS)

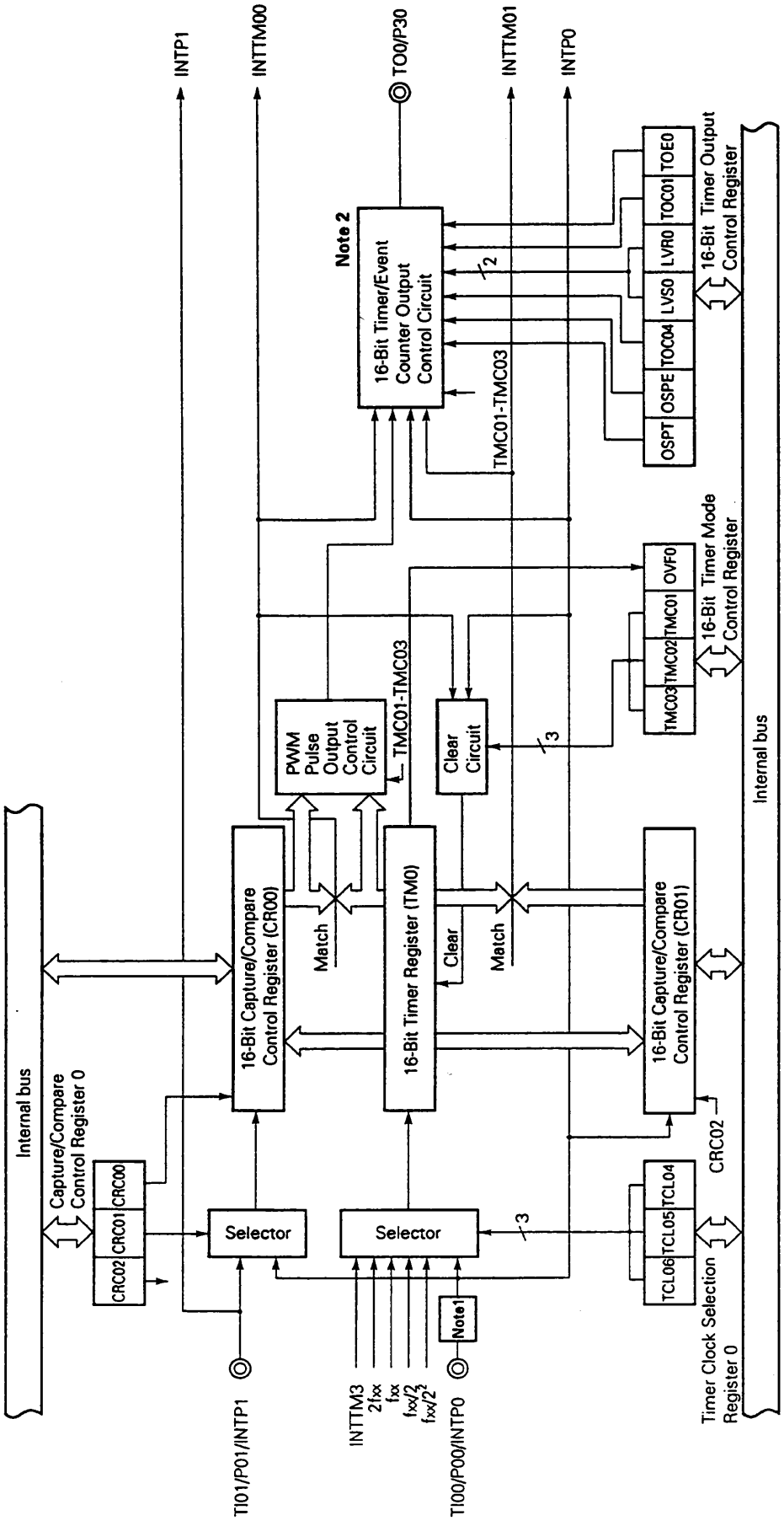


Fig. 6-1 16-Bit Timer/Event Counter Block Diagram

Notes:

(1) Edge detection circuit

(2) The configuration of the 16-bit timer/event counter output control circuit is shown in Fig. 6-2.

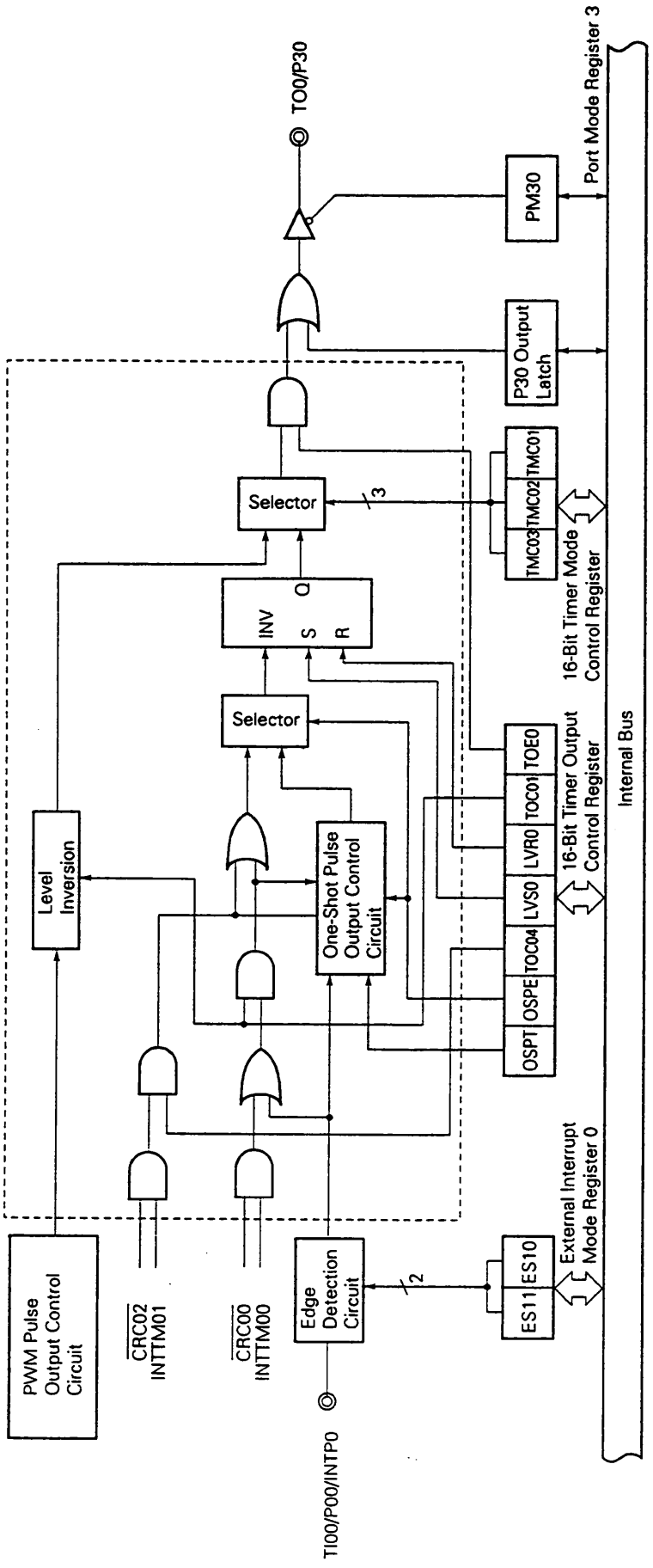


Fig. 6-2 16-Bit Timer/Event Counter Output Control Circuit Block Diagram

Remark: The circuitry enclosed by the dotted line is the output control circuit.

(1) Capture/compare register (CR00)

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

When CR00 is used as a compare register, the value set in the CR00 is constantly compared with the 16-bit timer register (TM0) count value, and an interrupt request (INTTM00) is generated if they match. It can also be used as the register which holds the interval time when TM0 is set to interval timer operation, and as the register which sets the pulse width in the PWM operating mode.

When CR00 is used as a capture register, it is possible to select the valid edge of the INTP0/TI00 pin or the INTP1/TI01 pin as the capture trigger. Setting of the INTP0/TI00 or INTP1/TI01 valid edge is performed by means of external interrupt mode register 0.

If CR00 is specified as the capture register and if the capture trigger is specified as the valid edge of the INTP0/TI00 pin, the situation is as shown in the following table:

Table 6-5 INTP0/TI00 Pin Valid Edge and CR00 Capture Trigger Valid Edge

ES11	ES10	INTP0/TI00 Pin Valid Edge	CR00 Capture Trigger Valid Edge
0	0	Falling edge	Rising edge
0	1	Rising edge	Falling edge
1	0	Setting prohibited	
1	1	Both rising and falling edges	No capture operation

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

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

(2) Capture/compare register (CR01)

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

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

When CR01 is used as a capture register, it is possible to select the valid edge of the INTP0/TI00 pin as the capture trigger. Setting of the INTP0/TI00 valid edge is performed by means of external interrupt mode register 0.

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

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

(3) 16-bit timer register (TM0)

TM0 is a 16-bit register which counts the count pulses.

TM0 is read by a 16-bit memory manipulation instruction. When TM0 is read, capture/compare register (CR01) should first be set as a capture register.

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

Caution: As reading of the value of TM0 is performed via CR01, the previously set value of CR01 is lost.

6.3 16-Bit Timer/Event Counter Control Registers

The following seven 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)
- Capture/compare control register 0 (CRC0)
- 16-bit timer output control register (TOC0)
- Port mode register 3 (PM3)
- External interrupt mode register 0 (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 manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL0 value 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. 6-3 Timer Clock Selection Register 0 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	When 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	0	1	f_{XX} (4.0 MHz)
0	1	1	0	$f_{XX}/2$ (2.0 MHz)
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
Other than above				Setting prohibited

TCL06	TCL05	TCL04	16-Bit Timer Register Count Clock Selection
0	0	0	TI00 (Valid edge specifiable)
0	0	1	$2f_{XX}$ **Note**
0	1	0	f_{XX} (4.0 MHz)
0	1	1	$f_{XX}/2$ (2.0 MHz)
1	0	0	$f_{XX}/2^2$ (1.0 MHz)
1	1	1	Watch timer output (INTTM3)
Other than above			Setting prohibited

CLOE	PCL Output Control
0	Output disabled
1	Output enabled

Note: Setting prohibited when operated at f_{XX} > 2.5 MHz.

Cautions:

(1) Setting of the TI00/INTP0 pin valid edge is performed by external interrupt mode register 0, and selection of the sampling clock frequency is performed by the sampling clock selection register.

(2) When enabling PCL output, set TCL00 to TCL03, then set 1 in CLOE with a 1-bit memory manipulation instruction.

(3) To read the count value when TI00 has been specified as the TM0 count clock, the value should be read from TM0, not from capture/compare register CR01.

(4) When rewriting TCL0 to data other than the same data, stop the timer operation first. ★

Remarks:

(1) f_{XX} : Main system clock frequency

(2) f_{XT} : Subsystem clock oscillation frequency

(3) TI00 : 16-bit timer/event counter input pin

(4) TM0 : 16-bit timer register

(5) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz or f_{XT} = 32.768 kHz.

(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 value to 00H.

★ **Caution: The 16-bit timer register starts operating at the point where values other than 0, 0, 0 (operation stop mode) are set to TMC01 to TMC03. To stop operations, set 0, 0, 0 to TMC01 to TMC03.**

Fig. 6-4 16-Bit Timer Mode Control Register Format

Symbol	7	6	5	4	3	2	1	⓪	Address	When 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	Operating Mode & Clear Mode Selection	TO0 Output Timing Selection	Interrupt Generation
0	0	0	Operation stop (TM0 cleared to 0)	No change	Not generated
0	0	1	PWM mode (free running)	PWM pulse output	Generated on match between TM0 and CR00, and match between TM0 and CR01
0	1	0	Free running mode	Match between TM0 and CR00 or match between TM0 and CR01	
0	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	
1	0	0	Clear & start on TI00 valid edge	Match between TM0 and CR00 or match between TM0 and CR01	
1	0	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	
1	1	0	Clear & start on match between TM0 and CR00	Match between TM0 and CR00 or match between TM0 and CR01	
1	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	

Cautions:

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

(2) Set the valid edge of the TI00/INTP0 pin with an external interrupt mode register 0 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.

(4) If clear & start mode on match between TM0 and CR00 is selected, the set value of CR00 is 0FFFFH, and when TM0 value changes from 0FFFFH to 0000H, OVF0 flag is set to 1.

Remarks:

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

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

This register controls the operation of the capture/compare registers (CR00, CR01).
CRC0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CRC0 value to 04H.

Fig. 6-5 Capture/Compare Control Register 0 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
CRC0	0	0	0	0	0	CRC02	CRC01	CRC00	FF4CH	04H	R/W

CRC00	CR00 Operating Mode Selection
0	Operates as compare register
1	Operates as capture register

CRC01	CR00 Capture Trigger Selection
0	Captures on valid edge of TI01
1	Captures on valid edge of TI00

CRC02	CR01 Operating Mode Selection
0	Operates as compare register
1	Operates as capture register

Cautions:

(1) Timer operation must be stopped before setting CRC0.
(2) When clear & start mode on a match between TM0 and CR00 is selected with the 16-bit timer mode control register, CR00 should not be specified as a capture register.

(4) 16-bit timer output control register (TOC0)

This register controls the operation of the 16-bit timer/event counter output control circuit. It sets R-S type flip-flop (LV0) setting/resetting, the active level in PWM mode, inversion enabling/disabling in modes other than PWM mode, 16-bit timer/event counter timer output enabling/disabling, one-shot pulse output operation enabling/disabling, and output trigger for a one-shot pulse by software.

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

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

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

Symbol	7	⑥	⑤	4	③	②	1	⓪	Address	When Reset	R/W
TOC0	0	OSPT	OSPE	TOC04	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	In PWM Mode	In Other Modes
	Active level selection	Timer output F/F control by match of CR00 and TM0
0	Active high	Inversion operation disabled
1	Active low	Inversion operation enabled

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

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

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

OSPT	Control of One-Shot Pulse Output Trigger by Software
0	No one-shot pulse trigger
1	One-shot pulse trigger used

Cautions:

(1) Timer operation must be stopped before setting TOC0.

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

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

(5) 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 output latch of P30 to 0.

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

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

Fig. 6-7 Port Mode Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(6) External interrupt mode register 0 (INTM0)

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

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

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

Fig. 6-8 External Interrupt Mode Register 0 Format

Symbol	7	6	5	4	3	2	1	0	Address	When 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 falling and rising edges

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

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

(7) Sampling clock select registers (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 manipulation instruction.

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

Fig. 6-9 Sampling Clock Select Register Format

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

SCS1	SCS0	INTP0 Sampling Clock Selection
0	0	$f_{XX}/2^N$
0	1	$f_{XX}/2^7$ (31.3kHz)
1	0	$f_{XX}/2^5$ (125.0kHz)
1	1	$f_{XX}/2^6$ (62.5kHz)

Caution: $f_{XX}/2^N$ is the clock supplied to the CPU, and $f_{XX}/2^5$, $f_{XX}/2^6$, and $f_{XX}/2^7$ are clocks supplied to peripheral hardware. $f_{XX}/2^N$ is stopped in HALT mode.

Remarks:

(1) N : Value set in bit 0 to bit 2 (PCC0 to PCC2) of the processor clock control register (N = 0 to 4)

(2) f_{XX} : Main system clock frequency

(3) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz.

6.4 16-Bit Timer/Event Counter Operations

6.4.1 Interval timer operations

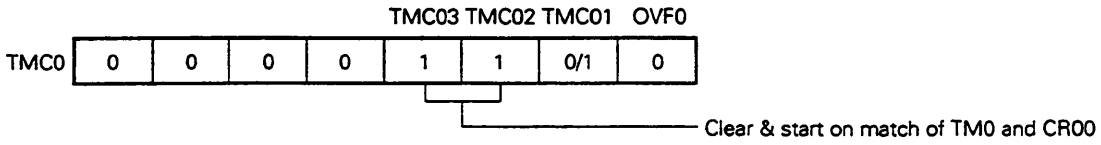
Setting the 16-bit timer mode control register (TMC0) and capture/compare control register 0 (CRC0) as shown in Fig. 6-10 allows operation as an interval timer. Interrupts are generated repeatedly using the count value set in 16-bit capture/compare register 00 (CR00) beforehand is used as the interval.

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

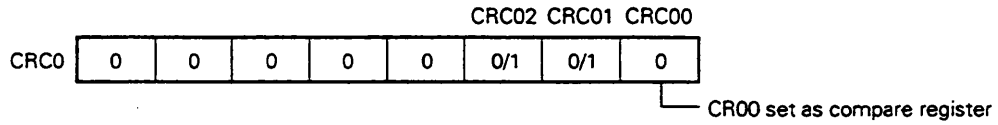
Count clock of the 16-bit timer/event counter can be selected with bits 4 to 6 (TCL04 to TCL06) of the timer clock select register 0 (TCL0).

Fig. 6-10 Control Register Settings for Interval Timer Operation

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



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



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

Fig. 6-11 Interval Timer Configuration Diagram

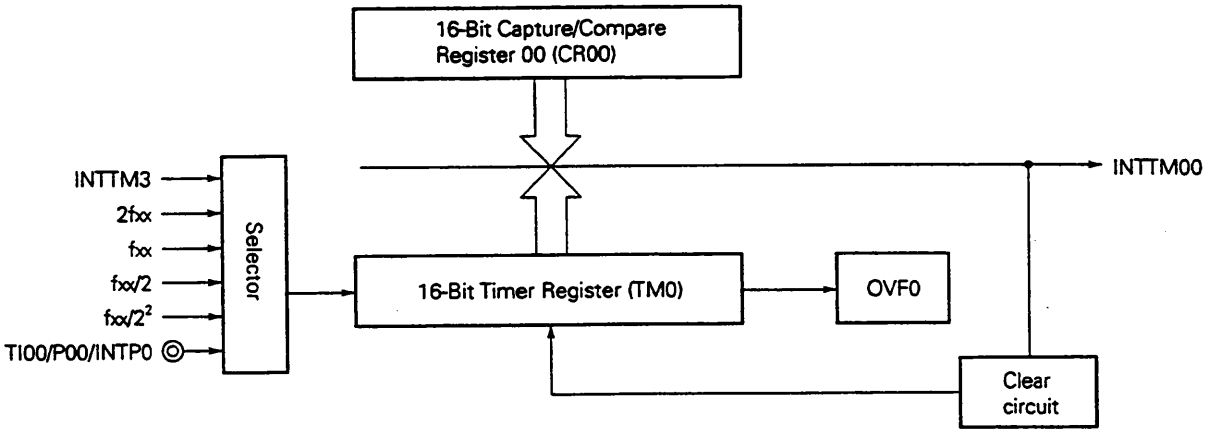
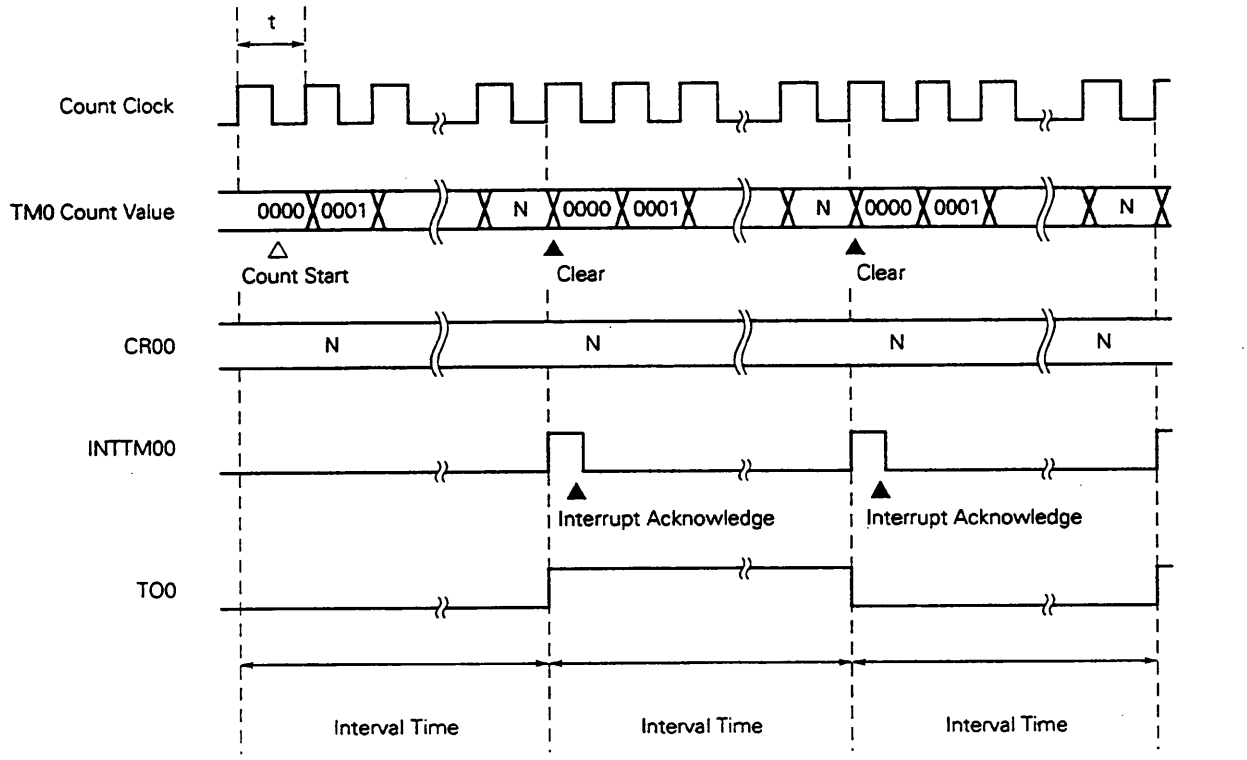


Fig. 6-12 Interval Timer Operation Timings



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

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

TCL06	TCL05	TCL04	Minimum interval time	Maximum interval time	Resolution
0	0	0	2 x TI00 input cycle	2^{16} x TI00 input cycle	TI00 input edge cycle
0	0	1	1/fxx (250 ns)	2^{15} x 1/fxx (8.19 ms)	1/2fxx (125 ns)
0	1	0	2 x 1/fxx (500 ns)	2^{16} x 1/fxx (16.4 ms)	1/fxx (250 ns)
0	1	1	2^2 x 1/fxx (1.0 μs)	2^{17} x 1/fxx (32.8 ms)	2 x 1/fxx (500 ns)
1	0	0	2^3 x 1/fxx (2.0 μs)	2^{18} x 1/fxx (65.5 ms)	2^2 x 1/fxx (1.0 μs)
1	1	1	2 x watch timer output cycle	2^{16} x watch timer output cycle	Watch timer output edge cycle
Other than above			Setting prohibited		

Remarks:

(1) fxx : Main system clock oscillation frequency

(2) Figures in parentheses apply to operation with fxx = 4.0 MHz

6.4.2 PWM output operations

Setting the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) as shown in Fig. 6-13 allows operation as PWM output. Pulses with the duty rate determined by the value set in 16-bit capture/compare register 00 (CR00) beforehand 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-pass 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 Φ can be selected with bits 4 to 6 (TCL04 to TCL06) of the timer clock select register 0 (TCL0).

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

Cautions:

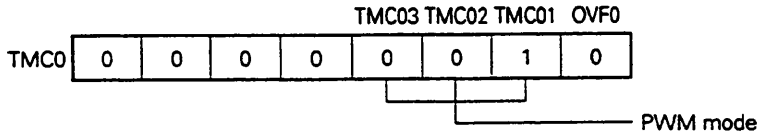
(1) PWM operation mode should be selected before setting CR00.

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

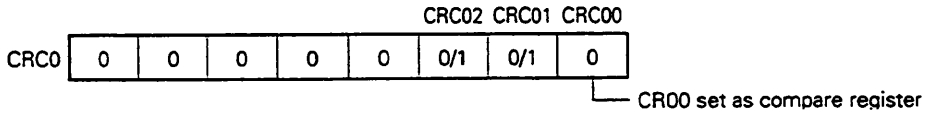
(3) Do not select PWM operation mode for external clock input from the TI00/P00 pin.

Fig. 6-13 Control Register Settings for PWM Output Operation

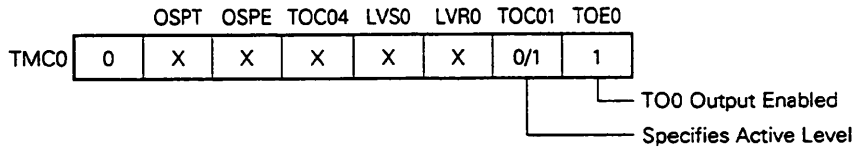
(a) 16-bit timer mode control register (TMC0)



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



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



Remarks:

(1) 0/1 : Setting 0 or 1 allows another function to be used simultaneously with PWM output. See the description of the respective control registers for details.

(2) × : Don't care

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. 6-14 is as follows.

$$V_{AN} = V_{REF} \times \frac{\text{capture/compare register 00 (CR00) value}}{2^{16}}$$

V_{REF} : External switching circuit reference voltage

Fig. 6-14 Example of D/A Converter Configuration with PWM Output

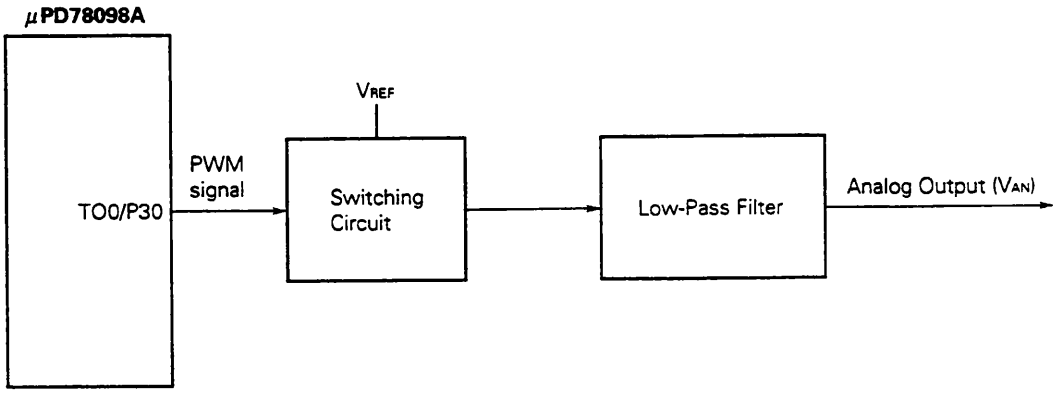
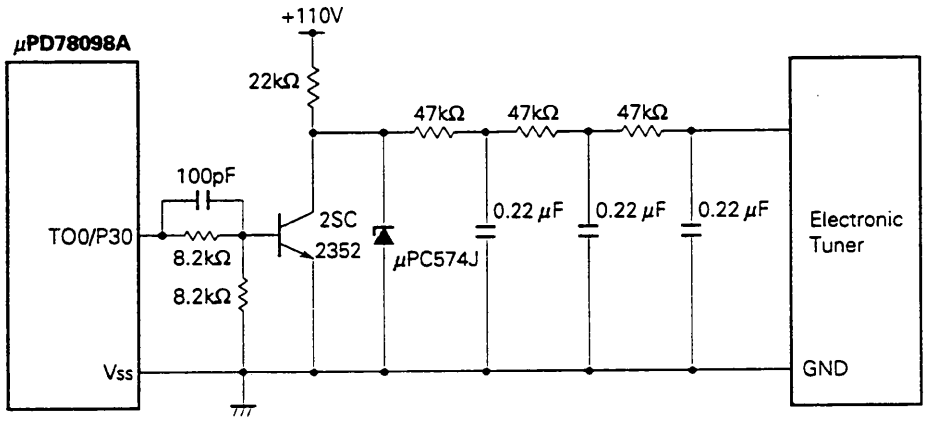


Fig. 6-15 shows an example in which PWM output is converted to an analog voltage and used in a voltage synthesizer type TV tuner.

Fig. 6-15 TV Tuner Application Circuit Example



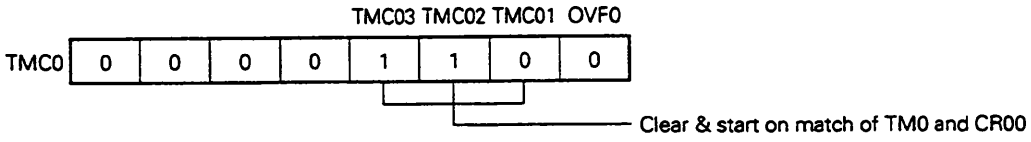
6.4.3 PPG output operations

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

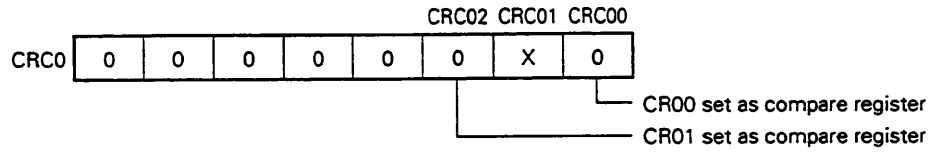
In the PPG output operation, square waves are output from the TO0/P30 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit capture/compare register 01 (CR01) and in 16-bit capture/compare register 00 (CR00), respectively.

Fig. 6-16 Control Register Settings for PPG Output Operation

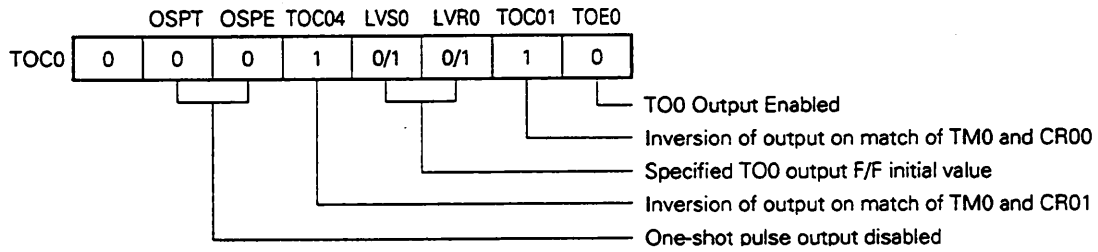
(a) 16-bit timer mode control register (TMC0)



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



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



Remark: × : Don't care

Caution: Values in the following range should be set in CR00 and CR01.

0000H ≤ CR01 < CR00 ≤ FFFFH

6.4.4 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/P00 pin and TI01/P01 pin using the 16-bit timer register (TM0).

There are two measurement methods: measuring with TM0 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00/P00 pin.

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

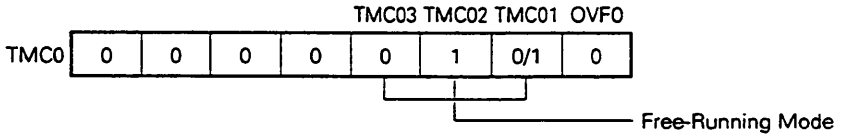
When the 16-bit timer register (TM0) is operated in free-running mode for PWM output, etc., (see register settings in Fig. 6-17), and the edge specified by external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTP0) is set.

Any of three edge specifications can be selected - rising, falling, or both edges - by means of bits 2 and 3 (ES10 and ES11) of INTM0.

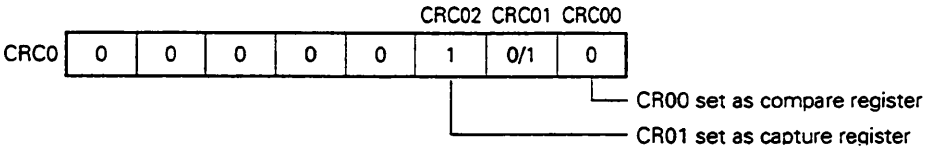
For valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Fig. 6-17 Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

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



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



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

Fig. 6-18 Configuration Diagram for Pulse Width Measurement by Free-Running Counter

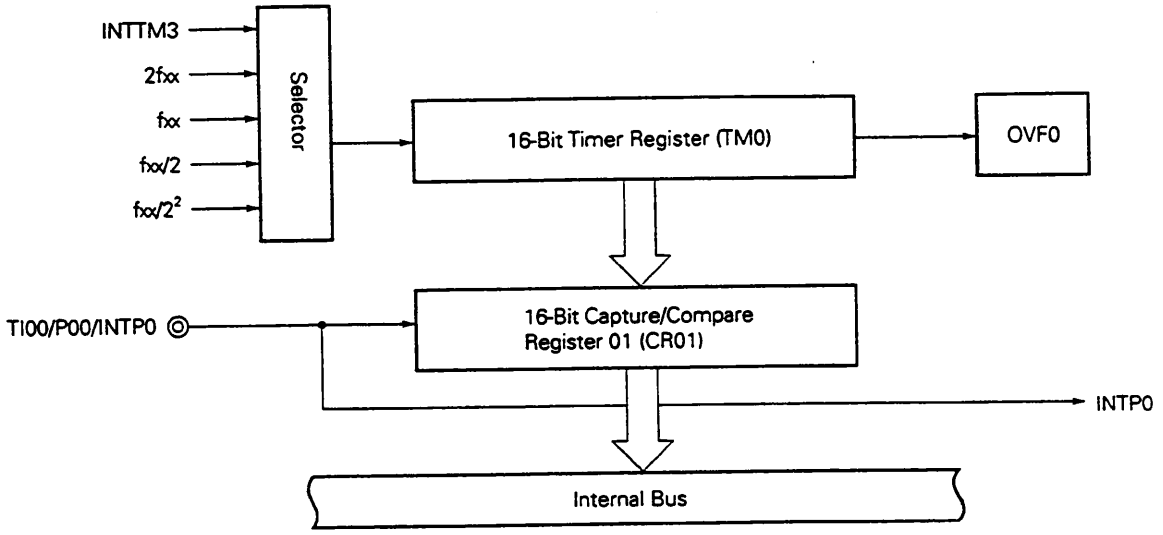
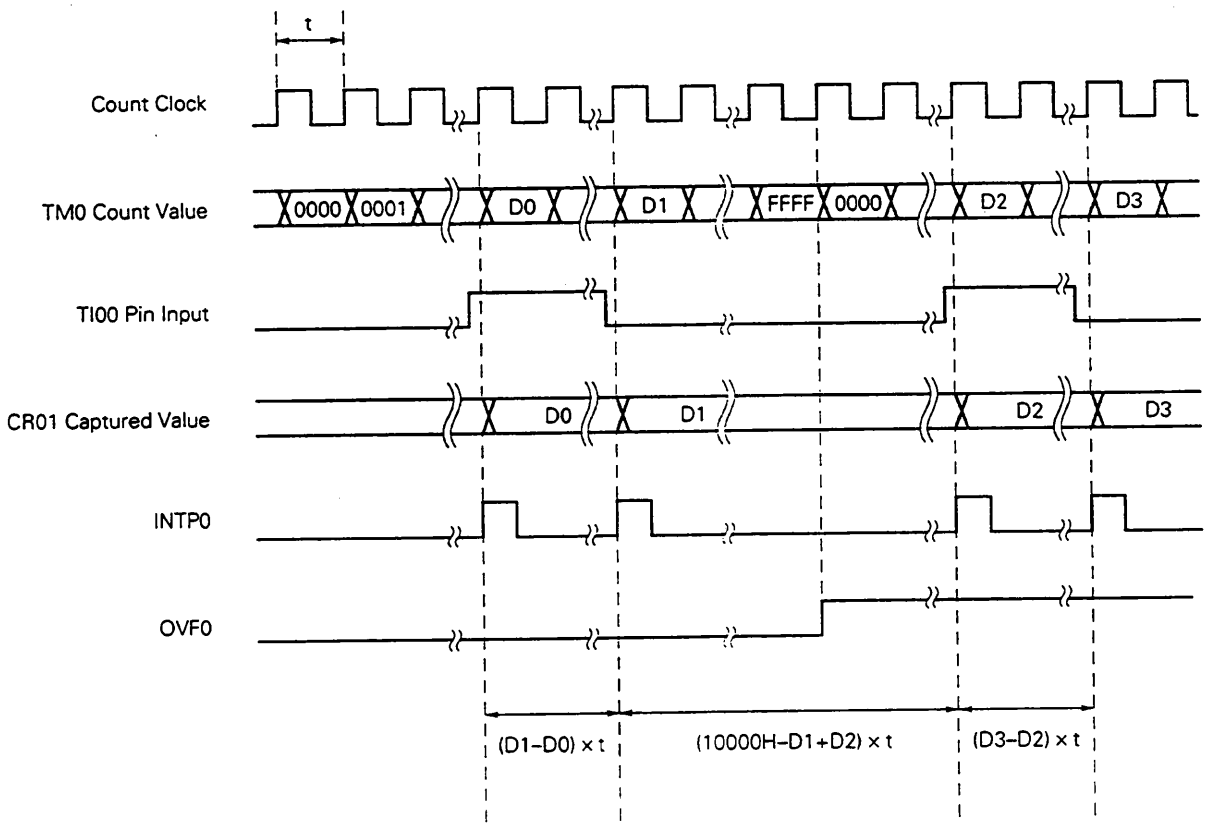


Fig. 6-19 Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



(2) Two pulse width measurements with free-running counter

When the 16-bit timer register (TM0) is operated in free-running mode for PWM output, etc., (see register settings in Fig. 6-20), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/P00 pin and the TI01/P01 pin.

When the edge specified by bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTP0) is set.

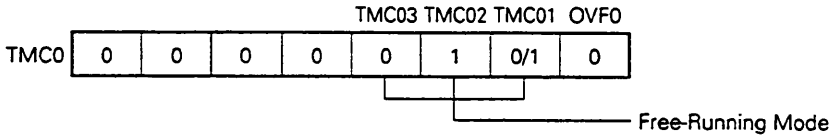
Also, when the edge specified by bits 4 and 5 (ES20 and ES21) of INTM0 is input to the TI01/P01 pin, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00) and an external interrupt request signal (INTP1) is set.

Any of three edge specifications can be selected - rising, falling, or both edges - as the valid edges for the TI00/P00 pin and the TI01/P01 pin by means of bits 2 and 3 (ES10 and ES11) and bits 4 and 5 (ES20 and ES21) of INTM0, respectively.

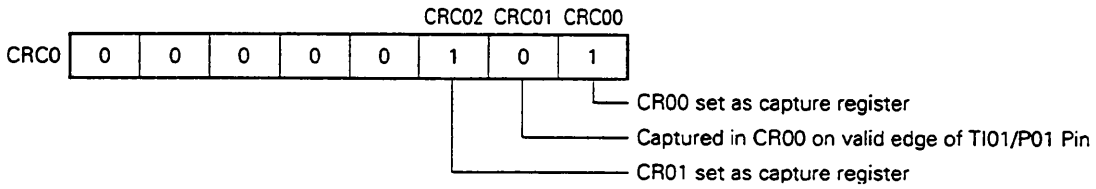
For TI00/P00 pin valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Fig. 6-20 Control Register Settings for Two Pulse Width Measurements with Free-Running Counter

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

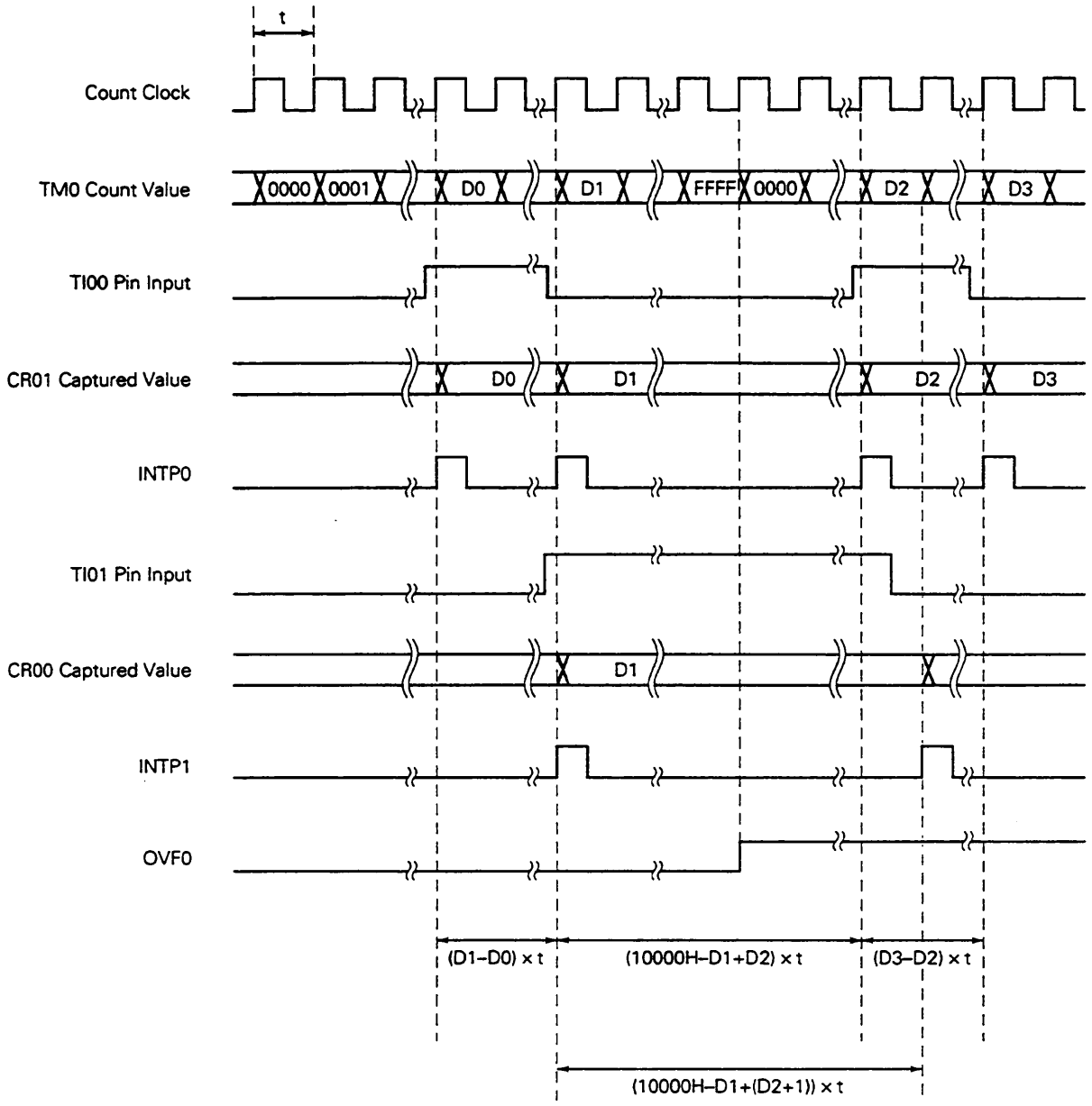


(b) Capture/compare control register 0 (CRC0)



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

Fig. 6-21 Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



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

When the 16-bit timer register (TM0) is operated in free-running mode for PWM output, etc., (see register settings in Fig. 6-22), it is possible to measure the pulse width of the signal input to the TI00/P00 pin.

When the edge specified by bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTP0) is set.

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

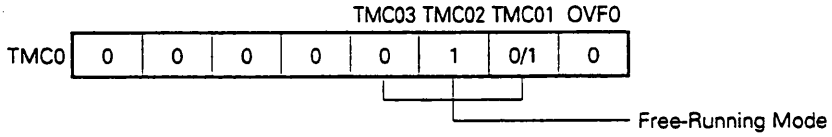
★ Any of two edge specifications can be selected - rising or falling edges - as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of INTM0.

For TI00/P00 pin valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

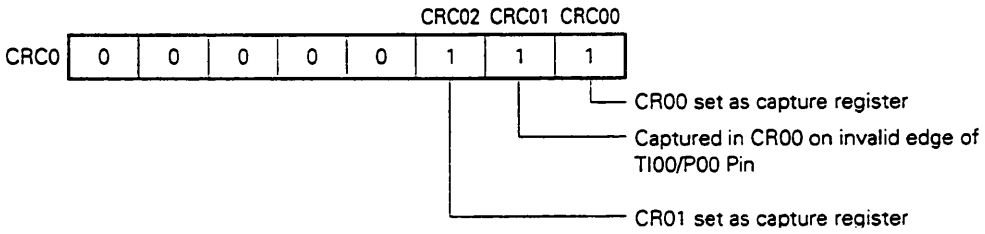
★ **Caution: When both the rising and falling edges are specified as the valid edges for Pin TI00/P00, the capture/compare register 00 (CR00) does not perform capture operations.**

Fig. 6-22 Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

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

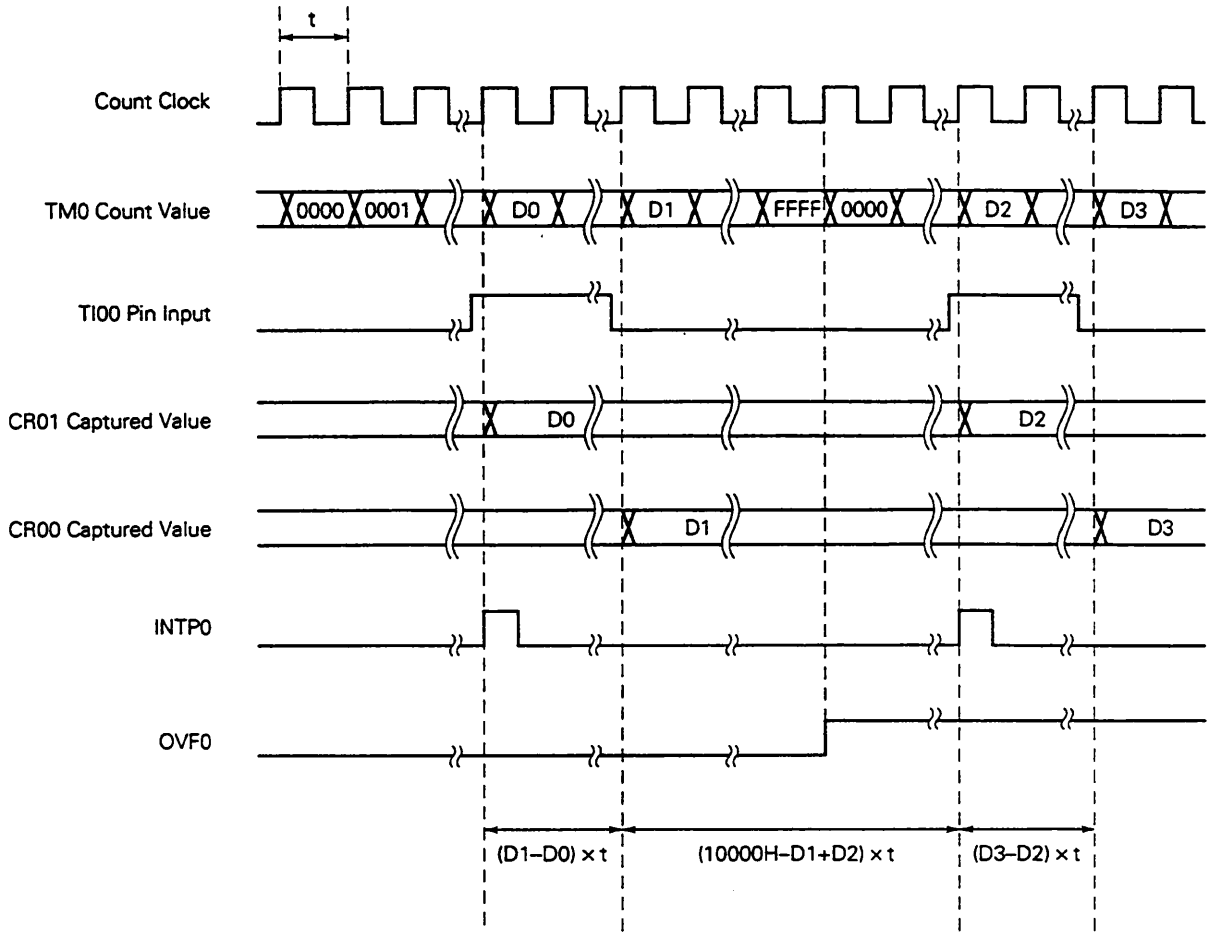


(b) Capture/compare control register 0 (CRC0)



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

Fig. 6-23 Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



(4) Pulse width measurement by means of restart

When input of a valid edge to the TI00/P00 pin is detected, the count value of the 16-bit timer register (TM0) is taken into 16-bit capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/P00 pin is measured by clearing TM0 and restarting the count (see register settings in Fig. 6-24).

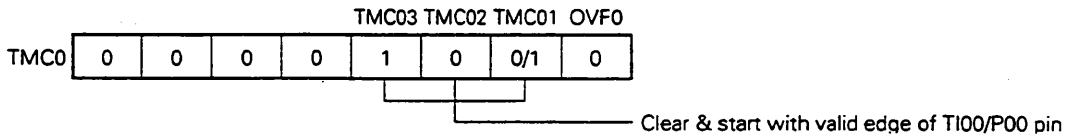
★ The edge specification can be selected from two types, rising and falling edges by INTM0 bit 2 and bit 3 (ES10 and ES11).

In a valid edge detection, the sampling is performed by a cycle selected by the sampling clock selection register (SCS), and a capture operation is not performed before detecting valid levels twice allowing short pulse width noise to be eliminated.

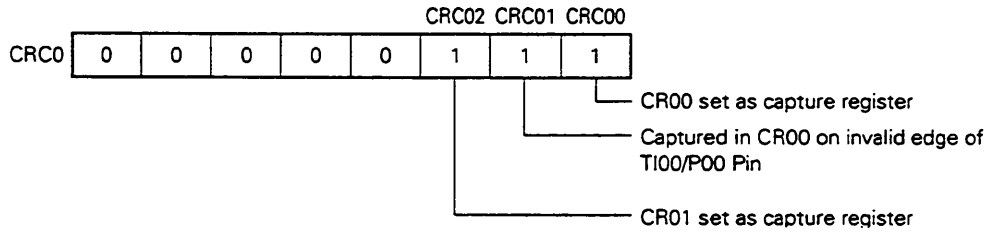
★ **Caution: When both the rising and falling edges are specified as the valid edges for Pin TI00/P00, the capture/compare register 00 (CR00) does not perform capture operations.**

Fig. 6-24 Control Register Settings for Pulse Width Measurement by Means of Restart

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

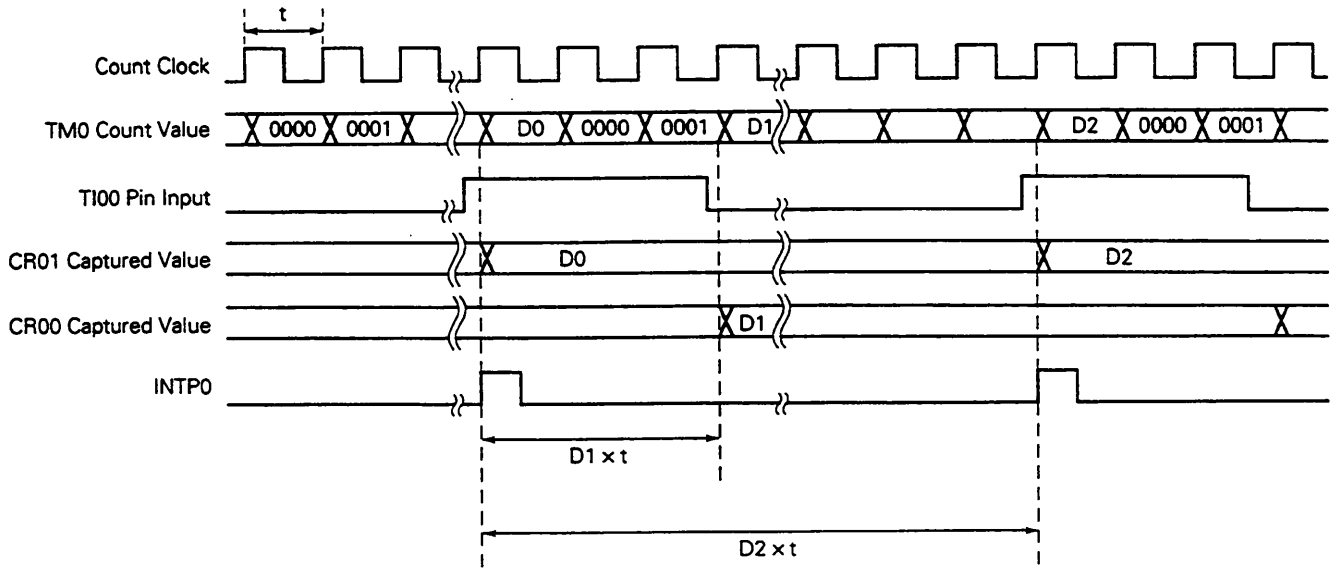


(b) Capture/compare control register 0 (CRC0)



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

Fig. 6-25 Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.4.5 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/P00 pin with the 16-bit timer register (TM0).

TM0 is incremented each time the valid edge specified with the external interrupt mode register 0 (INTM0) is input.

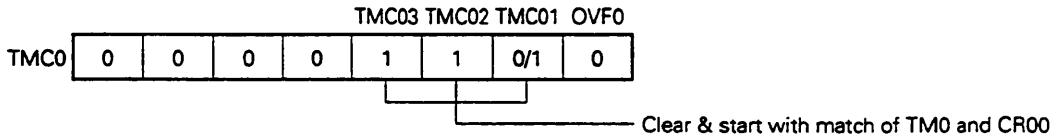
When the TM0 counted value matches the 16-bit capture/ compare register 00 (CR00) value, TM0 is cleared to 0 and the interrupt request signal (INTTM00) is generated.

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

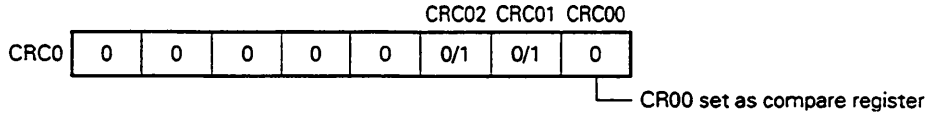
Because operation is carried out only after the valid edge is detected twice by sampling at the cycle selected with the sampling clock select register (SCS), noise with short pulse widths can be removed.

Fig. 6-26 Control Register Settings in External Event Counter Mode

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



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



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

Fig. 6-27 External Event Counter Configuration Diagram

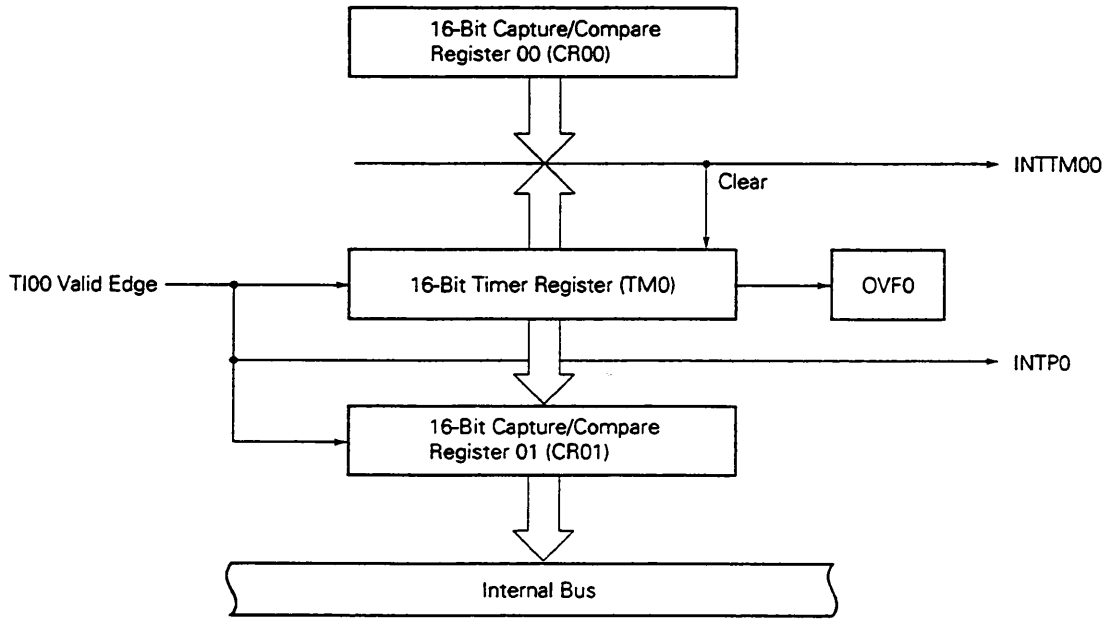
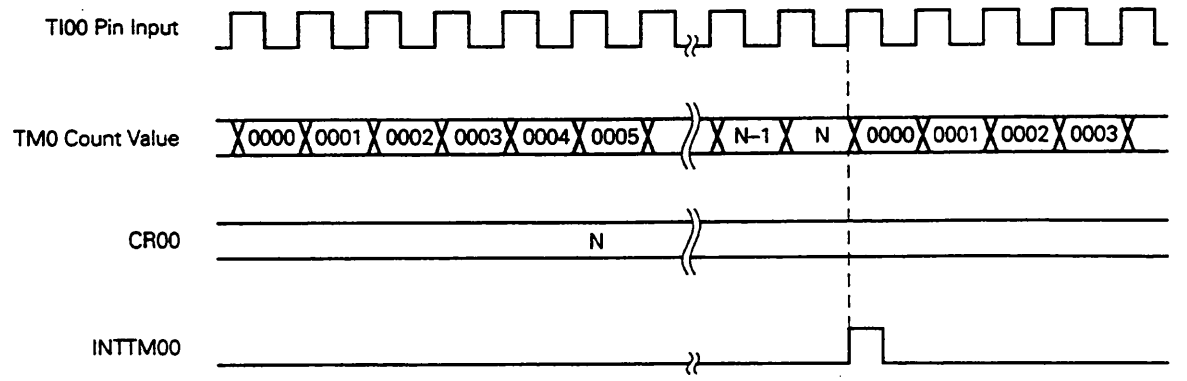


Fig. 6-28 External Event Counter Operation Timings (with Rising Edge Specified)



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

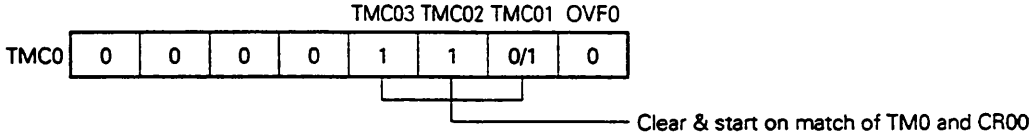
6.4.6 Square-wave output operation

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

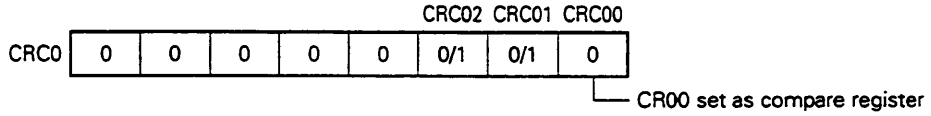
The TO0/P30 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of the 16-bit timer output control register to 1. This enables a square with any selected frequency to be output.

Fig. 6-29 Control Register Settings in Square Wave Output Mode

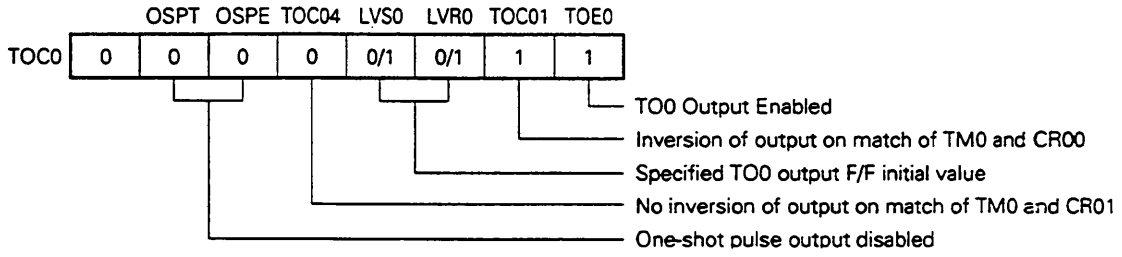
(a) 16-bit timer mode control register (TMC0)



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



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



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

Fig. 6-30 Square Wave Output Operation Timing

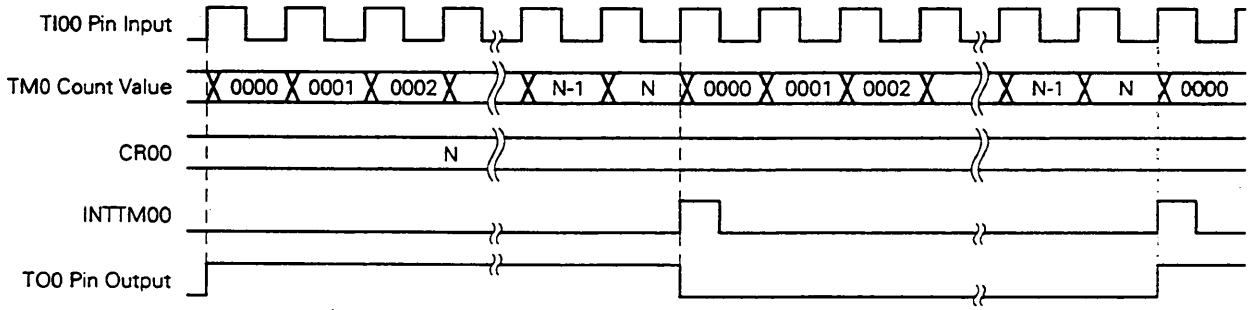


Table 6-7 16-Bit Timer/Event Count Square-Wave Output Ranges

Minimum pulse time	Maximum pulse time	Resolution
2 x TI00 input cycle	2^{16} x TI00 input cycle	TI00 input edge cycle
1/fxx (250 ns)	2^{15} x 1/fxx (8.19 ms)	1/2fxx (125 ns)
2 x 1/fxx (500 ns)	2^{16} x 1/fxx (16.4 ms)	1/fxx (250 ns)
2^2 x 1/fxx (1.0 μs)	2^{17} x 1/fxx (32.8 ms)	2 x 1/fxx (500 ns)
2^3 x 1/fxx (2.0 μs)	2^{18} x 1/fxx (65.5 ms)	2^2 x 1/fxx (1.0 μs)
2 x watch timer output cycle	2^{16} x watch timer output cycle	Watch timer output edge cycle

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 MHz

6.4.7 One-shot pulse output operation

It is possible to output one-shot pulses synchronized with a software trigger or an external trigger (TI00/P00 pin input).

(1) One-shot pulse output using software trigger

If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Fig. 6-31, and 1 is set in bit 6 (OSPT) of TOC0 by software, a one-shot pulse is output from the TO0/P30 pin.

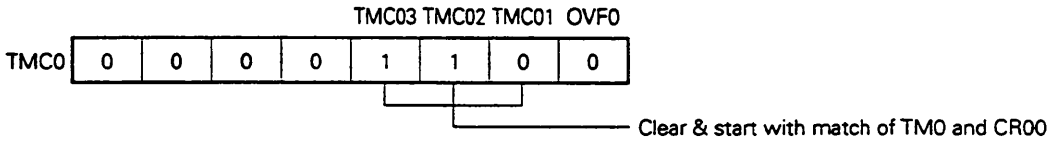
By setting 1 in OSPT, the 16-bit timer/event counter is cleared and started, and output is activated by the count value set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

TM0 continues operations after the one-shot pulse output. To stop TM0, 00H must be set to TMC0. ★

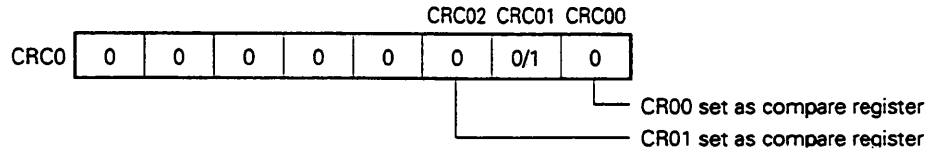
Caution: When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, execute after the INTTM00, or interrupt match signal with CR00, is generated.

Fig. 6-31 Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger

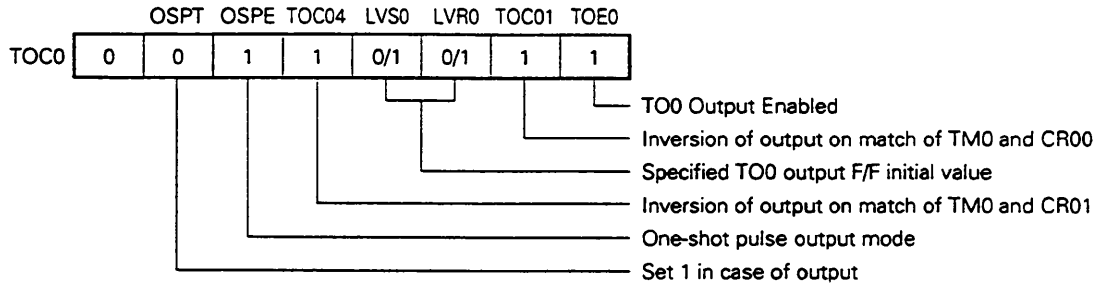
(a) 16-bit timer mode control register (TMC0)



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



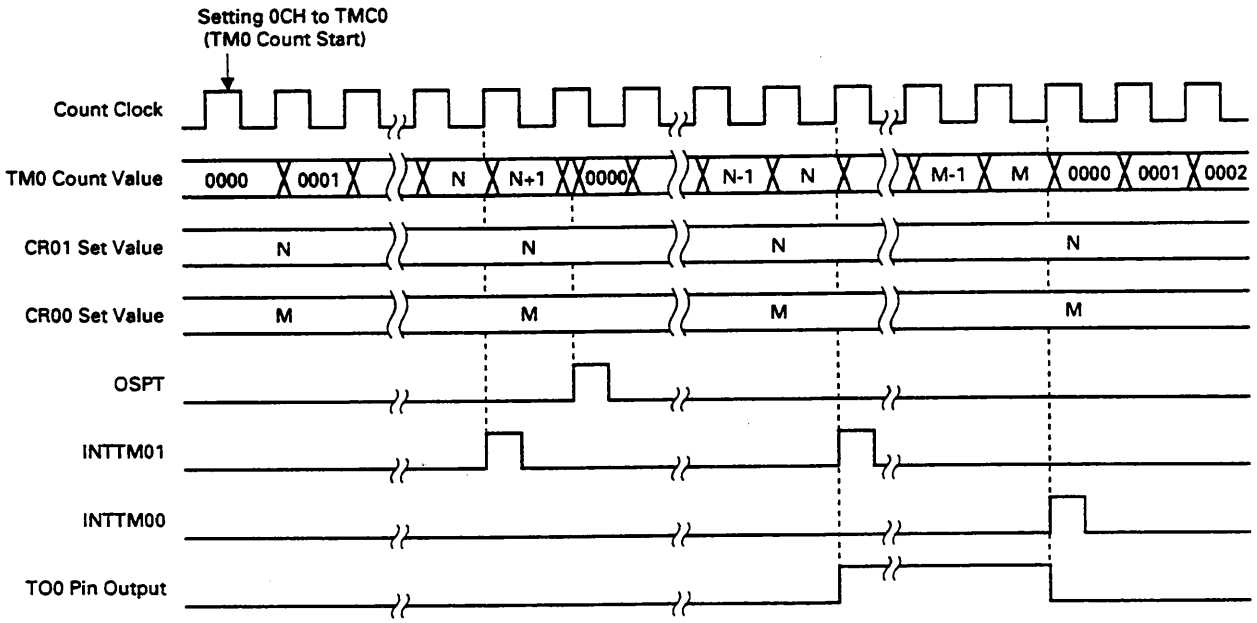
(c) 16-bit timer output control register (TOC0)



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

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

Fig. 6-32 Timing of One-Shot Pulse Output Operation Using Software Trigger ★



Caution: The 16-bit timer register starts operating at the point where values other than 0, 0, 0 (operation stop mode) are set to TMC01 to TMC03.

(2) One-shot pulse output using external trigger

If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Fig. 6-33, a one-shot pulse is output from the TO0/P30 pin with a TI00/P00 valid edge as an external trigger.

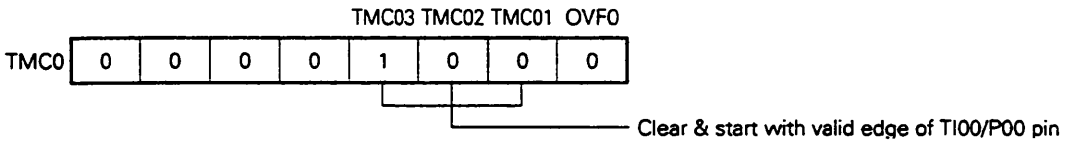
Any of three edge specifications can be selected - rising, falling, or both edges - as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0).

When a valid edge is input to the TI00/P00 pin, the 16-bit timer/event counter is cleared and started, and output is activated by the count value set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

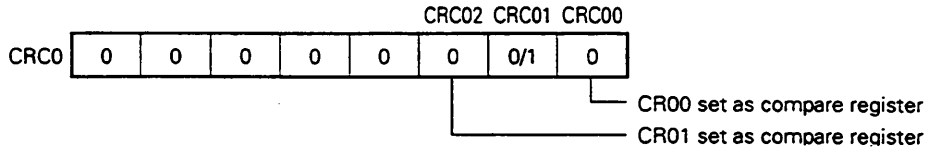
Caution: When outputting one-shot pulses, external trigger is ignored if generated again.

Fig. 6-33 Control Register Settings for One-Shot Pulse Output Operation Using External Trigger

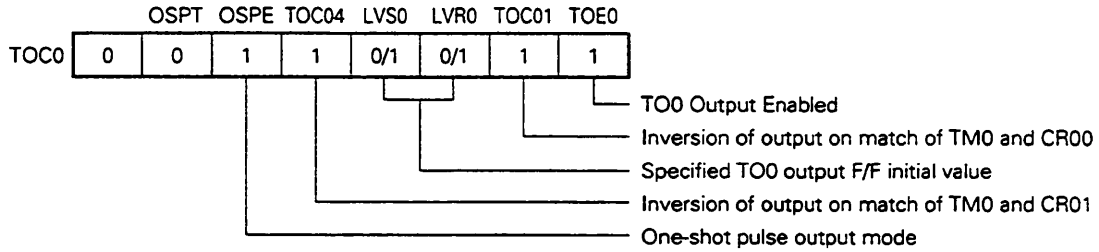
(a) 16-bit timer mode control register (TMC0)



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



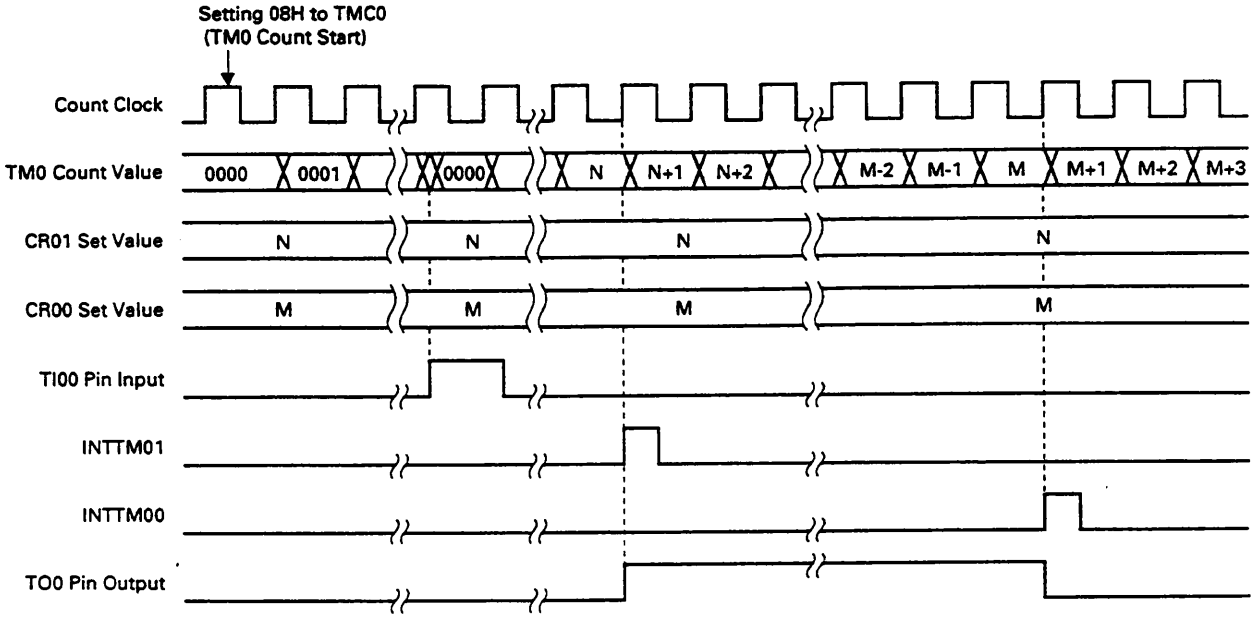
(c) 16-bit timer output control register (TOC0)



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

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

Fig. 6-34 Timing of One-Shot Pulse Output Operation Using External Trigger (with Rising Edge Specified) ★



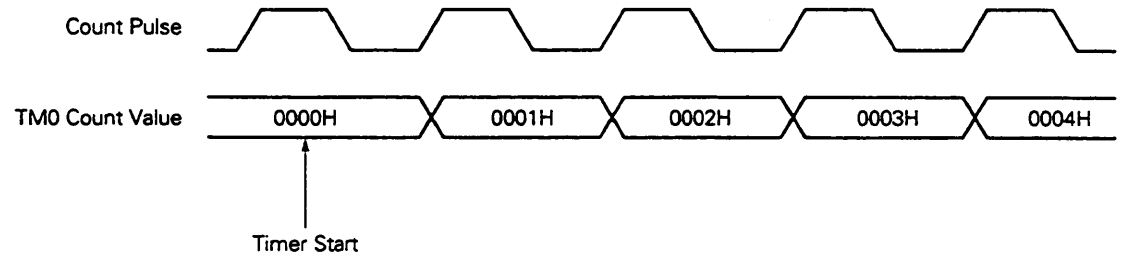
Caution: The 16-bit timer register starts operating at the point where values other than 0, 0, 0 (operation stop mode) are set to TMC01 to TMC03.

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. 6-35 16-Bit Timer Register Start Timings



(2) 16-bit compare register set

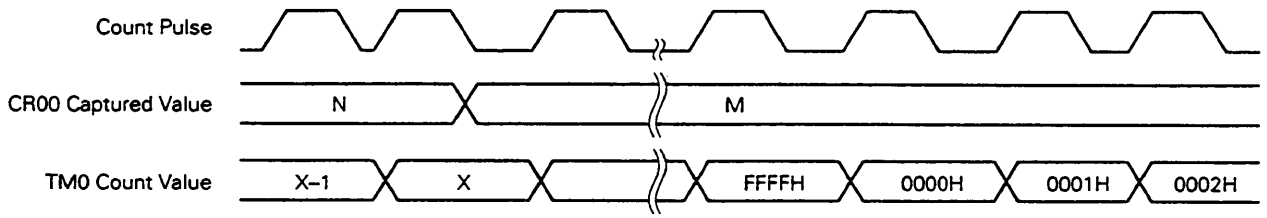
Set a value other than 0000H to the 16-bit capture/compare register 00 (CR00).

Thus, when using the 16-bit capture/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 capture/compare register 00 (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 after CR00 change (M) is smaller than that before change (N), it is necessary to restart the timer after changing CR00.

Fig. 6-36 Timings after Change of Compare Register during Timer Count Operation

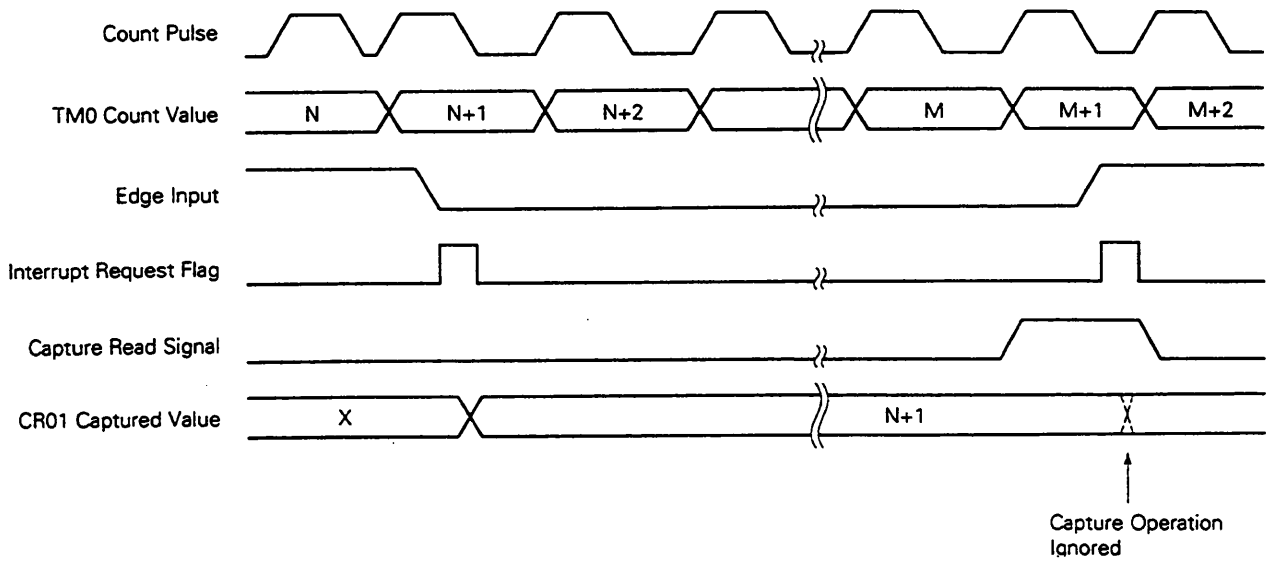


Remark: N > X > M

(4) Capture register data retention timings

If the valid edge of the TI00/P00 pin is input during 16-bit capture/compare register 01 (CR01) read, CR01 holds data without carrying out capture operation. However, the interrupt request flag (PIF0) is set upon detection of the valid edge.

Fig. 6-37 Capture Register Data Retention Timings



(5) Valid edge set

Set the valid edge of the TI00/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 0.

(6) Re-trigger of one-shot pulse

(a) One-shot pulse output using software

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

(b) One-shot pulse output using external trigger

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

(7) Operation of OVF0 flag

OVF0 flag is set to 1 in the following case.

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

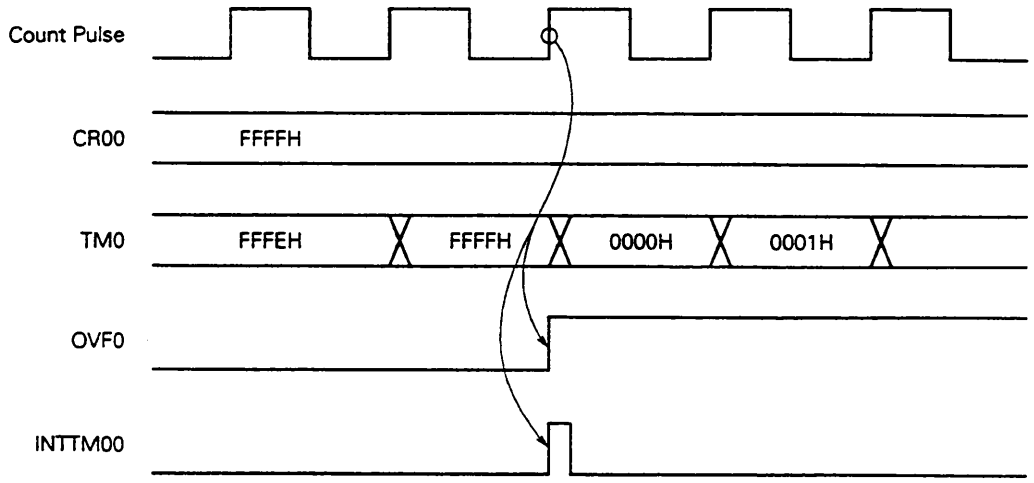
↓

CR00 is set to FFFFH

↓

When TM0 is counted up from FFFFH to 0000H

Fig. 6-38 Operation Timing of OVF0 Flag



CHAPTER 7 8-BIT TIMER/EVENT COUNTER

7.1 8-Bit Timer/Event Counter Functions

For the 8-bit timer/event counter incorporated in the μPD78098 sub-series, two modes are available. One is a mode for two-channel 8-bit timer/event counters to be used separately (the 8-bit timer/event counter mode) and the other is a mode for the 8-bit timer/event counter to be used as 16-bit timer/event counter (the 16-bit timer/event counter mode).

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

Interrupts are generated at the preset time intervals.

Table 7-1 8-Bit Timer/Event Counter Interval Times

Minimum interval time	Maximum interval time	Resolution
$2 \times 1/f_{XX}$ (500 ns)	$2^9 \times 1/f_{XX}$ (128.0 μs)	$2 \times 1/f_{XX}$ (500 ns)
$2^2 \times 1/f_{XX}$ (1.0 μs)	$2^{10} \times 1/f_{XX}$ (256.0 μs)	$2^2 \times 1/f_{XX}$ (1.0 μs)
$2^3 \times 1/f_{XX}$ (2.0 μs)	$2^{11} \times 1/f_{XX}$ (512.0 μs)	$2^3 \times 1/f_{XX}$ (2.0 μs)
$2^4 \times 1/f_{XX}$ (4.0 μs)	$2^{12} \times 1/f_{XX}$ (1.02 ms)	$2^4 \times 1/f_{XX}$ (4.0 μs)
$2^5 \times 1/f_{XX}$ (8.0 μs)	$2^{13} \times 1/f_{XX}$ (2.05 ms)	$2^5 \times 1/f_{XX}$ (8.0 μs)
$2^6 \times 1/f_{XX}$ (16.0 μs)	$2^{14} \times 1/f_{XX}$ (4.10 ms)	$2^6 \times 1/f_{XX}$ (16.0 μs)
$2^7 \times 1/f_{XX}$ (32.0 μs)	$2^{15} \times 1/f_{XX}$ (8.19 ms)	$2^7 \times 1/f_{XX}$ (32.0 μs)
$2^8 \times 1/f_{XX}$ (64.0 μs)	$2^{16} \times 1/f_{XX}$ (16.4 ms)	$2^8 \times 1/f_{XX}$ (64.0 μs)
$2^9 \times 1/f_{XX}$ (128.0 μs)	$2^{17} \times 1/f_{XX}$ (32.8 ms)	$2^9 \times 1/f_{XX}$ (128.0 μs)
$2^{11} \times 1/f_{XX}$ (512.0 μs)	$2^{19} \times 1/f_{XX}$ (131.1 ms)	$2^{11} \times 1/f_{XX}$ (512.0 μs)

Remarks:

(1) f_{XX}: Main system clock frequency

(2) Values in parentheses when operated at f_{XX} = 4.0 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 pulse time	Maximum pulse time	Resolution
$2 \times 1/fxx$ (500 ns)	$2^9 \times 1/fxx$ (128.0 μs)	$2 \times 1/fxx$ (500 ns)
$2^2 \times 1/fxx$ (1.0 μs)	$2^{10} \times 1/fxx$ (256.0 μs)	$2^2 \times 1/fxx$ (1.0 μs)
$2^3 \times 1/fxx$ (2.0 μs)	$2^{11} \times 1/fxx$ (512.0 μs)	$2^3 \times 1/fxx$ (2.0 μs)
$2^4 \times 1/fxx$ (4.0 μs)	$2^{12} \times 1/fxx$ (1.02 ms)	$2^4 \times 1/fxx$ (4.0 μs)
$2^5 \times 1/fxx$ (8.0 μs)	$2^{13} \times 1/fxx$ (2.05 ms)	$2^5 \times 1/fxx$ (8.0 μs)
$2^6 \times 1/fxx$ (16.0 μs)	$2^{14} \times 1/fxx$ (4.10 ms)	$2^6 \times 1/fxx$ (16.0 μs)
$2^7 \times 1/fxx$ (32.0 μs)	$2^{15} \times 1/fxx$ (8.19 ms)	$2^7 \times 1/fxx$ (32.0 μs)
$2^8 \times 1/fxx$ (64.0 μs)	$2^{16} \times 1/fxx$ (16.4 ms)	$2^8 \times 1/fxx$ (64.0 μs)
$2^9 \times 1/fxx$ (128.0 μs)	$2^{17} \times 1/fxx$ (32.8 ms)	$2^9 \times 1/fxx$ (128.0 μs)
$2^{11} \times 1/fxx$ (512.0 μs)	$2^{19} \times 1/fxx$ (131.1 ms)	$2^{11} \times 1/fxx$ (512.0 μs)

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 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 16-Bit Timer/Event Counter

Minimum interval time	Maximum interval time	Resolution
2 × 1/fxx (500 ns)	2^{17} × 1/fxx (32.8 ms)	2 × 1/fxx (500 ns)
2^2 × 1/fxx (1.0 μs)	2^{18} × 1/fxx (65.5 ms)	2^2 × 1/fxx (1.0 μs)
2^3 × 1/fxx (2.0 μs)	2^{19} × 1/fxx (131.1 ms)	2^3 × 1/fxx (2.0 μs)
2^4 × 1/fxx (4.0 μs)	2^{20} × 1/fxx (262.1 ms)	2^4 × 1/fxx (4.0 μs)
2^5 × 1/fxx (8.0 μs)	2^{21} × 1/fxx (524.3 ms)	2^5 × 1/fxx (8.0 μs)
2^6 × 1/fxx (16.0 μs)	2^{22} × 1/fxx (1.0 s)	2^6 × 1/fxx (16.0 μs)
2^7 × 1/fxx (32.0 μs)	2^{23} × 1/fxx (2.1 s)	2^7 × 1/fxx (32.0 μs)
2^8 × 1/fxx (64.0 μs)	2^{24} × 1/fxx (4.2 s)	2^8 × 1/fxx (64.0 μs)
2^9 × 1/fxx (128.0 μs)	2^{25} × 1/fxx (8.4 s)	2^9 × 1/fxx (128.0 μs)
2^{11} × 1/fxx (512.0 μs)	2^{27} × 1/fxx (33.6 s)	2^{11} × 1/fxx (512.0 μs)

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 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 pulse time	Maximum pulse time	Resolution
$2 \times 1/f_{XX}$ (500 ns)	$2^{17} \times 1/f_{XX}$ (32.8 ms)	$2 \times 1/f_{XX}$ (500 ns)
$2^2 \times 1/f_{XX}$ (1.0 μs)	$2^{18} \times 1/f_{XX}$ (65.5 ms)	$2^2 \times 1/f_{XX}$ (1.0 μs)
$2^3 \times 1/f_{XX}$ (2.0 μs)	$2^{19} \times 1/f_{XX}$ (131.1 ms)	$2^3 \times 1/f_{XX}$ (2.0 μs)
$2^4 \times 1/f_{XX}$ (4.0 μs)	$2^{20} \times 1/f_{XX}$ (262.1 ms)	$2^4 \times 1/f_{XX}$ (4.0 μs)
$2^5 \times 1/f_{XX}$ (8.0 μs)	$2^{21} \times 1/f_{XX}$ (524.3 ms)	$2^5 \times 1/f_{XX}$ (8.0 μs)
$2^6 \times 1/f_{XX}$ (16.0 μs)	$2^{22} \times 1/f_{XX}$ (1.0 s)	$2^6 \times 1/f_{XX}$ (16.0 μs)
$2^7 \times 1/f_{XX}$ (32.0 μs)	$2^{23} \times 1/f_{XX}$ (2.1 s)	$2^7 \times 1/f_{XX}$ (32.0 μs)
$2^8 \times 1/f_{XX}$ (64.0 μs)	$2^{24} \times 1/f_{XX}$ (4.2 s)	$2^8 \times 1/f_{XX}$ (64.0 μs)
$2^9 \times 1/f_{XX}$ (128.0 μs)	$2^{25} \times 1/f_{XX}$ (8.4 s)	$2^9 \times 1/f_{XX}$ (128.0 μs)
$2^{11} \times 1/f_{XX}$ (512.0 μs)	$2^{27} \times 1/f_{XX}$ (33.6 s)	$2^{11} \times 1/f_{XX}$ (512.0 μs)

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 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 × 2 (TM1, TM2)
Register	Compare register: 8 bits × 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)

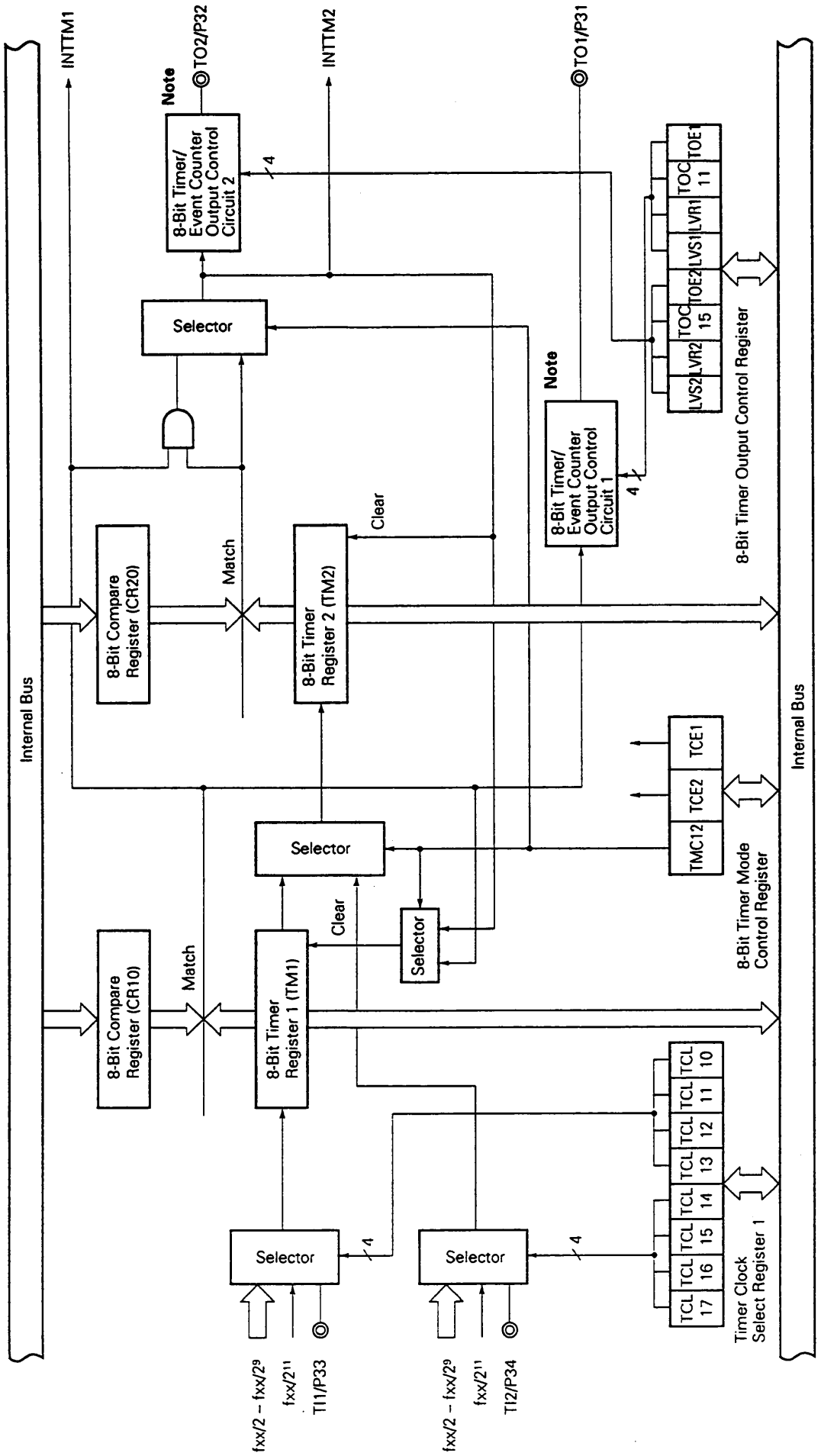
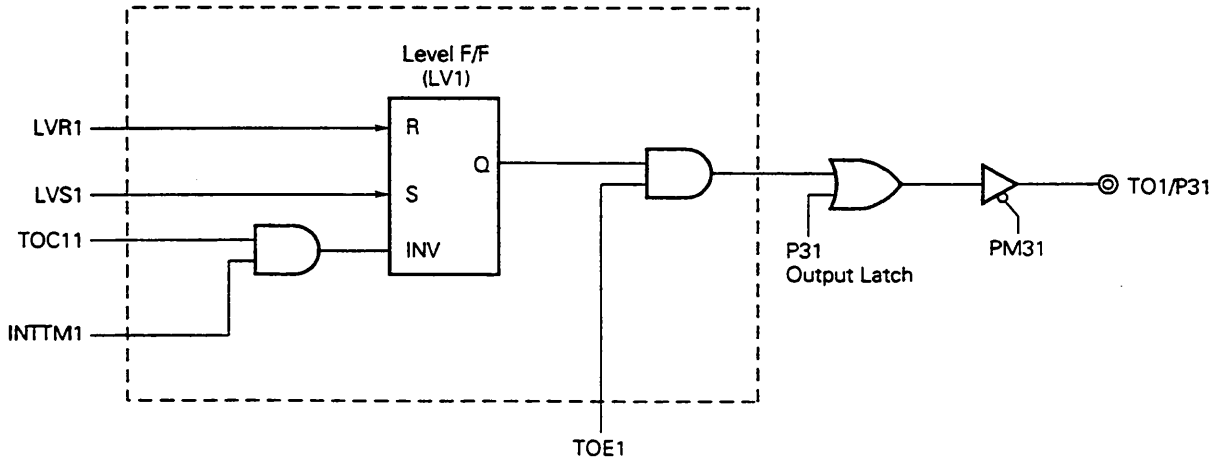


Fig. 7-1 8-Bit Timer/Event Counter Block Diagram

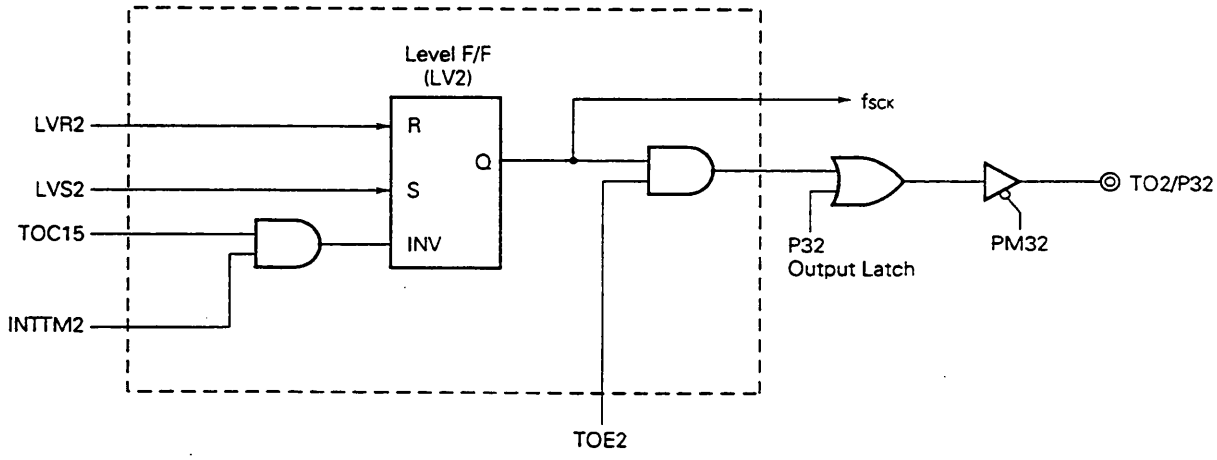
Note: Refer to Figs. 7-2 and 7-3 for details of 8-bit timer/event counter output control circuits 1 and 2, respectively.

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) fsck: Serial clock frequency

(1) 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 manipulation instruction. They cannot be set with a 16-bit memory manipulation 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, 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 manipulation instruction. When TM1 and TM2 are used as 16-bit timer × 1-channel mode, 16-bit timer (TMS) is read with a 16-bit memory manipulation 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 manipulation 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	When 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	1	0	$f_{XX}/2$ (2.0 MHz)
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)
Other than 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	1	0	$f_{XX}/2$ (2.0 MHz)
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)
Other than above				Setting prohibited

Remarks:

(1) f_{XX} : Main system clock frequency

(2) TI1 : 8-bit timer register 1 input pin

(3) TI2 : 8-bit timer register 2 input pin

(4) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz.

★ **Caution: When rewriting TCL1 to data other than the same data, stop the timer operation first.**

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

This register enables/stops 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 manipulation 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	When 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 clear to 0)
1	Operation enable

TCE2	8-Bit Timer Register 2 Operation Control
0	Operation stop (TM2 clear to 0)
1	Operation enable

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 operating mode after stopping timer operation.

(2) When used as 16-bit timer register, TCE1 should be used for operation enable/stop.

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

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

It 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 manipulation 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	When 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 disable (port mode)
1	Output enable

TOE11	8-Bit Timer/Event Counter 1 Timer Output F/F Control
0	Inverted operation disable
1	Inverted operation enable

LVS1	LVR1	8-Bit Timer/Event Counter 1 Timer Output F/F Status Set
0	0	Unchanged
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 disable (port mode)
1	Output enable

TOC15	8-Bit Timer/Event Counter 2 Timer Output F/F Control
0	Inverted operation disable
1	Inverted operation enable

LVS2	LVR2	8-Bit Timer/Event Counter 2 Timer Output F/F Status Set
0	0	Unchanged
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 input/output in 1-bit units.

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

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

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

Fig. 7-7 Port Mode Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

7.4 8-Bit Timer/Event Counter Operations

7.4.1 8-bit timer/event counter mode

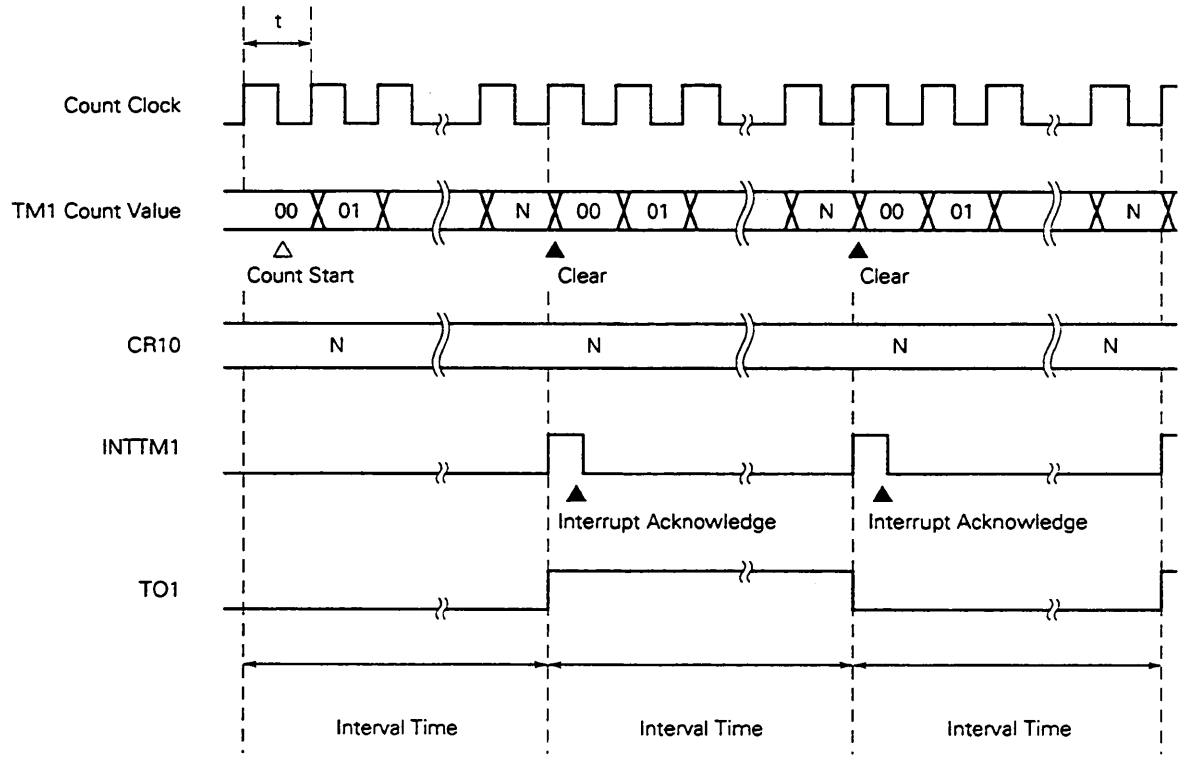
(1) Interval timer operations

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 timer clock select register 1 (TCL1).

Fig. 7-8 Interval Timer Operation Timings



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

Table 7-6 8-Bit Timer/Event Counter 1 Interval Time

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	1	0	2 x 1/fxx (500 ns)	2^9 x 1/fxx (128.0 μs)	2 x 1/fxx (500 ns)
0	1	1	1	2^2 x 1/fxx (1.0 μs)	2^{10} x 1/fxx (256.0 μs)	2^2 x 1/fxx (1.0 μs)
1	0	0	0	2^3 x 1/fxx (2.0 μs)	2^{11} x 1/fxx (512.0 μs)	2^3 x 1/fxx (2.0 μs)
1	0	0	1	2^4 x 1/fxx (4.0 μs)	2^{12} x 1/fxx (1.02 ms)	2^4 x 1/fxx (4.0 μs)
1	0	1	0	2^5 x 1/fxx (8.0 μs)	2^{13} x 1/fxx (2.05 ms)	2^5 x 1/fxx (8.0 μs)
1	0	1	1	2^6 x 1/fxx (16.0 μs)	2^{14} x 1/fxx (4.10 ms)	2^6 x 1/fxx (16.0 μs)
1	1	0	0	2^7 x 1/fxx (32.0 μs)	2^{15} x 1/fxx (8.19 ms)	2^7 x 1/fxx (32.0 μs)
1	1	0	1	2^8 x 1/fxx (64.0 μs)	2^{16} x 1/fxx (16.4 ms)	2^8 x 1/fxx (64.0 μs)
1	1	1	0	2^9 x 1/fxx (128.0 μs)	2^{17} x 1/fxx (32.8 ms)	2^9 x 1/fxx (128.0 μs)
1	1	1	1	2^{11} x 1/fxx (512.0 μs)	2^{19} x 1/fxx (131.1 ms)	2^{11} x 1/fxx (512.0 μs)
Other than above				Setting prohibited		

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 MHz.

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

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	1	0	2 x 1/fxx (500 ns)	2^9 x 1/fxx (128.0 μs)	2 x 1/fxx (500 ns)
0	1	1	1	2^2 x 1/fxx (1.0 μs)	2^{10} x 1/fxx (256.0 μs)	2^2 x 1/fxx (1.0 μs)
1	0	0	0	2^3 x 1/fxx (2.0 μs)	2^{11} x 1/fxx (512.0 μs)	2^3 x 1/fxx (2.0 μs)
1	0	0	1	2^4 x 1/fxx (4.0 μs)	2^{12} x 1/fxx (1.02 ms)	2^4 x 1/fxx (4.0 μs)
1	0	1	0	2^5 x 1/fxx (8.0 μs)	2^{13} x 1/fxx (2.05 ms)	2^5 x 1/fxx (8.0 μs)
1	0	1	1	2^6 x 1/fxx (16.0 μs)	2^{14} x 1/fxx (4.10 ms)	2^6 x 1/fxx (16.0 μs)
1	1	0	0	2^7 x 1/fxx (32.0 μs)	2^{15} x 1/fxx (8.19 ms)	2^7 x 1/fxx (32.0 μs)
1	1	0	1	2^8 x 1/fxx (64.0 μs)	2^{16} x 1/fxx (16.4 ms)	2^8 x 1/fxx (64.0 μs)
1	1	1	0	2^9 x 1/fxx (128.0 μs)	2^{17} x 1/fxx (32.8 ms)	2^9 x 1/fxx (128.0 μs)
1	1	1	1	2^{11} x 1/fxx (512.0 μs)	2^{19} x 1/fxx (131.1 ms)	2^{11} x 1/fxx (512.0 μs)
Other than above				Setting prohibited		

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 MHz.

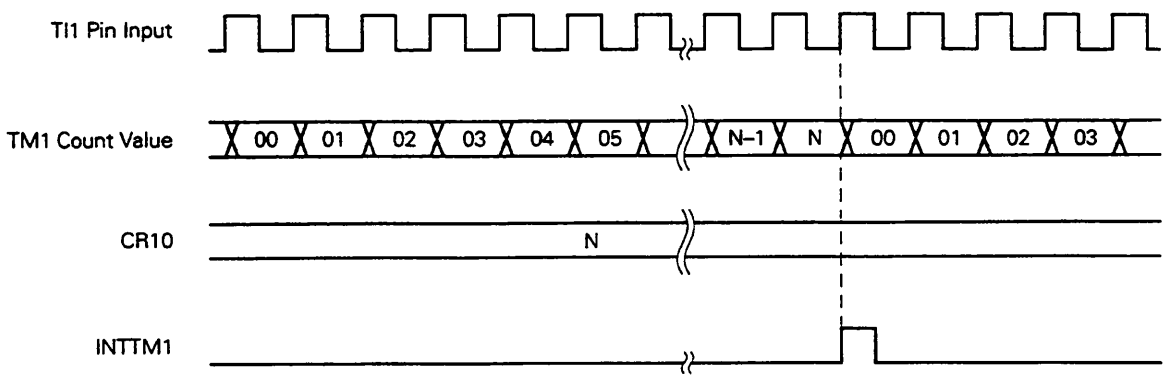
(2) External event counter operation

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 1 (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 reversed 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 wave with any selected frequency to be output.

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

Minimum pulse time	Maximum pulse time	Resolution
$2 \times 1/fxx$ (500 ns)	$2^9 \times 1/fxx$ (128.0 μs)	$2 \times 1/fxx$ (500 ns)
$2^2 \times 1/fxx$ (1.0 μs)	$2^{10} \times 1/fxx$ (256.0 μs)	$2^2 \times 1/fxx$ (1.0 μs)
$2^3 \times 1/fxx$ (2.0 μs)	$2^{11} \times 1/fxx$ (512.0 μs)	$2^3 \times 1/fxx$ (2.0 μs)
$2^4 \times 1/fxx$ (4.0 μs)	$2^{12} \times 1/fxx$ (1.02 ms)	$2^4 \times 1/fxx$ (4.0 μs)
$2^5 \times 1/fxx$ (8.0 μs)	$2^{13} \times 1/fxx$ (2.05 ms)	$2^5 \times 1/fxx$ (8.0 μs)
$2^6 \times 1/fxx$ (16.0 μs)	$2^{14} \times 1/fxx$ (4.10 ms)	$2^6 \times 1/fxx$ (16.0 μs)
$2^7 \times 1/fxx$ (32.0 μs)	$2^{15} \times 1/fxx$ (8.19 ms)	$2^7 \times 1/fxx$ (32.0 μs)
$2^8 \times 1/fxx$ (64.0 μs)	$2^{16} \times 1/fxx$ (16.4 ms)	$2^8 \times 1/fxx$ (64.0 μs)
$2^9 \times 1/fxx$ (128.0 μs)	$2^{17} \times 1/fxx$ (32.8 ms)	$2^9 \times 1/fxx$ (128.0 μs)
$2^{11} \times 1/fxx$ (512.0 μs)	$2^{19} \times 1/fxx$ (131.1 ms)	$2^{11} \times 1/fxx$ (512.0 μs)

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 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 bit 0 (TCE1) of TMC1.

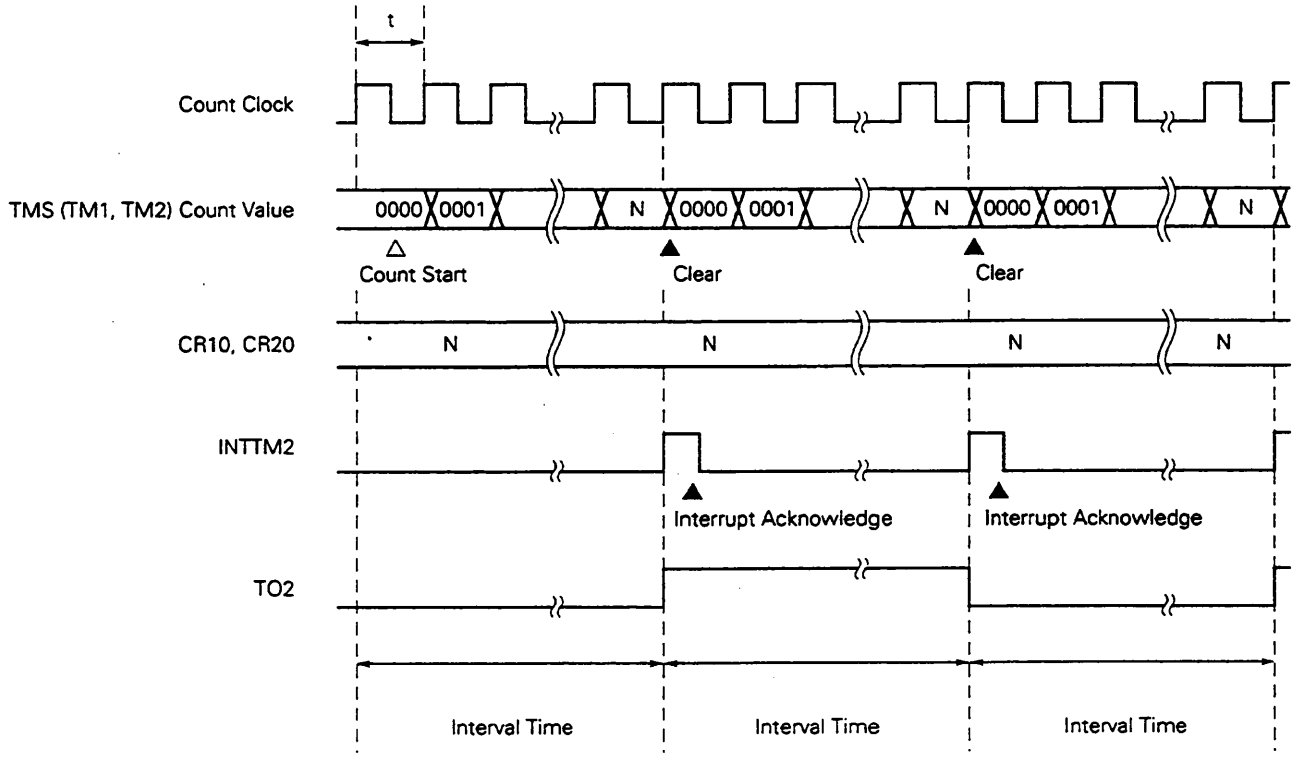
(1) Interval timer

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 1 (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 inverted. Thus, when using 8-bit timer/event counter as 16-bit interval timer, set the INTTM1 mask flag TMMK1 to 1 to disable INTTM1 acknowledgment.

When reading 16-bit timer (TMS) count value, use 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	2^9 × TI1 input cycle	TI1 input edge cycle
0	1	1	0	2 × 1/fxx (500 ns)	2^{17} × 1/fxx (32.8 ms)	2 × 1/fxx (500 ns)
0	1	1	1	2^2 × 1/fxx (1.0 μs)	2^{18} × 1/fxx (65.5 ms)	2^2 × 1/fxx (1.0 μs)
1	0	0	0	2^3 × 1/fxx (2.0 μs)	2^{19} × 1/fxx (131.1 ms)	2^3 × 1/fxx (2.0 μs)
1	0	0	1	2^4 × 1/fxx (4.0 μs)	2^{20} × 1/fxx (262.1 ms)	2^4 × 1/fxx (4.0 μs)
1	0	1	0	2^5 × 1/fxx (8.0 μs)	2^{21} × 1/fxx (524.3 ms)	2^5 × 1/fxx (8.0 μs)
1	0	1	1	2^6 × 1/fxx (16.0 μs)	2^{22} × 1/fxx (1.0 s)	2^6 × 1/fxx (16.0 μs)
1	1	0	0	2^7 × 1/fxx (32.0 μs)	2^{23} × 1/fxx (2.1 s)	2^7 × 1/fxx (32.0 μs)
1	1	0	1	2^8 × 1/fxx (64.0 μs)	2^{24} × 1/fxx (4.2 s)	2^8 × 1/fxx (64.0 μs)
1	1	1	0	2^9 × 1/fxx (128.0 μs)	2^{25} × 1/fxx (8.4 s)	2^9 × 1/fxx (128.0 μs)
1	1	1	1	2^{11} × 1/fxx (512.0 μs)	2^{27} × 1/fxx (33.6 s)	2^{11} × 1/fxx (512.0 μs)
Other than above				Setting prohibited		

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses when operated at fxx = 4.0 MHz.

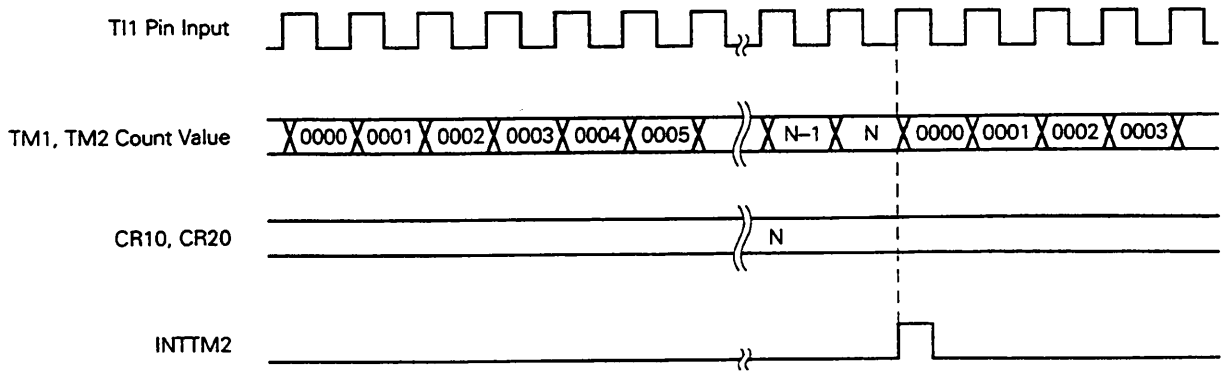
(2) External event counter operations

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 1 (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 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 inverted. Thus, when using 8-bit timer/event counter as 16-bit interval timer, set the INTTM1 mask flag TMMK1 to 1 to disable INTTM1 acknowledgment.

When reading 16-bit timer (TMS) count value, use the 16-bit memory manipulation instruction.

(3) Square-wave output operation

A square wave with any selected frequency is output at intervals of the 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 bit 4 (TOE2) of the 8-bit timer output control register (TOC1) to 1. This enables a square wave with any selected frequency to be output.

★ **Table 7-10 Square-Wave Output Ranges when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter**

Minimum pulse time	Maximum pulse time	Resolution
2 × 1/fxx (500 ns)	2^{17} × 1/fxx (32.8 ms)	2 × 1/fxx (500 ns)
2^2 × 1/fxx (1.0 μs)	2^{18} × 1/fxx (65.5 ms)	2^2 × 1/fxx (1.0 μs)
2^3 × 1/fxx (2.0 μs)	2^{19} × 1/fxx (131.1 ms)	2^3 × 1/fxx (2.0 μs)
2^4 × 1/fxx (4.0 μs)	2^{20} × 1/fxx (262.1 ms)	2^4 × 1/fxx (4.0 μs)
2^5 × 1/fxx (8.0 μs)	2^{21} × 1/fxx (524.3 ms)	2^5 × 1/fxx (8.0 μs)
2^6 × 1/fxx (16.0 μs)	2^{22} × 1/fxx (1.0 s)	2^6 × 1/fxx (16.0 μs)
2^7 × 1/fxx (32.0 μs)	2^{23} × 1/fxx (2.1 s)	2^7 × 1/fxx (32.0 μs)
2^8 × 1/fxx (64.0 μs)	2^{24} × 1/fxx (4.2 s)	2^8 × 1/fxx (64.0 μs)
2^9 × 1/fxx (128.0 μs)	2^{25} × 1/fxx (8.4 s)	2^9 × 1/fxx (128.0 μs)
2^{11} × 1/fxx (512.0 μs)	2^{27} × 1/fxx (33.6 s)	2^{11} × 1/fxx (512.0 μs)

Remarks:

(1) fxx: Main system clock frequency

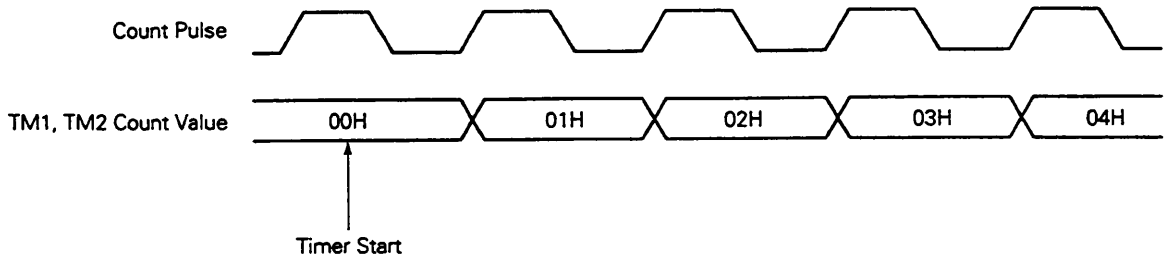
(2) Values in parentheses when operated at fxx = 4.0 MHz.

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 Timings



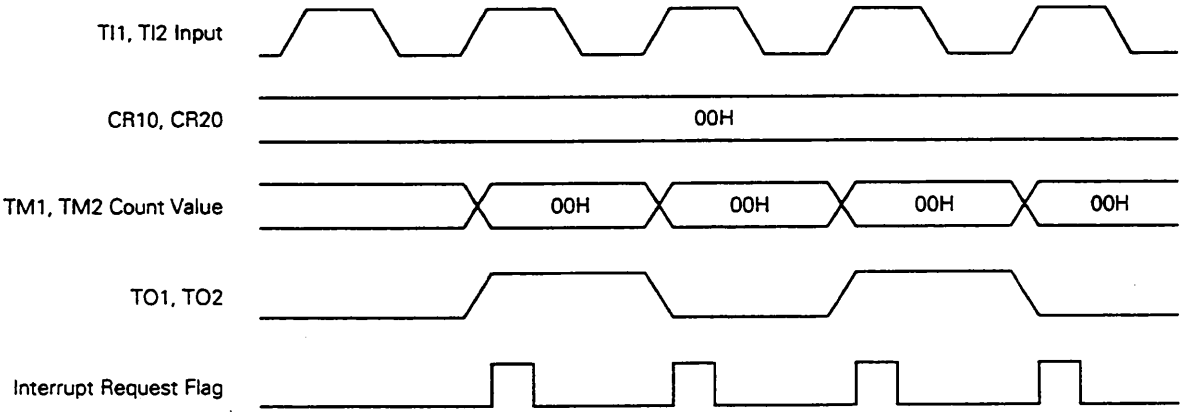
(2) 8-bit compare registers 1 and 2 sets

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 after setting bit 0 (TCE1) of the 8-bit timer mode control register to 0 and stopping timer operation.

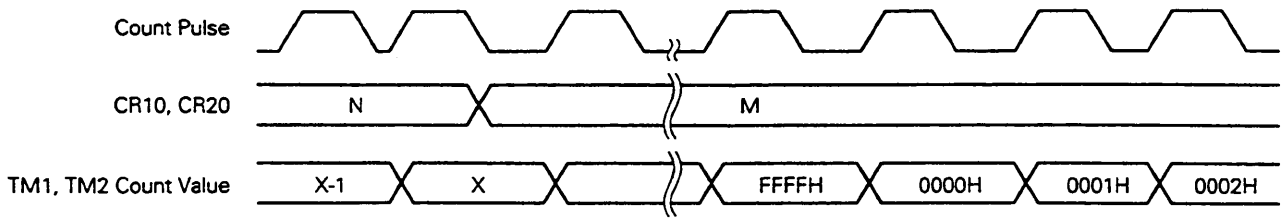
Fig. 7-13 External Event Counter Operation Timings



(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 restart counting from 0. Thus, if the value after CR10 and CR20 (M) change is smaller than that before change (N), it is necessary to restart the timer after changing CR10 and CR20.

Fig. 7-14 Timings 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 the interval timer can be used simultaneously.

(1) Watch timer

When the 32.768 kHz subsystem clock is used, a flag (WTIF) is set at 0.5 second or 0.25 second intervals.

Caution: 0.5 second intervals cannot be generated with the 6.0 MHz main system clock. You should switch to the 32.768 kHz subsystem clock to generate 0.5 second intervals.

(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 f_{XX} = 4.0 MHz	When operated at f_{XT} = 32.768 kHz
$2^4 \times 1/fw$	512 μs	488 μs
$2^5 \times 1/fw$	1.02 ms	977 μs
$2^6 \times 1/fw$	2.05 ms	1.95 ms
$2^7 \times 1/fw$	4.10 ms	3.91 ms
$2^8 \times 1/fw$	8.19 ms	7.81 ms
$2^9 \times 1/fw$	16.4 ms	15.6 ms

Remarks:

(1) f_{XX} : Main system clock frequency

(2) f_{XT} : Subsystem clock oscillation frequency

(3) fw : Watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})

8.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 8-2 Watch Timer Configuration

Item	Configuration
Counter	5 bits x 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 manipulation instruction.

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

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

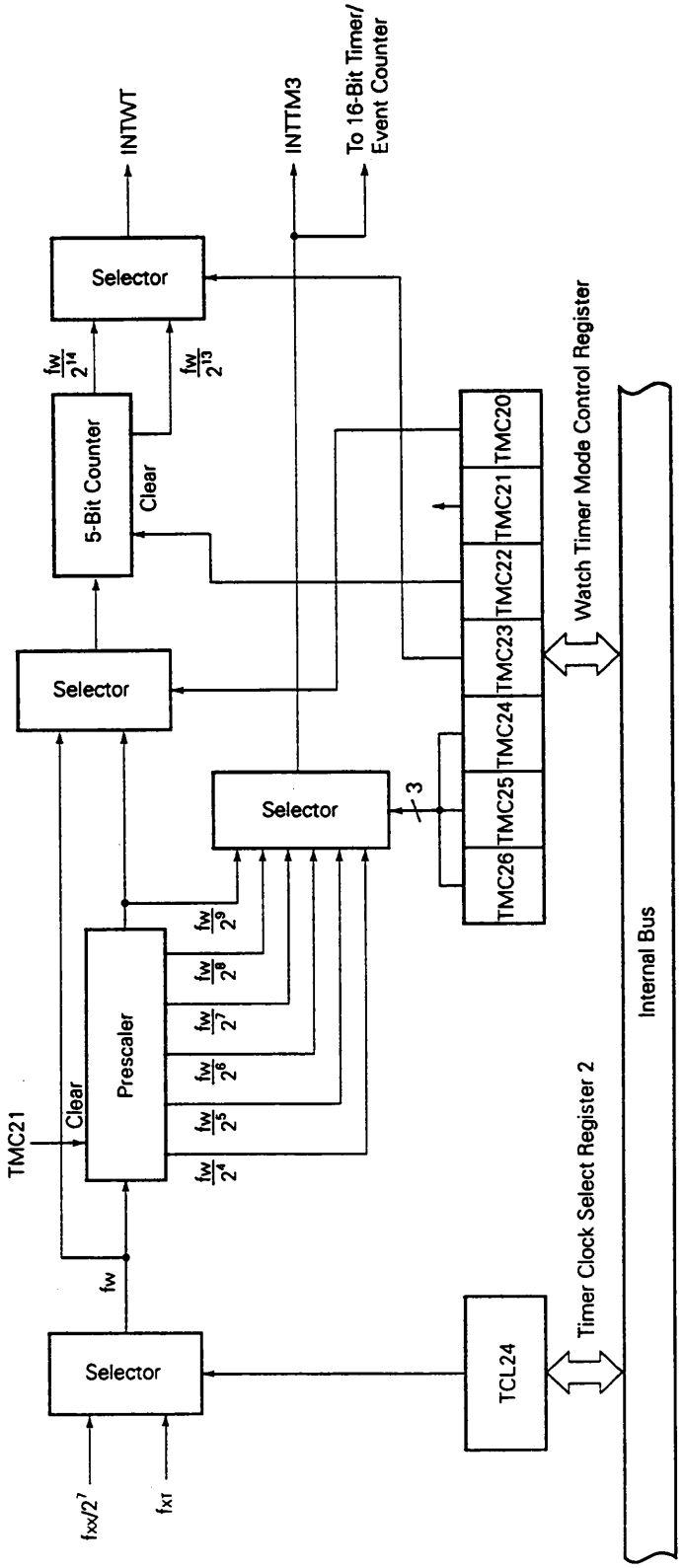


Fig. 8-1 Watch Timer Block Diagram

Fig. 8-2 Timer Clock Select Register 2 Format

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

TCL22	TCL21	TCL20	Watchdog Timer Count Clock Selection
0	0	0	$f_{XX}/2^3$ (500 kHz)
0	0	1	$f_{XX}/2^4$ (250 kHz)
0	1	0	$f_{XX}/2^5$ (125 kHz)
0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)

TCL24	Watch Timer Count Clock Selection
0	$f_{XX}/2^7$ (31.3 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disable
1	0	0	$f_{XX}/2^9$ (7.8 kHz)
1	0	1	$f_{XX}/2^{10}$ (3.9 kHz)
1	1	0	$f_{XX}/2^{11}$ (1.95 kHz)
1	1	1	Setting prohibited

Remarks:

(1) f_{XX} : Main system clock frequency

(2) f_{XT} : Subsystem clock oscillation frequency

(3) × : Don't care

(4) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz or f_{XT} = 32.768 kHz.

★ **Caution: When rewriting TCL2 to data other than the same data, stop the timer operation first.**

(2) Watch timer mode control register (TMC2)

This register sets the watch timer operating mode, watch flag set time and prescaler interval time and enables/disables prescaler and 5-bit counter operations. TMC2 is set with a 1-bit or 8-bit memory manipulation 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	When 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
0	Clear after operation stops
1	Operation enable

TMC22	5-Bit Counter Operation Control
0	Clear after operation stops
1	Operation enable

TMC26	TMC25	TMC24	Prescaler Interval Time Selection
0	0	0	2^{4}/fw (488 μs)
0	0	1	2^{5}/fw (977 μs)
0	1	0	2^{6}/fw (1.95 ms)
0	1	1	2^{7}/fw (3.91 ms)
1	0	0	2^{8}/fw (7.81 ms)
1	0	1	2^{9}/fw (15.6 ms)
Other than above			Setting prohibited

Caution: Do not frequently clear the prescaler when using the watch timer.

Remarks:

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

(2) Figures in ( ) applies when fw = 32.768 kHz.

8.4 Watch Timer Operations

8.4.1 Watch timer operation

When the 32.768 kHz subsystem clock is used, the timer operates as a watch timer with a 0.5 second or 0.25 second interval.

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: 32.8 ms when operated at fxx = 4.0 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 fxx = 4.0 MHz	When operated at fxT = 32.768 kHz
0	0	0	2^4 x 1/fw	512 μs	488 μs
0	0	1	2^5 x 1/fw	1.02 ms	977 μs
0	1	0	2^6 x 1/fw	2.05 ms	1.95 ms
0	1	1	2^7 x 1/fw	4.10 ms	3.91 ms
1	0	0	2^8 x 1/fw	8.19 ms	7.81 ms
1	0	1	2^9 x 1/fw	16.4 ms	15.6 ms
Other than above			Setting prohibited		

Remarks:

(1) fxx : Main system clock frequency

(2) fxT : Subsystem clock oscillation frequency

(3) fw : Watch timer clock frequency (fxx/2^7 or fxT)

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.

(2) Interval timer mode

Interrupts are generated at the preset time intervals.

9.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 9-1 Watchdog Timer Configuration

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

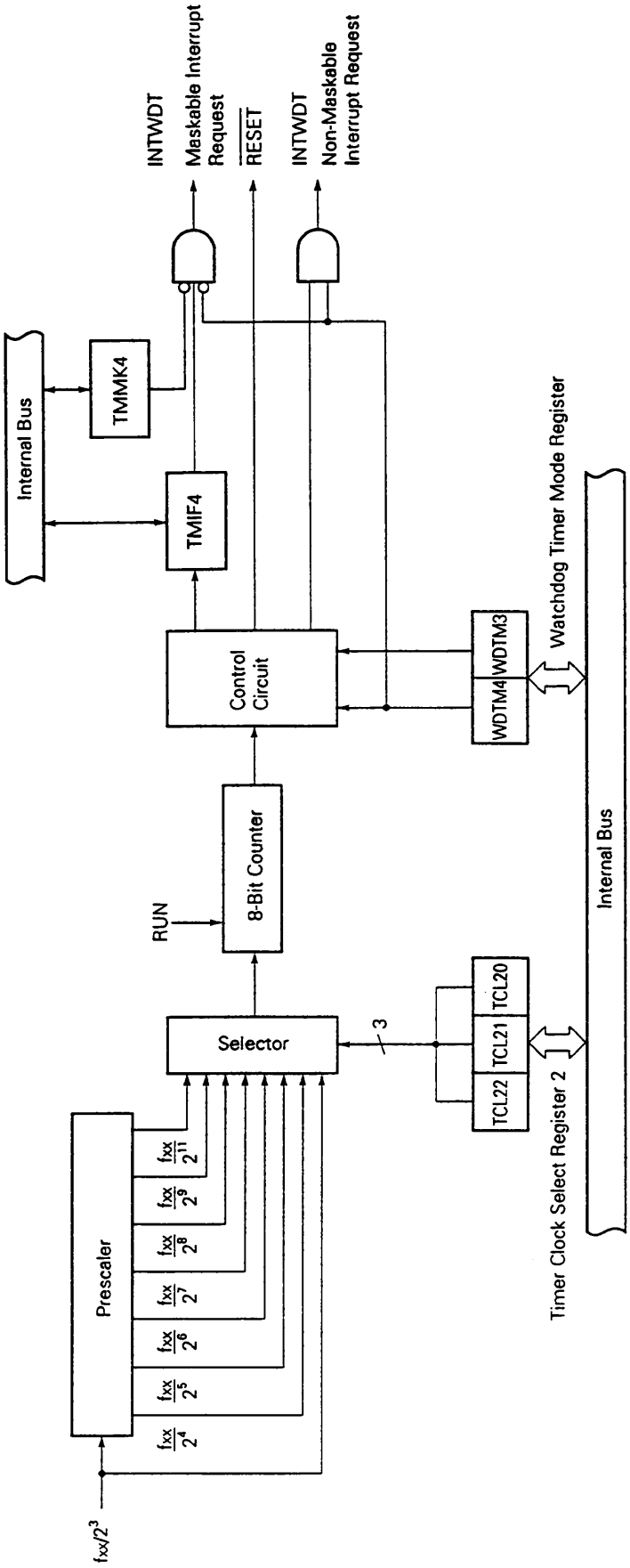


Fig. 9-1 Watchdog Timer Block Diagram

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 an 8-bit memory manipulation instruction.

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

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

Fig. 9-2 Timer Clock Select Register 2 Format

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

TCL22	TCL21	TCL20	Watchdog Timer Count Clock Selection
0	0	0	$f_{XX}/2^3$ (500 kHz)
0	0	1	$f_{XX}/2^4$ (250 kHz)
0	1	0	$f_{XX}/2^5$ (125 kHz)
0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)

TCL24	Watch Timer Count Clock Selection
0	$f_{XX}/2^7$ (31.3 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disable
1	0	0	$f_{XX}/2^9$ (7.8 kHz)
1	0	1	$f_{XX}/2^{10}$ (3.9 kHz)
1	1	0	$f_{XX}/2^{11}$ (1.95 kHz)
1	1	1	Setting prohibited

Remarks

(1) f_{XX} : Main system clock frequency

(2) f_{XT} : Subsystem clock oscillation frequency

(3) × : Don't care

(4) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz or f_{XT} = 32.768 kHz.

★ **Caution: When rewriting TCL2 to data other than the same data, stop the timer operation first.**

(2) Watchdog timer mode register (WDTM)

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

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

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

Fig. 9-3 Watchdog Timer Mode Register Format

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

WDTM4	WDTM3	Watchdog Timer Operating Mode Selection[Note1]
0	0	Operation stop
0	1	Interval timer Mode (Maskable interrupt occurs upon generation of an overflow.)
1	0	Watchdog timer mode 1 (Non-maskable interrupt occurs upon generation of an overflow.)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow.)

RUN	Watchdog Timer Operation Selection[Note2]
0	Count stop
1	Counter is 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 1 is set in RUN so that watchdog timer is cleared, the actual overflow time is up to 0.5% shorter than the time set by timer clock select 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).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set overrun time interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the inadvertent program loop detection time is past, system reset or a non-maskable interrupt is generated according to the WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions:

(1) The actual overrun detection time may be shorter than the set time by a maximum of 0.5%.

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

Table 9-2 Watchdog Timer Overrun Detection Time

TCL22	TCL21	TCL20	Runaway Detection Time
0	0	0	2^{11} x 1/fxx (512 μs)
0	0	1	2^{12} x 1/fxx (1.02 ms)
0	1	0	2^{13} x 1/fxx (2.05 ms)
0	1	1	2^{14} x 1/fxx (4.10 ms)
1	0	0	2^{15} x 1/fxx (8.19 ms)
1	0	1	2^{16} x 1/fxx (16.4 ms)
1	1	0	2^{17} x 1/fxx (32.8 ms)
1	1	1	2^{19} x 1/fxx (131.1 ms)

Remarks:

(1) fxx: Main system clock frequency

(2) Figures in parentheses apply to operation with fxx = 4.0 MHz.

9.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupts repeatedly at intervals of a preset count value when bit 3 (WDTM3) of the watchdog timer mode register (WDTM) is set to 1 and bit 4 (WDTM4) is cleared to 0.

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specify flag (TMPR4) are validated 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 9-3 Interval Timer Interval Time

TCL22	TCL21	TCL20	Interval Time
0	0	0	2^{11} x 1/fxx (512 μs)
0	0	1	2^{12} x 1/fxx (1.02 ms)
0	1	0	2^{13} x 1/fxx (2.05 ms)
0	1	1	2^{14} x 1/fxx (4.10 ms)
1	0	0	2^{15} x 1/fxx (8.19 ms)
1	0	1	2^{16} x 1/fxx (16.4 ms)
1	1	0	2^{17} x 1/fxx (32.8 ms)
1	1	1	2^{19} x 1/fxx (131.1 ms)

Remarks:

(1) fxx: Main system clock frequency

(2) Figures in parentheses apply to operation with fxx = 4.0 MHz.

[MEMO]

CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT

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.

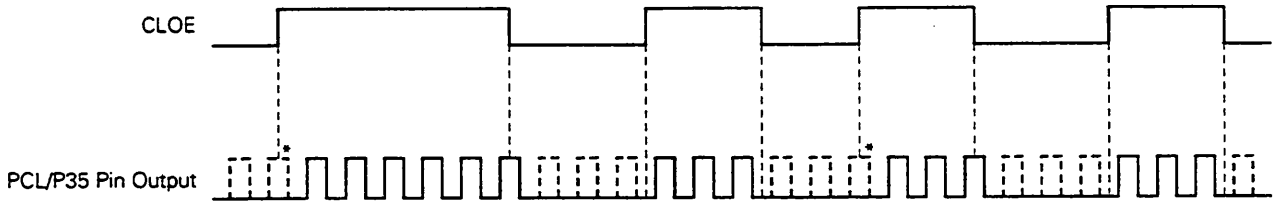
Follow the procedure below to output clock pulses.

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

Caution: Clock output cannot be used when setting P35 output latch to 1.

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

Fig. 10-1 Remote Controlled Output Application Example



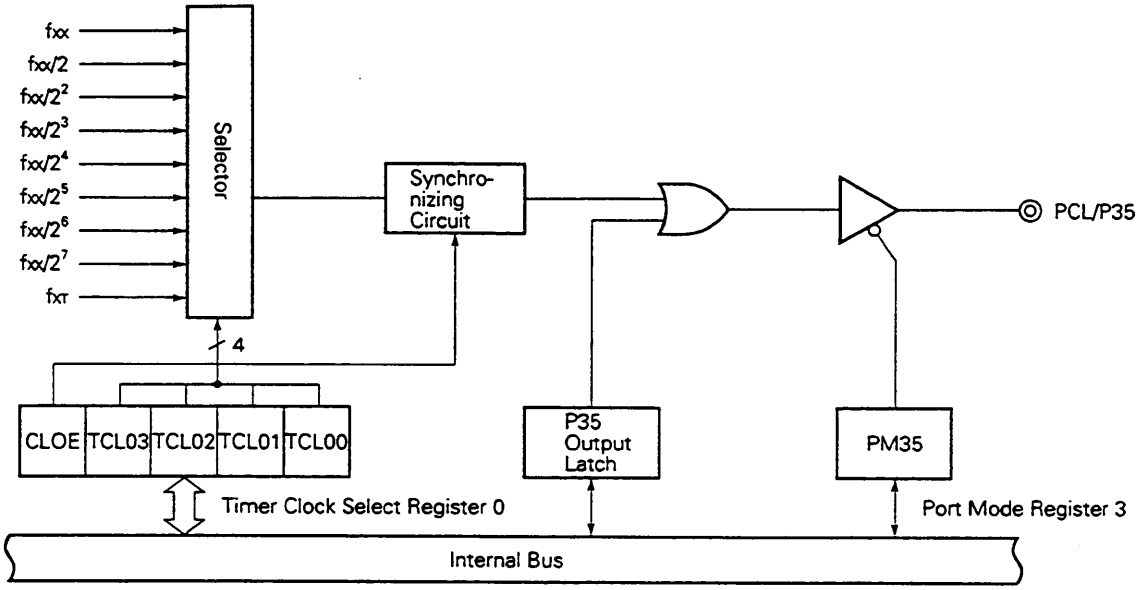
10.2 Clock Output Control Circuit Configuration

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 manipulation 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	When 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	0	1	f_{XX} (4.0 MHz)
0	1	1	0	$f_{XX}/2$ (2.0 MHz)
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
Other than above				Setting prohibited

TCL06	TCL05	TCL04	16-Bit Timer Register Count Clock Selection
0	0	0	TI00 (Valid edge specifiable)
0	0	1	$2f_{XX}$ Note
0	1	0	f_{XX} (4.0 MHz)
0	1	1	$f_{XX}/2$ (2.0 MHz)
1	0	0	$f_{XX}/2^2$ (1.0 MHz)
1	1	1	Watch Timer Output (INTTM3)
Other than above			Setting prohibited

CLOE	PCL Output Control
0	Output disable
1	Output enable

Note: Setting prohibited when fxx > 2.5 MHz.

Cautions:

(1) Setting of the TI00/INTP0 pin valid edge is performed by external interrupt mode register 0, and selection of the sampling clock frequency is performed by the sampling clock selection register.

(2) When enabling PCL output, set TCL00 to TCL03, then set 1 in CLOE with a 1-bit memory manipulation instruction.

(3) To read the count value when TI00 has been specified as the TM0 count clock, the value should be read from TM0, not from capture/compare register CR01.

(4) When rewriting TCL0 to data other than the same data, stop the timer operation first. ★

Remarks:

(1) fxx : Main system clock frequency

(2) fxT : Subsystem clock oscillation frequency

(3) TI00 : 16-bit timer/event counter input pin

(4) TM0 : 16-bit timer register

(5) Figures in parentheses apply to operation with fxx = 4.0 MHz or fxT = 32.768 kHz.

(2) Port mode register 3 (PM3)

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

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

PM3 is set with a 1-bit or 8-bit memory manipulation 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	When Reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

CHAPTER 11 BUZZER OUTPUT CONTROL CIRCUIT

11.1 Buzzer Output Control Circuit Functions

The buzzer output control circuit outputs 977 Hz, 1.95 kHz, 3.9 kHz, or 7.8 kHz frequency square-wave. 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 (Set to output mode).

Caution: Buzzer output cannot be used when setting P36 output latch to 1.

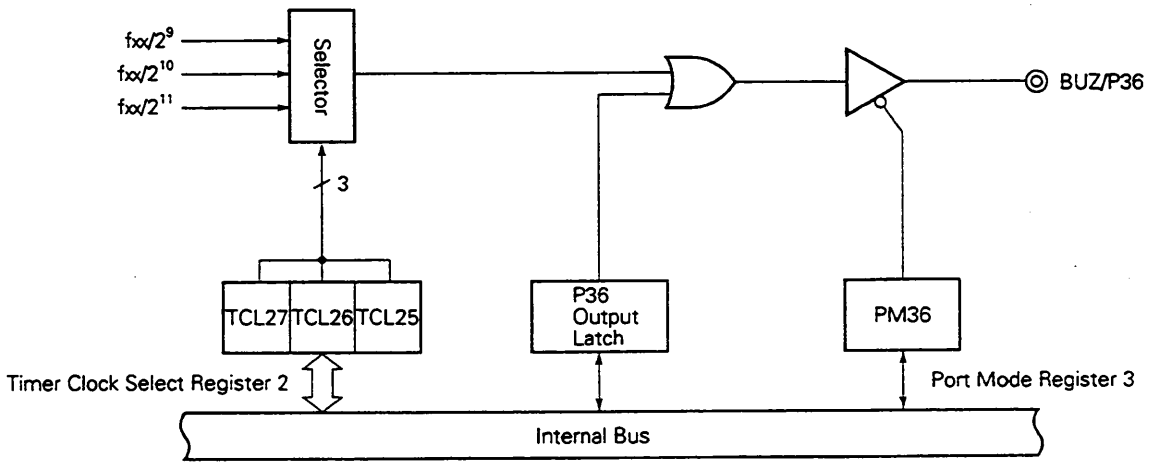
11.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 11-1 Buzzer Output Control Circuit Configuration

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 manipulation instruction.

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

Remark: Besides setting the buzzer output frequency, TCL2 sets the watch 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	When Reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Watchdog Timer Count Clock Selection
0	0	0	$f_{XX}/2^3$ (500 kHz)
0	0	1	$f_{XX}/2^4$ (250 kHz)
0	1	0	$f_{XX}/2^5$ (125 kHz)
0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)

TCL24	Watch Timer Count Clock Selection
0	$f_{XX}/2^7$ (31.3 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disable
1	0	0	$f_{XX}/2^9$ (7.8 kHz)
1	0	1	$f_{XX}/2^{10}$ (3.9 kHz)
1	1	0	$f_{XX}/2^{11}$ (1.95 kHz)
1	1	1	Setting prohibited

Remarks:

(1) f_{XX} : Main system clock frequency

(2) f_{XT} : Subsystem clock oscillation frequency

(3) × : Don't care

(4) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz or f_{XT} = 32.768 kHz.

Caution: When rewriting TCL2 to data other than the same data, stop the timer operation first. ★

(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 output latch of P36 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	When Reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n Pin Input/Output Mode Selection (n=0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

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 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 is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

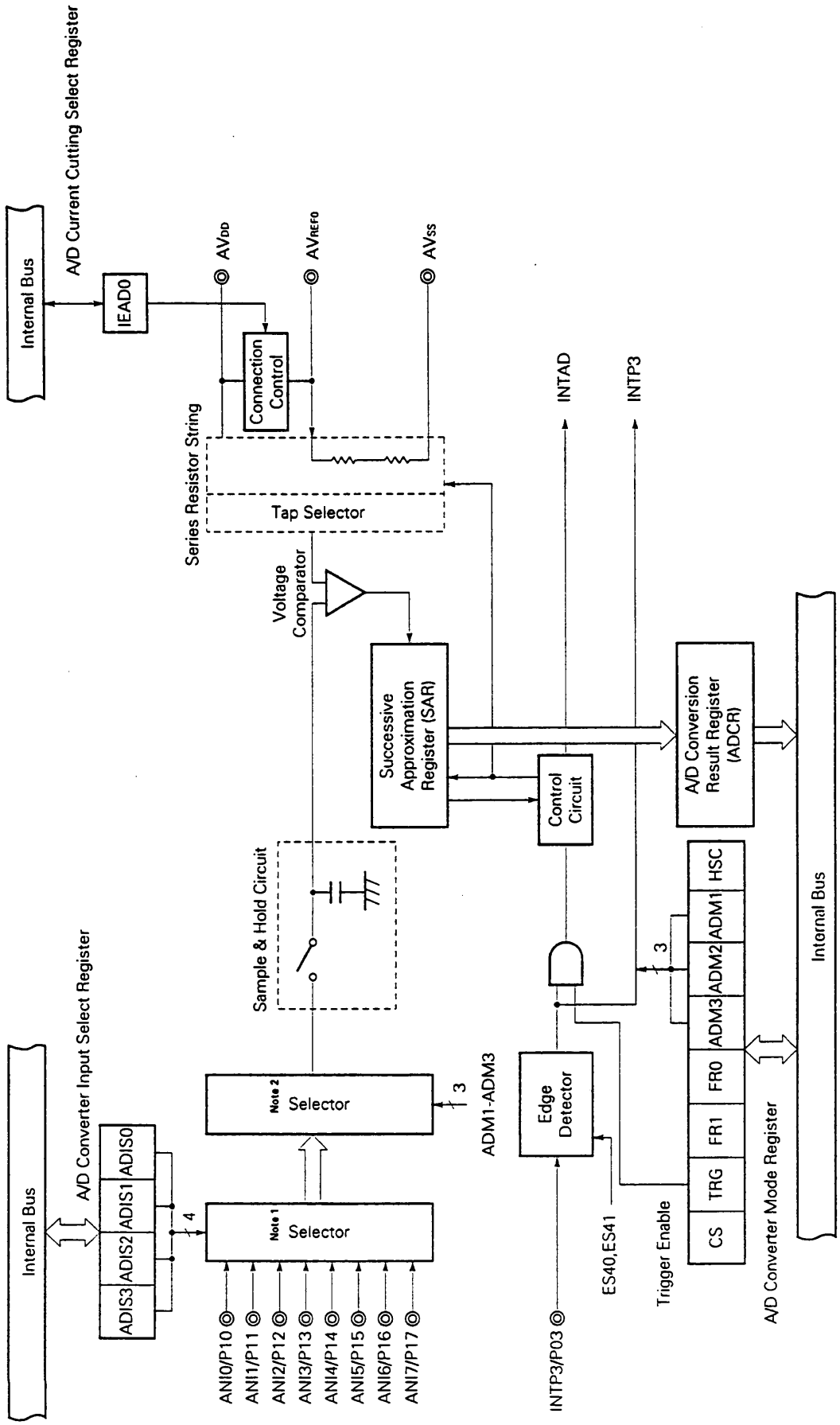
12.2 A/D Converter Configuration

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) External interrupt mode register 1 (INTM1) A/D current cutting select register (IEAD)
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 resistor 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 manipulation instruction.

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

(3) Sample & hold circuit

The sample & hold circuit samples each analog input sequentially applied from the input circuit and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is in AV_{REF0} to AVss and 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.

Pins other than those selected as analog input pins by the A/D converter input select register (ADIS) can be used as I/O port pins.

Caution: Use ANI0 to ANI7 input voltages within the specified range. If a voltage higher than AV_{REF0} 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_{REF0} 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_{REF0} and AVss.

The current flowing in the series resistor string can be reduced by setting the voltage to be input to the AV_{REF0} pin to AVss level in standby mode.

(8) AVss pin

Ground potential pin of the A/D converter. Always keep this pin at the same potential as the Vss pin even when the A/D converter is not used.

(9) AV_{DD} pin

Analog power supply pin of the A/D converter. Always keep this pin 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 four types of registers are used to control the A/D converter.

- A/D converter mode register (ADM)
- A/D converter input select register (ADIS)
- External interrupt mode register 1 (INTM1)
- A/D current cutting select register (IEAD)

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

This register sets the analog input channel for A/D conversion, conversion time, conversion start/stop and external trigger.

ADM is set with a 1-bit or 8-bit memory manipulation 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	When Reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	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	HSC	A/D Conversion Time Selection[Note1]
0	0	1	80/fxx (20.0 μs)
0	1	1	40/fxx (setting prohibited [Note2])
1	0	0	50/fxx (setting prohibited [Note2])
1	0	1	100/fxx (25.0 μs)
Other than above			Setting prohibited

TRG	External Trigger Selection
0	No external trigger (software starts)
1	Conversion started by external trigger (hardware starts)

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

Notes:

(1) Set so that the A/D conversion time is 19.1 μs or more.

(2) Setting prohibited because A/D conversion time is less than 19.1 μs.

Cautions:

(1) The CS bit should be cleared to 0 to stop the A/D conversion operation before executing a STOP instruction.

(2) When restarting the stopped A/D conversion operation, start the A/D conversion operation after clearing the interrupt request flag (ADIF) to 0.

Remarks:

(1) fxx: Main system clock frequency

(2) Figures in parentheses apply to operation with fxx = 4.0 MHz.

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

This register determines whether the ANI0/P10 to ANI7/P17 pins should be used for analog input channels or ports. Pins other than those selected as analog input pins 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 among the channels which is set for analog input with ADIS.

(2) No internal pull-up resistor can be connected to 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 Converter Input Select Register Format

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

ADIS3	ADIS2	ADIS1	ADIS0	Number of Analog Input Channel 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 ANI6, P17)
1	0	0	0	8 channels (ANI0 to ANI7)
Other than above				Setting prohibited

(3) External interrupt mode register 1 (INTM1)

This register sets the valid edge for INTP3 to INTP6.

INTM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM1 to 00H.

Fig. 12-4 External Interrupt Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
INTM1	ES71	ES70	ES61	ES60	ES51	ES50	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	INTP3 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES51	ES50	INTP4 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES61	ES60	INTP5 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES71	ES70	INTP6 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

(4) A/D current cutting select register (IEAD)

This register specifies whether AV_{DD} and AV_{REF0} are connected.

In a system where $AV_{DD} \fallingdotseq AV_{REF0}$ and high accuracy is not demanded, the power dissipation in the standby mode can be further reduced by leaving the AV_{REF0} pin open, connecting AV_{DD} and AV_{REF0} in the normal operation mode, and disconnecting them in the standby mode.

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

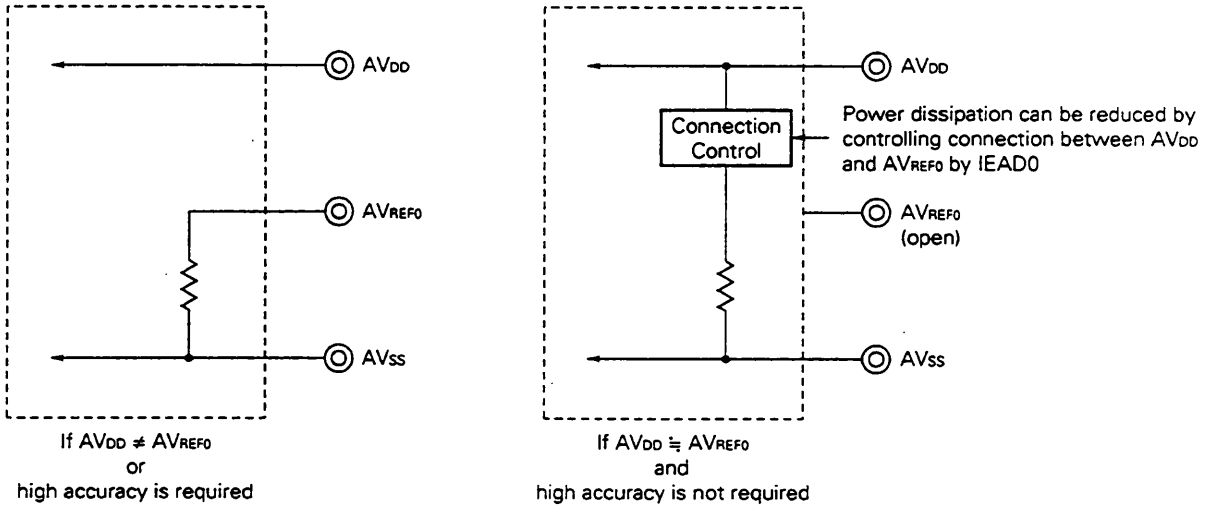
$\overline{RESET}$ input sets IEAD to 00H.

Fig. 12-5 A/D Current Cutting Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
IEAD	0	0	0	0	0	0	0	IEAD0	F8E2H	00H	R/W

IEAD0	Connection Control between AV_{DD} and AV_{REF0}
0	Disconnects AV_{DD} from AV_{REF0}
1	Connects AV_{DD} and AV_{REF0}

Fig. 12-6 Function of A/D Current Cutting Select Register



12.4 A/D Converter Operations

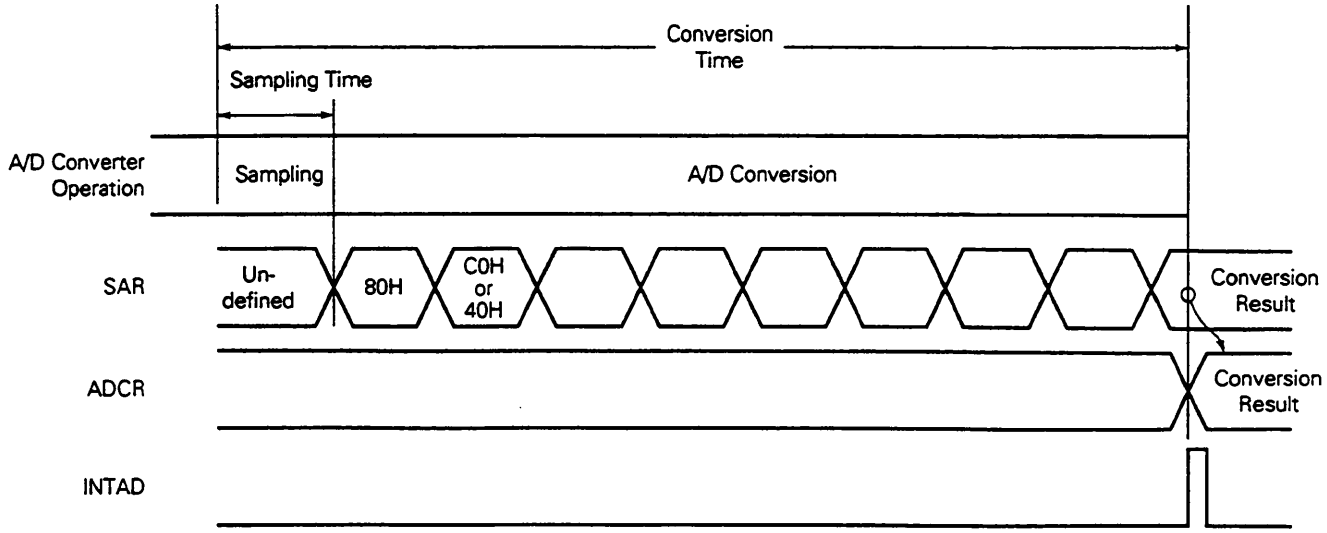
12.4.1 Basic operations of A/D converter

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 holds 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 resistor string voltage tap to (1/2) AV_{REF0}.
⑥ The voltage difference between the series resistor string voltage tap and analog input is compared with a voltage comparator. If the analog input is larger than (1/2) AV_{REF0}, the MSB of SAR remains set. If the input is smaller than (1/2) AV_{REF0}, 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 resistor string voltage tap is selected according to the preset value of bit 7 as described below.
- Bit 7 = 1: (3/4) AV_{REF0}
- Bit 7 = 0: (1/4) AV_{REF0}

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

Fig. 12-7 A/D Converter Basic Operation



A/D conversion operations are performed continuously until the CS bit is reset (0) by software.
If a write to the ADM register is performed during an A/D conversion operation, the conversion operation is initialized, and if the CS bit is set (1), conversion starts again from the beginning.
After $\overline{\text{RESET}}$ input, the value of ADCR is undefined.

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (the value stored in ADCR) is shown by the following expression.

$$ADCR = INT\left(\frac{V_{IN}}{AV_{REF0}} \times 256 + 0.5\right)$$

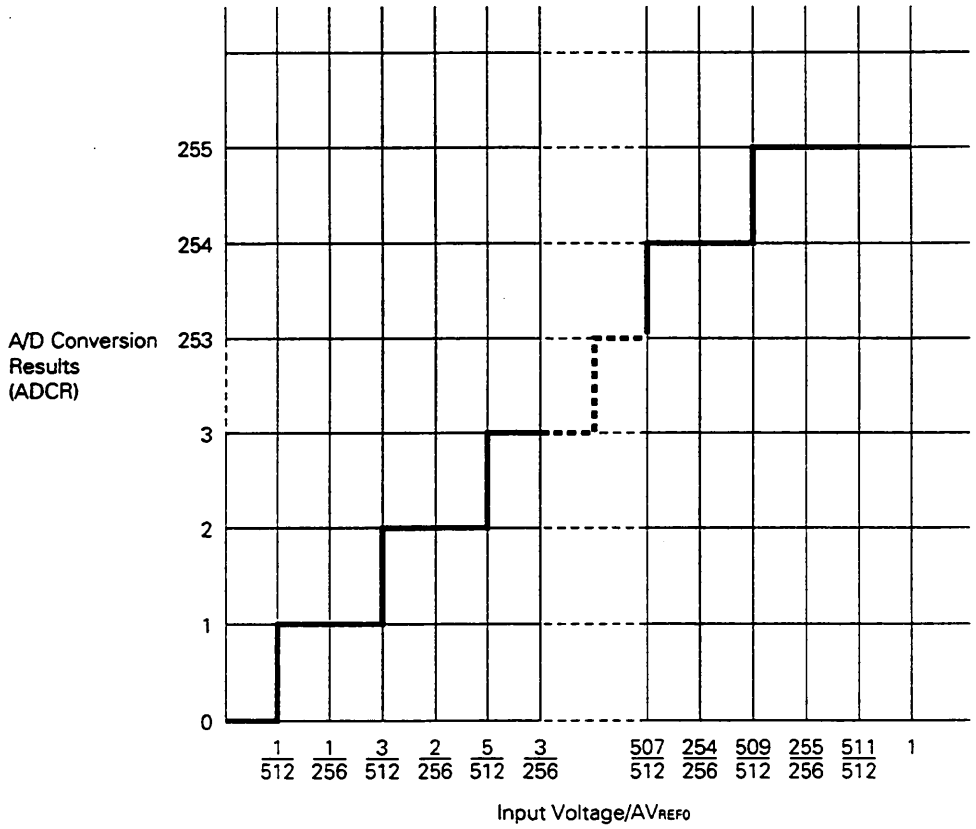
or

$$(ADCR - 0.5) \times \frac{AV_{REF0}}{256} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{REF0}}{256}$$

Remark: INT ( ) : Function which returns integer parts of value in parentheses.
V_{IN} : Analog input voltage
AV_{REF0} : AV_{REF0} pin voltage
ADCR : ADCR register value

Fig. 12-8 shows the relationship between the analog input voltage and the A/D conversion result.

Fig. 12-8 Relationship 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

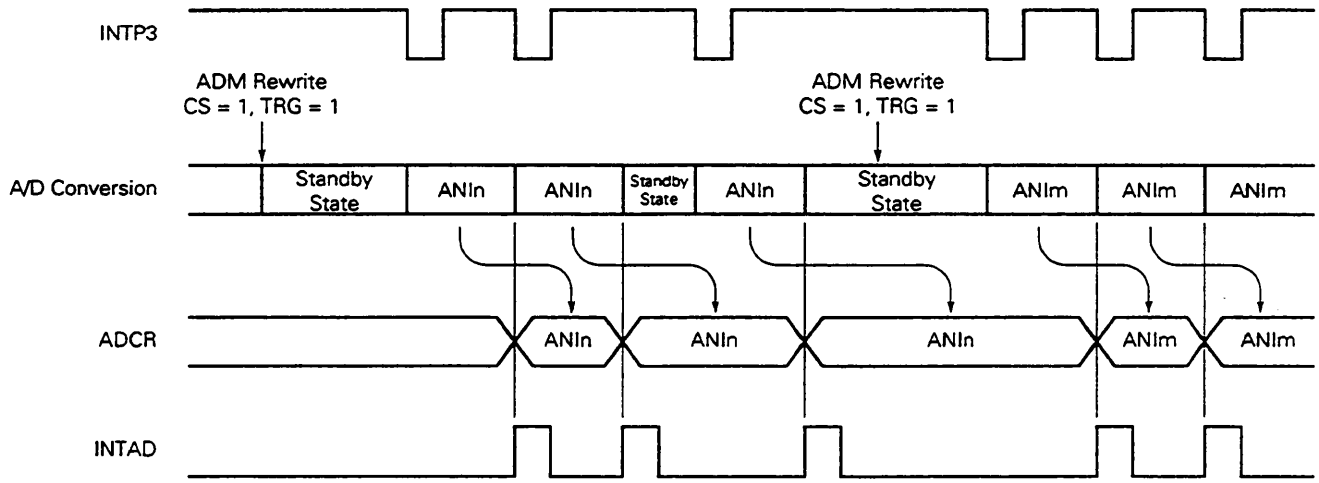
When bit 6 (TRG) and bit 7 (CS) of ADM are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is input, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

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 to ADM 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. 12-9 A/D Conversion by Hardware Start



Remarks:

(1) n = 0, 1, ... 7

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

(2) A/D conversion by software start

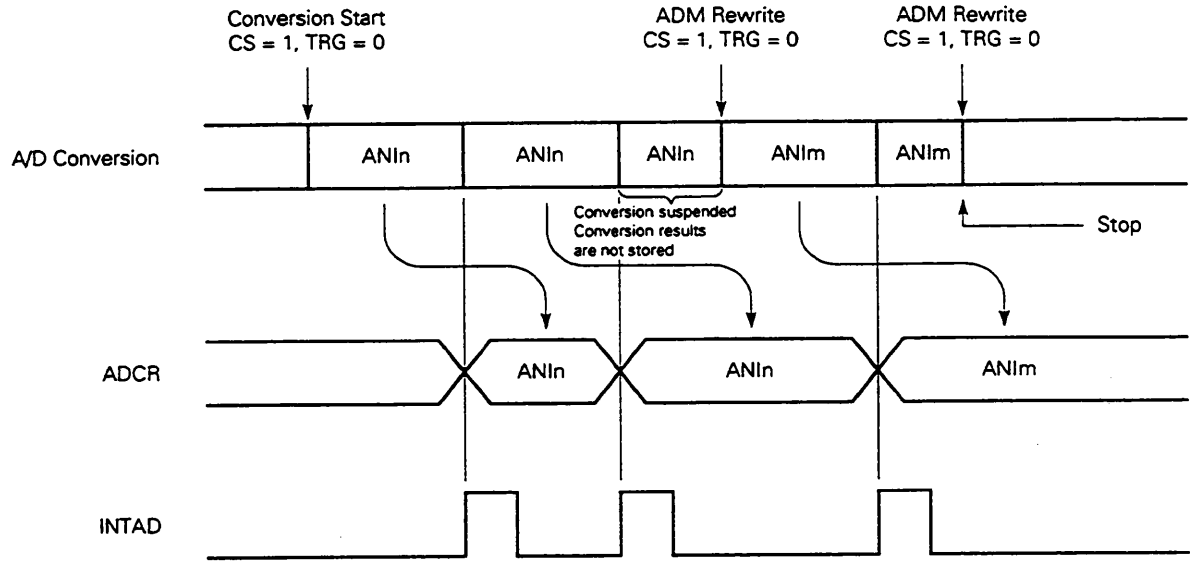
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

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 to ADM 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 ADM during A/D conversion, the A/D conversion operation stops immediately.

Fig. 12-10 A/D Conversion by Software Start



Remarks:

(1) n = 0, 1, ... 7

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

12.5 A/D Converter Precautions

(1) Current consumption in standby mode

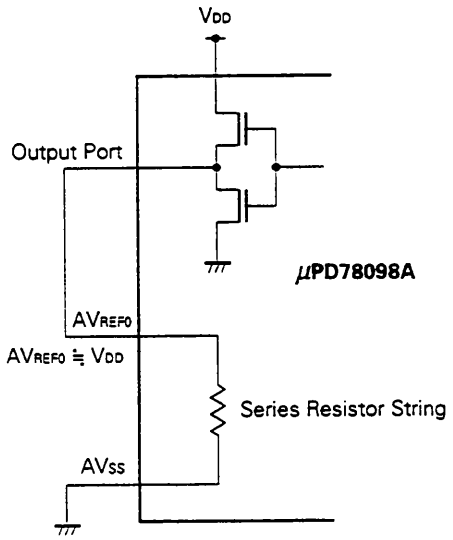
The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode or in HALT mode with the subsystem clock. As a current still flows in the AV_{REFO} pin at this time, this current must be cut in order to minimize the overall system power dissipation.

(a) To use port

Fig. 12-11 shows an example in which a port is used.

In this example, the power dissipation can be reduced if a low level is output to the output port in the standby mode. However, the actual AV_{REFO} voltage is not so accurate and, accordingly, the converted value is not accurate and should be used for relative comparison only.

Fig. 12-11 Example of Method of Reducing Power Dissipation in Standby Mode (when using port)



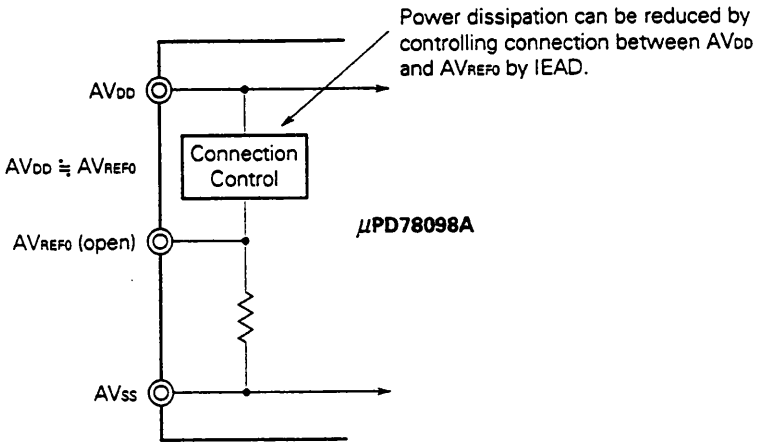
(b) Using A/D current cutting select register (IEAD)

Fig. 12-12 illustrates how the A/D current cutting select register (IEAD) is used.

In a system where $AV_{DD} \fallingdotseq AV_{REF0}$, the power dissipation in the standby mode can be further reduced by leaving the AV_{REF0} pin open, connecting AV_{DD} and AV_{REF0} (IEAD = 00H) in the normal operation mode, and disconnecting them (IEAD = 01H) in the standby mode.

In this case, however, the conversion accuracy is less than when a voltage is applied to the AV_{REF0} pin because the characteristics of the internal switch are reflected.

Fig. 12-12 Example of Method of Reducing Power Dissipation in Standby Mode (using A/D current cutting select register)



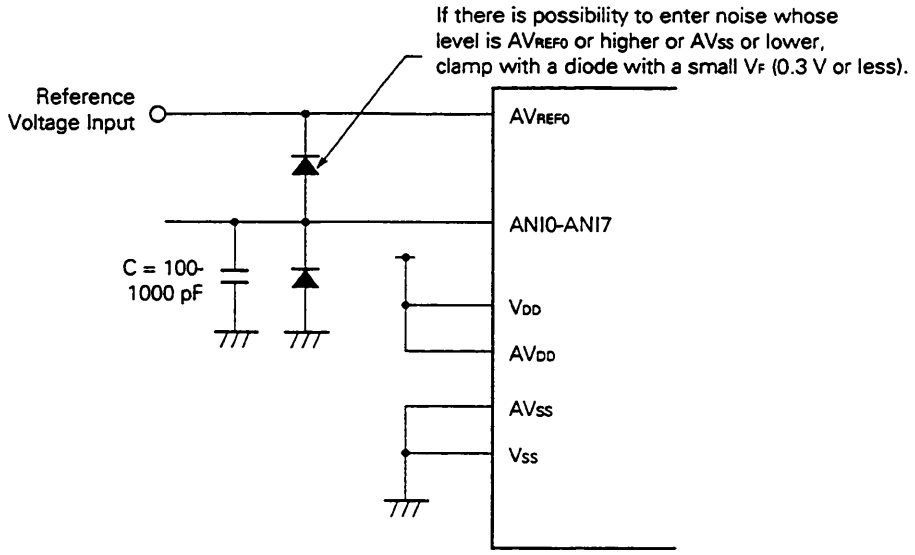
(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage above AV_{REF0} or below AVss is input (even if within the absolute maximum rating range), the conversion value for that channel will be indeterminate. 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 AV_{REF0} and ANI0 to ANI7. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor is connected externally as shown in Fig. 12-13 in order to reduce noise.

Fig. 12-13 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. 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) AV_{REF0} pin input impedance

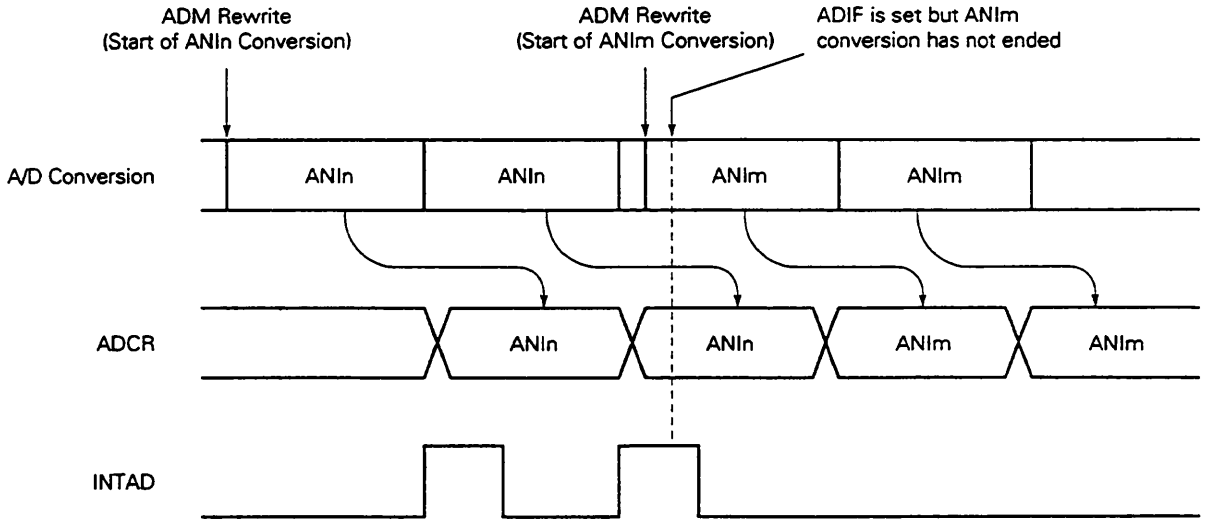
A series resistor string of approximately 10 kΩ is connected between the AV_{REF0} pin and the AVss pin. Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{REF0} pin and the AVss pin, and there will be a large 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. Caution is therefore required since, if a change of analog input pin is performed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADM rewrite, and when ADIF is read immediately after the ADM rewrite, ADIF may be set despite the fact that the A/D conversion for the post-change analog input has not ended.

To stop A/D conversion once and restart it again, clear the interrupt request flag (ADIF) before restarting.

Fig. 12-14 A/D Conversion End Interrupt Generation Timing



(7) AV_{DD} pin

The AV_{DD} pin is the analog circuit power supply pin, and supplies power to the input circuits of ANI0/P10 to ANI7/P17.

Therefore, connection to V_{DD} is necessary even if the A/D converter is not used.

[MEMO]

CHAPTER 13 D/A CONVERTER

13.1 D/A Converter Functions

The D/A converter converts a digital input into an analog value. It consists of two 8-bit resolution channels of voltage output type D/A converter.

The conversion method used is a R-2R resistor ladder method.

D/A conversion is started when DACE0 and DACE1 of the D/A converter mode register (DAM) are set to 1.

The D/A converter operates in the following two modes:

(1) Normal mode

In this mode, the D/A converter outputs an analog voltage immediately after conversion.

(2) Real-time output mode

The D/A converter outputs an analog voltage in synchronization with an output trigger after conversion. Because square wave can be generated in this mode, an MSK modem for cordless telephones can be easily realized in this mode.

Caution: When $AV_{REF} < V_{DD}$ and the D/A converter is used for only one channel, perform the one of the following procedures for pins not used as analog output. ★

- **Set the port mode register (PM13x) to 1 (input mode) and connect to Vss.**
- **Set the port mode register (PM13x) to 0 (output mode), set the output latch to 0, and output low level.**

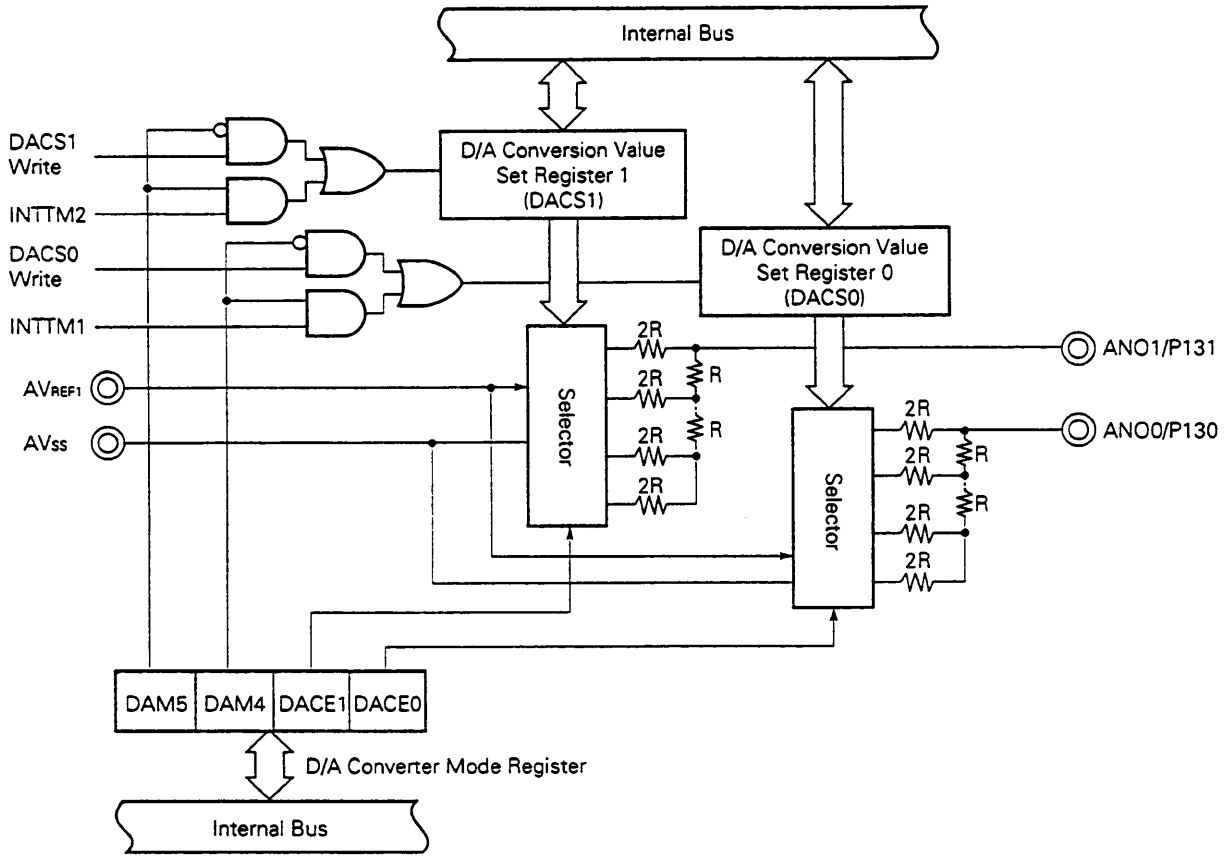
13.2 D/A Converter Configuration

The D/A converter consists of the following hardware.

Table 13-1 D/A Converter Configuration

Item	Configuration
Register	D/A conversion value set register 0 (DACS0) D/A conversion value set register 1 (DACS1)
Control register	D/A converter mode register (DAM)

★ **Fig. 13-1 D/A Converter Block Diagram**



(1) D/A conversion value set register 0, 1 (DACS0, DACS1)

DACS0 and DACS1 are registers that set analog voltage values output to the ANO0 and ANO1 pins, respectively.

DACS0 and DACS1 are set with 8-bit memory manipulation instructions.

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

Analog voltage output to the ANO0 and ANO1 pins is determined by the following expression.

$$\text{ANOn output voltage} = AV_{REF1} \times \frac{DACSn}{256}$$

Remark: n = 0, 1

Cautions:

(1) In the real-time output mode, when data that are set in DACS0 and DACS1 are read before an output trigger is generated, the previous data are read rather than the set data.

(2) In the real-time output mode, data should be set to DACS0 and DACS1 after an output trigger and before the next output trigger.

13.3 D/A Converter Control Registers

The D/A converter mode register (DAM) controls the D/A converter. This register sets D/A converter operation enable/stop.

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

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

Fig. 13-2 D/A Converter Mode Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	When Reset	R/W
DAM	0	0	DAM5	DAM4	0	0	DACE1	DACE0	FF98H	00H	R/W

DACE0	D/A Converter Channel 0 Control
0	D/A conversion stop
1	D/A conversion enable

DACE1	D/A Converter Channel 1 Control
0	D/A conversion stop
1	D/A conversion enable

DAM4	D/A Converter Channel 0 Operating Mode
0	Normal mode
1	Real-time output mode

DAM5	D/A Converter Channel 1 Operating Mode
0	Normal mode
1	Real-time output mode

Cautions:

(1) When using the D/A converter, a dual-function port pin should be set to the input mode, and a pull-up resistor should be disconnected.

(2) Always set bits 2, 3, 6, and 7 to 0.

(3) When D/A conversion is stopped, the output state is high-impedance.

(4) The output trigger in the real-time output mode is INTTM1 for channel 0 and INTTM2 for channel 1.

13.4 D/A Converter Operation

① Select an operation mode for channel 0 by DAM4 of the D/A converter mode register (DAM) and a mode for channel 1 by DAM5.

② Set data corresponding to the analog voltage values to be output to ANO0/P130 and ANO1/P131 pins to D/A conversion value setting registers 0 and 1 (DACS0 and DACS1).

③ By setting DACE0 and DACE1 of DAM, D/A conversion of channels 0 and 1 starts.

④ In the normal mode, analog voltages are output to the ANO0/P130 and ANO1/P131 pins immediately after conversion. In the real-time output mode, analog voltages are output in synchronization with the output trigger.

⑤ The analog voltage values to be output are held by DACS0 and DACS1 in the normal mode until new data is set. In the real-time output mode, they are held until the next output trigger is generated after new data have been set to DACS0 and DACS1.

Caution: Before setting DACE0 and DACE1, set data to DACS0 and DACS1.

CHAPTER 14 SERIAL INTERFACE CHANNEL 0

The μPD78098 sub-series incorporates three channels of serial interfaces. Differences between channels 0, 1, and 2 are as follows (Refer to **CHAPTER 15 SERIAL INTERFACE CHANNEL 1** for details of the serial interface channel 1. Refer to **CHAPTER 16 SERIAL INTERFACE CHANNEL 2** for details of the serial interface channel 2) .

Table 14-1 Differences between Channels 0, 1, and 2

Serial Transfer Mode		Channel 0	Channel 1	Channel 2
3-wire serial I/O	Clock selection	$f_{XX}/2^{Note}$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$, external clock, TO2 output	$f_{XX}2^{Note}$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$, external clock, TO2 output	External clock, baud rate generator output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit Automatic transmit/ receive function	MSB/LSB switchable as the start bit
	Transfer end flag	Serial transfer end interrupt request flag (INTCSI0)	Serial transfer end interrupt request flag (INTCSI1)	Serial transfer end interrupt request flag (INTCSI2)
SBI (serial bus interface)		Use possible	None	None
2-wire serial I/O				
UART (Asynchronous serial interface)		None		Use possible

Note: Can be set only when the main system clock frequency is 5.0 MHz or less.

14.1 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 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 ($\overline{SCK0}$), 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 clocked 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 serial interface control portions to be simplified.

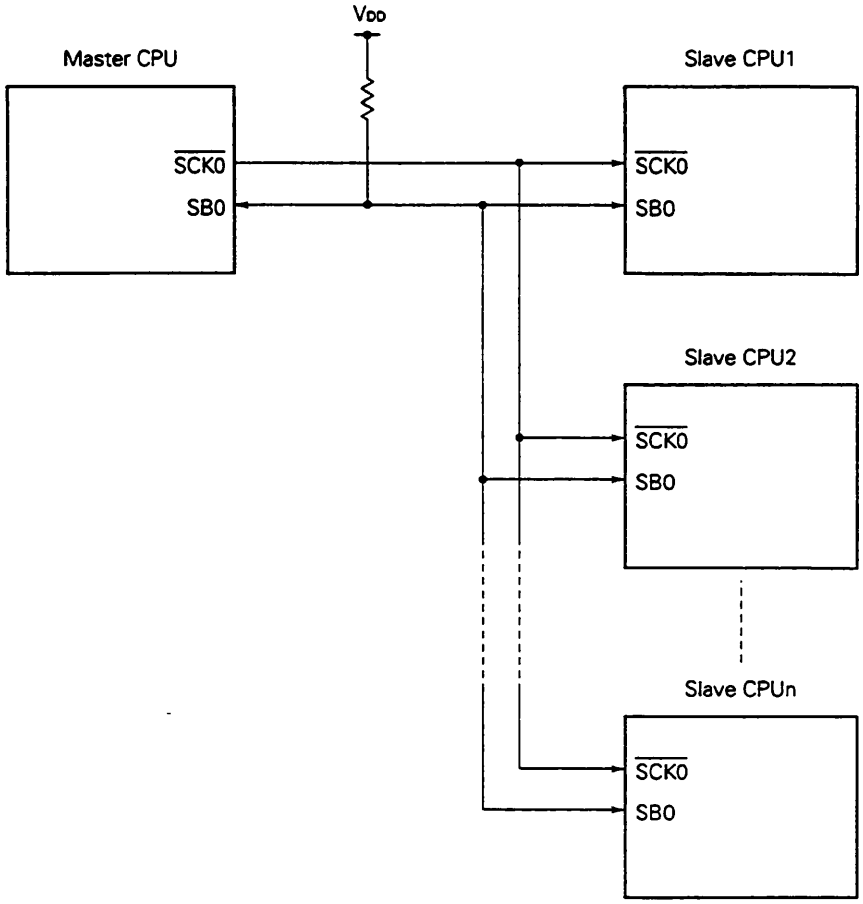
In this mode, the wake-up function for handshake and the output function of acknowledge signal and busy signal can also be used.

(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 by software. Thus, the handshake lines previously necessary for connection of two or more devices can be removed and the input/output ports can be used effectively.

Fig. 14-1 Serial Bus Interface (SBI) System Configuration Example



14.2 Serial Interface Channel 0 Configuration

Serial interface channel 0 consists of the following hardware.

Table 14-2 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) Port mode register 2 (PM2)

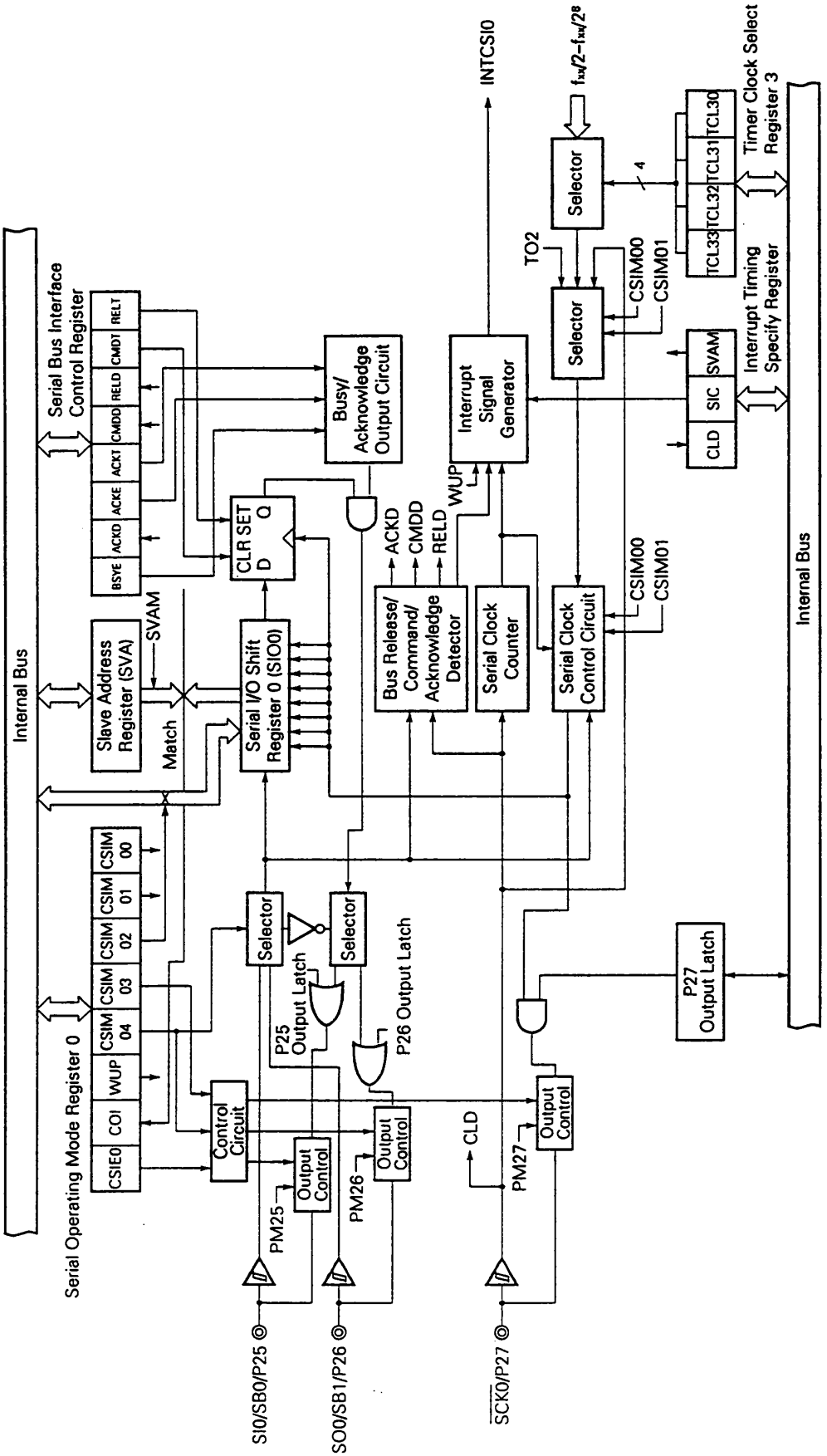


Fig. 14-2 Serial Interface Channel 0 Block Diagram

Remark: Output Control performs selection between CMOS output and 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 manipulation 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 (SI0) 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 manipulation 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 $\overline{\text{SCK0}}$/P27 pin.

(6) Interrupt signal generator

This circuit controls 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 operating mode register 0 (CSIM0).

(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.

14.3 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 (CSIM0)
- 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 manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Fig. 14-3 Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	When 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	$f_{XX}/2$ Note
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
Other than above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection
0	1	1	0	$f_{XX}/2$ Note
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
Other than above				Setting prohibited

Note: Can only be set when the main system clock frequency is 5.0 MHz or less.

Remarks:

(1) fxx: Main system clock frequency

(2) Figures in parentheses apply to operation with fxx = 4.0 MHz.

Caution: When rewriting TCL3 to data other than the same data, stop the timer operation first. ★

(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 manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

★ **Fig. 14-4 Serial Operating Mode Register 0 Format (1/2)**

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	When 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	x	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$/P27 Pin Function
	0	x	0	Note2 1	Note2 x	0	0	0	1	3-wire serial I/O mode	MSB	SI0 Note2 (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1								LSB			
	1	0	0	Note3 x	Note3 x	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1	0	0	Note3 x	Note3 x	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/ output)	
	1	1	0	Note3 x	Note3 x	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (N-ch open-drain input/ output)
			1	0	0	Note3 x	Note3 x	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

Notes:

(1) Bit 6 (COI) is a Read-Only bit.

(2) Can be used as P25 (CMOS input/output) when used only for transmission.

(3) Can be used freely as port function.

Remark: x: Don't care

Fig. 14-4 Serial Operating Mode Register 0 Format (2/2)

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 in SBI mode

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 Channel 0 Operation Control
	0	Operation stopped
	1	Operation enable

Note: When CSIE0 = 0, COI becomes 0.

(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 manipulation instruction.

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

Fig. 14-5 Serial Bus Interface Control Register Format (1/2)

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W Note 1

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 cleared to (0). After SO latch clearance, 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 clock of $\overline{\text{SCK0}}$ just after execution of the instruction to be set to 1, and after acknowledge signal output, automatically cleared to 0. Used as ACKE=0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

(Cont'd)

Note: Bits 2, 3 and 6 (RELD, CMDD and ACKD) are Read-Only bits.

Remark: Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when read after data setting.

Fig. 14-5 Serial Bus Interface Control Register Format (2/2)

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 9th clock falling edge of $\overline{SCK0}$ (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{SCK0}$ 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)
• At the falling edge of $\overline{SCK0}$ immediately after the busy mode has been released when a transfer start instruction is executed • 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

BSYE[Note]	Synchronizing Busy Signal Output Control
0	Disables busy signal which is output in synchronization with the falling edge of $\overline{SCK0}$ clock just after execution of the instruction to be cleared to 0.
1	Outputs busy signal at the falling edge of $\overline{SCK0}$ clock following the acknowledge signal.

Note: Busy mode can be cleared by start of serial interface transfer or reception of address signal. However, 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 manipulation instruction.

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

Fig. 14-6 Interrupt Timing Specify Register Format

Symbol	7	⑥	⑤	④	3	2	1	0	Address	When 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 Factor Selection Note 2
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ Pin Level Note 3
0	Low Level
1	High Level

Caution: Be sure to set bits 0 to 3 to 0.

Notes:

(1) Bit 6 (CLD) is a Read-Only bit.

(2) When using wake-up function, set SIC to 0.

(3) When CSIE0 = 0, CLD becomes 0.

Remark: SVA: Slave address register

14.4 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

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 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 setting

The operation stop mode is set with the serial operating mode register 0 (CSIM0).

CSIM0 is set with a 1-bit or 8-bit memory manipulation 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	When 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 stopped
1	Operation enable

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 clocked 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 setting

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 manipulation 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	When Reset	R/W
CSIM0	CSIE0	COI							FF60H	00H	R/W Note 1

R/W	CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
	0	×	Input clock to $\overline{SCK0}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{SCK0}$/P27 Pin Function
	0	×	0	1 Note2	× Note2	0	0	0	1	3-wire serial I/O mode	MSB	SI0 (Input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS input/output)
			1								LSB			
	1	0	SBI mode (refer to 14.4.3 SBI mode operation)											
	1	1	2-wire serial I/O mode (refer to 14.4.4 2-wire serial I/O mode operation)											

R/W	WUP	Wake-up Function Control Note3
	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 in SBI mod

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enable

Notes:

(1) Bit 6 (COI) is a Read-Only bit.

(2) Can be used as P25 (CMOS input/output) when used only for transmission.

(3) Always set 0 to WUP when using the 3-wire serial I/O mode.

Remark: ×: Don't care

(b) Serial bus interface control register (SBIC)

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

$\overline{RESET}$ 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	When 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 cleared to 0. After SO latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

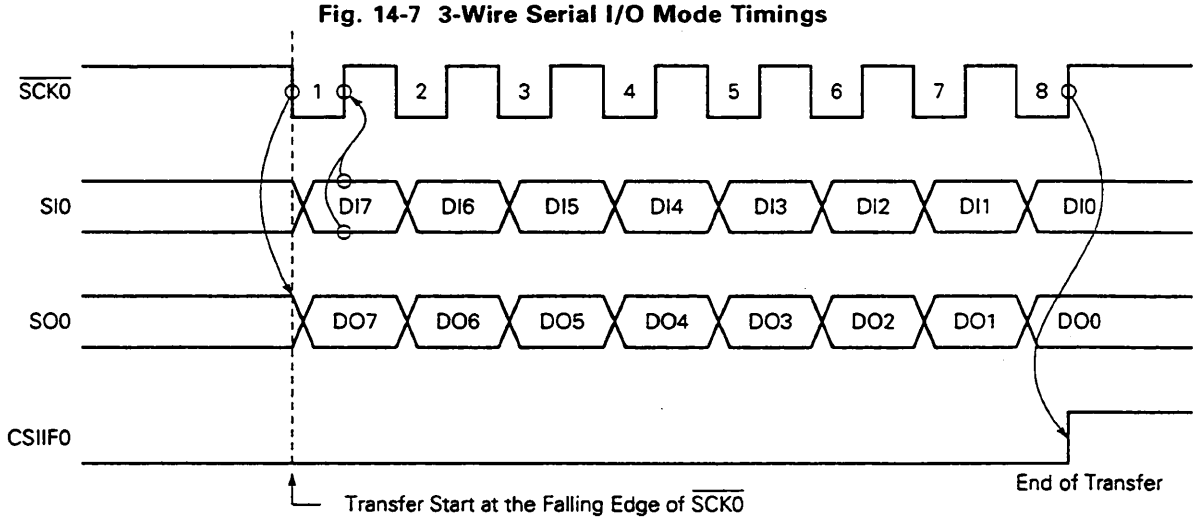
(2) Communication operation

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. 14-7 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.

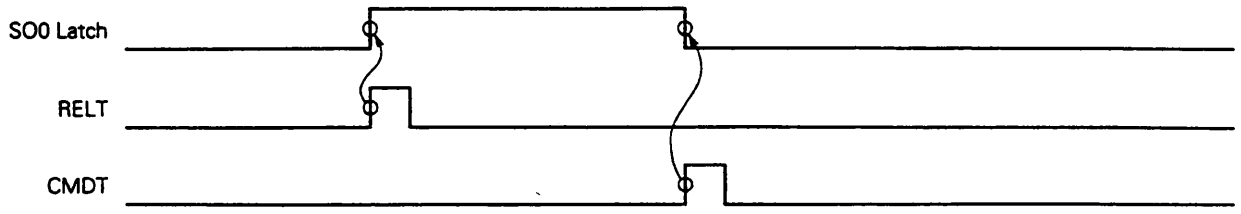
However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **14.4.5 $\overline{SCK0}$ pin output manipulation**).

(3) Various signals

Fig. 14-8 shows RELT and CMDT operations.

Fig. 14-8 RELT and CMDT Operations



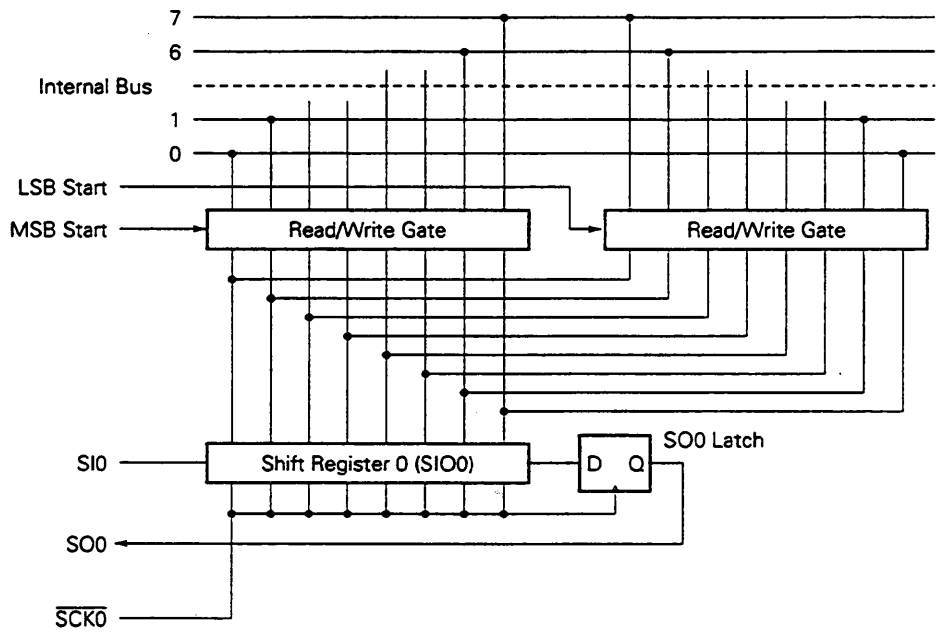
(4) MSB/LSB switching as the start bit

The 3-wire serial I/O mode enables to select transfer to start at MSB or LSB.

Fig. 14-9 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. 14-9 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.

14.4.3 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.

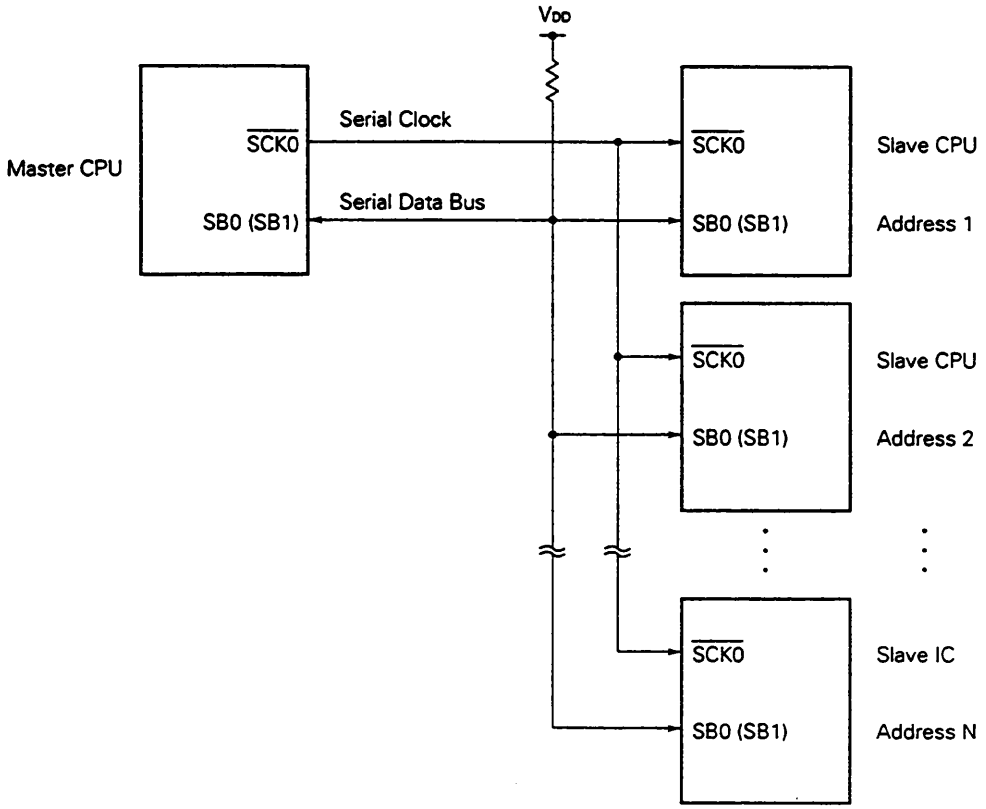
The SBI function is incorporated into various devices including 75X-series devices and 78K-series 8-bit and 16-bit single-chip microcomputers.

Fig. 14-10 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.

Refer to **(10) SBI mode precautions (d)** described later when the SBI mode is used.

Fig. 14-10 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 (INTCSI0) 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 signal ($\overline{ACK}$) 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.

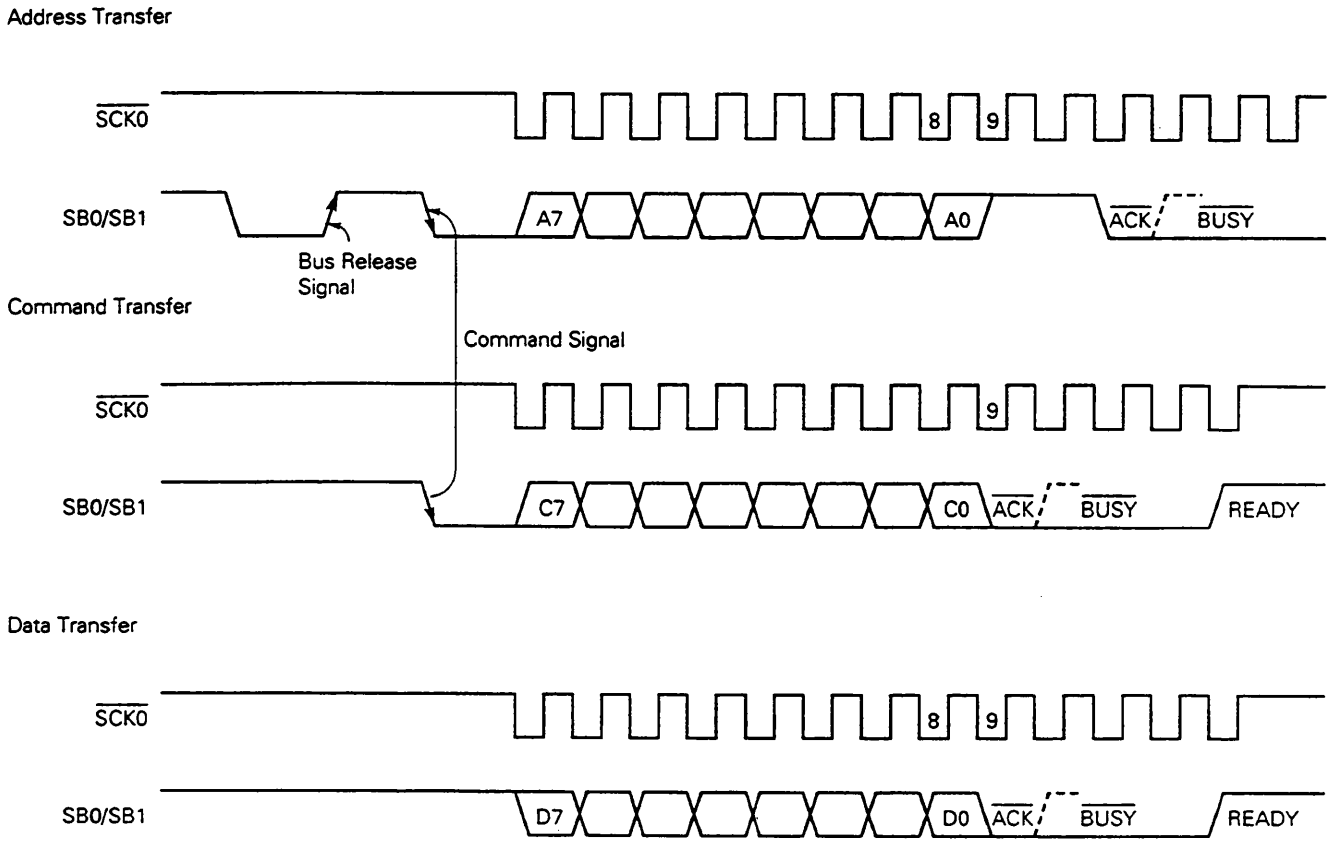
(2) SBI definition

The SBI serial data format is defined as follows.

Serial data to be transferred with SBI is distinguished into three types, "address", "command" and "data".

Fig. 14-11 shows the address, command and data transfer timings.

Fig. 14-11 SBI Transfer Timings



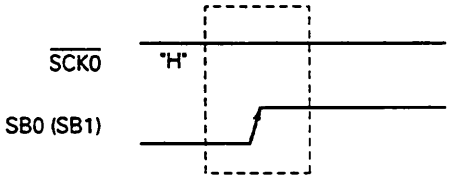
The bus release signal and the command signal are output by the master device. $\overline{\text{BUSY}}$ is output by the slave signal. $\overline{\text{ACK}}$ can be output by either the master or slave device (normally, the 8-bit data receiver outputs).

Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{\text{BUSY}}$ 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. 14-12 Bus Release Signal

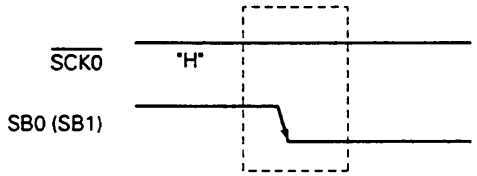


The bus release signal indicates that the master device is going to transmit an address to the slave device. The slave device incorporates hardware to detect the bus release signal.

(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. 14-13 Command Signal

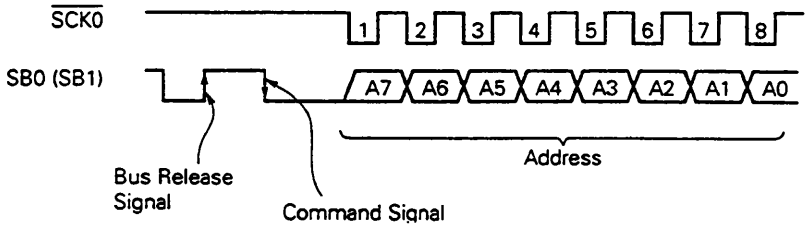


The slave device incorporates hardware to detect the command signal.

(c) Address

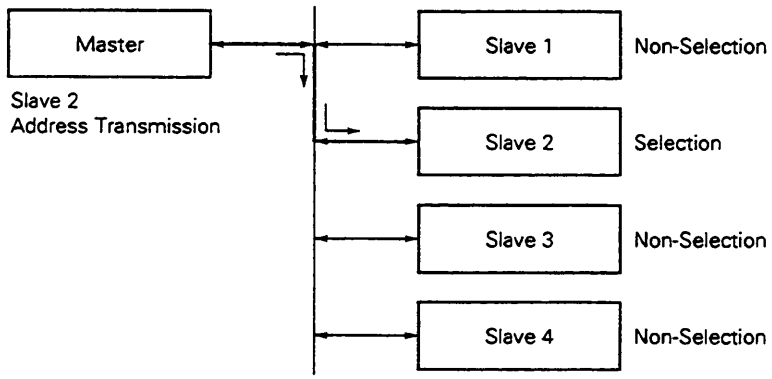
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Fig. 14-14 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. 14-15 Slave Selection with Address



(d) Command and data

The master device transmits commands to, and transmits/receives data to/from the slave device selected by address transmission.

Fig. 14-16 Commands

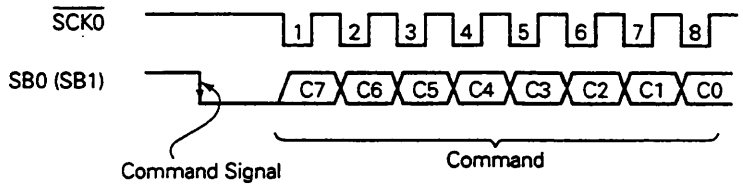
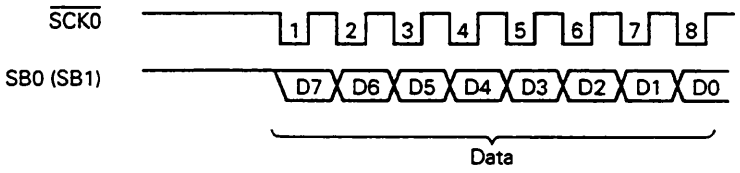


Fig. 14-17 Data



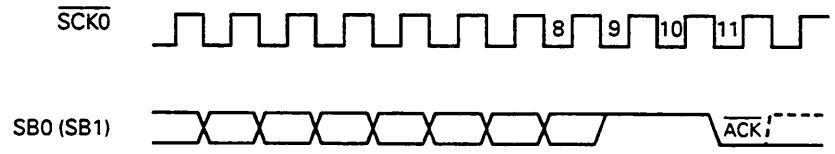
8-bit data following a command signal is defined as a "command". 8-bit data without command signal is defined as "data". Command and data operation procedures can be determined arbitrarily according to communication specifications.

(e) Acknowledge signal ($\overline{ACK}$)

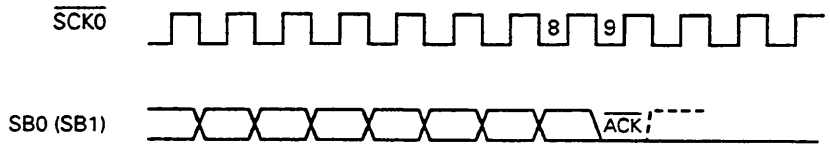
The acknowledge signal is used to check serial data reception between transmitter and receiver.

Fig. 14-18 Acknowledge Signal

[When output in synchronization with 11th clock $\overline{SCK0}$]



[When output in synchronization with 9th clock $\overline{SCK0}$]



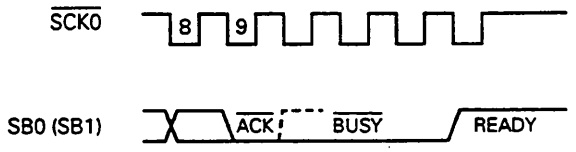
The acknowledge signal is one-shot pulse to be generated at the falling edge of $\overline{SCK0}$ after 8-bit data transfer. It can be positioned anywhere and can be synchronized with any clock $\overline{SCK0}$.
After 8-bit data transmission, the transmitter checks whether the receiver has returned 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 ($\overline{\text{BUSY}}$) and ready signal (READY)

The $\overline{\text{BUSY}}$ 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. 14-19 $\overline{\text{BUSY}}$ and READY Signals



In SBI, the slave device notifies the master device of the busy state by setting SB0 (SB1) line to the low level.

The $\overline{\text{BUSY}}$ signal is output in the wake of the acknowledge signal output from the master or slave device. It is set/reset at the falling edge of $\overline{\text{SCK0}}$. When the $\overline{\text{BUSY}}$ signal is reset, the master device automatically terminates the output of $\overline{\text{SCK0}}$ serial clock.

When the $\overline{\text{BUSY}}$ signal is reset and the READY signal is set, the master device can start the next transfer.

(3) Register setting

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 manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

The shaded area is used in the SBI mode.

Symbol

	⑦	⑥	⑤	4	3	2	1	0	Address	When Reset	R/W
CSIM0	CSIM07	CSIM06	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
	0	×	Input clock to $\overline{SCK0}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{SCK0}$/P27 Pin Function
	0	×	3-wire serial I/O mode (refer to 14.4.2 3-wire serial I/O mode operation)											
	1	0	0	× Note2	× Note2	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{SCK0}$ (CMOS input/output)
			1	0	0	× Note2	× Note2	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	
	1	1	2-wire serial I/O mode (refer to 14.4.4 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 in SBI mode

R	COI	Slave Address Comparison Result Flag [Note3]
	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 stopped
	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 manipulation 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	When 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 cleared to (0). After SO latch clearance, 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

(Cont'd)

Note: Bits 2, 3 and 6 (RELD, CMDD and ACKD) are Read-Only bits.

Remark: Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when read after data setting.

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 clock of $\overline{\text{SCK0}}$ just after execution of the instruction to be set to (1), and after acknowledge signal output, automatically cleared to 0. Used as ACKE=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 9th clock falling edge of $\overline{\text{SCK0}}$ (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ 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)
• At the falling edge of $\overline{\text{SCK0}}$ just after busy mode is cleared when transfer start instruction is executed • 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

BSYE[Note]	Synchronizing Busy Signal Output Control
0	Disables busy signal which is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction to be cleared to (0).
1	Outputs busy signal at the falling edge of $\overline{\text{SCK0}}$ clock following the acknowledge signal.

Note: Busy mode can be cleared by start of serial interface transfer or reception of address signal. However, BSYE flag is not cleared to 0.

(c) Interrupt timing specify register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulation 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	When 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 Factor Selection Note 2
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ Pin Level Note 3
0	Low Level
1	High Level

Caution: Be sure to set bits 0 to 3 to 0.

Notes:

(1) Bit 6 (CLD) is a Read-Only bit.

(2) When using wake-up function, set SIC to 0.

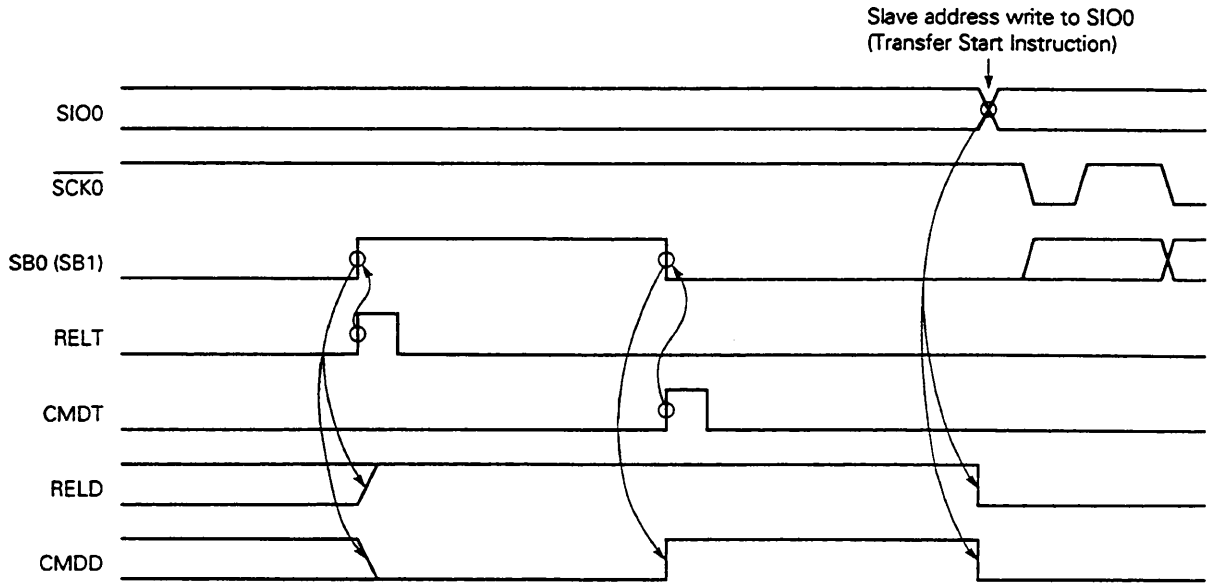
(3) When CSIE0 = 0, CLD becomes 0.

Remark: SVA: Slave address register

(4) Various signals

Figs. 14-20 to 14-25 show various signals and flag operations in SBI. Table 14-3 lists various signals in SBI.

Fig. 14-20 RELT, CMDT, RELD and CMDD Operations (Master)



★ **Fig. 14-21 RELT, CMDT, RELD and CMDD Operations (Slave)**

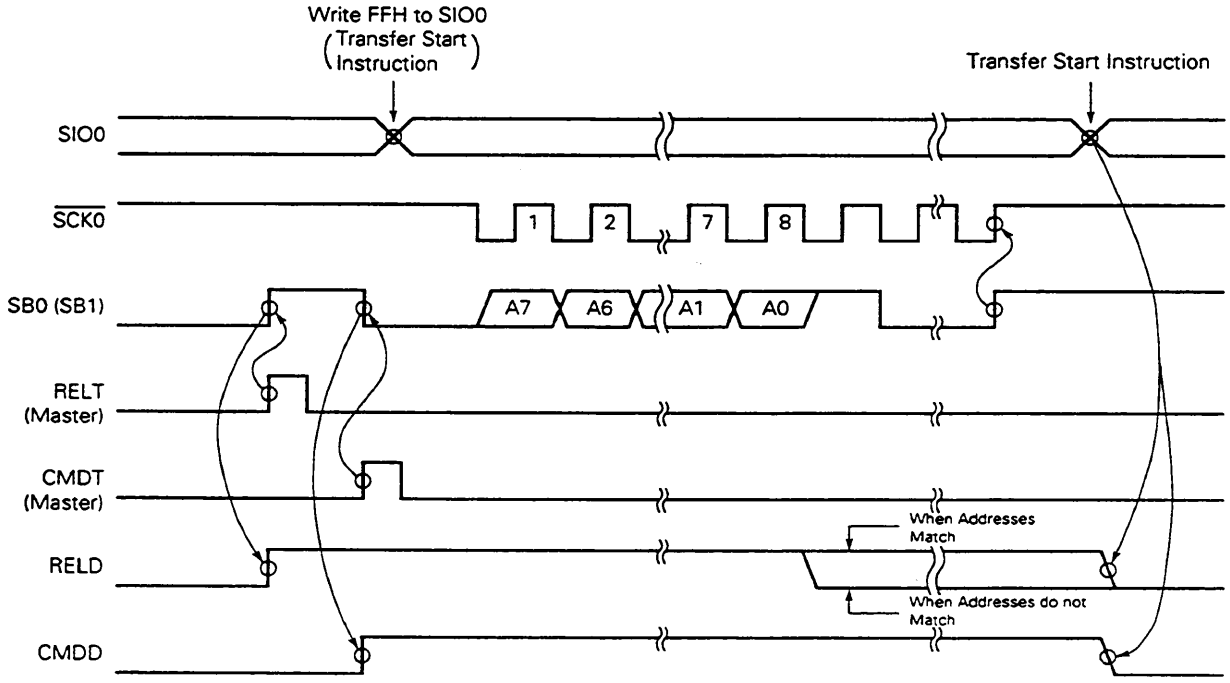
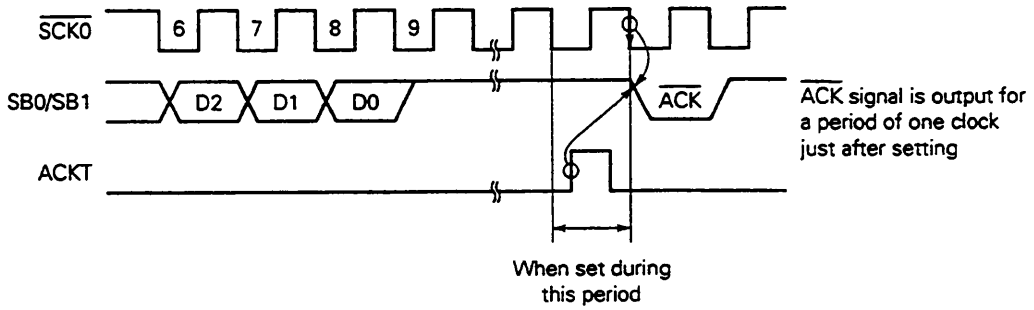


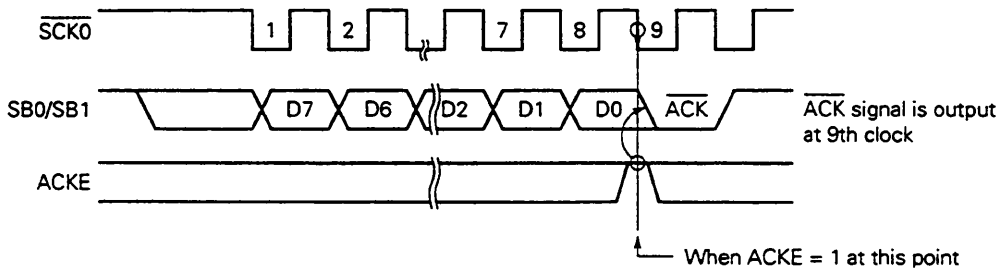
Fig. 14-22 ACKT Operation



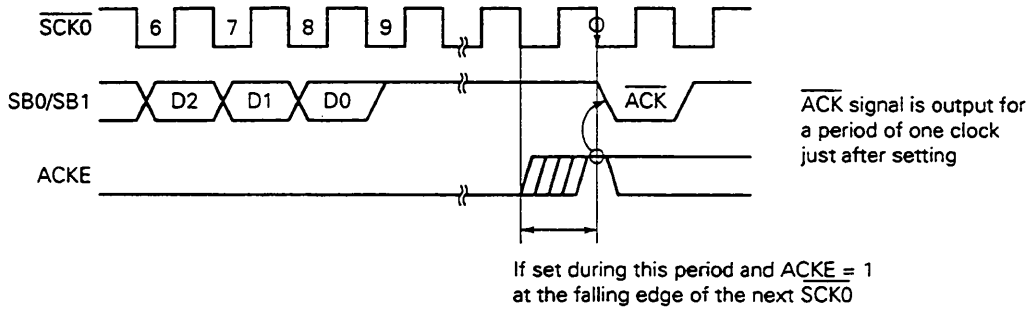
Caution: Do not set ACKT before termination of transfer.

Fig. 14-23 ACKE Operations

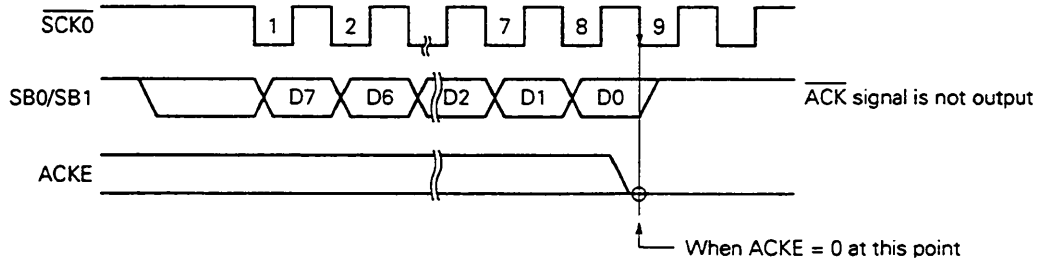
(a) When ACKE = 1 upon completion of transfer



(b) When set after completion of transfer



(c) When ACKE = 0 upon completion of transfer



(d) When ACKE = 1 period is short

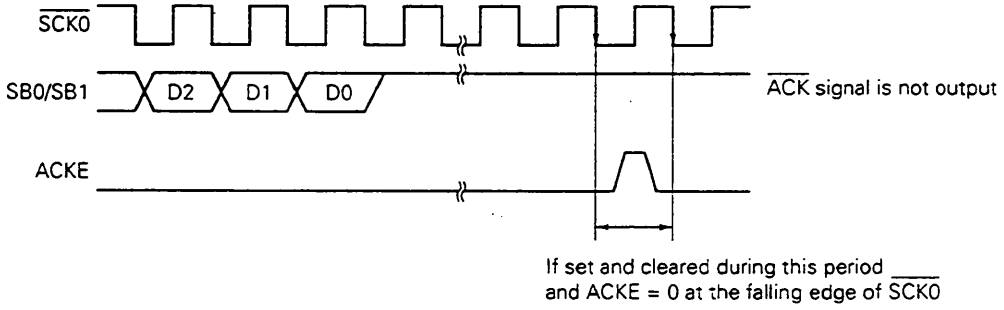
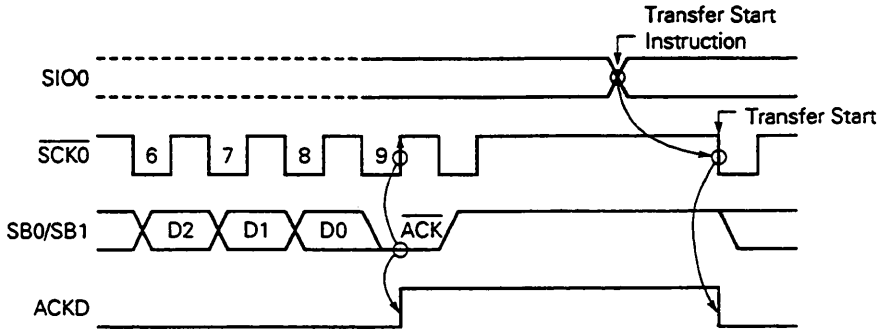
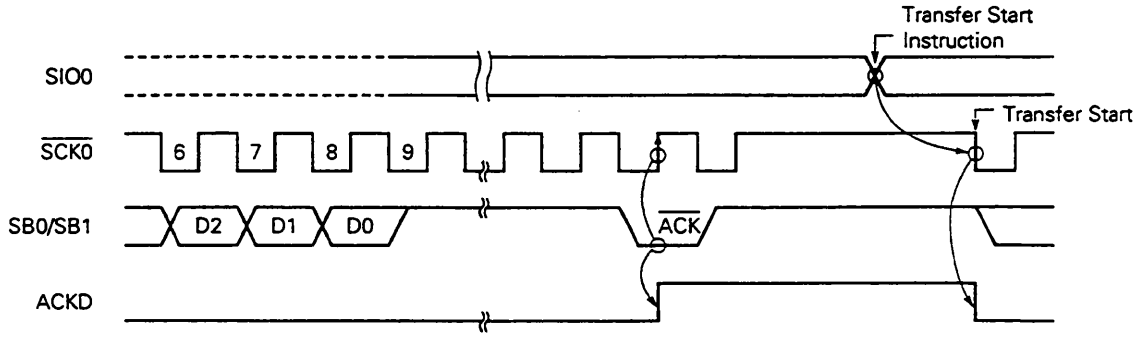


Fig. 14-24 ACKD Operations

(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 start is instructed in BUSY

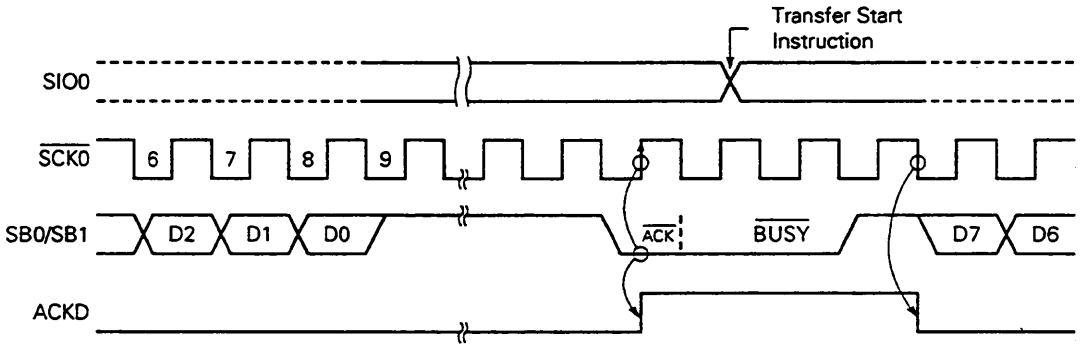


Fig. 14-25 BSYE Operation

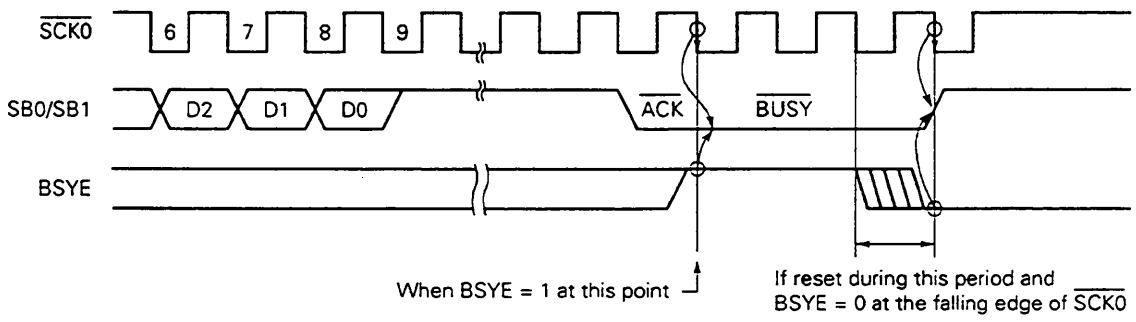


Table 14-3 Various Signals in SBI Mode (1/2)

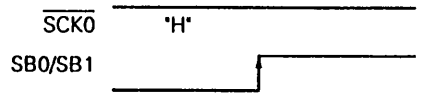
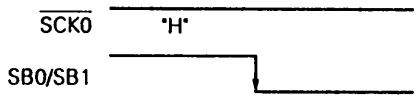
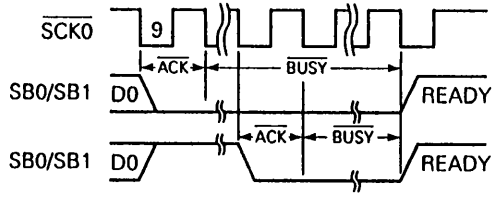
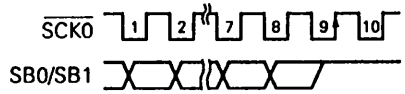
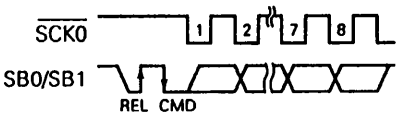
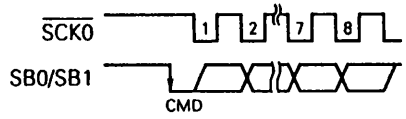
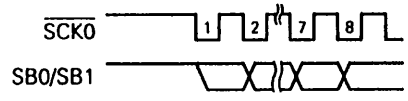
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0/SB1 rising edge when $\overline{SCK0}$ = 1		• RELT set	• RELD set • CMDD clear	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0/SB1 falling edge when $\overline{SCK0}$ = 1		• CMDT set	• CMDD set	i) Transmit data is an address after REL signal output. ii) REL signal is not output and transmit data is a command.
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]** 	① ACKE = 1 ② ACKT set	• ACKD set	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 receive 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 receive enable

Table 14-3 Various 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) Note2	CSIIF0 set (rising edge of 9th clock of $\overline{SCK0}$) Note1	Timing of signal output to serial data bus
Address (A7 to 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 to C0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK0}$ after output of only CMD signal without REL signal output				Istructions and messages to the slave device
Data (D7 to 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 starts after the READY state is set.

(5) Pin configuration

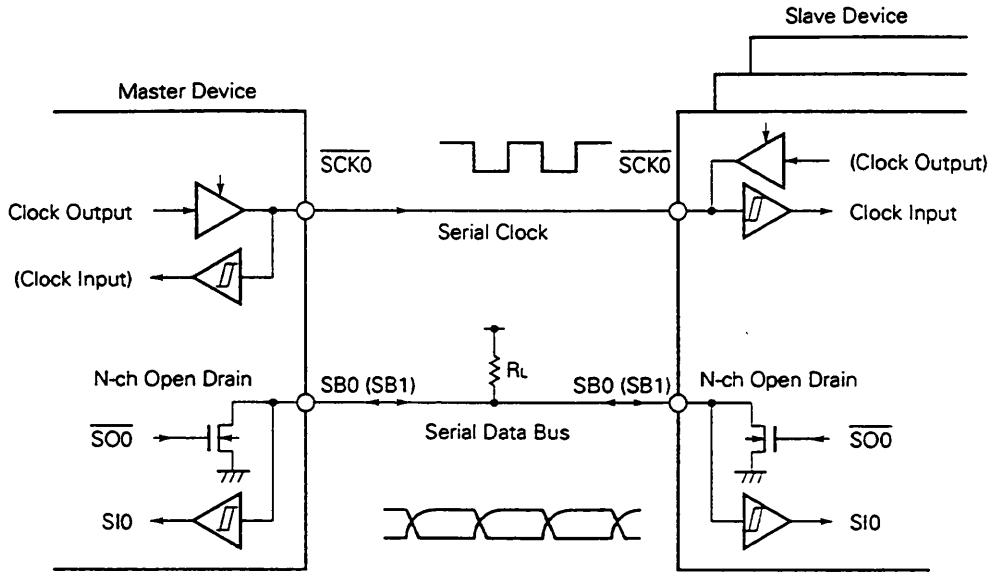
The serial clock pin ($\overline{SCK0}$) and serial data bus pin SB0 (SB1) have the following configurations.

(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. 14-26 Pin Configuration



Caution: Because the N-ch open-drain must be turned OFF at time of data reception, write FFH to SIO0 in advance. The N-ch open-drain can be turned OFF at any time of 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 matching 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. 14-27 to 14-30 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) pin at the rising edge of $\overline{SCK0}$ is latched into the shift register.

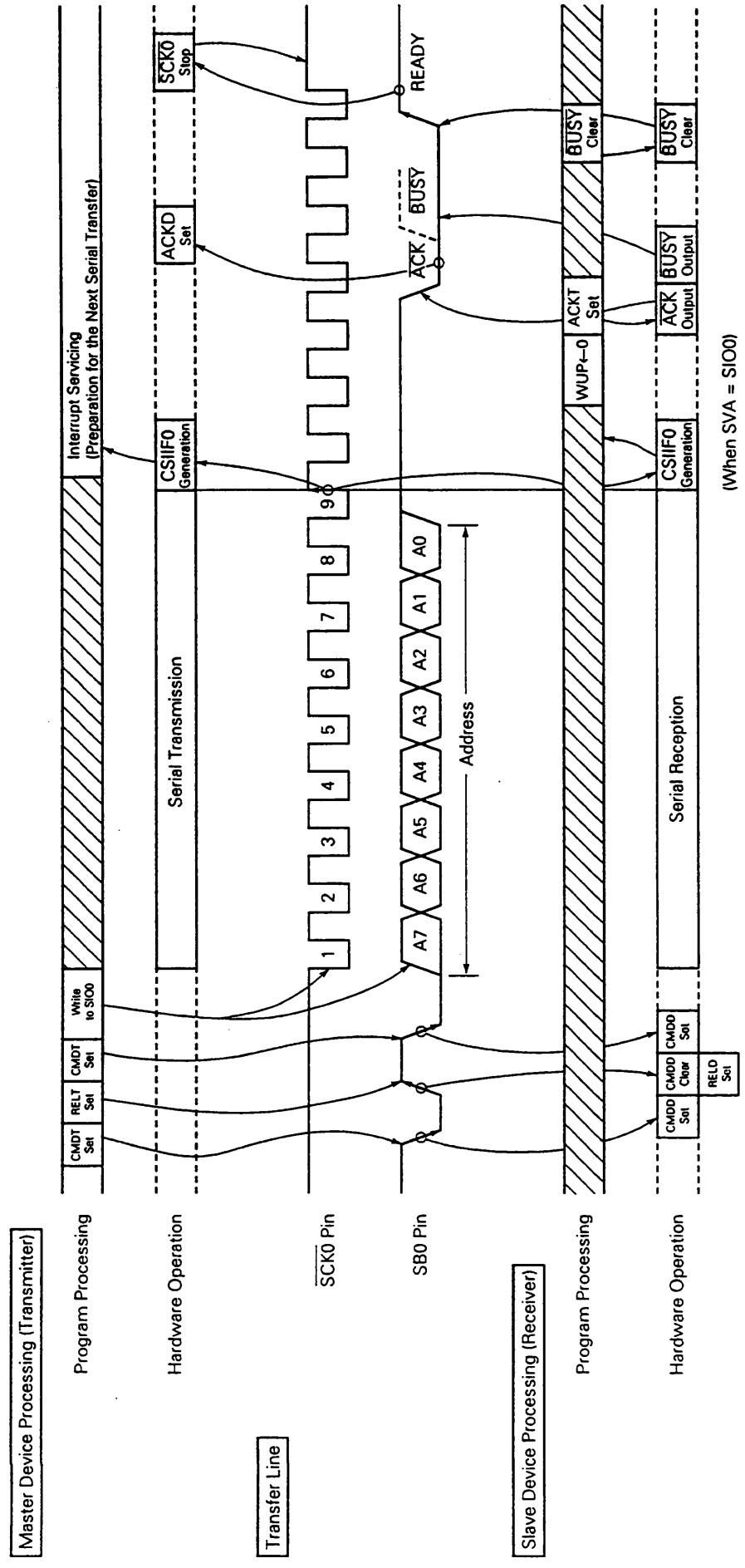


Fig. 14-27 Address Transmission from Master Device to Slave Device (WUP = 1)

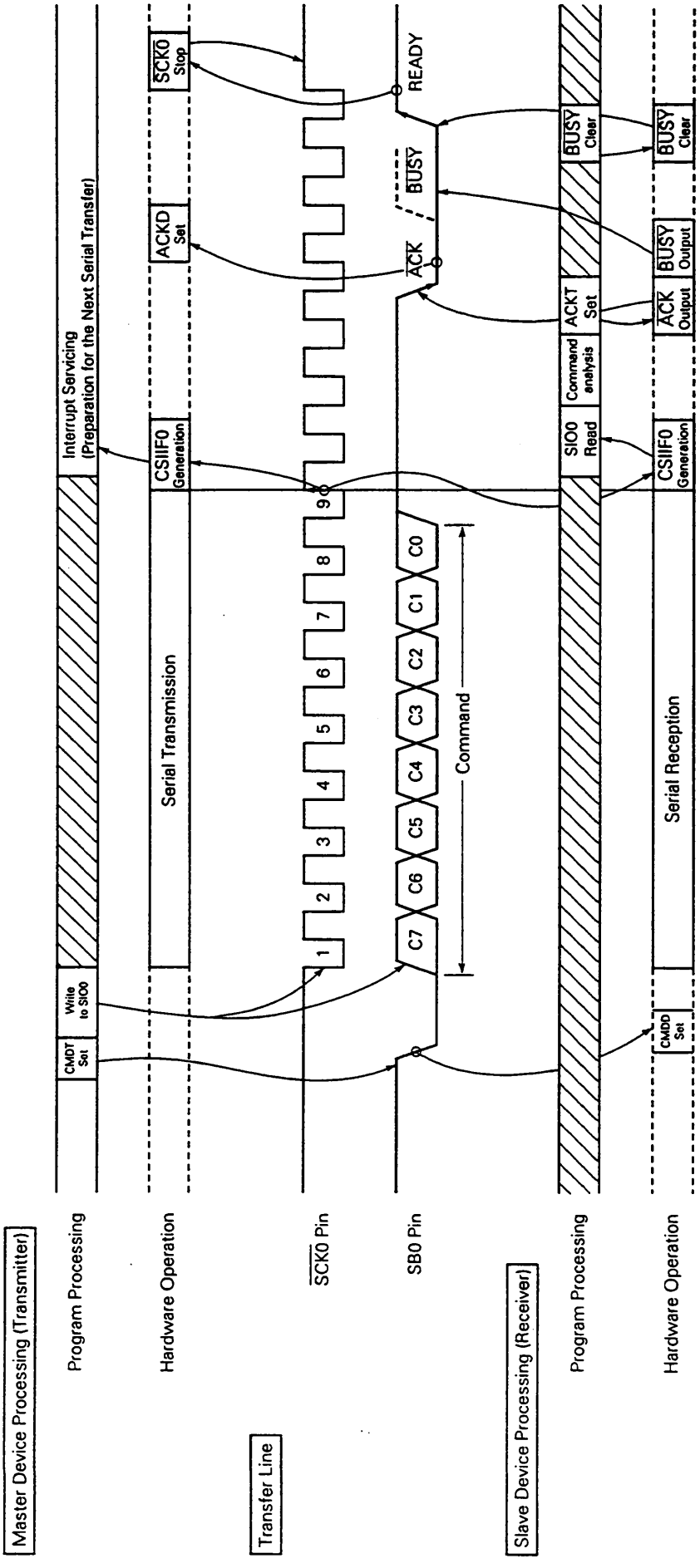


Fig. 14-28 Command Transmission from Master Device to Slave Device

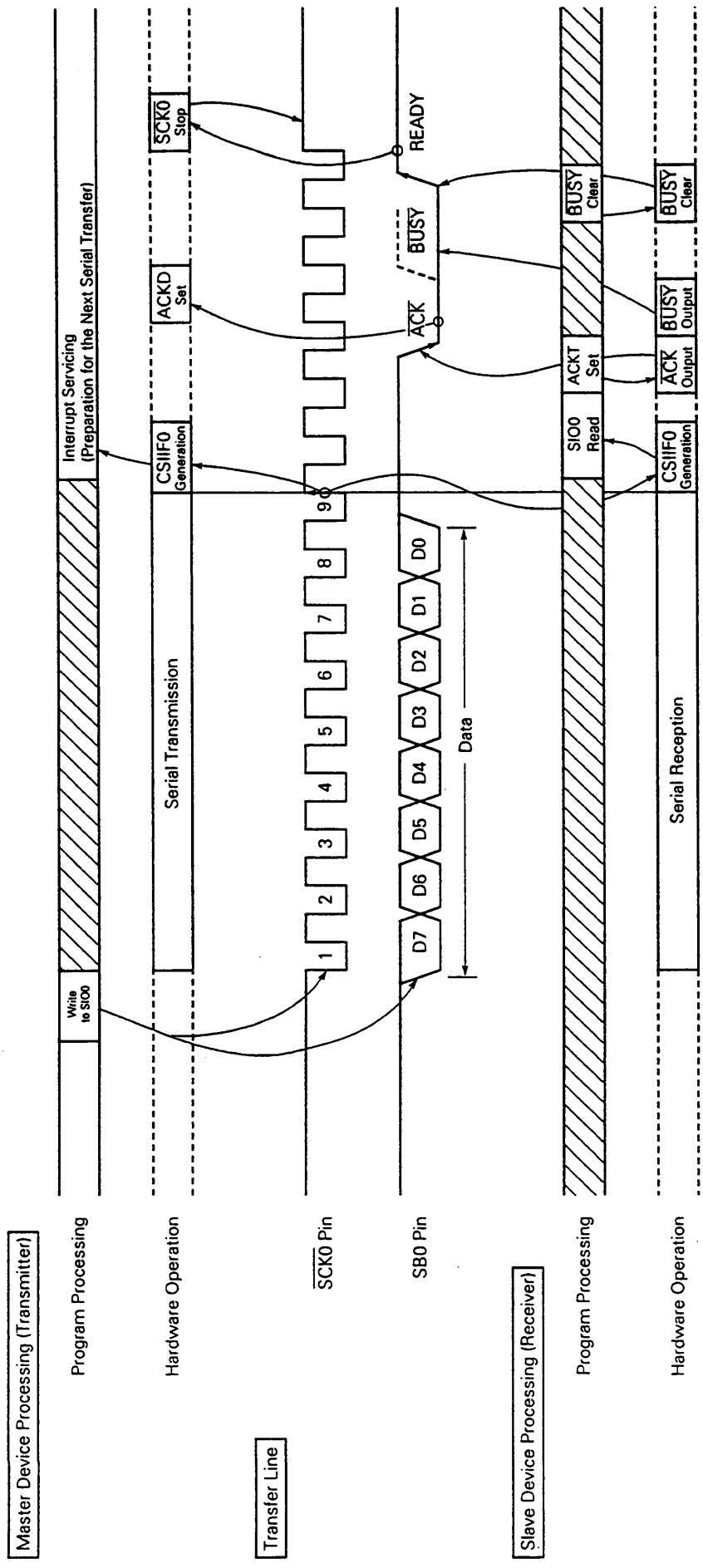


Fig. 14-29 Data Transmission from Master Device to Slave Device

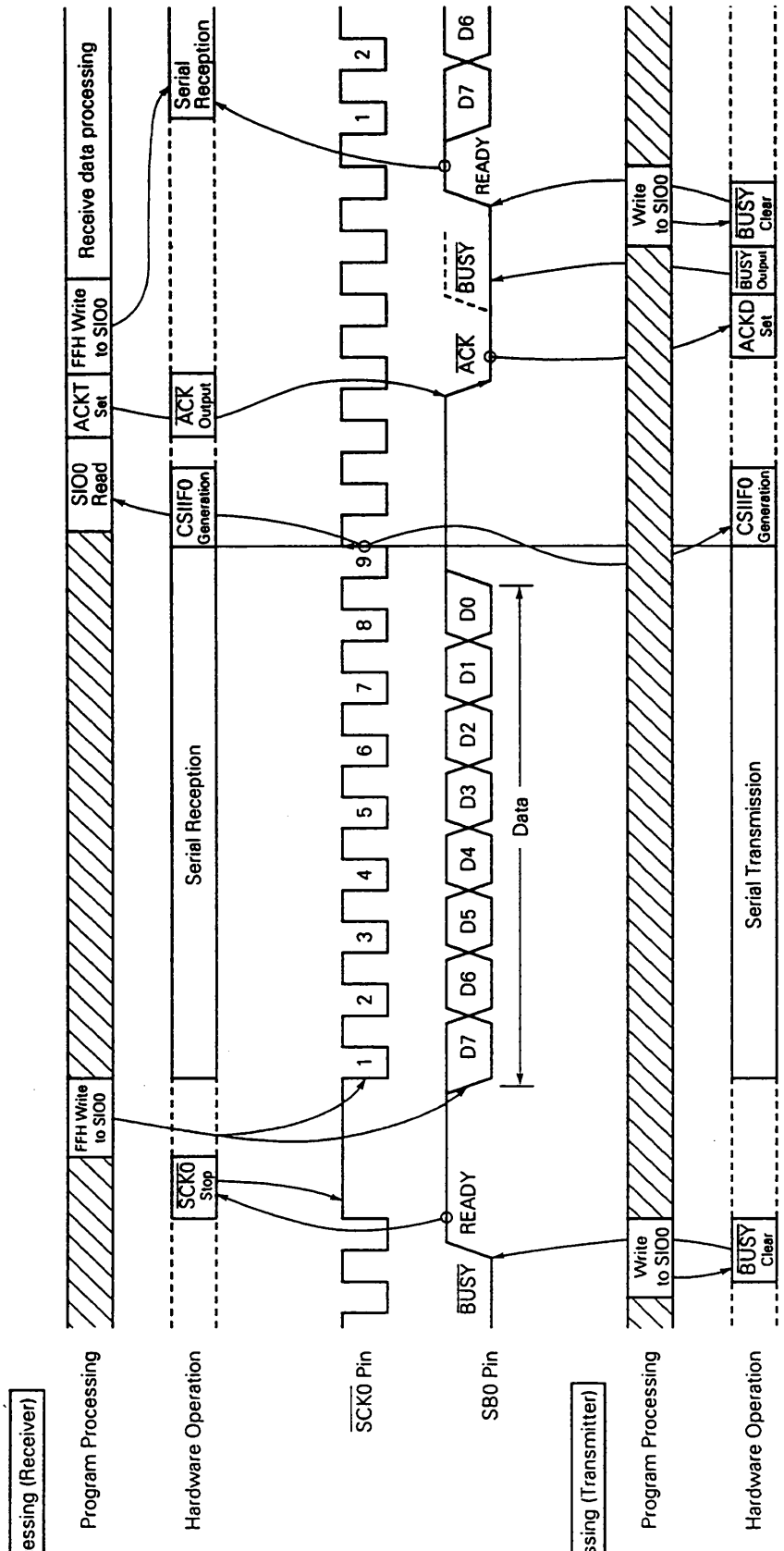


Fig. 14-30 Data Transmission from Slave Device to Master Device

(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 transistor 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 (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.

(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 $\overline{\text{BUSY}}$ signal output, $\overline{\text{BUSY}}$ is not cleared. In SBI, the $\overline{\text{BUSY}}$ signal continues to be output after $\overline{\text{BUSY}}$ clear instruction generation to the falling edge of the next serial clock ($\overline{\text{SCK0}}$). Before setting WUP to 1, be sure to clear $\overline{\text{BUSY}}$ 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 device is in the master mode, follow the method below to judge whether slave device is in the busy state or not.

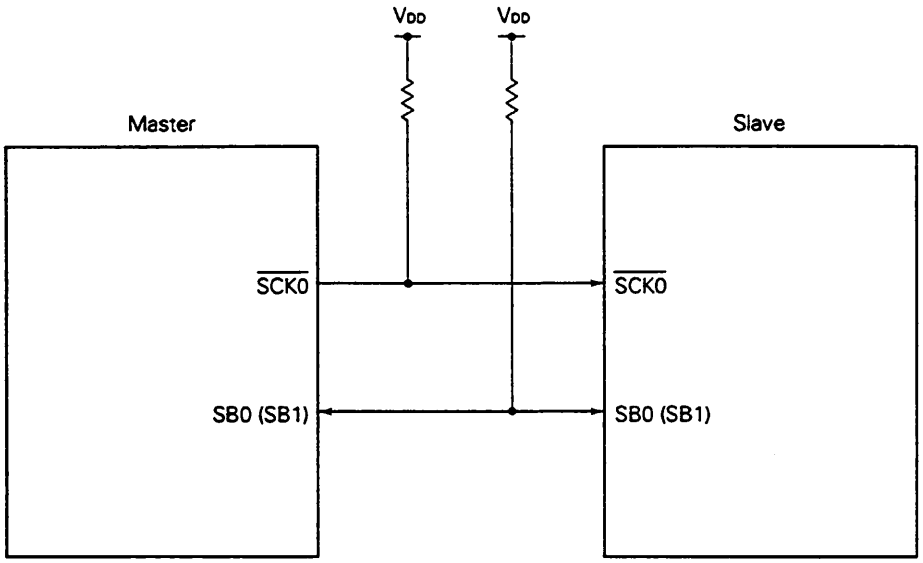
① Detect acknowledge signal ($\overline{\text{ACK}}$) or interrupt signal generation.
② Set the port mode register PM25 (or PM26) of the SB0/P25 (or SB1/P26) pin into the input mode.
③ Read out the pin state (when the pin level is high, the READY state is set).

After the detection of the READY state, set the port mode register to 0 and return to the output mode.

14.4.4 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 ($\overline{SCK0}$) and serial data input/output (SB0 or SB1).



(1) Register setting

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 manipulation instruction.

$\overline{RESET}$ 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	When Reset	R/W
CSIM0									FF60H	00H	R/W Note 1

R/W	CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
	0	×	Input clock to $\overline{SCK0}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{SCK0}$/P27 Pin Function
	0	×	3-wire serial I/O mode (refer to 14.4.2 3-wire serial I/O mode operation)											
	1	0	SBI mode (refer to 14.4.3 SBI mode operation)											
	1	1	0	Note2 ×	Note2 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{SCK0}$ (N-ch open-drain input/ output)
			1	0	0	Note2 ×	Note2 ×	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

R/W	WUP	Wake-up Function Control Note3
	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 in SBI mode

R

COI	Slave Address Comparison Result Flag Note4
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 stopped
1	Operation enable

Notes:

(1) Bit 6 (COI) is a Read-Only bit.

(2) Can be used freely as port function.

(3) Always set 0 to WUP when using the 2-wire serial I/O mode.

(4) When CSIE0 = 0, COI becomes 0.

Remark: x: Don't care

(b) Serial bus interface control register (SBIC)

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

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	When Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	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

CMDT	When CMDT = 1, SO latch is cleared to 0. After SO latch clearance, 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 manipulation 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	When 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 Factor Selection
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ Pin Level Note 2
0	Low Level
1	High Level

Caution: Be sure to set bits 0 to 3 to 0.

Notes:

(1) Bit 6 (CLD) is a Read-Only bit.

(2) When CSIE0 = 0, CLD becomes 0.

Remark: SVA: Slave address register

(2) Communication operation

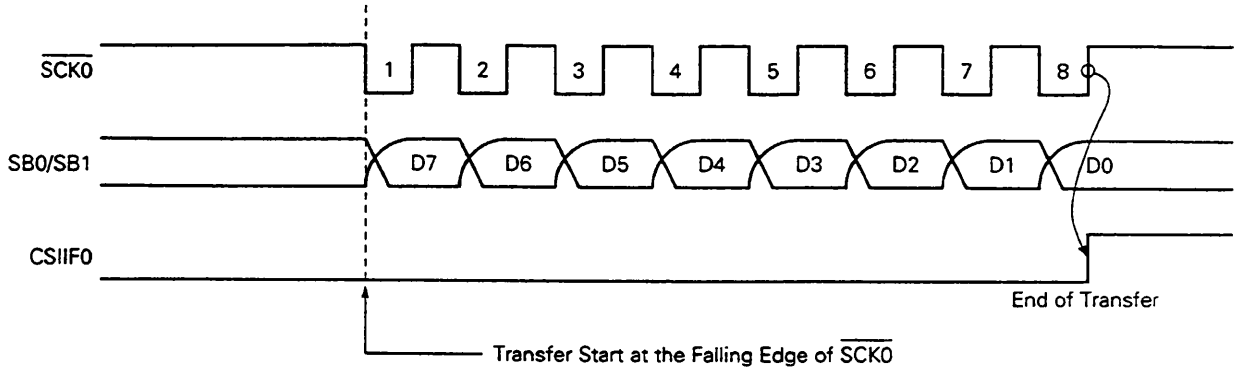
The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out in synchronization with 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. 14-31 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 for data reception, 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.

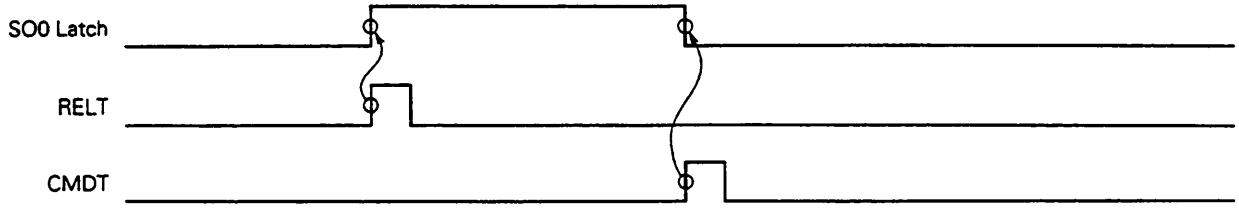
However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **14.4.5 $\overline{SCK0}$ pin output manipulation**).

(3) Various signals

Fig. 14-32 shows RELT and CMDT operations.

Fig. 14-32 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 (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 transistor 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.

14.4.5 $\overline{\text{SCK0}}$ pin output manipulation

Because the $\overline{\text{SCK0}}$/P27 pin incorporates an output latch, static output is also possible by software in addition to normal serial clock output.

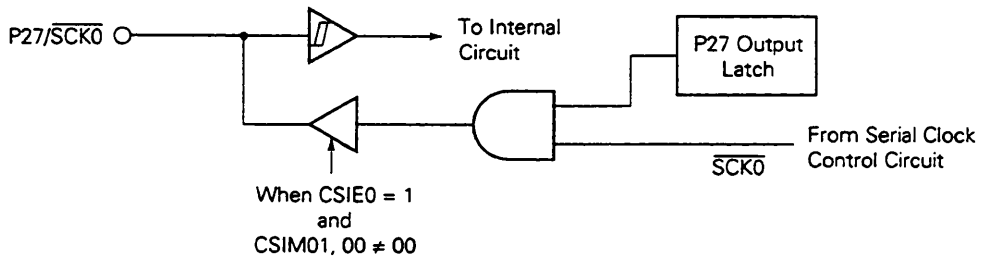
P27 output latch manipulation enables any number of $\overline{\text{SCK0}}$ to be set by software (SI0/SB0 and SO0/SB1 pin to be controlled with the RELT and CMDT bits of SBIC).

$\overline{\text{SCK0}}$/P27 pin output manipulating procedure is described below.

① Set the serial operating mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin enabled for serial operation in the output mode). $\overline{\text{SCK0}}$ = 1 with serial transfer suspended.

② Manipulate the P27 output latch with a bit manipulation instruction.

Fig. 14-33 $\overline{\text{SCK0}}$/P27 Pin Configuration



CHAPTER 15 SERIAL INTERFACE CHANNEL 1

15.1 Serial Interface Channel 1 Functions

Serial interface channel 1 employs the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

(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.

(3) 3-wire serial I/O mode with automatic transmit/receive function

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 automatic transmit/receive function is used to transmit/receive data with a maximum of 32 bytes. This function enabled the hardware to transmit/receive data to/from the OSD (On Screen Display) device and a device with on-chip display controller/driver independently of the CPU thus the software load can be alleviated.

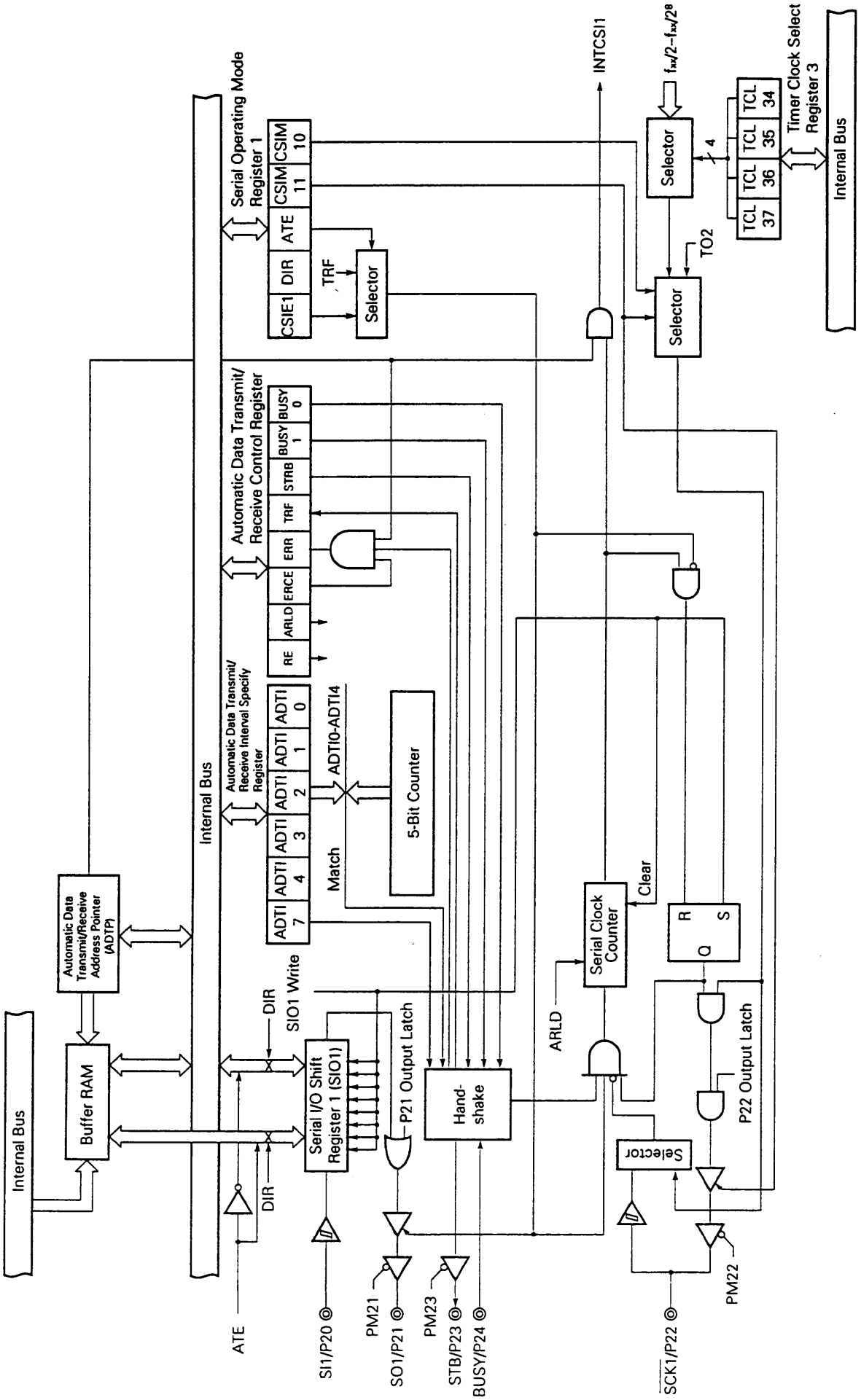
15.2 Serial Interface Channel 1 Configuration

Serial interface channel 1 consists of the following hardware.

Table 15-1 Serial Interface Channel 1 Configuration

Item	Configuration
Register	Serial I/O shift register 1 (SIO1) Automatic data transmit/receive address pointer (ADTP)
Control register	Timer clock select register 3 (TCL3) Serial operating mode register 1 (CSIM1) Automatic data transmit/receive control register (ADTC) Automatic data transmit/receive interval specify register (ADTI) Port mode register 2 (PM2)

Fig. 15-1 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 manipulation instruction.

When value in bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is 1, writing data to SIO1 starts 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 makes SIO1 undefined.

Caution: Do not write data to SIO1 while the automatic transmit/receive function is activated.

(2) Automatic data transmit/receive address pointer (ADTP)

This register stores buffer RAM addresses while the automatic transmit/receive function is activated.

ADTP is set with an 8-bit memory manipulation instruction. The high-order 3 bits must be set to 0.

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

Caution: Do not write data to ADTP while the automatic transmit/receive function is activated.

(3) 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.

15.3 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)
- Automatic data transmit/receive control register (ADTC)
- Automatic data transmit/receive interval specify register (ADTI)

(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 manipulation 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. 15-2 Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	When 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	$f_{XX}/2$ Note
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
Other than above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection
0	1	1	0	$f_{XX}/2$ Note
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
Other than above				Setting prohibited

Note: Can only be set when the main system clock frequency is 5.0 MHz or less.

Remarks:

(1) fxx: Main system clock frequency

(2) Figures in parentheses apply to operation with fxx = 4.0 MHz

★ **Caution: When rewriting TCL3 to data other than the same data, stop the timer operation first.**

(2) Serial operating mode register 1 (CSIM1)

This register sets serial interface channel 1 serial clock, operating mode, operation enable/stop and automatic transmit/receive operation enable/stop.

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM1 to 00H.

Fig. 15-3 Serial Operation Mode Register 1 Format

Symbol	⑦	6	⑤	4	3	2	1	0	Address	When Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial Interface Channel 1 Clock Selection
0	x	Clock externally input to $\overline{\text{SCK1}}$ pin Note1
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin function
0	MSB	SI1/P20 (Input)	SO1 (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	$\overline{\text{SCK1}}$/P22 Pin Function
0	×	Note2 ×	Note2 ×	Note2 ×	Note2 ×	Note2 ×	Note2 ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note3 1	Note3 ×	0	0	1	×	Operation enable	Count operation	SI1 Note3 (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes:

(1) If the external clock input has been selected with CSIM11 set to 0, set STRB and BUSY1 of the automatic data transmit/receive control register to 0, 0.

(2) Can be used freely as port function.

(3) Can be used as P20 (CMOS input/output) when only transmitter is used.

Remark: ×: Don't care

(3) Automatic data transmit/receive control register (ADTC)

This register sets automatic receive enable/disable, the operating mode, strobe control enable/disable, busy control enable/disable and displays automatic transmit/receive execution.

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

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

Fig. 15-4 Automatic Data Transmit/Receive Control Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When Reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Busy Input Control
0	×	Not using busy input
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB	Strobe Output Control
0	Strobe output disable
1	Strobe output enable

R

TRF	Status of Automatic Transmit/Receive Function Note 2
0	Detection of termination of automatic transmission/reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error Detection of Automatic Transmit/Receive Function
0	No error (This bit is set to 0 when data is written to SIO1)
1	Error occurred

R/W

ERCE	Error Check Control of Automatic Transmit/Receive Function
0	Error check disable
1	Error check enable (only when BUSY1 = 1)

R/W

ARLD	Operating Mode Selection of Automatic Transmit/Receive Function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive Control of Automatic Transmit/Receive Function
0	Receive disable
1	Receive enable

Notes:

(1) Bits 3 and 4 (TRF and ERR) are Read-Only bit.

(2) The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1.

Caution: When an external clock input is selected with CSIM11 of the serial operating mode register 1 set to 0, set STRB and BUSY1 of ADTC to 0, 0.

Remark: ×: Don't care

(4) Automatic data transmit/receive interval specify register (ADTI)

This register sets the automatic data transmit/receive function data transfer interval.

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

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

Fig. 15-5 Automatic Data Transmit/Receive Interval Specify Register Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADTI Note 1
1	Control of interval by ADTI (ADTI0 to ADTI4)

★

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (fxx = 4.0 MHz Operation)	
					Minimum Note 2	Maximum Note 2
0	0	0	0	0	46.0 μs + 0.5/fsck	50.0 μs + 1.5/fsck
0	0	0	0	1	78.0 μs + 0.5/fsck	82.0 μs + 1.5/fsck
0	0	0	1	0	110.0 μs + 0.5/fsck	114.0 μs + 1.5/fsck
0	0	0	1	1	142.0 μs + 0.5/fsck	146.0 μs + 1.5/fsck
0	0	1	0	0	174.0 μs + 0.5/fsck	178.0 μs + 1.5/fsck
0	0	1	0	1	206.0 μs + 0.5/fsck	210.0 μs + 1.5/fsck
0	0	1	1	0	238.0 μs + 0.5/fsck	242.0 μs + 1.5/fsck
0	0	1	1	1	270.0 μs + 0.5/fsck	274.0 μs + 1.5/fsck
0	1	0	0	0	302.0 μs + 0.5/fsck	306.0 μs + 1.5/fsck
0	1	0	0	1	334.0 μs + 0.5/fsck	338.0 μs + 1.5/fsck
0	1	0	1	0	366.0 μs + 0.5/fsck	370.0 μs + 1.5/fsck
0	1	0	1	1	398.0 μs + 0.5/fsck	402.0 μs + 1.5/fsck
0	1	1	0	0	430.0 μs + 0.5/fsck	434.0 μs + 1.5/fsck
0	1	1	0	1	462.0 μs + 0.5/fsck	466.0 μs + 1.5/fsck
0	1	1	1	0	494.0 μs + 0.5/fsck	498.0 μs + 1.5/fsck
0	1	1	1	1	526.0 μs + 0.5/fsck	530.0 μs + 1.5/fsck

Notes:

(1) The interval is dependent only on CPU processing.

(2) The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/fsck, the minimum interval time is 2/fsck.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_{XX}} + \frac{56}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_{XX}} + \frac{72}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions:

(1) ADTI should not be written to during operation of the automatic data transmit/receive function.

(2) Zero must be set in bits 5 and 6.

Remarks:

(1) fxx : Main system clock frequency

(2) fsck : Serial clock frequency

Fig. 15-5 Automatic Data Transmit/Receive Interval Specify Register Format (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (fxx = 4.0 MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	558.0 μs + 0.5/fsck	562.0 μs + 1.5/fsck
1	0	0	0	1	590.0 μs + 0.5/fsck	594.0 μs + 1.5/fsck
1	0	0	1	0	622.0 μs + 0.5/fsck	626.0 μs + 1.5/fsck
1	0	0	1	1	654.0 μs + 0.5/fsck	658.0 μs + 1.5/fsck
1	0	1	0	0	686.0 μs + 0.5/fsck	690.0 μs + 1.5/fsck
1	0	1	0	1	718.0 μs + 0.5/fsck	722.0 μs + 1.5/fsck
1	0	1	1	0	750.0 μs + 0.5/fsck	754.0 μs + 1.5/fsck
1	0	1	1	1	782.0 μs + 0.5/fsck	786.0 μs + 1.5/fsck
1	1	0	0	0	814.0 μs + 0.5/fsck	818.0 μs + 1.5/fsck
1	1	0	0	1	846.0 μs + 0.5/fsck	850.0 μs + 1.5/fsck
1	1	0	1	0	878.0 μs + 0.5/fsck	882.0 μs + 1.5/fsck
1	1	0	1	1	910.0 μs + 0.5/fsck	914.0 μs + 1.5/fsck
1	1	1	0	0	942.0 μs + 0.5/fsck	946.0 μs + 1.5/fsck
1	1	1	0	1	974.0 μs + 0.5/fsck	978.0 μs + 1.5/fsck
1	1	1	1	0	1.006 ms + 0.5/fsck	1.010 ms + 1.5/fsck
1	1	1	1	1	1.038 ms + 0.5/fsck	1.042 ms + 1.5/fsck

Note: The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/fsck, the minimum interval time is 2/fsck.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{fxx} + \frac{56}{fxx} + \frac{0.5}{fsck}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{fxx} + \frac{72}{fxx} + \frac{1.5}{fsck}$$

Cautions:

(1) ADTI should not be written to during operation of the automatic data transmit/receive function.

(2) Zero must be set in bits 5 and 6.

Remarks:

(1) fxx : Main system clock frequency

(2) fsck : Serial clock frequency

15.4 Serial Interface Channel 1 Operations

The following three operating modes are available to the serial interface channel 1.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

15.4.1 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 an ordinary 8-bit register.

In the operation stop mode, the P20/SI1, P21/SO1, P22/$\overline{SCK1}$, P23/STB and P24/BUSY pins can be used as ordinary input/output ports.

(1) Register setting

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 manipulation 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
CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10

Address	When Reset	R/W
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	×	Note1 ×	Note1 ×	Note1 ×	Note1 ×	Note1 ×	Note1 ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
★ 1	0	Note2 1	Note2 ×	0	0	1	×	Operation enable	Count operation	SI1 Note2 (Input)	SO1 (CMOS output)	$\overline{SCK1}$ (Input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes:

(1) Can be used freely as port function.

(2) Can be used as P20 (CMOS input/output) when only transmitter is used.

Remark: ×: Don't care

15.4.2 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 setting

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 manipulation 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	When Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial Interface Channel 1 Clock Selection
0	x	Clock externally input to $\overline{SCK1}$ pin Note
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

(Cont'd)

Note: If the external clock input has been selected with CSIM11 set to 0, set STRB and BUSY1 of the automatic data transmit/receive control register to 0, 0.

Remark: x: Don't care

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	×	Note1 ×	Note1 ×	Note1 ×	Note1 ×	Note1 ×	Note1 ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note2 1	Note2 ×	0	0	1	×	Operation enable	Count operation	SI1 Note2 (Input)	SO1 (CMOS output)	$\overline{SCK1}$ (Input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes:

(1) Can be used freely as port function.

(2) Can be used as P20 (CMOS input/output) 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 interrupt request flag (CSIIF1) is set.

(3) Transfer start

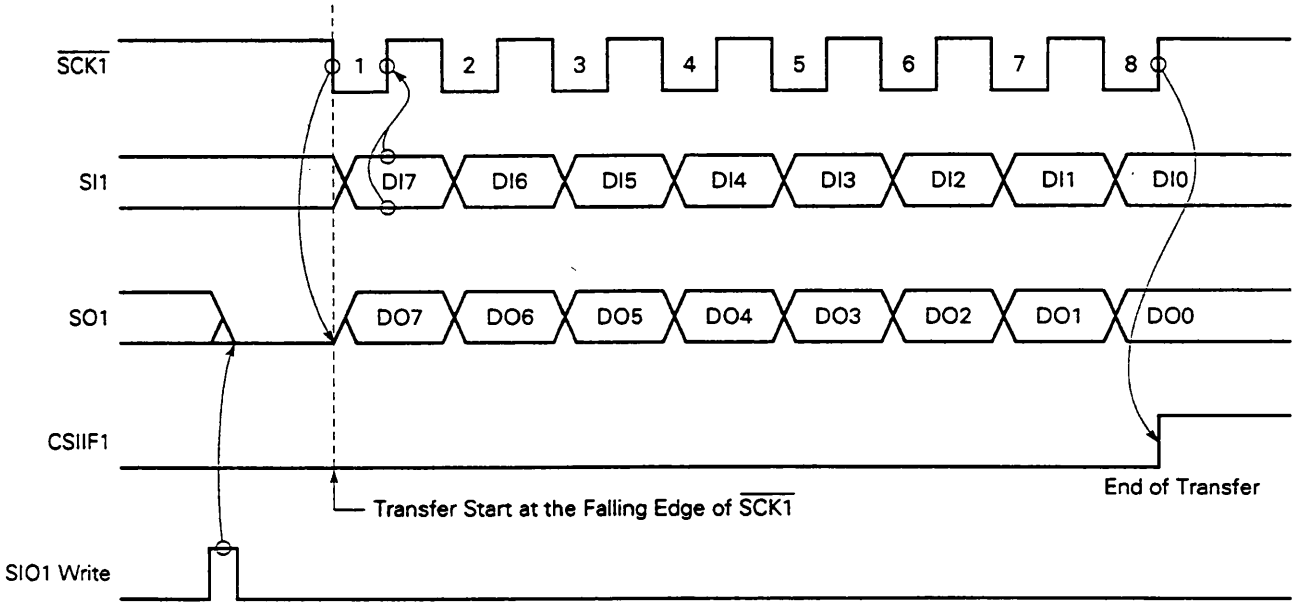
Serial transfer is started if the following two conditions are satisfied when the transfer data is set to the serial I/O shift register 1 (SIO1):

- Operation control bit of serial interface 1 (CSIE1) = 1
- When internal serial clock is stopped or when $\overline{SCK1}$ is high after 8-bit serial transfer has been completed

Caution: Transfer is not started even if CSIE1 is set to "1" after data has been written to SIO1.

Upon termination of 8-bit transfer, serial transfer stops automatically and the interrupt request flag (CSIIF1) is set.

Fig. 15-6 3-Wire Serial I/O Mode Timings



Caution: SO1 pin becomes low level by SIO1 write.

15.4.3 3-wire serial I/O mode operation with automatic transmit/receive function

This 3-wire serial I/O mode is used for transmission/reception of a maximum of 32-byte data without the use of software. Once transfer is started, the data prestored in the RAM can be transmitted by the set number of bytes, and data can be received and stored in the RAM by the set number of bytes.

Handshake signals (STB and BUSY) are supported by hardware to transmit/receive data continuously. OSD (On Screen Display) LSI and peripheral LSI including LCD controller/driver can be connected without difficulty.

(1) Register setting

The 3-wire serial I/O mode with automatic transmit/receive function is set with the serial operating mode register 1 (CSIM1), the automatic data transmit/receive control register (ADTC) and the automatic data transmit/receive interval specify register (ADTI).

(a) Serial operating mode register 1 (CSIM1)

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM1 to 00H.

The shaded area is used in the 3-wire serial I/O mode with automatic transmit/receive function.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	When Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial Interface Channel 1 Clock Selection
0	x	Clock externally input to $\overline{SCK1}$ pin Note1
1	0	8-bit timer register 2 output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin function
0	MSB	SI1/P20 (Input)	SO1 (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	×	Note2 ×	Note2 ×	Note2 ×	Note2 ×	Note2 ×	Note2 ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note3 1	Note3 ×	0	0	1	×	Operation enable	Count operation	SI1 Note3 (Input)	SO1 (CMOS output)	$\overline{SCK1}$ (Input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes:

(1) If the external clock input has been selected with CSIM11 set to 0, set STRB and BUSY1 of the automatic data transmit/receive control register to 0, 0.

(2) Can be used freely as port function.

(3) Can be used as P20 (CMOS input/output) when only transmitter is used.

Remark: ×: Don't care

(b) Automatic data transmit/receive control register (ADTC)

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

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

The shaded area is used in the 3-wire serial I/O mode with automatic transmit/receive function.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When Reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Busy Input Control
0	×	Not using busy input
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB	Strobe Output Control
0	Strobe output disable
1	Strobe output enable

R

TRF	Status of Automatic Transmit/Receive Function Note 2
0	Detection of termination of automatic transmission/reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error Detection of Automatic Transmit/Receive Function
0	No error (This bit is set to 0 when data is written to SIO1)
1	Error occurred

R/W

ERCE	Error Check Control of Automatic Transmit/Receive Function
0	Error check disable
1	Error check enable (only when BUSY1 = 1)

R/W

ARLD	Operating Mode Selection of Automatic Transmit/Receive Function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive Control of Automatic Transmit/Receive Function
0	Receive disable
1	Receive enable

Notes:

(1) Bits 3 and 4 (TRF and ERR) are Read-Only bit.

(2) The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1.

Caution: When an external clock input is selected with CSIM11 of the serial operating mode register 1 set to 0, set STRB and BUSY1 of ADTC to 0, 0.

Remark: ×: Don't care

(c) Automatic data transmit/receive interval specify register (ADTI)

This register sets the data transfer interval of the automatic data transmit/receive function.

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

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

The shaded area indicates the bits used in automatic data transmit/receive 3-wire serial I/O mode.

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADTI Note 1
★ 1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (fxx = 4.0 MHz Operation)	
					Minimum Note 2	Maximum Note 2
0	0	0	0	0	46.0 μs + 0.5/fsck	50.0 μs + 1.5/fsck
0	0	0	0	1	78.0 μs + 0.5/fsck	82.0 μs + 1.5/fsck
0	0	0	1	0	110.0 μs + 0.5/fsck	114.0 μs + 1.5/fsck
0	0	0	1	1	142.0 μs + 0.5/fsck	146.0 μs + 1.5/fsck
0	0	1	0	0	174.0 μs + 0.5/fsck	178.0 μs + 1.5/fsck
0	0	1	0	1	206.0 μs + 0.5/fsck	210.0 μs + 1.5/fsck
0	0	1	1	0	238.0 μs + 0.5/fsck	242.0 μs + 1.5/fsck
0	0	1	1	1	270.0 μs + 0.5/fsck	274.0 μs + 1.5/fsck
0	1	0	0	0	302.0 μs + 0.5/fsck	306.0 μs + 1.5/fsck
0	1	0	0	1	334.0 μs + 0.5/fsck	338.0 μs + 1.5/fsck
0	1	0	1	0	366.0 μs + 0.5/fsck	370.0 μs + 1.5/fsck
0	1	0	1	1	398.0 μs + 0.5/fsck	402.0 μs + 1.5/fsck
0	1	1	0	0	430.0 μs + 0.5/fsck	434.0 μs + 1.5/fsck
0	1	1	0	1	462.0 μs + 0.5/fsck	466.0 μs + 1.5/fsck
0	1	1	1	0	494.0 μs + 0.5/fsck	498.0 μs + 1.5/fsck
0	1	1	1	1	526.0 μs + 0.5/fsck	530.0 μs + 1.5/fsck

Notes:

(1) The interval is dependent only on CPU processing.

(2) The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/fsck, the minimum interval time is 2/fsck.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_{XX}} + \frac{56}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_{XX}} + \frac{72}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions:

(1) ADTI should not be written to during operation of the automatic data transmit/receive function.

(2) Zero must be set in bits 5 and 6.

Remarks:

(1) fxx : Main system clock frequency

(2) fsck : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (fxx = 5.0 MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	558.0 μs + 0.5/fsck	562.0 μs + 1.5/fsck
1	0	0	0	1	590.0 μs + 0.5/fsck	594.0 μs + 1.5/fsck
1	0	0	1	0	622.0 μs + 0.5/fsck	626.0 μs + 1.5/fsck
1	0	0	1	1	654.0 μs + 0.5/fsck	658.0 μs + 1.5/fsck
1	0	1	0	0	686.0 μs + 0.5/fsck	690.0 μs + 1.5/fsck
1	0	1	0	1	718.0 μs + 0.5/fsck	722.0 μs + 1.5/fsck
1	0	1	1	0	750.0 μs + 0.5/fsck	754.0 μs + 1.5/fsck
1	0	1	1	1	782.0 μs + 0.5/fsck	786.0 μs + 1.5/fsck
1	1	0	0	0	814.0 μs + 0.5/fsck	818.0 μs + 1.5/fsck
1	1	0	0	1	846.0 μs + 0.5/fsck	850.0 μs + 1.5/fsck
1	1	0	1	0	878.0 μs + 0.5/fsck	882.0 μs + 1.5/fsck
1	1	0	1	1	910.0 μs + 0.5/fsck	914.0 μs + 1.5/fsck
1	1	1	0	0	942.0 μs + 0.5/fsck	946.0 μs + 1.5/fsck
1	1	1	0	1	974.0 μs + 0.5/fsck	978.0 μs + 1.5/fsck
1	1	1	1	0	1.006 ms + 0.5/fsck	1.010 ms + 1.5/fsck
1	1	1	1	1	1.038 ms + 0.5/fsck	1.042 ms + 1.5/fsck

Note: The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/fsck, the minimum interval time is 2/fsck.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{fxx} + \frac{56}{fxx} + \frac{0.5}{fsck}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{fxx} + \frac{72}{fxx} + \frac{1.5}{fsck}$$

Cautions:

(1) ADTI should not be written to during operation of the automatic data transmit/receive function.

(2) Zero must be set in bits 5 and 6.

Remarks:

(1) fxx : Main system clock frequency

(2) fsck : Serial clock frequency

(2) Automatic transmit/receive data setting

(a) Transmit data setting

① Write transmit data from the least significant address FAC0H of buffer RAM (up to FADFH at maximum).
The transmit data should be in the order from high-order address to low-order address.
② Set to the automatic data transmit/receive address pointer (ADTP) the value obtained by subtracting 1 from the number of transmit data bytes.

(b) Automatic transmit/receive mode setting

① Set CSIE1 and ATE of the serial operating mode register 1 (CSIM1) to 1.
② Set RE of the automatic data transmit/receive control register (ADTC) to 1.
③ Set a data transmit/receive interval in the automatic data transmit /receive interval specify register (ADTI).
④ Write any value to the serial shift register 1 (SIO1) (transfer start trigger).

Caution: Writing any value to SIO1 orders the start of automatic transmit/receive operation and the written value has no meaning.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data specified with ADTP is transferred to SIO1, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address specified with ADTP.
- ADTP is decremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTP decremental output becomes 00H and address FAC0H data is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, TRF is cleared to 0.

(3) Communication operation

(a) Basic transmission/reception mode

This transmission/reception mode is the same as the 3-wire serial I/O mode in which specified number of data are transmitted/received in 8-bit units.

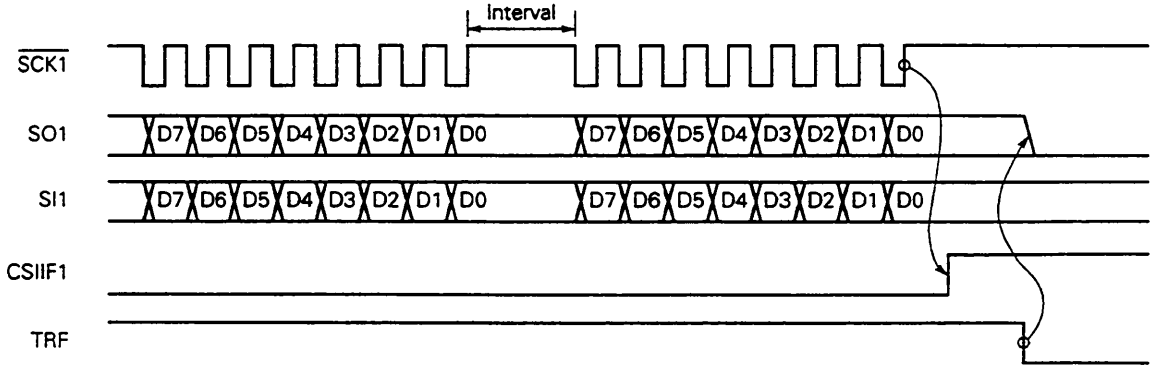
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set.

If busy control and strobe control are not executed, the P23/STB and P24/BUSY pins can be used as normal input/output ports.

Fig. 15-7 shows the basic transmission/reception mode operation timings, Fig. 15-8 shows the operation flowchart.

Fig. 15-7 Basic Transmission/Reception Mode Operation Timings



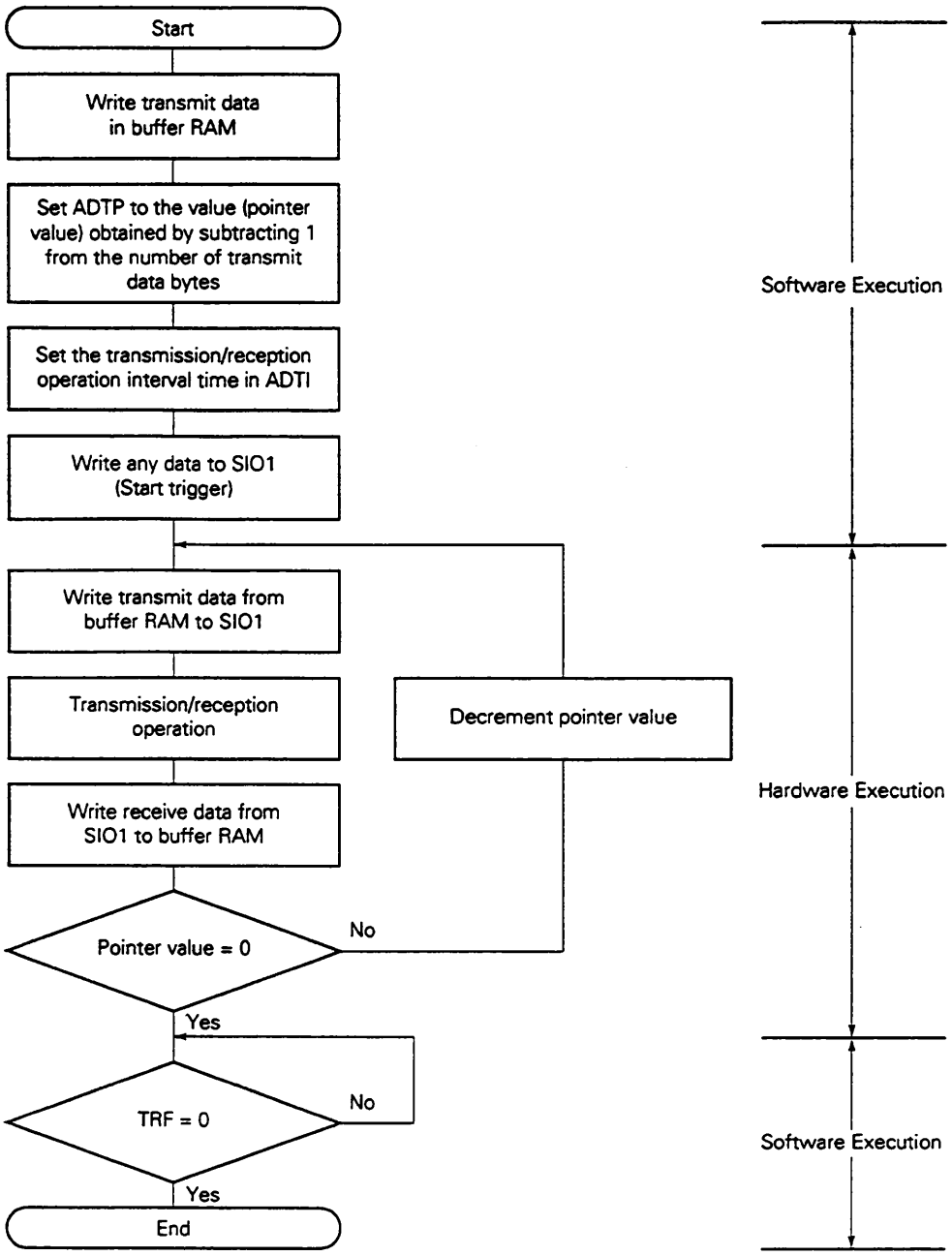
Cautions:

(1) Because, in the basic transmission/reception mode, the automatic transmit/receive function writes/reads data to/from the buffer RAM after 1-byte transmission/reception, an interval is inserted till the next transmission/reception.

As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (h) Automatic data transmit/receive interval).

(2) When TRF is cleared, SO1 pin becomes low level.

Fig. 15-8 Basic Transmission/Reception Mode Flowchart



Remarks:

(1) ADTP : Automatic data transmit/receive address pointer

(2) ADTI : Automatic data transmit/receive interval specify register

(3) SIO1 : Serial I/O shift register 1

In 6-byte transmission/reception (ARLD = 0, RE = 1) in basic transmit/receive mode, buffer RAM operates as follows.

(i) Before transmission/reception

After arbitrary data has been written to SIO1 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the receive data (R1) is transferred from SIO1 to the buffer RAM, and ADTP is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission/reception point

Transmission/reception of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When transmission of the fourth byte is completed, the receive data (R4) is transferred from SIO1 to the buffer RAM, and ADTP is decremented.

(iii) Completion of trnasmission/reception

When transmission of the sixth byte is completed, the receive data (R6) is transferred from SIO1 to the buffer RAM, and the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Fig. 15-9 Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (1/2)

(a) Before transmission/reception

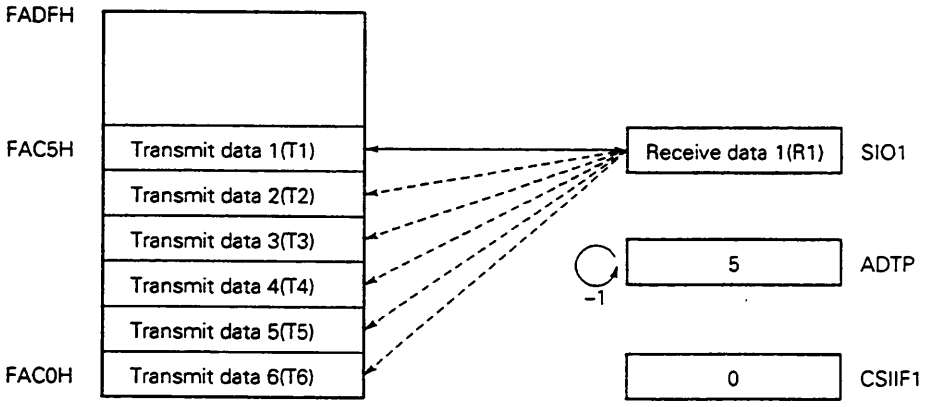
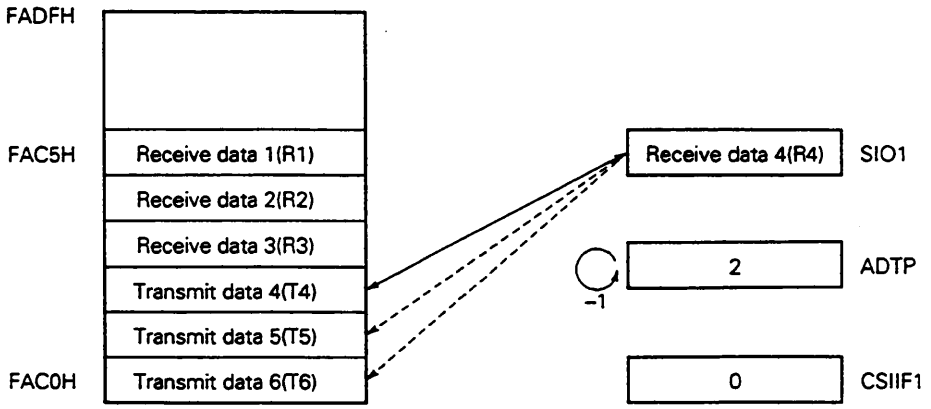
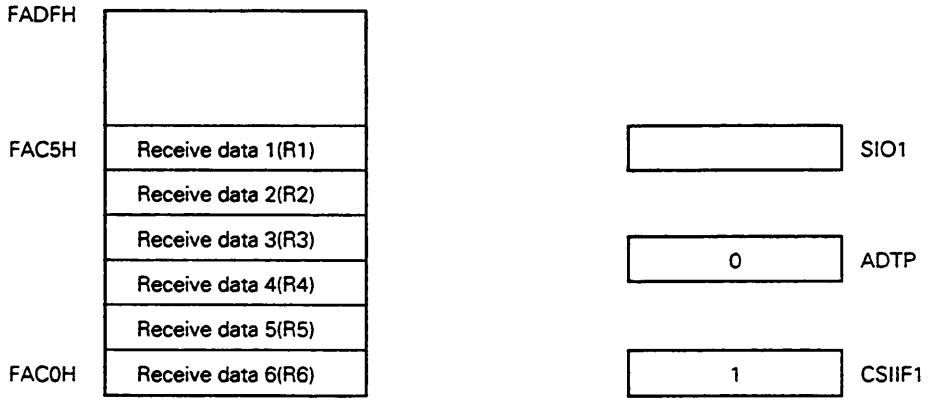


Fig. 15-9 Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (2/2)

(b) 4th byte transmission/reception point



(c) Completion of transmission/reception



(b) Basic transmission mode

In this mode, specified number of 8-bit unit data are transmitted.

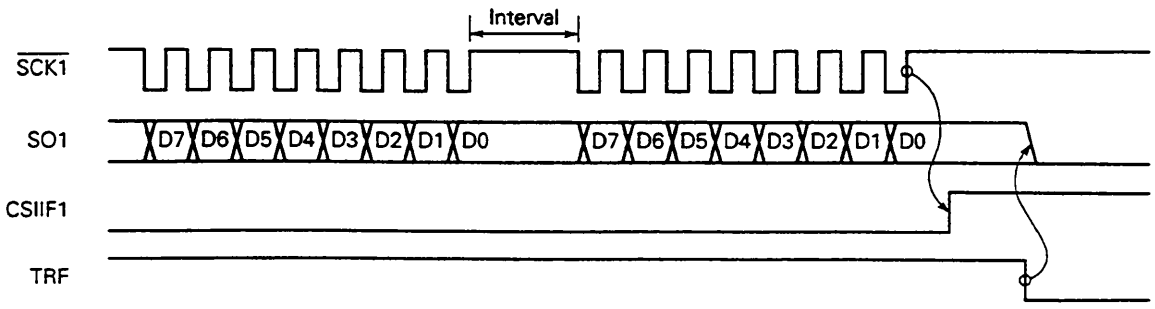
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set.

If receive operation, busy control and strobe control are not executed, the P20/SI1, P23/STB and P24/BUSY pins can be used as normal input/output ports.

Fig. 15-10 shows the basic transmission mode operation timings, Fig. 15-11 shows the operation flowchart.

Fig. 15-10 Basic Transmission Mode Operation Timings

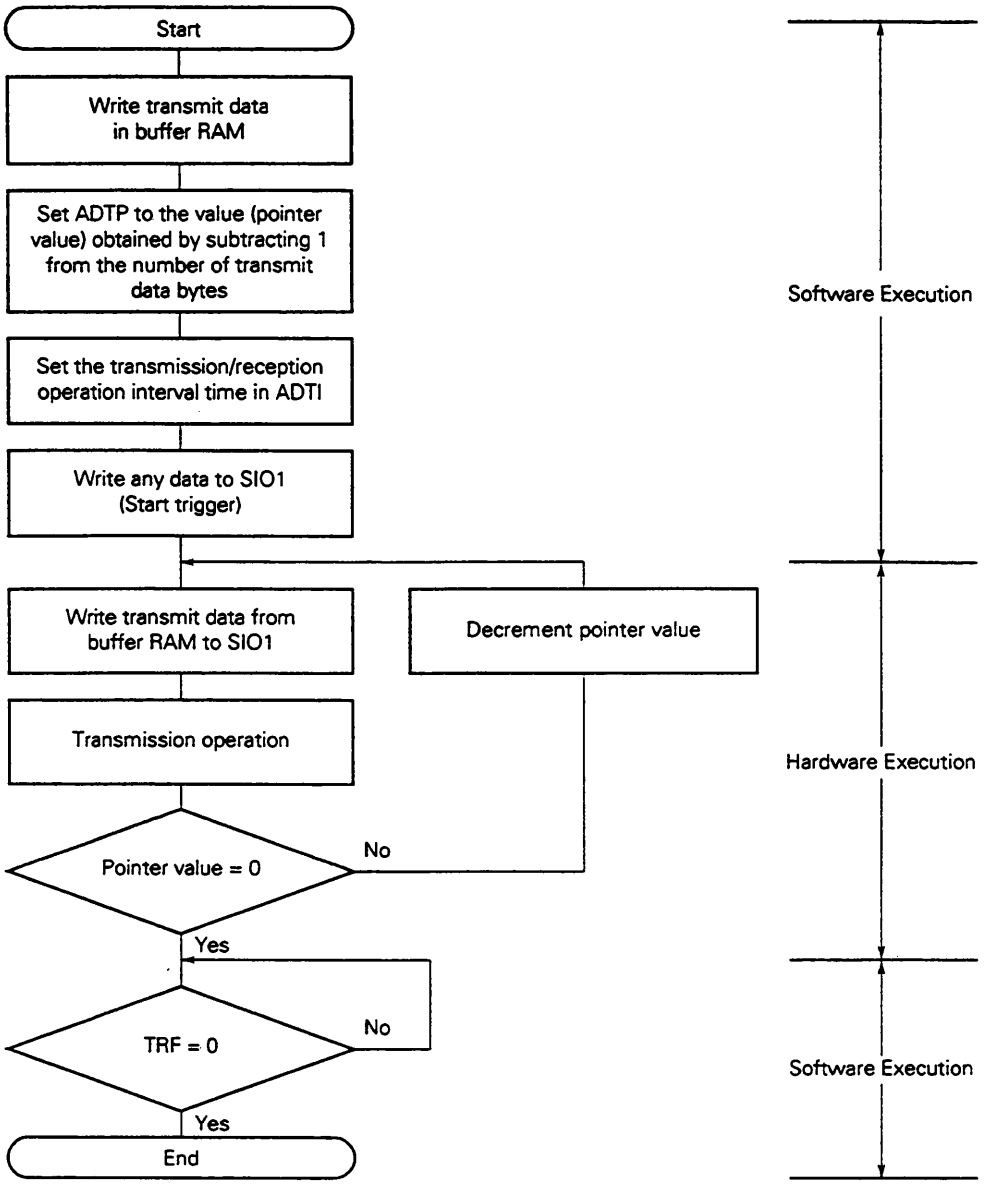


Cautions:

(1) Because, in the basic transmission mode, the automatic transmit/receive function reads data from the buffer RAM after 1-byte transmission, an interval is inserted till the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (h) Automatic data transmit/receive interval).

(2) When TRF is cleared, SO1 pin becomes low level.

Fig. 15-11 Basic Transmission Mode Flowchart



Remarks:

(1) ADTP : Automatic data transmit/receive address pointer

(2) ADTI : Automatic data transmit/receive interval specify register

(3) SIO1 : Serial I/O shift register 1

In 6-byte transmission (ARLD = 0, RE = 0) in basic transmit mode, buffer RAM operates as follows.

(i) Before transmission

After arbitrary data has been written to SIO1 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, ADTP is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission point

Transmission of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When transmission of the fourth byte is completed, ADTP is decremented.

(iii) Completion of trnasmission/reception

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Fig. 15-12 Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (1/2)

(a) Before transmission

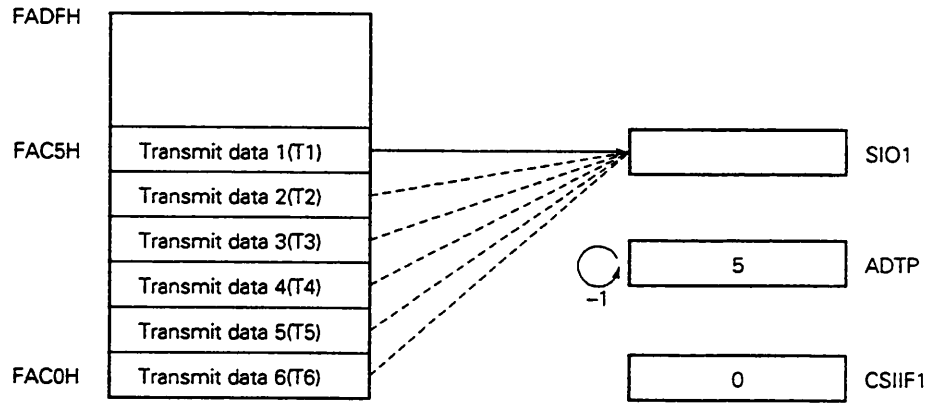
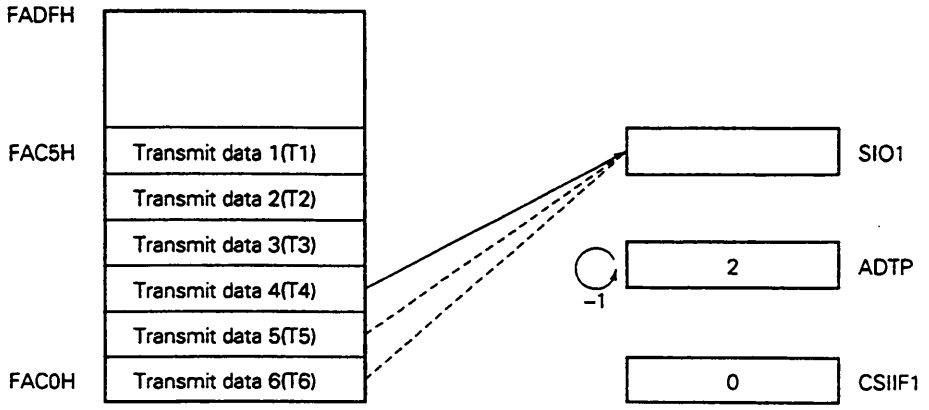
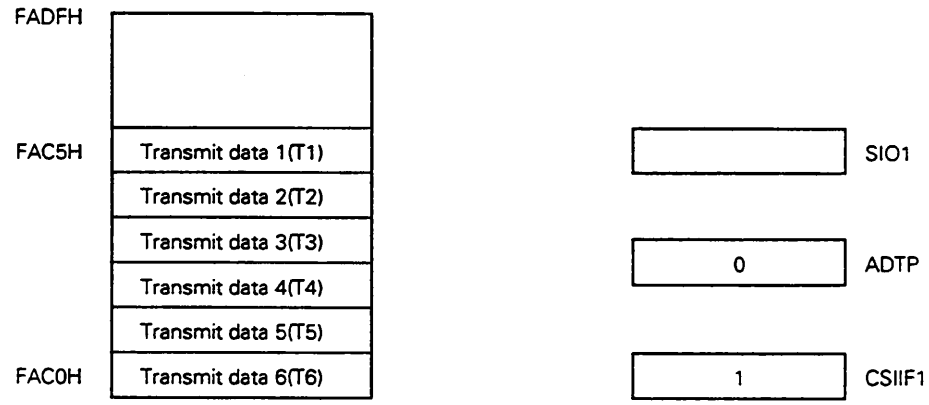


Fig. 15-12 Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (2/2)

(b) 4th byte transmission point



(c) Completion of transmission/reception



(c) Repeat transmission mode

In this mode, data stored in the buffer RAM is transmitted repeatedly.

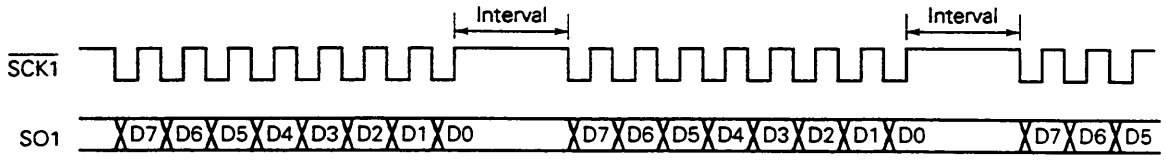
Serial transfer is started by writing any data to the serial I/O shift register 1 (SIO1) when 1 is set in bit 7 (CSIE1) of serial operating mode register 1 (CSIM1).

Unlike the basic transmission mode, after the last byte (data in address FAC0H) has been transmitted, the interrupt request flag (CSIIF1) is not set, the value at the time when the transmit/receive function was started is set in the automatic data transmit/receive address pointer (ADTP) again, and the buffer RAM contents are transmitted again.

When a reception operation, busy control and strobe control are not performed, the P20/SI1, P23/STB and P24/BUSY pins can be used as normal input/output ports.

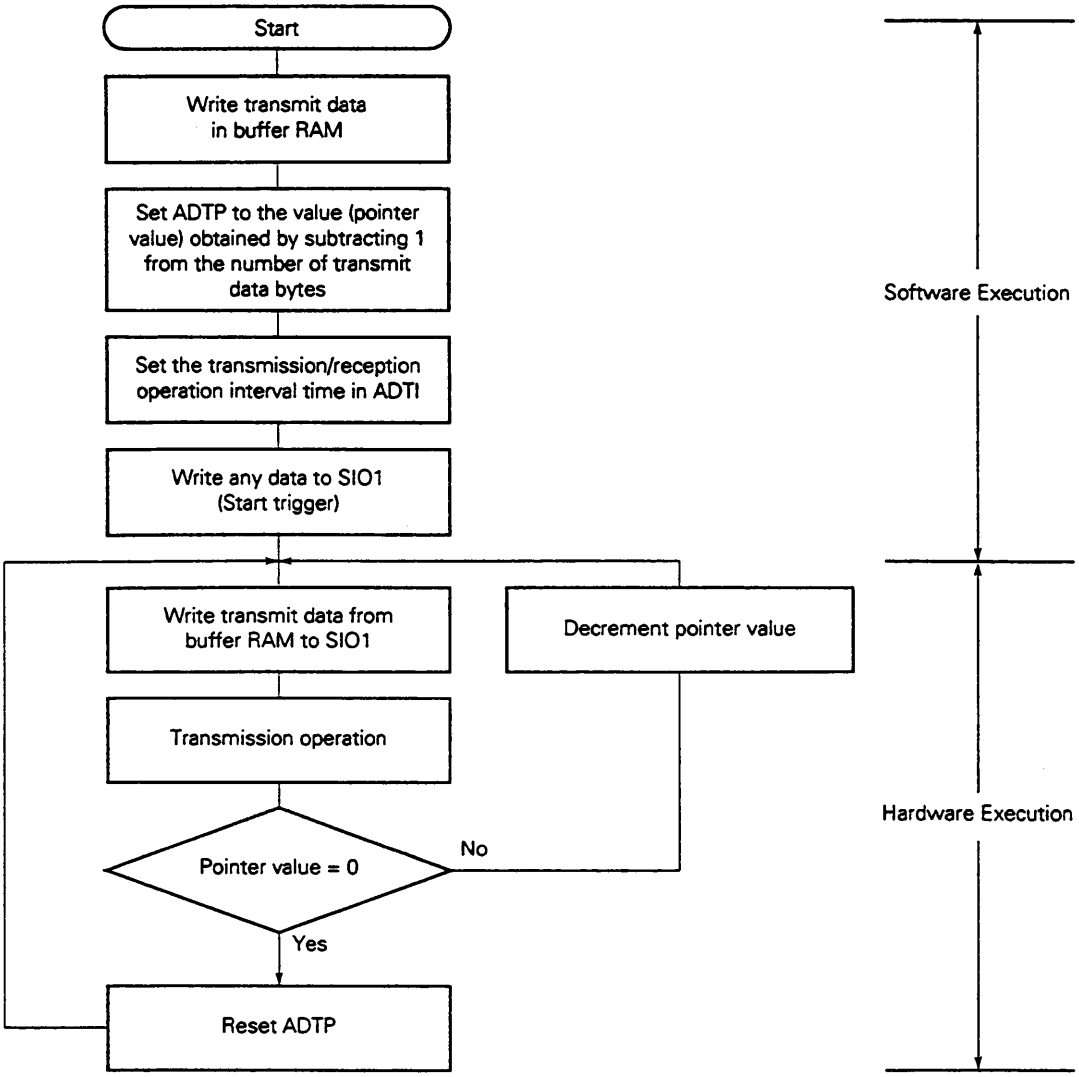
The repeat transmission mode operation timing is shown in Fig. 15-13, and the operation flowchart in Fig. 15-14.

Fig. 15-13 Repeat Transmission Mode Operation Timings



Caution: Since, in the repeat transmission mode, a read is performed on the buffer RAM after the transmission of one byte, the interval is included in the period up to the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon the CPU operation and the value of the automatic data transmit/receive interval specify register (ADTI) (see (h) Automatic data transmit/receive interval).

Fig. 15-14 Repeat Transmission Mode Flowchart



Remarks:

(1) ADTP : Automatic data transmit/receive address pointer

(2) ADTI : Automatic data transmit/receive interval specify register

(3) SIO1 : Serial I/O shift register 1

In 6-byte transmission (ARLD = 1, RE = 0) in repeat transmit mode, buffer RAM operates as follows.

(i) Before transmission

After arbitrary data has been written to SIO1 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, ADTP is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) Upon completion of transmission of 6th bytes

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is not set.

(iii) 7th byte transmission point

Transmit data (T1) is transferred from the buffer RAM to SIO1 again. When transmission of the first byte is completed, ADTP is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

Fig. 15-15 Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (1/2)

(a) Before transmission

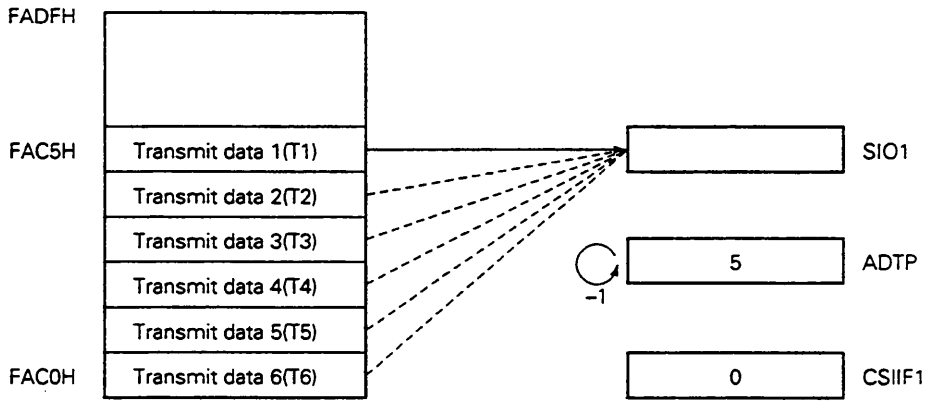
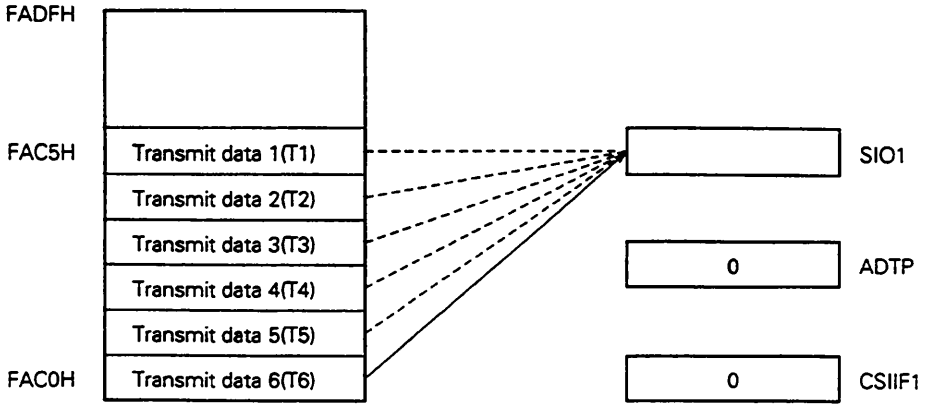
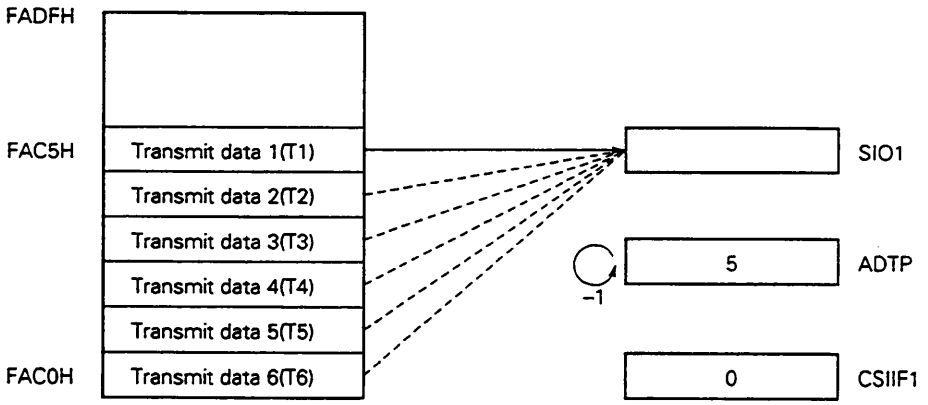


Fig. 15-15 Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (2/2)

(b) Upon completion of transmission of 6 bytes



(c) 7th byte transmission point



(d) Busy control option

When bit 5 (ATE) of the serial operating mode register (CSIM1) and bit 1 (BUSY1) of the automatic data transmit/receive control register (ADTC) are both set to 1, busy control can be used. At this time, the BUSY/P24 pin is sampled. While the BUSY signal is input, transmission/reception can be put to a wait state.

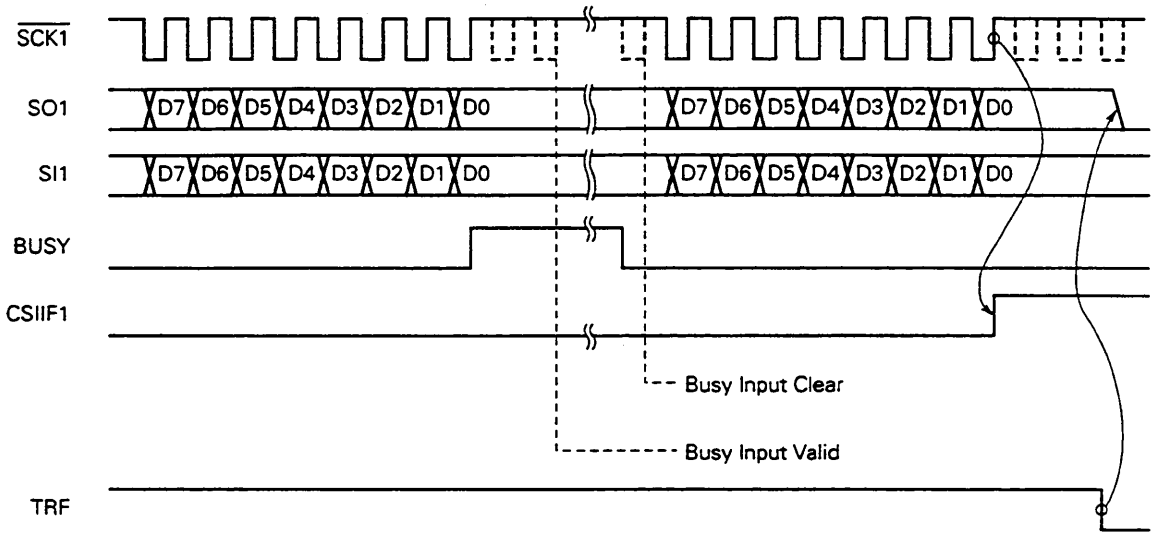
When bit 0 (BUSY0) of ADTC is cleared to 0, the BUSY signal active level is judged to be high. When BUSY0 is set to 1, the active level is judged to be low.

Since strobe control is not carried out, the P23/STB pin can be used as normal input/output port.

Fig. 15-16 shows the operation timings when using the busy control option.

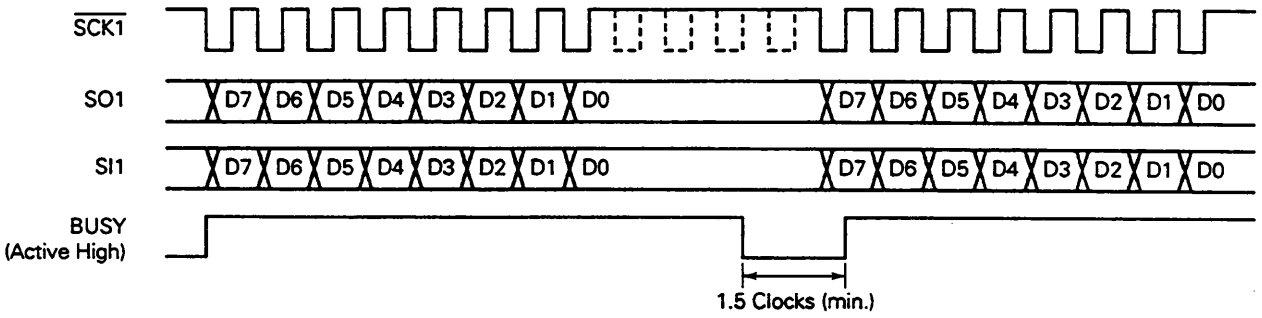
Caution: Select an internal clock for serial clock. The busy control transmit/receive mode cannot be operated with an external clock.

Fig. 15-16 Operation Timings when Using Busy Control Option (BUSY0 = 0)



Caution: When TRF is cleared, SO1 pin becomes low level.

Upon start of transmission/reception, the BUSY signal can be made active.
In this case, if 10th BUSY signals onward are active, a wait is applied in accordance with the active period. To clear the wait, input the inactive level during a minimum period of 1.5 clocks.



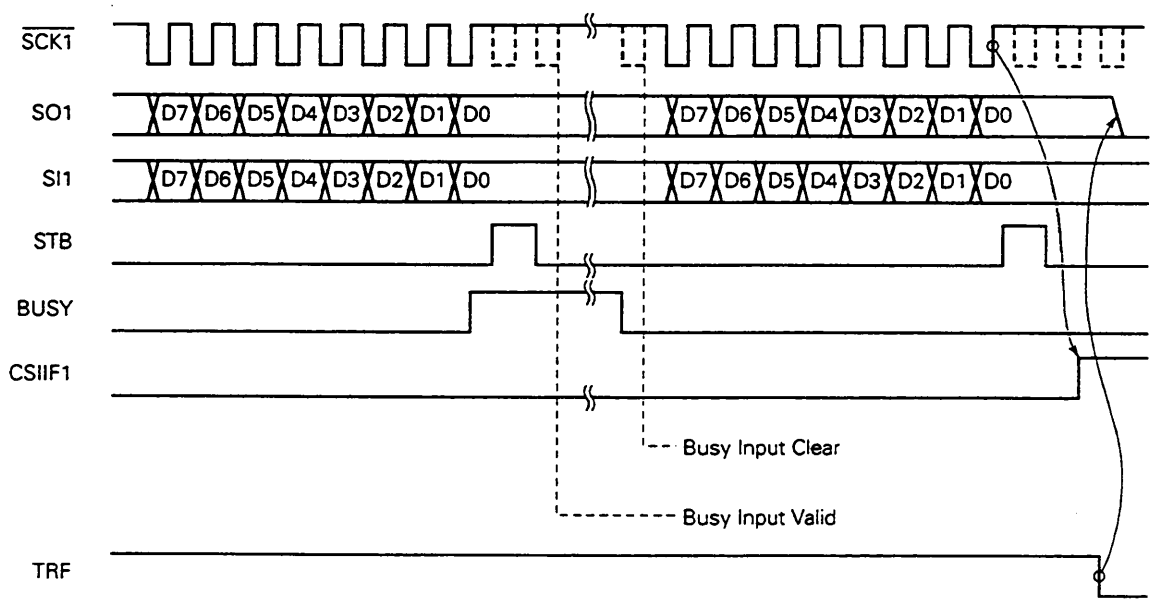
(e) Busy & strobe control option

When bit 5 (ATE) of the serial operating mode register 1 (CSIM1) and bits 1 (BUSY1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) are all set to 1, the busy & strobe control can be used. At this time, the strobe signal is output from the STB/P23 pin. While the BUSY/P24 pin is sampled and the BUSY signal is input in this mode, transmission/reception can be put to a wait state.

When bit 0 (BUSY0) of ADTC is cleared to 0, the BUSY signal active level is judged to be high. When BUSY0 is set to 1, the active level is judged to be low.

Fig. 15-17 shows operation timings when using the busy & strobe control option.

Fig. 15-17 Operation Timings when Using Busy & Strobe Control Option (BUSY0 = 0)



Caution: When TRF is cleared, SO1 pin becomes low level.

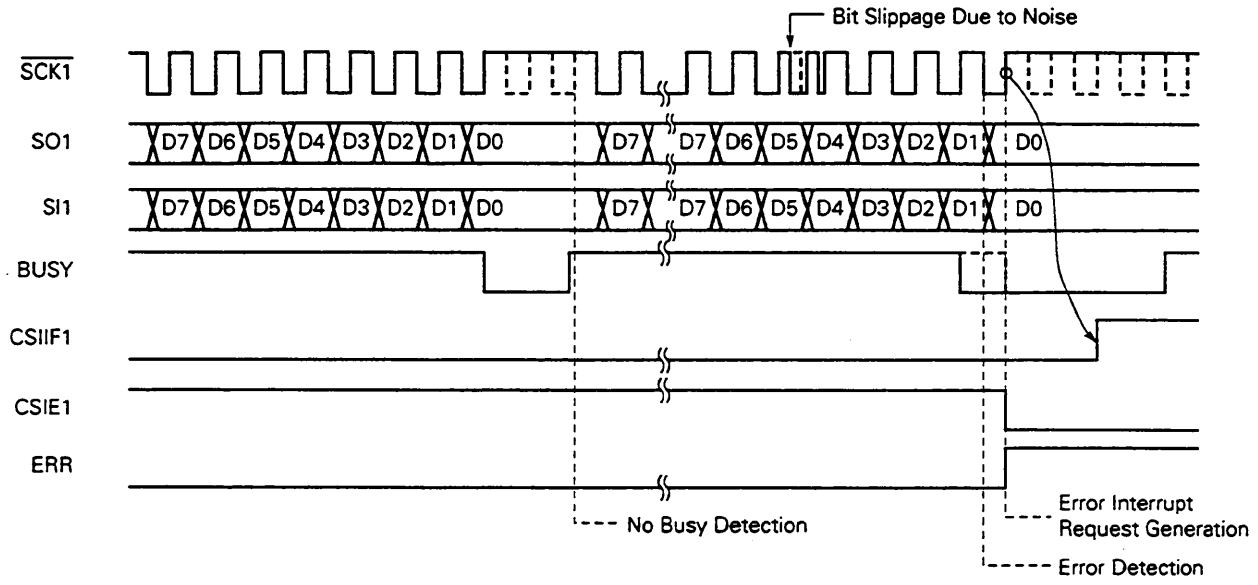
(f) Bit slippage detection function

If noise is carried on the serial clock signal during automatic transmission, causing bit slippage, when the strobe control option is used, synchronization is achieved and therefore the bit slippage has no effect when the next byte is transmitted, but there is an effect when the strobe control option is not used.

When the strobe control option is not used, the transmission side can detect bit slippage by checking the BUSY signal during transmission by means of the BUSY control option. The BUSY signal is output after the 8th rise of the serial clock, and the BUSY signal is taken in synchronization with the fall of the serial clock signal, and if active, this is interpreted as indicating bit slippage, and error handling is performed (bit 4 (ERR) of the automatic data transmit/receive control register is set to 1).

The bit slippage detection function operation timing is shown in Fig. 15-18.

Fig. 15-18 Bit Slippage Detection Function Operation Timing



(g) Automatic transmission/reception suspending and restart

Automatic transmission/reception can be temporarily suspended by setting bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) to 0.

If during 8-bit data transfer, the transmission/reception is not suspended if bit 7 (CSIE1) is set to 0. It is suspended upon completion of 8-bit data transfer.

When suspended, bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) is set to 0 after transfer of the 8th bit, and all the port pins used with the serial interface pins for dual function (P20/SI1, P21/SO1, P22/$\overline{SCK1}$, P23/STB and P24/BUSY) are set to the port mode.

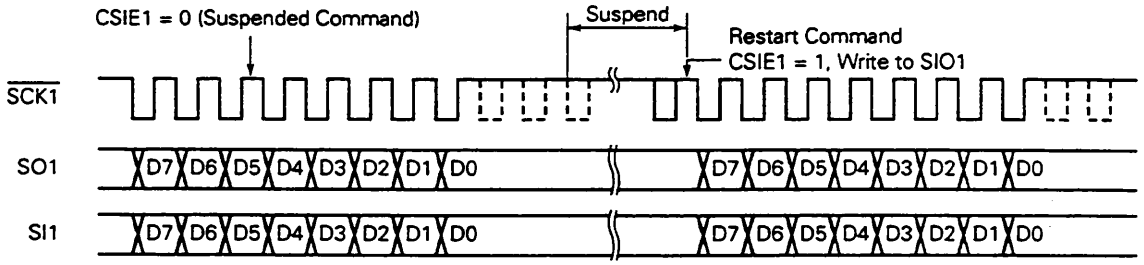
Automatic transmission/reception can be restarted and the remaining data can be transferred by setting CSIE1 to 1 and writing any data to the serial I/O shift register 1 (SIO1).

Cautions:

(1) If the HALT instruction is executed during automatic transmission/reception, transfer is suspended and the HALT mode is set if during 8-bit data transfer. When the HALT mode is cleared, automatic transmission/reception is restarted at the suspended point.

(2) When suspending automatic transmission/reception, do not change the operating mode to 3-wire serial I/O mode during TRF = 1.

Fig. 15-19 Automatic Transmission/Reception Suspension and Restart



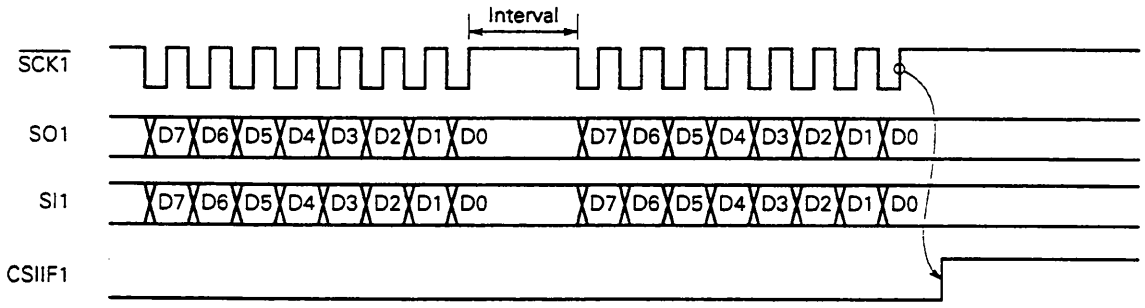
★ **(h) Automatic data transmit/receive interval**

When the automatic data transmit/receive function is used, an interval is set until the next transmit/receive after 1 byte is transmitted/received for writing/reading in/from the buffer RAM to be performed.

When the automatic transmit/receive function is operated using the internal clock, because buffer RAM writing/reading is performed at the same time as the CPU processing in the 8th rising edge timing of the serial clock, the interval time depends on the values set for CPU processing and automatic data transmit/receive interval specify register (ADTI). Whether to depend on ADTI can be selected by setting bit 7 (ADTI7) of ADTI. When ADTI7 is set to 0, the interval time will depend only on the CPU processing. When 1 is set to ADTI7, the interval time will be the interval time determined by the contents set to ADTI or the interval time by CPU processing, whichever is the greater one.

When the automatic transmit/receive function is operated using the external clock, an external clock which will cause the interval time to be greater than the time shown by (ii) has to be input.

Fig. 15-20 Automatic Data Transmit/Receive Interval



(i) When the automatic transmit/receive function is operated by the internal clock (CSIM11=1)

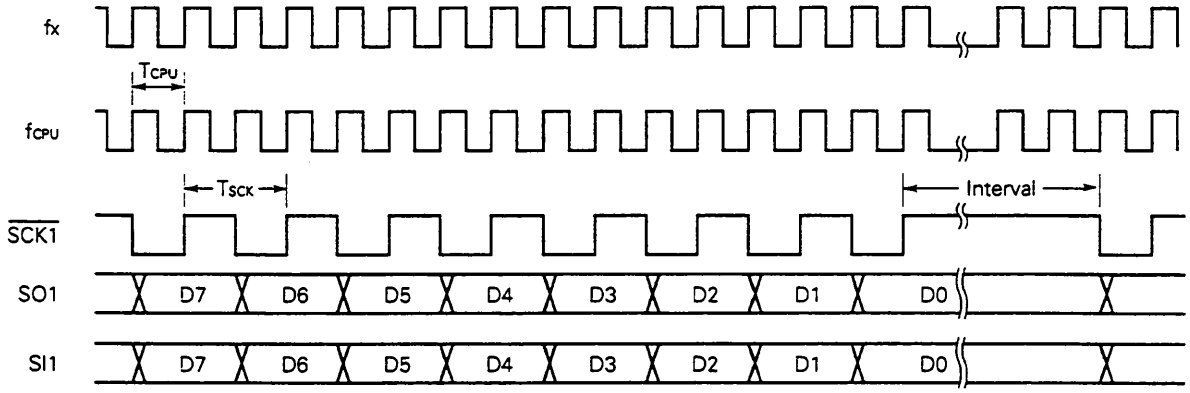
When the automatic transmit/receive function is operated by the internal clock, the interval time by CPU processing will be as follows.

When 0 is set to ADTI7, the interval itme will be the interval time by CPU processing. When 1 is set to ADTI7, the interval time will be the interval time determined by the contents set to ADTI or the interval time by CPU processing, whichever is the greater one.

For the interval times by ADTI, refer to **Fig.15-5 Automatic Data Transmit/Receive Interval Specify Register Format**.

CPU Processing	Interval Time
When multiply instruction is used	MAX. (2.5T_{SCK}, 13T_{CPU})
When divide instruction is used	MAX. (2.5T_{SCK}, 20T_{CPU})
External access 1 wait mode	MAX. (2.5T_{SCK}, 9T_{CPU})
Other than above	MAX. (2.5T_{SCK}, 7T_{CPU})

T_{SCK} : 1/f_{SCK}
f_{SCK} : Serial clock frequency
T_{CPU} : 1/f_{CPU}
f_{CPU} : CPU clock (set with processor clock control register bits 0 to 2 (PCC0 to PCC2), oscillation mode selection register bit 0 (MCS))
MAX. (a, b) : Greater value of a, or b
CSIM11 : Serial operation mode register 1 (CSIM1) bit 1



(ii) When the automatic transmit/receive function is operated by the external clock (CSIM11=0)

When the automatic transmit/receive function is operated by the external clock, an external clock which will cause the interval time to be greater than the time shown below has to be input.

CPU Processing	Interval Time
When multiply instruction is used	Above 13T_{CPU}
When divide instruction is used	Above 20T_{CPU}
External access 1 wait mode	Above 9T_{CPU}
Other than above	Above 7T_{CPU}

T_{CPU} : 1/f_{CPU}
f_{CPU} : CPU clock (set with processor clock control register bits 0 to 2 (PCC0 to PCC2), oscillation mode selection register bit 0 (MCS))
CSIM11 : Serial operation mode register 1 (CSIM1) bit 1

[MEMO]

CHAPTER 16 SERIAL INTERFACE CHANNEL 2

16.1 Serial Interface Channel 2 Functions

Serial interface channel 2 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) 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) Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

(3) 3-wire serial I/O mode (MSB-first/LSB-first switchable)

In this mode, 8-bit data transfer is performed using three lines: the serial clock ($\overline{SCK2}$), and serial data lines (SI2, SO2).

In the 3-wire serial I/O mode, simultaneous transmission and reception are possible, increasing the data transfer processing speed.

Either the MSB or LSB can be specified as the start bit for an 8-bit data serial transfer, allowing connection to devices using either as the start bit.

The 3-wire serial I/O mode is useful for connection to peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous clocked serial interface, such as the 75X series, 78K series 17K series, etc.

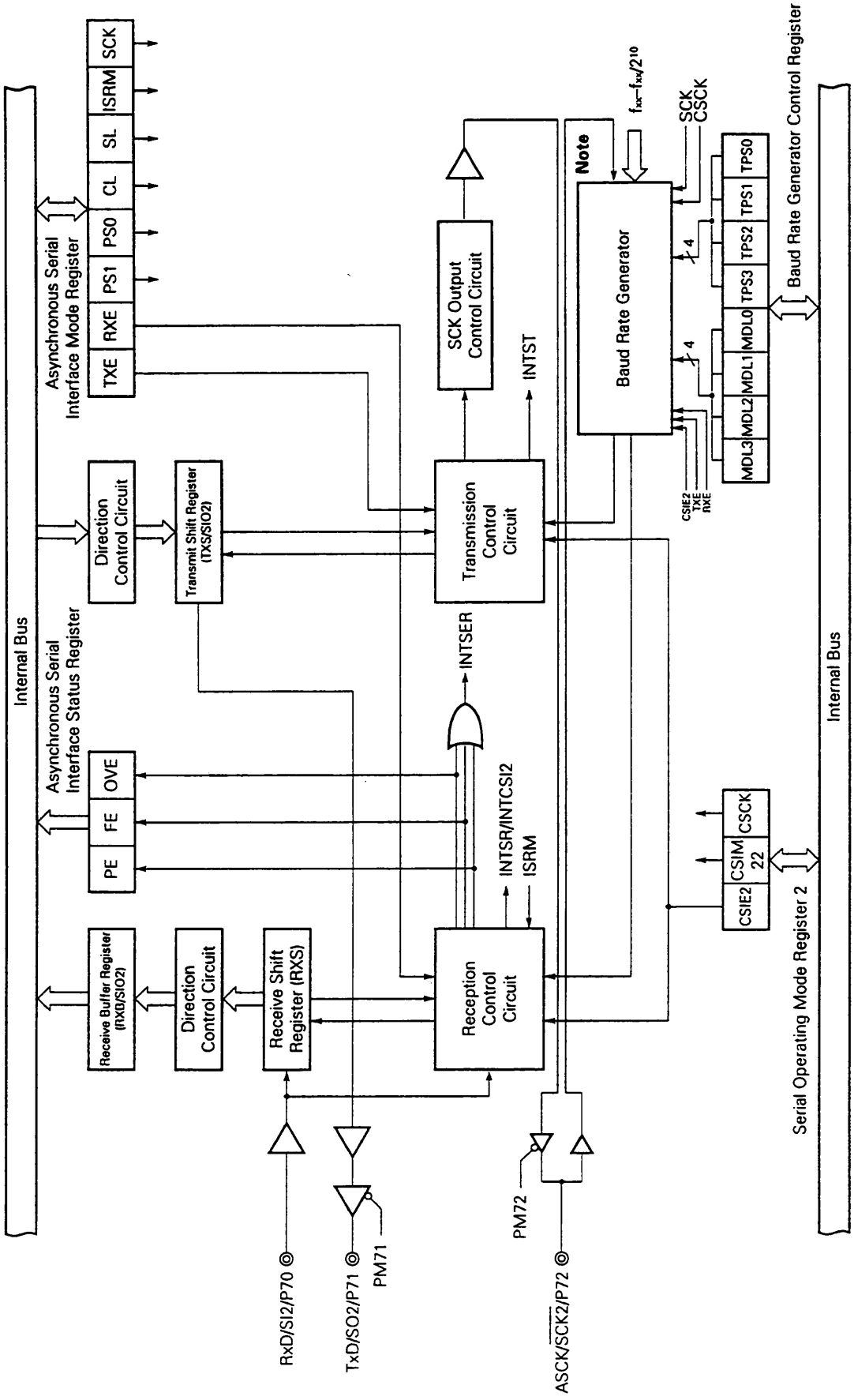
16.2 Serial Interface Channel 2 Configuration

Serial interface channel 2 consists of the following hardware.

Table 16-1 Serial Interface Channel 2 Configuration

Item	Configuration
Registers	Transmit shift register (TXS) Receive shift register (RXS) Receive buffer register (RXB)
Control registers	Serial operating mode register 2 (CSIM2) Asynchronous serial interface mode register (ASIM) Asynchronous serial interface status register (ASIS) Baud rate generator control register (BRGC)

Fig. 16-1 Serial Interface Channel 2 Block Diagram ★



Note: See Fig. 16-2 for the baud rate generator configuration.

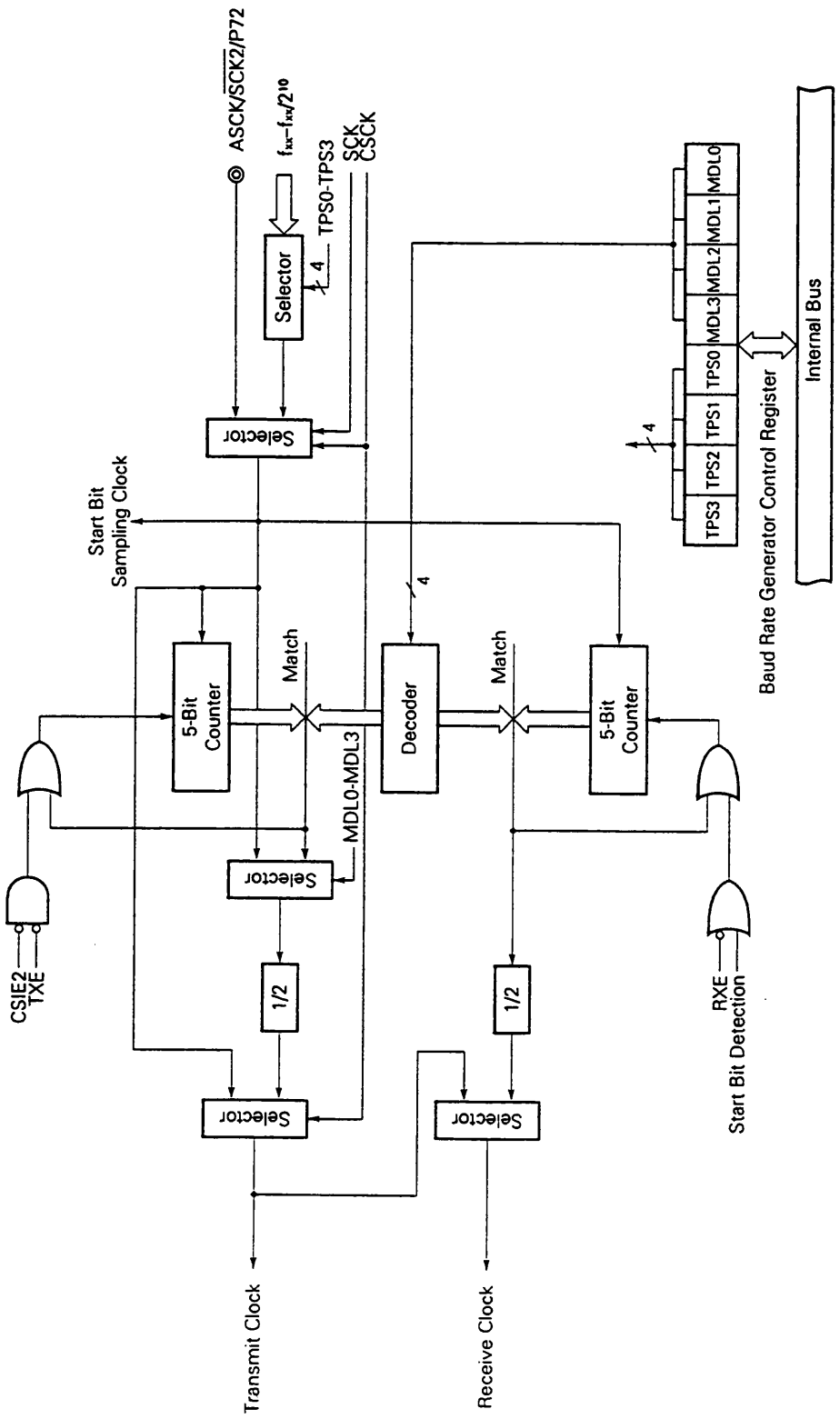


Fig. 16-2 Baud Rate Generator Block Diagram

(1) Transmit shift register (TXS)

This register is used to set the transmit data. The data written in TXS is transmitted as serial data.

If the data length is specified as 7 bits, bits 0 to 6 of the data written in TXS is transferred as transmit data. Writing data to TXS starts the transmit operation.

TXS is written to with an 8-bit memory manipulation instruction. It cannot be read.

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

Caution: TXS must not be written to during a transmit operation.
TXS and the receive buffer register (RXB) are allocated to the same address, and when a read is performed, the value of RXB is read.

(2) Receive shift register (RXS)

This register is used to convert serial data input to the RxD pin to parallel data. When one byte of data is received, the receive data is transferred to the receive buffer register (RXB).

RXS cannot be directly manipulated by a program.

(3) Receive buffer register (RXB)

This register holds receive data. Each time one byte of data is received, new receive data is transferred from the receive shift register (RXS).

If the data length is specified as 7 bits, the receive data is transferred to bits 0 to 6 of RXB, and the MSB of RXB is always set to 0.

RXB is read with an 8-bit memory manipulation instruction. It cannot be written to.

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

Caution: RXB and the transmit shift register (TXS) are allocated to the same address and when a write is performed, the value is written to TXS.

(4) Transmission control circuit

This circuit performs transmit operation control such as the addition of a start bit, parity bit and stop bit to data written in the transmit shift register (TXS) in accordance with the contents set in the asynchronous serial interface mode register (ASIM).

(5) Reception control circuit

This circuit controls receive operations in accordance with the contents set in the asynchronous serial interface mode register (ASIM). It performs error checks for parity errors, etc., during a receive operation, and if an error is detected, sets a value in the asynchronous serial interface status register (ASIS) in accordance with the error contents.

16.3 Serial Interface Channel 2 Control Registers

Serial interface channel 2 is controlled by the following four registers.

- Serial Operating Mode Register 2 (CSIM2)
- Asynchronous Serial Interface Mode Register (ASIM)
- Asynchronous Serial Interface Status Register (ASIS)
- Baud Rate Generator Control Register (BRGC)

(1) Serial operating mode register 2 (CSIM2)

This register is set when serial interface channel 2 is used in the 3-wire serial I/O mode.

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM2 to 00H.

Fig. 16-3 Serial Operating Mode Register 2 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	When Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{\text{SCK2}}$ pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Cautions:

(1) Ensure that 0 is set in bits 0 and 3 to 6.

(2) When UART mode is selected, 00H should be set in CSIM2.

(2) Asynchronous serial interface mode register (ASIM)

This register is set when serial interface channel 2 is used in the asynchronous serial interface mode.

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

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

Fig. 16-4 Asynchronous Serial Interface Mode Register Format

Symbol	⑦	⑥	5	4	3	2	1	0	Address	When Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output Note

ISRM	Control of Reception Completion Interrupt in Case of Error Generation
0	Reception completion interrupt generated in case of error generation
1	Reception completion interrupt not generated in case of error generation

SL	Transmit Data Stop Bit Length Specification
0	1 bit
1	2 bits

CL	Character Length Specification
0	7 bits
1	8 bits

PS1	PS0	Parity Bit Specification
0	0	No Parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

Note: When SCK is set to 1 and the baud rate generator output is selected, the ASCK pin can be used as an input/output port.

Cautions:

(1) When the 3-wire serial I/O mode is selected, 00H should be set in ASIM.

(2) The serial transmit/receive operation must be stopped before changing the operating mode.

Table 16-2 Serial Interface Channel 2 Operating Mode Settings ★

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	Operating Mode	Start Bit	Shift Clock	P70/SI2/RxD Pin Function	P71/SO2/TxD Pin Function	P72/$\overline{SCK2}$/ASCK Pin Function
TXE	RXE	SCK	CSIE2	CSIM22	CSCK												
0	0	×	0	×	×	×[Note1]	×[Note1]	×[Note1]	×[Note1]	×[Note1]	×[Note1]	Operation stopped	–	–	P70	P71	P72
0	0	0	1	0	0	1[Note2]	×[Note2]	0	1	1	×	3-wire serial I/O mode	MSB	External clock	SI2[Note2]	SO2 (CMOS output)	$\overline{SCK2}$ input
					1					0	1			Internal clock			$\overline{SCK2}$ output
			1	1	0					1	×		LSB	External clock	SI2[Note2]	SO2 (CMOS output)	$\overline{SCK2}$ input
					1					0	1			Internal clock			$\overline{SCK2}$ output
1	0	0	0	0	0	×[Note1]	×	0	1	1	×	Asynchronous serial interface mode	LSB	External clock	P70	TxD (CMOS output)	ASCK input
		1								×[Note1]	×[Note1]			Internal clock			P72
0	1	0	0	0	0	1	×	×[Note1]	×[Note1]	1	×			External clock	RxD	P71	ASCK input
		1								×[Note1]	×[Note1]			Internal clock			P72
1	1	0	0	0	0	1	×	0	1	1	×			External clock		TxD (CMOS output)	ASCK input
		1								×[Note1]	×[Note1]			Internal clock			P72
Other than above												Setting prohibited					

Notes:

(1) Can be used freely as port function.

(2) Can be used as P70 (CMOS input/output) when only transmitter is used.

Remark: ×: Don't care

(3) Asynchronous serial interface status register (ASIS)

This is a register which displays the type of error when a reception error is generated in the asynchronous serial interface mode.

ASIS is read with a 1-bit or 8-bit memory manipulation instruction.

In 3-wire serial I/O mode, the contents of the ASIS are undefined.

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

Fig. 16-5 Asynchronous Serial Interface Status Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
ASIS	0	0	0	0	0	PE	FE	OVE	FF71H	00H	R

OVE	Overrun Error Flag
0	Overrun error not generated
1	Overrun error generated Note 1 (When next receive operation is completed before data from receive buffer register is read)

FE	Framing Error Flag
0	Framing error not generated
1	Framing error generated Note 2 (When stop bit is not detected)

PE	Parity Error Flag
0	Parity error not generated
1	Parity error generated (When transmit data parity does not match)

Notes:

(1) The receive buffer register (RXB) must be read when an overrun error is generated. Overrun errors will continue to be generated until RXB is read.

(2) Even if the stop bit length has been set as 2 bits by bit 2 (SL) of the asynchronous serial interface mode register, only single stop bit detection is performed during reception.

(4) Baud rate generator control register (BRGC)

This register sets the serial clock when the asynchronous serial interface mode is used.

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

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

Fig. 16-6 Baud Rate Generator Control Register Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	f_{SCK} Note	—

Note: Can only be used in 3-wire serial I/O mode.

Remarks:

(1) f_{SCK} : 5-bit counter source clock

(2) k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Fig. 16-6 Baud Rate Generator Control Register Format (2/2)

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection	n
0	0	0	0	$f_{XX}/2^{10}$ (3.91 kHz)	11
0	1	0	1	f_{XX} (4.0 MHz)	1
0	1	1	0	$f_{XX}/2$ (2.0 MHz)	2
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)	3
1	0	0	0	$f_{XX}/2^3$ (500 kHz)	4
1	0	0	1	$f_{XX}/2^4$ (250 kHz)	5
1	0	1	0	$f_{XX}/2^5$ (125 kHz)	6
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)	7
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)	8
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)	9
1	1	1	0	$f_{XX}/2^9$ (7.81 kHz)	10
Other than above				Setting prohibited	

Caution: When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.

Remarks:

(1) f_{XX} : Main system clock frequency

(2) n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

(3) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(a) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clock is generated by scaling the main system clock. The baud rate generated from the main system clock is found from the following expression.

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k + 16)} \text{ [Hz]}$$

Remarks:

(1) fxx : Main system clock frequency

(2) n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

(3) k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 16-3 Relationship between Main System Clock and Baud Rate

Baud Rate (bps)	fxx = 4.0 MHz	
	Set value of BRGC	Error (%)
75	0AH	0.16
110	02H	-1.36
150	EAH	0.16
300	DAH	0.16
600	CAH	0.16
1200	BAH	0.16
2400	AAH	0.16
4800	9AH	0.16
9600	8AH	0.16
19200	7AH	0.16
31250	70H	0
38400	6AH	0.16

(b) Generation of baud rate transmit/receive clock by means of external clock from ASCK pin

The transmit/receive clock is generated by scaling the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is found from the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{2 \times (k + 16)} \quad [\text{Hz}]$$

Remarks:

(1) f_{ASCK} : Frequency of clock input to ASCK pin

(2) k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 16-4 Relationship between ASCK Pin Input Frequency and Baud Rate (when BRGC is set to 00H)

Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1200	38.4 kHz
2400	76.8 kHz
4800	153.6 kHz
9600	307.2 kHz
19200	614.4 kHz
31250	1000.0 kHz
38400	1228.8 kHz

16.4 Serial Interface Channel 2 Operation

Serial interface channel 2 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

16.4.1 Operation stop mode

In the operation stop mode, serial transfer is not performed, and therefore power consumption can be reduced.

In the operation stop mode, the P70/SI2/RxD, P71/SO2/TxD and P72/$\overline{\text{SCK2}}$/ASCK pins can be used as normal input/output ports.

(1) Register setting

Operation stop mode settings are performed using serial operating mode register 2 (CSIM2) and the asynchronous serial interface mode register (ASIM).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM2 to 00H.

The bit used in the operation stop mode is indicated by shading.

Symbol	⑦	6	5	4	3	2	1	0	Address	When Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution: Ensure that 0 is set in bits 0 and 3 to 6.

(b) Asynchronous serial interface mode register (ASIM)

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

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

The bits used in the operation stop mode are indicated by shading.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	When Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

16.4.2 Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

(1) Register setting

UART mode settings are performed using serial operating mode register 2 (CSIM2), the asynchronous serial interface mode register (ASIM), the asynchronous serial interface status register (ASIS), and the baud rate generator control register (BRGC).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM2 to 00H.

The bits used in the UART mode are indicated by shading.

When the UART mode is selected, 00H should be set in CSIM2.

Symbol	⑦	6	5	4	3	2	1	0	Address	When Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{\text{SCK2}}$ pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution: Ensure that 0 is set in bits 0 and 3 to 6.

(b) Asynchronous serial interface mode register (ASIM)

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

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

The bits used in the UART mode are indicated by shading.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	When Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output Note

ISRM	Control of Reception Completion Interrupt in Case of Error Generation
0	Reception completion interrupt generated in case of error generation
1	Reception completion interrupt not generated in case of error generation

SL	Transmit Data Stop Bit Length Specification
0	1 bit
1	2 bits

CL	Character Length Specification
0	7 bits
1	8 bits

PSI	PS0	Parity Bit Specification
0	0	No Parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

Note: When SCK is set to 1 and the baud rate generator output is selected, the ASCK pin can be used as an input/output port.

Caution: The serial transmit/receive operation must be stopped before changing the operating mode.

(c) Asynchronous serial interface status register (ASIS)

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

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

The bits used in the UART mode are indicated by shading.

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
ASIS	0	0	0	0	0	PE	FE	OVE	FF71H	00H	R

OVE	Overrun Error Flag
0	Overrun error not generated
1	Overrun error generated Note 1 (When next receive operation is completed before data from receive buffer register is read)

FE	Framing Error Flag
0	Framing error not generated
1	Framing error generated Note 2 (When stop bit is not detected)

PE	Parity Error Flag
0	Parity error not generated
1	Parity error generated (When transmit data parity does not match)

Notes:

(1) The receive buffer register (RXB) must be read when an overrun error is generated. Overrun errors will continue to be generated until RXB is read.

(2) Even if the stop bit length has been set as 2 bits by bit 2 (SL) of the asynchronous serial interface mode register, only single stop bit detection is performed during reception.

(d) Baud rate generator control register (BRGC)

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

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

The bits used in the UART mode are indicated by shading.

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	fsck/16	0
0	0	0	1	fsck/17	1
0	0	1	0	fsck/18	2
0	0	1	1	fsck/19	3
0	1	0	0	fsck/20	4
0	1	0	1	fsck/21	5
0	1	1	0	fsck/22	6
0	1	1	1	fsck/23	7
1	0	0	0	fsck/24	8
1	0	0	1	fsck/25	9
1	0	1	0	fsck/26	10
1	0	1	1	fsck/27	11
1	1	0	0	fsck/28	12
1	1	0	1	fsck/29	13
1	1	1	0	fsck/30	14

(Cont'd)

Remarks:

(1) fsck : 5-bit counter source clock

(2) k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection	n
0	0	0	0	$f_{XX}/2^{10}$ (3.9 kHz)	11
0	1	0	1	f_{XX} (4.0 MHz)	1
0	1	1	0	$f_{XX}/2$ (2.0 MHz)	2
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)	3
1	0	0	0	$f_{XX}/2^3$ (500 kHz)	4
1	0	0	1	$f_{XX}/2^4$ (250 kHz)	5
1	0	1	0	$f_{XX}/2^5$ (125 kHz)	6
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)	7
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)	8
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)	9
1	1	1	0	$f_{XX}/2^9$ (7.8 kHz)	10
Other than above				Setting prohibited	

Caution: When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.

Remarks:

(1) f_{XX} : Main system clock frequency

(2) n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

(3) Figures in parentheses apply to operation with f_{XX} = 4.0 MHz

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(i) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clock is generated by scaling the main system clock. The baud rate generated from the main system clock is found from the following expression.

$$[\text{Baud rate}] = \frac{fxx}{2^n \times (k + 16)} \text{ [Hz]}$$

Remarks:

(1) fxx : Main system clock frequency

(2) n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

(3) k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 16-5 Relationship between Main System Clock and Baud Rate

Baud Rate (bps)	fxx = 4.0 MHz	
	Set value of BRGC	Error (%)
75	0AH	0.16
110	02H	-1.36
150	EAH	0.16
300	DAH	0.16
600	CAH	0.16
1200	BAH	0.16
2400	AAH	0.16
4800	9AH	0.16
9600	8AH	0.16
19200	7AH	0.16
31250	70H	0
38400	6AH	0.16

(ii) Generation of baud rate transmit/receive clock by means of external clock from ASCK pin

The transmit/receive clock is generated by scaling the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is found from the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{2 \times (k + 16)} \quad [\text{Hz}]$$

Remarks:

(1) f_{ASCK} : Frequency of clock input to ASCK pin

(2) k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 16-6 Relationship between ASCK Pin Input Frequency and Baud Rate (when BRGC is set to 00H)

Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1200	38.4 kHz
2400	76.8 kHz
4800	153.6 kHz
9600	307.2 kHz
19200	614.4 kHz
31250	1000.0 kHz
38400	1228.8 kHz

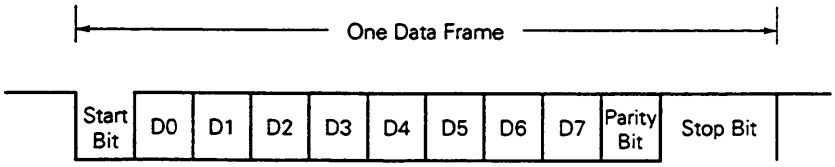
(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Fig. 16-7. One data frame consists of a start bit, character bits, parity bit and stop bit(s).

The specification of character bit length, parity selection, and specification of stop bit length for each data frame is carried out with asynchronous serial interace mode register (ASIM).

Fig. 16-7 Asynchronous Serial Interface Transmit/Receive Data Format



- Start bit ..... 1 bit
- Character bits ..... 7 bits/8 bits
- Parity bits ..... Even parity/odd parity/0 parity/no parity
- Stop bit(s) ..... 1 bit/2 bits

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission, the most significant bit (bit 7) is ignored, and in reception, the most significant bit (bit 7) is always "0".

The serial transfer rate between 75 bps and 38400 bps is selected by means of the asynchronous serial interface mode register and baud rate generator settings.

If a serial data receive error is generated, the receive error contents can be determined by reading the status of the asynchronous serial interface status register (ASIS).

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

- **Even parity**

  If the number of "1" bits in the transmit data is odd, the parity bit is set to "1". If the number of bits with a value of "1" is an even number, the parity bit is set to "0". Thus, control is executed so that the number of bits with a value of "1" contained in the transmit data and parity bit is an even number.

  In reception, the number of bits with a value of "1" contained in the receive data and parity bit are counted, and if this is an odd number, a parity error is generated.

- **Odd parity**

  Conversely to the situation with even parity, control is executed so that the number of bits with a value of "1" contained in the receive data and parity bit is an odd number.

  Similarly, in reception, the number of bits with a value of "1" contained in the receive data and parity bit are counted, and if this is an even number, a parity error is generated.

- **0 parity**

  When transmitting, the parity bit is set to "0" irrespective of the transmit data.

  In reception, a parity bit check is not performed. Therefore, a parity error is not generated irrespective of whether the parity bit is set to "0" or "1".

- **No parity**

  A parity bit is not added to the transmit data.

  In reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error is not generated.

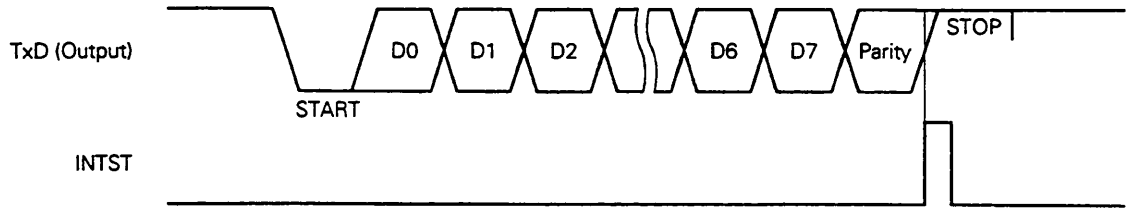
(c) Transmission

A transmit operation is started by writing transmit data to the transmit shift register (TXS). The start bit, parity bit and stop bit(s) are added automatically.

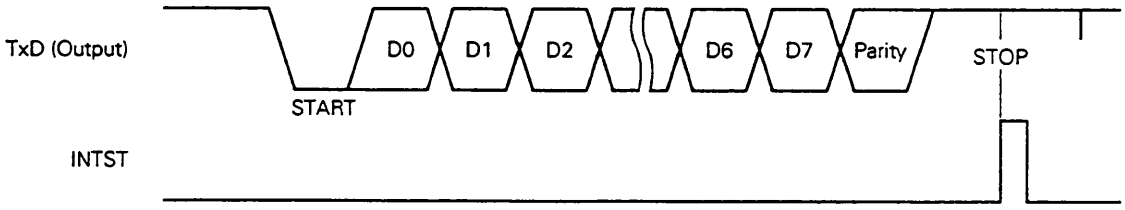
When the transmit operation starts, the data in the transmit shift register (TXS) is shifted out, and when the transmit shift register (TXS) is empty, a transmission completion interrupt (INTST) is generated.

Fig. 16-8 Asynchronous Serial Interface Transmission Completion Interrupt Timing

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution: Rewriting of the asynchronous serial interface mode register (ASIM) should not be performed during a transmit operation. If rewriting of the ASIM register is performed during transmission, subsequent transmit operations may not be possible (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt (INTST) or the interrupt request flag (STIF) set by the INTST.

(d) Reception

When the RXE bit of the asynchronous serial interface mode register (ASIM) is set (1), a receive operation is enabled and sampling of the RxD pin in put is performed.

RxD pin input sampling is performed using the serial clock specified by ASIM.

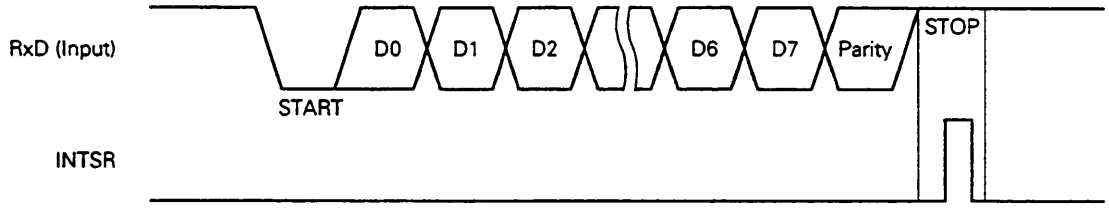
When the RxD pin input becomes low, the 5-bit counter starts counting, and at the time when half time determined by specified baud rate has passed, the data sampling start timing signal is output. If the RxD pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 5-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to the receive buffer register (RXB), and a reception completion interrupt (INTSR) is generated.

If an error is generated, the receive data in which the error was generated is still transferred to RXB, and INTSR is generated.

If the RXE bit is reset (0) during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB and ASIS are not changed, and INTSR and INTSER are not generated.

Fig. 16-9 Asynchronous Serial Interface Reception Completion Interrupt Timing



Caution: The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

Three kinds of errors can occur during a receive operation: a parity error, framing error, or overrun error. The data reception result error flag is set in the asynchronous serial interface status register (ASIS) and at the same time a receive error interrupt (INTSER) is generated. Receive error cause are shown in Table 16-7.

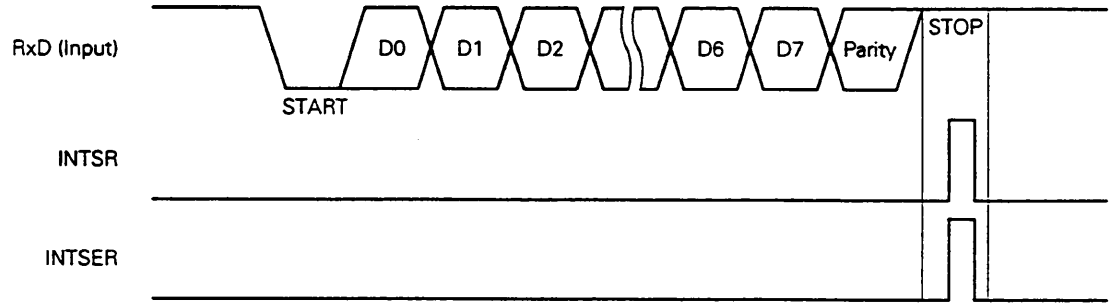
It is possible to determine what kind of error was generated during reception by reading the contents of the asynchronous serial interface status register (ASIS) in the reception error interrupt servicing (INTSER) (see Figs. 16-9 and 16-10).

The contents of ASIS are reset (0) by reading the receive buffer regsiter (RXB) or receiving the next data if there is an error in the next data, the corresponding error flag is set).

Table 16-7 Receive Error Causes

Receive Errors	Cause
Parity error	Transmission-time parity specification and reception data parity do not match
Framing error	Stop bit not detected
Overrun error	Reception of next data is completed before data is read from receive buffer

Fig. 16-10 Receive Error Timing



Cautions:

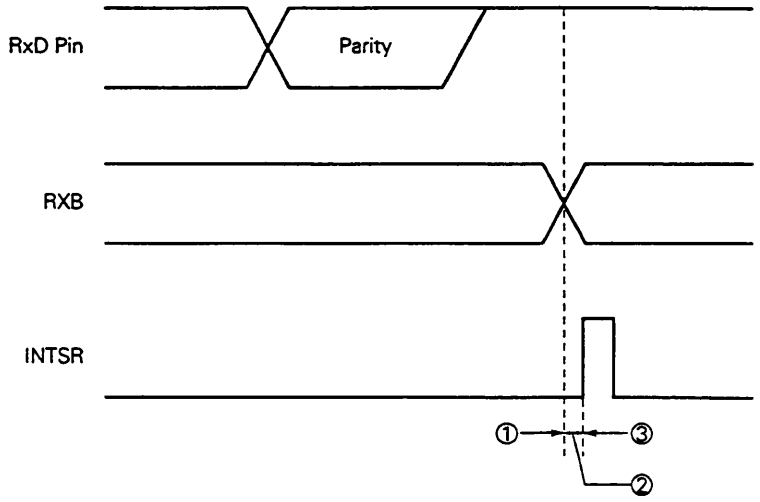
(1) The contents of the ASIS register are reset (0) by reading the receive buffer register (RXB) or receiving the next data. To ascertain the error contents, ASIS must be read before reading RXB.

(2) The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(3) UART mode cautions

(a) When bit 7 (TXE) of the asynchronous serial interface mode register (ASIM) is cleared during transmission, be sure to set the transmit shift register (TXS) to FFH, then set the TXE to 1 before executing the next transmission.

(b) When bit 6 (RXE) of the asynchronous serial interface mode register (ASIM) is cleared during reception, receive buffer register (RXB) and receive completion interrupt (INTSR) are as follows.



When RXE is set to 0 at a time indicated by ①, RXB holds the previous data and does not generate INTSR.
When RXE is set to 0 at a time indicated by ②, RXB renews the data and does not generate INTSR.
When RXE is set to 0 at a time indicated by ③, RXB renews the data and generates INTSR.

16.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous clocked serial interface, such as the 75X series, 78K series, 17K series, etc.

Communication is performed using three lines: the serial clock ($\overline{SCK2}$), serial output (SO2), and serial input (SI2).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operating mode register 2 (CSIM2) and the asynchronous serial interface mode register (ASIM).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM2 to 00H.

The bits used in the 3-wire serial I/O mode are indicated by shading.

Symbol	⑦	6	5	4	3	2	1	0	Address	When Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{SCK2}$ pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution: Ensure that 0 is set in bits 0 and 3 to 6.

(b) Asynchronous serial interface mode register (ASIM)

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

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

The bits used in the 3-wire serial I/O mode are indicated by shading.

When the 3-wire serial I/O mode is selected, 00H should be set in ASIM.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	When Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output

ISRM	Control of Reception Completion Interrupt in Case of Error Generation
0	Reception completion interrupt generated in case of error generation
1	Reception completion interrupt not generated in case of error generation

SL	Transmit Data Stop Bit Length Specification
0	1 bit
1	2 bits

CL	Character Length Specification
0	7 bits
1	8 bits

PS1	PS0	Parity Bit Specification
0	0	No Parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

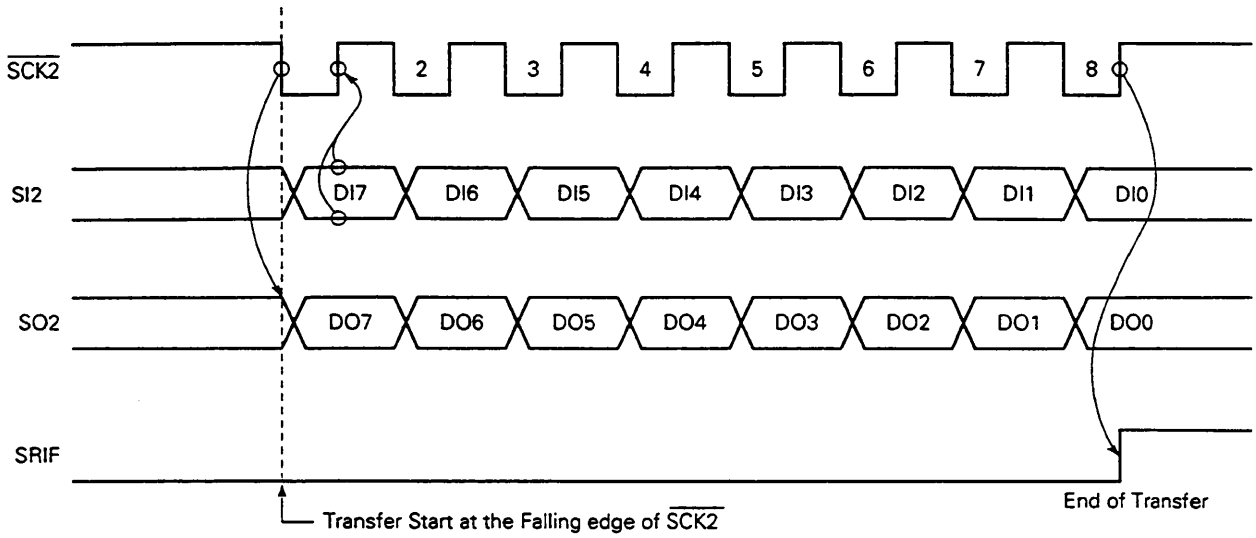
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmit shift register (TXS/SIO2) and receive shift register (RXS) shift operations are performed in synchronization with the fall of the serial clock ($\overline{SCK2}$). Then transmit data is held in the SO2 latch and output from the SO2 pin. Also, receive data input to the SI2 pin is latched in the receive buffer register (RXB/SIO2) on the rise of $\overline{SCK2}$.

At the end of an 8-bit transfer, the operation of the transmit shift register (TXS/SIO2) or receive shift register (RXS) stops automatically, and the interrupt request flag (SRIF) is set.

Fig. 16-11 3-Wire Serial I/O Mode Timing



(3) Transfer start

Serial transfer is started if the following two conditions are satisfied when the transfer data is set to the transmit shift register (TXS/SIO2):

- Operation control bit of serial interface 2 (CSIE2) = 1
- When internal serial clock is stopped or when $\overline{SCK2}$ is high after 8-bit serial transfer has been completed.

Caution: Transfer is not started even if CSIE2 is set to "1" after data has been written to SIO2.

At the end of an 8-bit transfer, serial transfer stops automatically and the interrupt request flag (SRIF) is set.

CHAPTER 17 REAL-TIME OUTPUT PORT

17.1 Real-Time Output Port Functions

Data set previously in the real-time output buffer register can be transferred to the output latch by hardware concurrently with timer interrupts or external interrupt generation, then output to external. This is called the real-time output function. The pins that output data to external are called real-time output ports.

By using a real-time output port, a signal which has no jitter can be output. This port is therefore suitable for control of stepping motors, etc.

Port mode/real-time output port mode can be specified bit-wise.

17.2 Real-Time Output Port Configuration

The real-time output port consists of the following hardware.

Table 17-1 Real-time Output Port Configuration

Item	Configuration
Register	Real-time output buffer register (RTBL, RTBH)
Control register	Real-time output port mode register (RTPM) Real-time output port control register (RTPC)

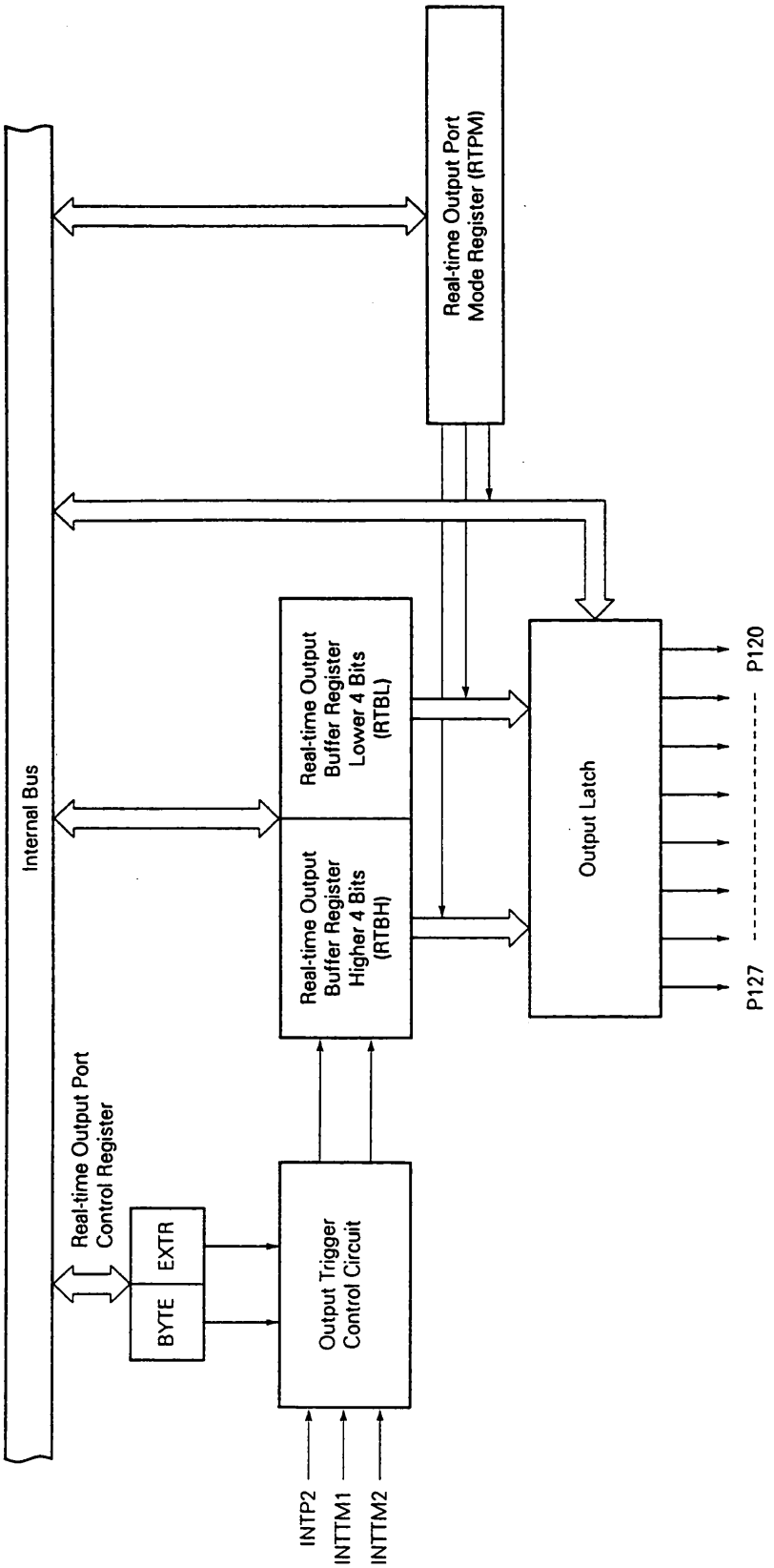


Fig. 17-1 Real-time Output Port Block Diagram

(1) Real-time output buffer register (RTBL, RTBH)

Addresses of RTBL and RTBH are mapped individually in the SFR area as shown in Fig. 17-2.

When specifying 4 bits × 2 channels as the operating mode, data are set individually in RTBL and RTBH.

When specifying 8 bits × 1 channel as the operating mode, data are set to both RTBL and RTBH by writing 8-bit data to either RTBL or RTBH.

Table 17-2 shows operations during manipulation of RTBL and RTBH.

Fig. 17-2 Real-time Output Buffer Register Configuration

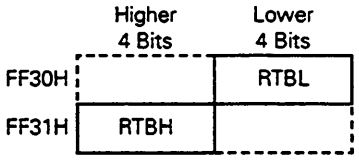


Table 17-2 Operation in Real-time Output Buffer Register Manipulation

Operating Mode	Register to be Manipulated	In Read Note1		In Write Note2	
		Higher 4 Bits	Lower 4 Bits	Higher 4 Bits	Lower 4 Bits
4 Bits × 2 Channels	RTBL	RTBH	RTBL	Invalid	RTBL
	RTBH	RTBH	RTBL	RTBH	Invalid
8 Bits × 1 Channel	RTBL	RTBH	RTBL	RTBH	RTBL
	RTBH	RTBH	RTBL	RTBH	RTBL

Notes:

(1) Only the bits set in the real-time output port mode can be read. Bits set in the port mode are 0.

(2) After setting data in the real-time output port, output data should be set in RTBL and RTBH by the time when a real-time output trigger is generated.

17.3 Real-Time Output Port Control Registers

The following two registers control the real-time output port.

- Real-time output port mode register (RTPM)
- Real-time output port control register (RTPC)

(1) Real-time output port mode register (RTPM)

This register selects the real-time output port mode/port mode bit-wise.

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

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

Fig. 17-3 Real-time Output Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
RTPM	RTPM7	RTPM6	RTPM5	RTPM4	RTPM3	RTPM2	RTPM1	RTPM0	FF34H	00H	R/W

RTPMn	Real-time Output Port Selection (n = 0 to 7)
0	Port mode
1	Real-time Output Port Mode

Cautions:

(1) When using these bits as a real-time output port, set the ports to which real-time output is performed to the output mode (set the bits of the port mode register to 0).

(2) In the port specified as a real-time output port, data cannot be set to the output latch. Therefore, when setting an initial value, data should be set to the output latch before setting the real-time output mode.

(2) Real-time output port control register (RTPC)

This register sets the real-time output port operating mode.

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

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

Fig. 17-4 Real-time Output Port Control Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	When Reset	R/W
RTPC	0	0	0	0	0	0	BYTE	EXTR	FF36H	00H	R/W

EXTR	Real-time Output Control by INTP2
0	INTP2 not specified as real-time output trigger
1	INTP2 specified as real-time output trigger

BYTE	Real-time Output Port Operating Mode
0	4 Bits × 2 Channels
1	8 Bits × 1 Channel

Table 17-3 Real-time Output Port Output Trigger

BYTE	EXTR	Output Mode	RTBH → Port Output	RTBL → Port Output
0	0	4 Bits × 2 Channels	INTTM2	INTTM1
	1		INTTM1	INTP2
1	0	8 Bits × 1 Channel	INTTM1	
	1		INTP2	

CHAPTER 18 INTERRUPT FUNCTIONS AND TEST FUNCTIONS

18.1 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 18-1**).

A standby release signal is generated.

Seven external interrupts and fourteen 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.

18.2 Interrupt Sources and Configuration

A total of 23 non-maskable, maskable and software interrupts are incorporated in the interrupt sources (see **Table 18-1**).

Table 18-1 Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	004H	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	INTP4			000EH	
	6	INTP5			0010H	
	7	INTP6			0012H	
	8	INTCSI0	End of serial interface channel 0 transfer	Internal	0014H	B
	9	INTCSI1	End of serial interface channel 1 transfer		0016H	
	10	INTSER	Serial interface channel 2 UART reception error generation		0018H	
	11	INTSR	End of serial interface channel 2 UART reception		001AH	
		INTCSI2	End of serial interface channel 2 3-wire transfer			
	12	INTST	End of serial interface channel 2 UART transfer		001CH	

Notes:

(1) Default priorities are intended for two or more simultaneously generated maskable interrupts. 0 is the highest priority and 19 is the lowest priority.

(2) Basic configuration types A to E correspond to A to E on Fig. 18-1.

Table 18-1 Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note2]
		Name	Trigger			
Maskable	13	INTTM3	Reference time interval signal from watch timer	Internal	001EH	B
	14	INTTM00	Generation of 16-bit timer register, capture/compare register (CR00) match signal		0020H	
	15	INTTM01	Generation of 16-bit timer register, capture/compare register (CR01) match signal		0022H	
	16	INTTM1	Generation of 8-bit timer/event counter 1 match signal		0024H	
	17	INTTM2	Generation of 8 bit timer/event counter 2 match signal		0026H	
	18	INTAD	End of A/D converter conversion		0028H	
	19	INTIE	When write operation from IEBus controller to return code register (RCR) (including same value) occurs or when runaway of IEBus interface is detected		002AH	
Software	—	BRK	BRK instruction execution	Internal	003EH	E

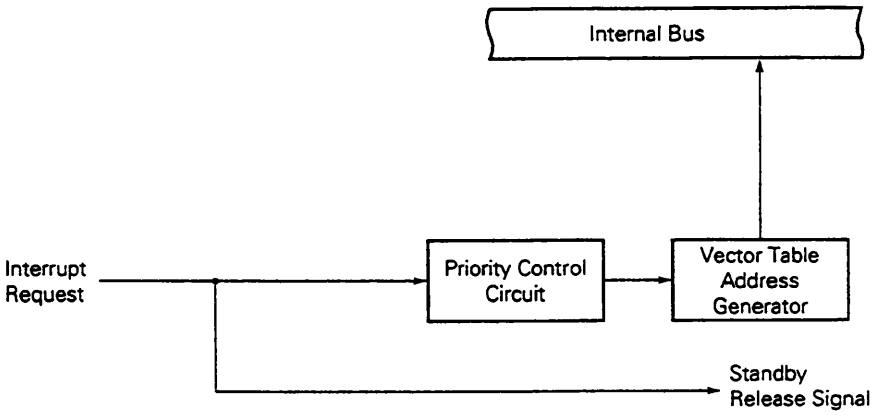
Notes:

(1) Default priorities are intended for two or more simultaneously generated maskable interrupts. 0 is the highest priority and 19 is the lowest priority.

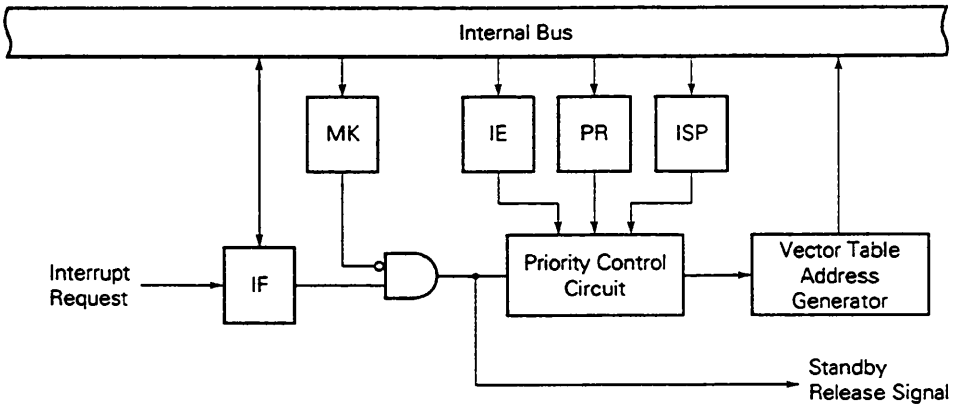
(2) Basic configuration types A to E correspond to A to E on Fig. 18-1.

Fig. 18-1 Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

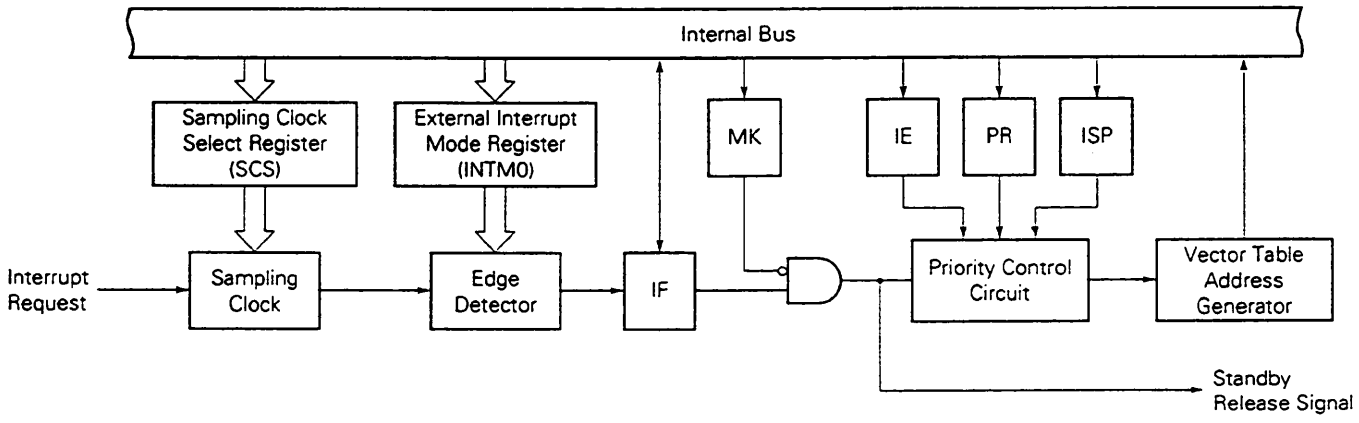
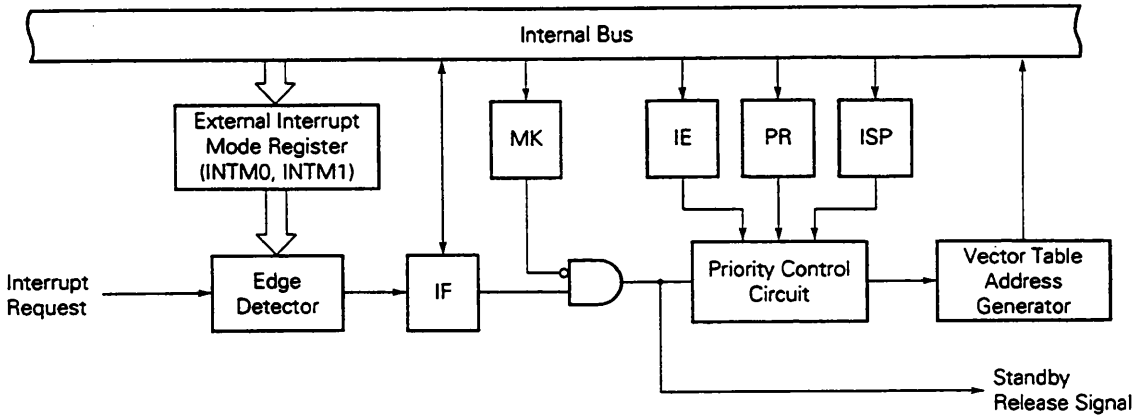
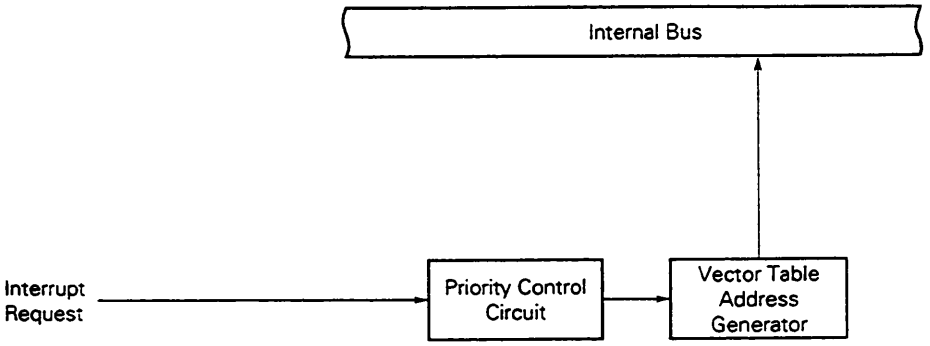


Fig. 18-1 Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



Remarks:

(1) IF : Interrupt request flag
(2) IE : Interrupt enable flag
(3) ISP : Inservice priority flag
(4) MK : Interrupt mask flag
(5) PR : Priority specify flag

18.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specify flag register (PR0L, PR0H, PR1L)
- External interrupt mode register (INTM0, INTM1)
- Sampling clock select register (SCS)
- Program status word (PSW)

Table 18-2 gives a listing of interrupt request flags, interrupt mask flags and priority specify flag names corresponding to interrupt request sources.

Table 18-2 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
INTP4	PIF4	PMK4	PPR4
INTP5	PIF5	PMK5	PPR5
INTP6	PIF6	PMK6	PPR6
INTTM00	TMIF00	TMMK00	TMPR00
INTTM01	TMIF01	TMMK01	TMPR01
INTTM1	TMIF1	TMMK1	TMPR1
INTTM2	TMIF2	TMMK2	TMPR2
INTTM3	TMIF3	TMMK3	TMPR3
INTWDT	TMIF4	TMMK4	TMPR4
INTCSI0	CSIIF0	CSIMK0	CSIPR0
INTCSI1	CSIIF1	CSIMK1	CSIPR1
INTSR/INTCSI2	SRIF	SRMK	SRPR
INTSER	SERIF	SERMK	SERPR
INTST	STIF	STMK	STPR
INTAD	ADIF	ADMK	ADPR
INTIE	IEIF	IEMK	IEPR

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H and IF1L are set with a 1-bit, 8-bit or 16-bit memory manipulation instruction.

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

Fig. 18-2 Interrupt Request Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When Reset	R/W
IF0L	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
IF0H	TMIF01	TMIF00	TMIF3	STIF	SRIF	SERIF	CSIIF1	CSIIF0	FFE1H	00H	R/W
	⑦	6	5	4	③	②	①	⓪			
IF1L	Note WTIF	0	0	0	IEIF	ADIF	TMIF2	TMIF1	FFE2H	00H	R/W

xxIFx	Interrupt Request Flag
0	No interrupt request signal
1	Interrupt request signal is generated; Interrupt request state

Note: WTIF is test input flag. Vectored interrupt is not generated.

Cautions:

(1) TMIF4 flag is R/W enabled only when a watchdog timer is used as an interval timer. If a watchdog timer is used as a non-maskable interrupt, set TMIF4 flag to 0.

(2) Because port 0 has a dual function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

(3) Set always 0 in IF1L bit 4 to bit 6.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L, MK0H and MK1L are set with a 1-bit, 8-bit or 16-bit memory manipulation instruction.

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

Fig. 18-3 Interrupt Mask Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When Reset	R/W
MK0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	TMMK4	FFE4H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
MK0H	TMMK 01	TMMK 00	TMMK 3	STMK	SRMK	SERMK	CSIMK 1	CSIMK 0	FFE5H	FFH	R/W
	⑦	6	5	4	③	②	①	⓪			
MK1L	Note WTMK	1	1	1	IEMK	ADMK	TMMK 2	TMMK 1	FFE6H	FFH	R/W

xxMKx	Interrupt Servicing and Standby Mode Control
0	Interrupt servicing enabled, standby mode release enabled
1	Interrupt servicing disabled, standby mode release disabled

Note: WTMK controls standby mode release enable/disable.

Cautions:

(1) If TMMK4 flag is read when a watchdog timer is used as a non-maskable interrupt, MK0 value becomes undefined.

(2) Because port 0 has a dual function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

(3) Set always 1 in MK1L bit 4 to bit 6.

(3) Priority specify flag registers (PR0L, PR0H and PR1L)

The priority specify flag is used to set the corresponding maskable interrupt priority orders.

PR0L, PR0H and PR1L are set with a 1-bit, 8-bit or 16-bit memory manipulation instruction.

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

Fig. 18-4 Priority Specify Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When Reset	R/W
PR0L	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	TMPR4	FFE8H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
PR0H	TMPR 01	TMPR 00	TMPR 3	STPR	SRPR	SERPR	CSIPR 1	CSIPR 0	FFE9H	FFH	R/W
	7	6	5	4	③	②	①	⓪			
PR1L	1	1	1	1	IEPR	ADPR	TMPR 2	TMPR 1	FFEAH	FFH	R/W

xxPRx	Priority Level Selection
0	High priority level
1	Low priority level

Cautions:

(1) When a watchdog timer is used as a non-maskable interrupt, set 1 in TMPR4 flag.

(2) Set always 1 in PR1L bit 4 to bit 7.

(4) External interrupt mode register (INTM0, INTM1)

These registers set the valid edge for INTP0 to INTP6.

INTM0 and INTM1 are set by 8-bit memory manipulation instructions.

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

Fig. 18-5 External Interrupt Mode Register 0 Format

Symbol	7	6	5	4	3	2	1	0	Address	When 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 falling and rising edges

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

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

Fig. 18-6 External Interrupt Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
INTM1	ES71	ES70	ES61	ES60	ES51	ES50	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	INTP3 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES51	ES50	INTP4 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES61	ES60	INTP5 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES71	ES70	INTP6 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising 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 manipulation instruction.

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

Fig. 18-7 Sampling Clock Select Register Format

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

SCS1	SCS0	INTP0 Sampling Clock Selection
0	0	$f_{XX}/2^N$
0	1	$f_{XX}/2^7$ (31.3kHz)
1	0	$f_{XX}/2^5$ (125.0kHz)
1	1	$f_{XX}/2^6$ (62.5kHz)

Caution: $f_{XX}/2^N$ is a clock to be supplied to the CPU and $f_{XX}/2^5$, $f_{XX}/2^6$ and $f_{XX}/2^7$ are clocks to be supplied to the peripheral hardware. $f_{XX}/2^N$ stops in the HALT mode.

Remarks:

(1) N : Value (N = 0 to 4) at bits 0 to 2 (PCC0 to PCC2) of processor clock control register

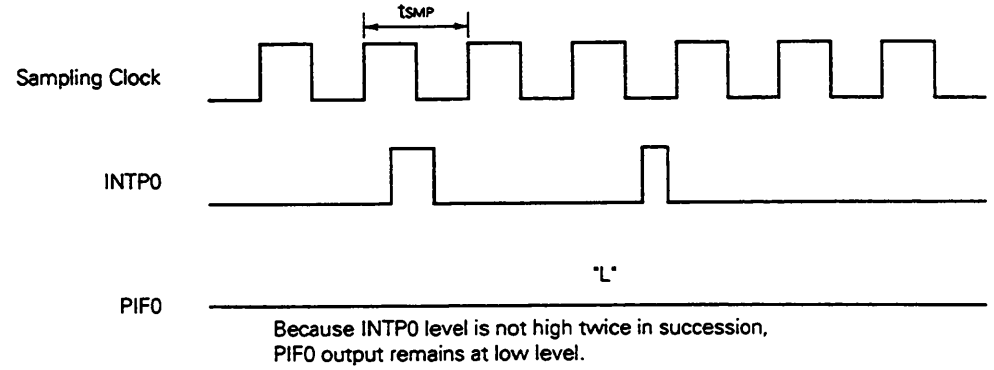
(2) f_{XX} : Main system clock frequency

(3) Values in parentheses when operated with f_{XX} = 4.0 MHz.

When the INTP0 input level is active twice in succession, the noise remover sets PIF0 flag to 1.

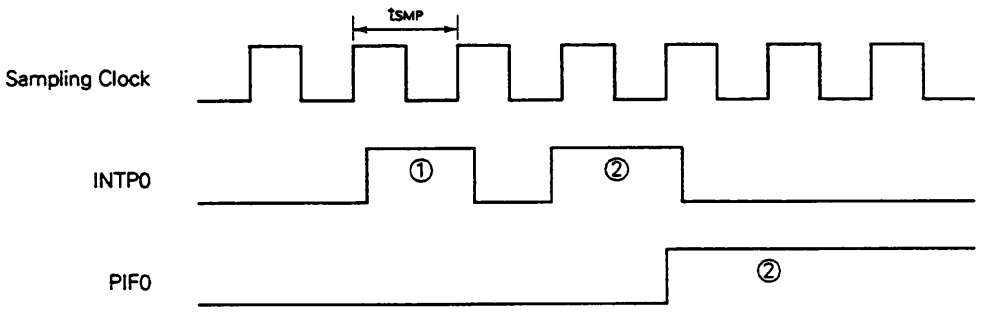
Fig. 18-8 Noise Remover Input/Output Timing (when rising edge is detected)

(a) When input is less than the sampling period (t_{SMP})



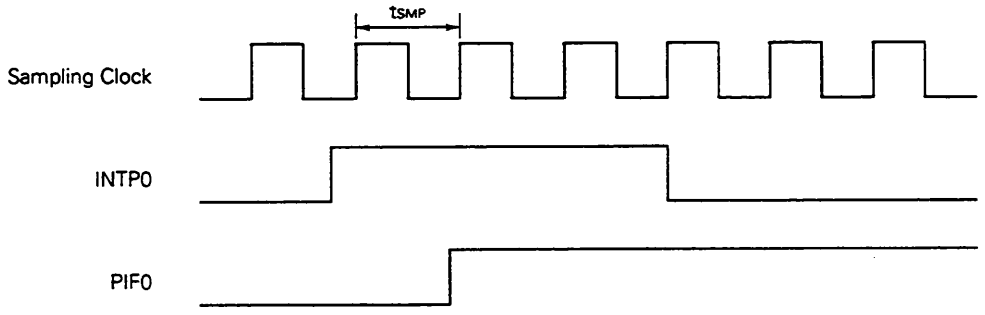
Because INTP0 level is not high twice in succession,
PIF0 output remains at low level.

(b) When input is equal to or twice the sampling period (t_{SMP})



Because INTP0 level is high twice in succession in ②,
PIF0 flag is set to 1.

(c) When input is twice or more than the sampling period (t_{SMP})



Because INTP0 level is high four times in succession,
PIF0 flag is set to 1.

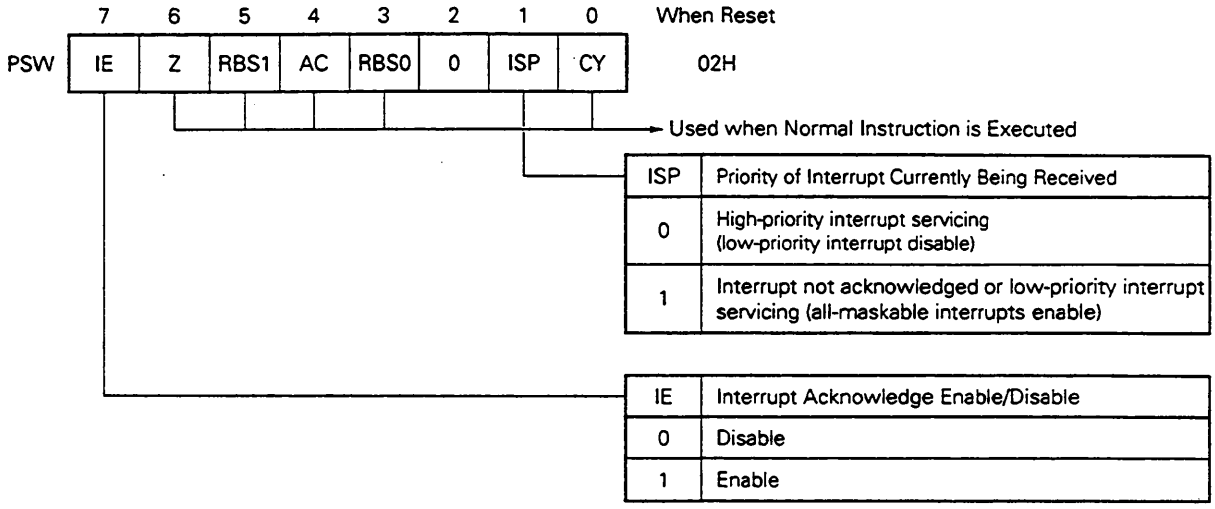
(6) Program status word (PSW)

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 manipulation 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. 18-9 Program Status Word Format



ISP	Priority of Interrupt Currently Being Received
0	High-priority interrupt servicing (low-priority interrupt disable)
1	Interrupt not acknowledged or low-priority interrupt servicing (all-maskable interrupts enable)

IE	Interrupt Acknowledge Enable/Disable
0	Disable
1	Enable

18.4 Interrupt Servicing Operations

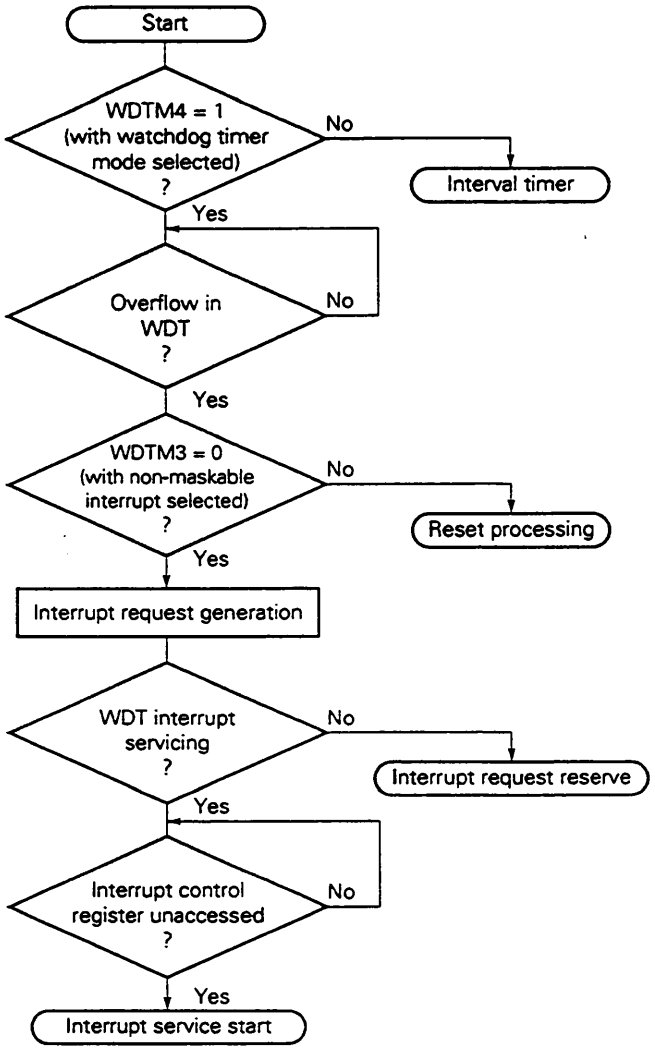
18.4.1 Non-maskable interrupt acknowledge operation

A non-maskable interrupt is unconditionally acknowledged even if in an interrupt acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the 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. 18-10 Non-Maskable Interrupt Acknowledge Flowchart



Remarks:

(1) WDTM : Watchdog timer mode register

(2) WDT : Watchdog timer

Fig. 18-11 Non-Maskable Interrupt Acknowledge Timing

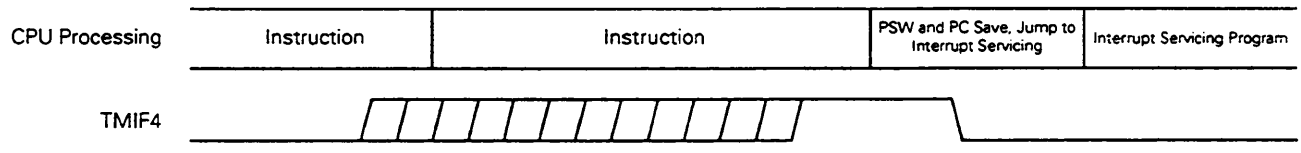
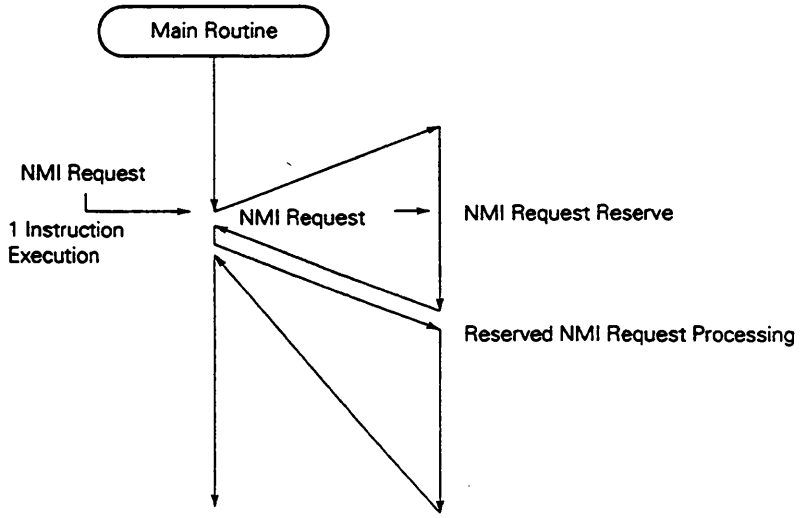
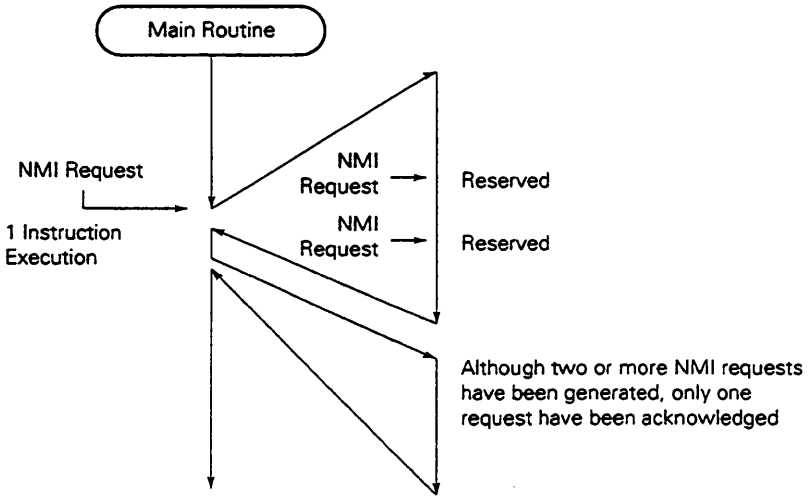


Fig. 18-12 Non-Maskable Interrupt Request Acknowledge Operation

(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



18.4.2 Maskable interrupt acknowledge operation

A maskable interrupt becomes acknowledgeable when an interrupt request flag 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 18-3 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 reserved interrupts are acknowledged when they become acknowledgeable.

Fig. 18-13 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 PC and branched.

Return from the interrupt is possible with the RETI instruction.

Fig. 18-13 Interrupt Acknowledge Processing Algorithm

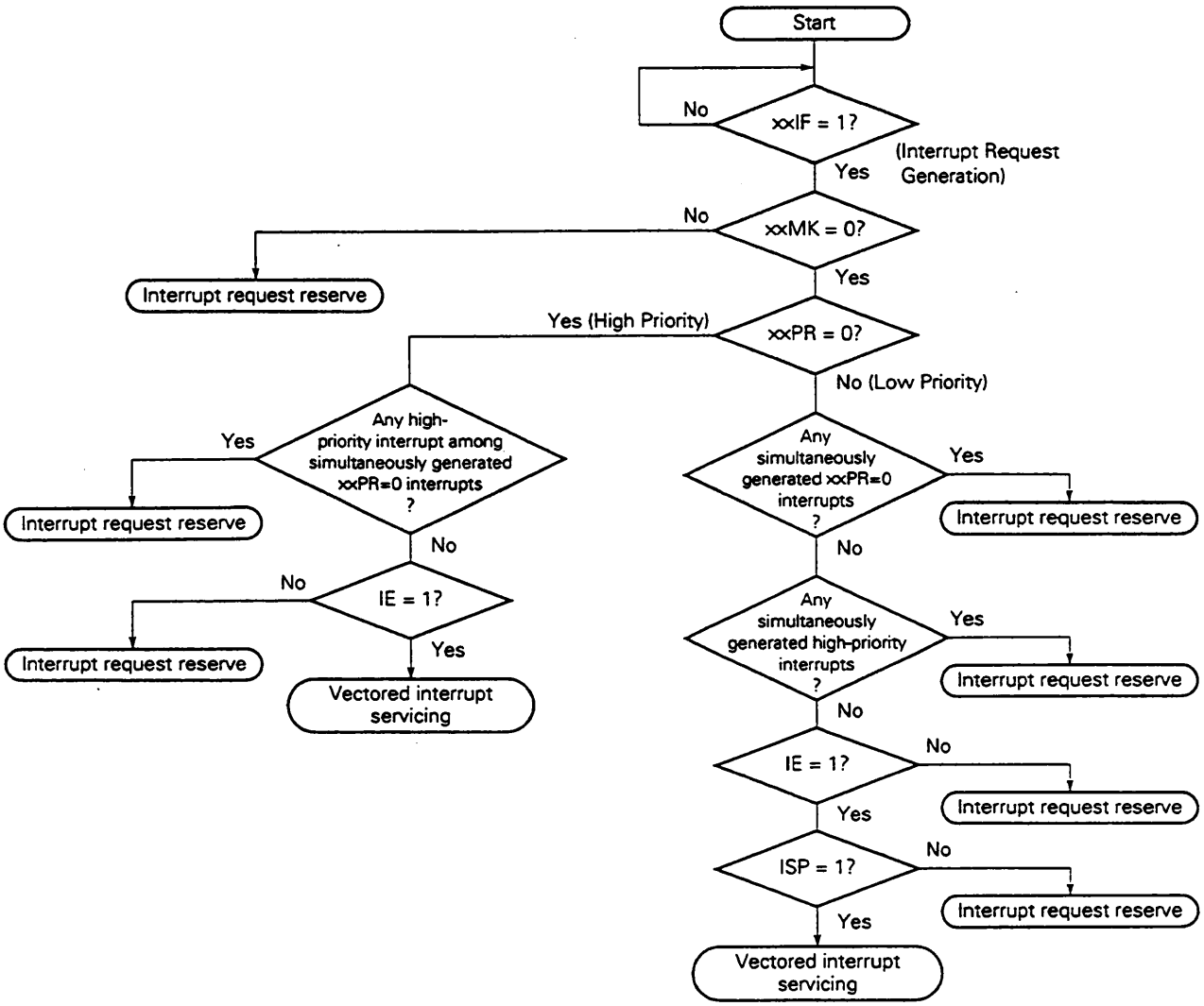


Fig. 18-14 Interrupt Acknowledge Timing (Minimum Time)

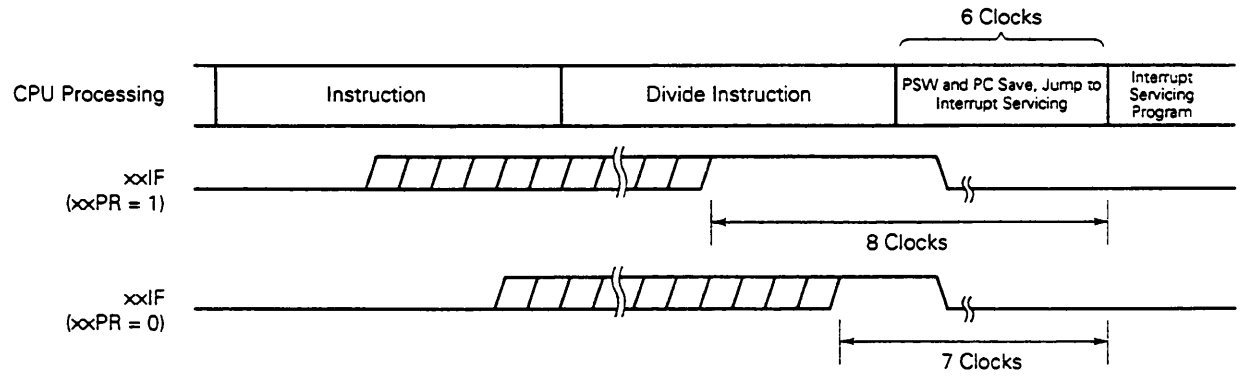
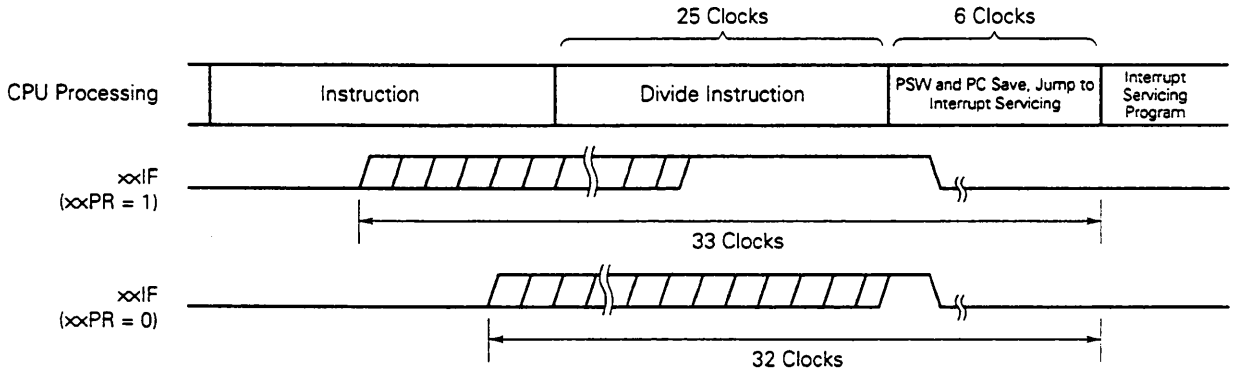


Fig. 18-15 Interrupt Acknowledge Timing (Maximum Time)



18.4.3 Software interrupt acknowledge operation

A software interrupt is acknowledged by BRK instruction execution. Software interrupt 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.

Return from the software interrupt is possible with the RETB instruction.

Caution: Do not use the RETI instruction for returning from the software interrupt.

18.4.4 Multiple interrupt servicing

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 registers (PR0L, PR0H and PR1L). 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 18-1**). In the latter, interrupt requests are divided into a high-priority group and a low-priority group by setting the bits corresponding to PR0L, PR0H and PR1L. The following are the interrupt requests enabled for multiple interrupts.

Table 18-4 Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing

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 servicing		×	×	×	×	×
Maskable interrupt servicing	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt servicing		○	○	×	○	×

Remarks:

(1) ○: Multiple interrupt enable

(2) ×: Multiple interrupt disable

(3) ISP and IE are flags included in PSW. ★

ISP=0: During high-priority interrupt.

ISP=1: Interrupt not accepted or during low-priority interrupt

IE=0 : Interrupt acceptance disable

IE=1 : Interrupt acceptance enable

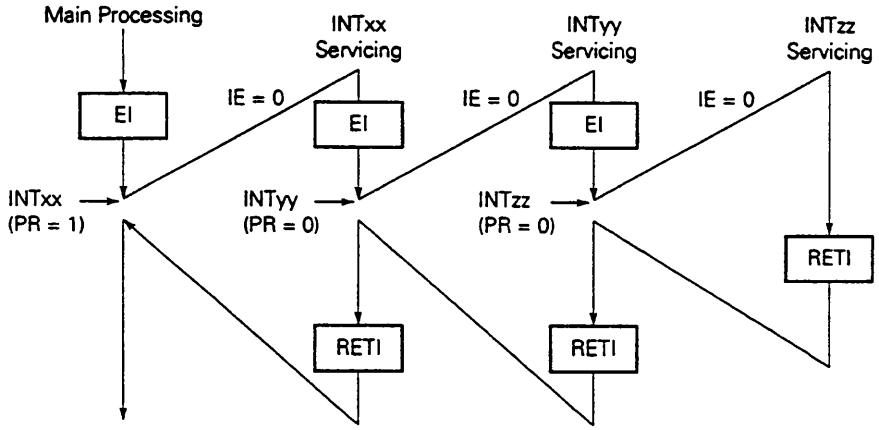
(4) xx PR is a flag included in PR0. ★

xx PR=0: High-priority level

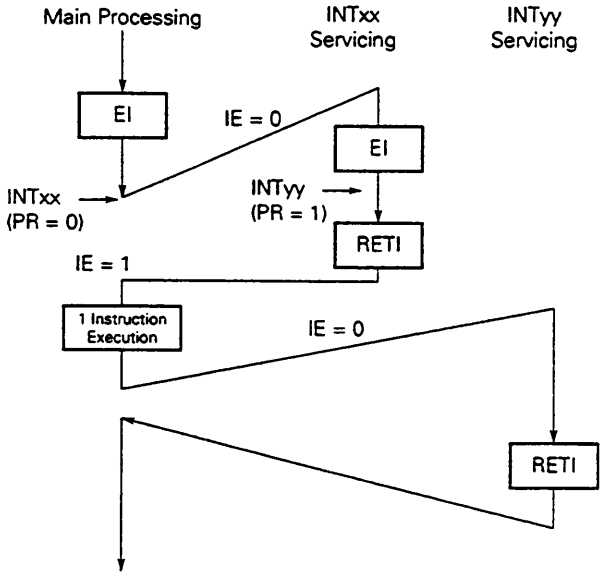
xx PR=1: Low-priority level

Fig. 18-16 Multiple Interrupt Example

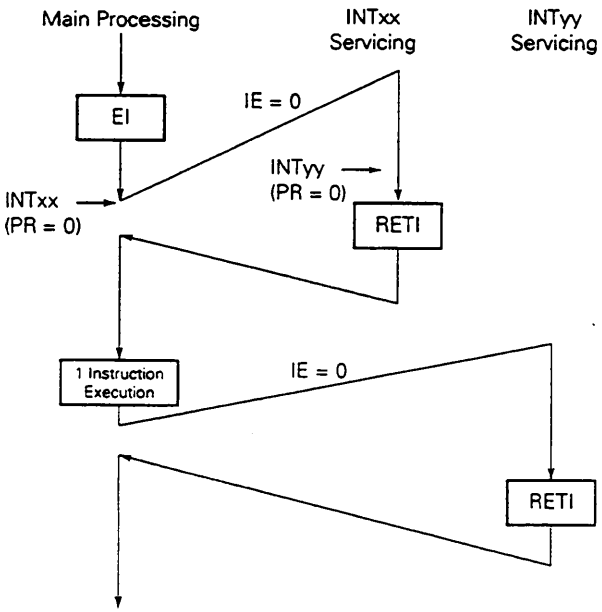
Example 1



Example 2



Example 3



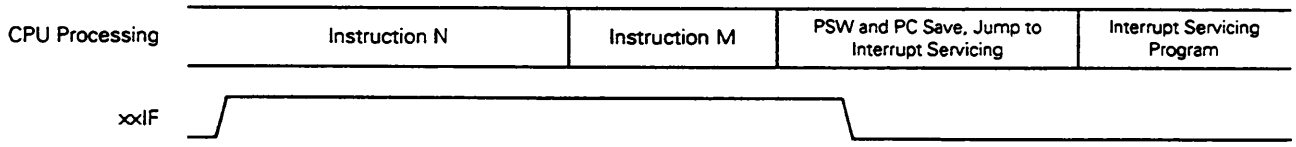
18.4.5 Interrupt reserve

Interrupt acknowledge is temporarily reserved 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
- Manipulation instructions for IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, PR1L, INTM0, and INTM1 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. 18-17 Interrupt Request Reserve



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.

★ 18.5 Test Functions

The vector process is not performed and the test input flag is set (1).

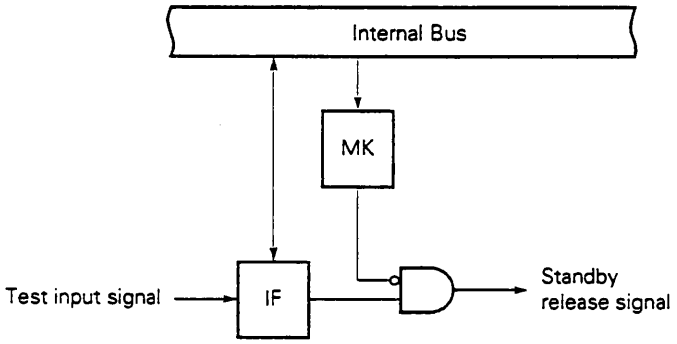
The standby release signal is generated.

There are two test input sources as shown in Table 18-5. The basic configuration is as shown in Fig. 18-18.

Table 18-5 Test Input Source List

Test Input Sources		Internal/External
Name	Trigger	
INTWT	Watch timer overflow	Internal
INTPT4	Port 4 falling edge detection	External

Fig. 18-18 Basic Configuration of Test Function



Remarks:

(1) IF: Test input flag

(2) MK: Test mask flag

18.5.1 Registers controlling test functions

The test function is controlled by the following three registers.

- Interrupt request flag register 1L (IF1L)
- Interrupt mask flag register 1L (MK1L)
- Key return mode register (KRM)

The names of test input flags and test mask flags corresponding to each test input signal are shown in Table 18-6.

Table 18-6 Various Flags Corresponding to Test Input Signal

Test Input Signal	Test Input Flag	Test Mask Flag
INTWT	WTIF	WTMK
INTPT4	KRIF	KRMK

(1) Interrupt request flag register 1L (IF1L)

This register indicates whether the watch timer overflow has been detected or not.

IF1L is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IF1L to 00H.

Fig. 18-19 Interrupt Request Flag Register 1L Format

Symbol	⑦	6	5	4	③	②	①	⓪	Address	When Reset	R/W
IF1L	WTIF	0	0	0	IEIF	ADIF	TMIF2	TMIF1	FFE2H	00H	R/W

WTIF	Watch Timer Overflow Detection Flag
0	Not detected
1	Detected

Caution: Always set 0 to bit 4 through bit 6.

(2) Interrupt mask flag register 1L (MK1L)

This register sets the standby mode release enable/disable by the watch timer.

MK1L is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK1L to FFH.

Fig. 18-20 Interrupt Mask Flag Register 1L Format

Symbol	⑦	6	5	4	③	②	①	⓪	Address	When Reset	R/W
MK1L	WTMK	1	1	1	IEMK	ADMK	TMMK2	TMMK1	FFE6H	FFH	R/W

WTMK	Standby Mode Control by Watch Timer
0	Standby mode release enabled
1	Standby mode release disabled

Caution: Always set 1 to bit 4 through bit 6.

(3) Key return mode register (KRM)

This register sets the enabling/disabling of standby function release by the key return signal (port 4 falling edge detection).

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

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

Fig. 18-21 Key Return Mode Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	When Reset	R/W
KRM	0	0	0	0	0	0	KRMK	KRIF	FFF6H	02H	R/W

KRIF	Key Return Signal Detection Flag
0	Not detected
1	Detected (Port 4 Falling Edge Detection)

KRMK	Standby Mode Control by Key Return Signal
0	Standby mode release enabled
1	Standby mode release disabled

Caution: When port 4 falling edge detection is used, be sure to clear KRIF to 0 (will not be cleared to 0 automatically).

18.5.2 Test input signal acknowledge operation

(1) Internal test input signal

The WTIF flag is set by watch timer overflow. By checking the WTIF flag using the period shorter than the watch timer overflow, the watch function is made possible.

(2) External test input signal

When the falling edge is input to port 4 (P40 to P47), KRIF is set. By using port 4 as the key matrix return signal input, the presence of key inputs can be checked in the KRIF state.

CHAPTER 19 EXTERNAL DEVICE EXPANSION FUNCTION

19.1 External Device Expansion Functions

The external device expansion functions connect external devices to areas other than the internal ROM, RAM, and SFR. Connection of external devices uses ports 4 to 6. Ports 4 to 6 control address/data, read/write strobe, wait, timing, etc.

Table 19-1 Pin Functions in External Memory Expansion Mode

Pin Function at External Device Connection		Dual-Function Pin
Name	Function	
AD0 to AD7	Multiplexed address/data bus	P40 to P47
A8 to A15	Address bus	P50 to P57
$\overline{RD}$	Read strobe signal	P64
$\overline{WR}$	Write strobe signal	P65
$\overline{WAIT}$	Wait signal	P66
ASTB	Address strobe signal	P67

Table 19-2 State of Port 4 to Port 6 Pins in External Memory Expansion Mode

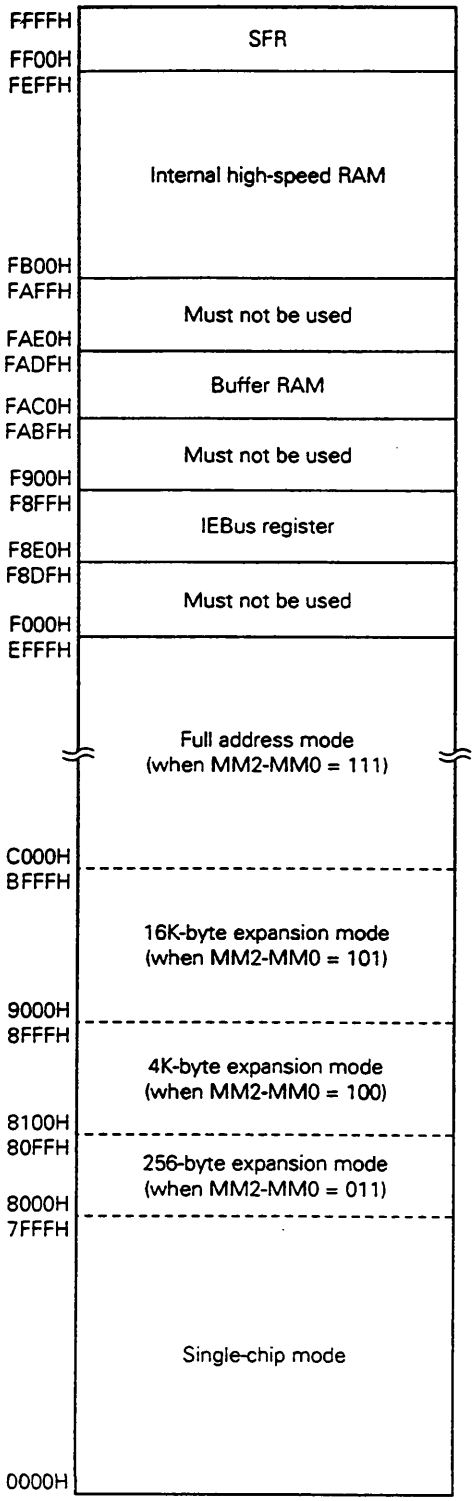
Port / External Expansion Mode	Port 4	Port 5								Port 6							
	0 to 7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Single-chip mode	Port	Port								Port							
256-byte expansion mode	Address/data	Port								Port				$\overline{RD}$,$\overline{WR}$,$\overline{WAIT}$,ASTB			
4K-byte expansion mode	Address/data	Address				Port				Port				$\overline{RD}$,$\overline{WR}$,$\overline{WAIT}$,ASTB			
16K-byte expansion mode	Address/data	Address						Port		Port				$\overline{RD}$,$\overline{WR}$,$\overline{WAIT}$,ASTB			
Full address mode	Address/data	Address								Port				$\overline{RD}$,$\overline{WR}$,$\overline{WAIT}$,ASTB			

Caution: When the external wait function is not used, the $\overline{WAIT}$ pin can be used as a port in all modes.

Memory maps when using the external device expansion function are as follows.

Fig. 19-1 Memory Map when Using External Device Expansion Function (1/3)

(a) μPD78094 Memory map when internal ROM size is 32K bytes



(b) μPD78095 Memory map when internal ROM size is 40K bytes

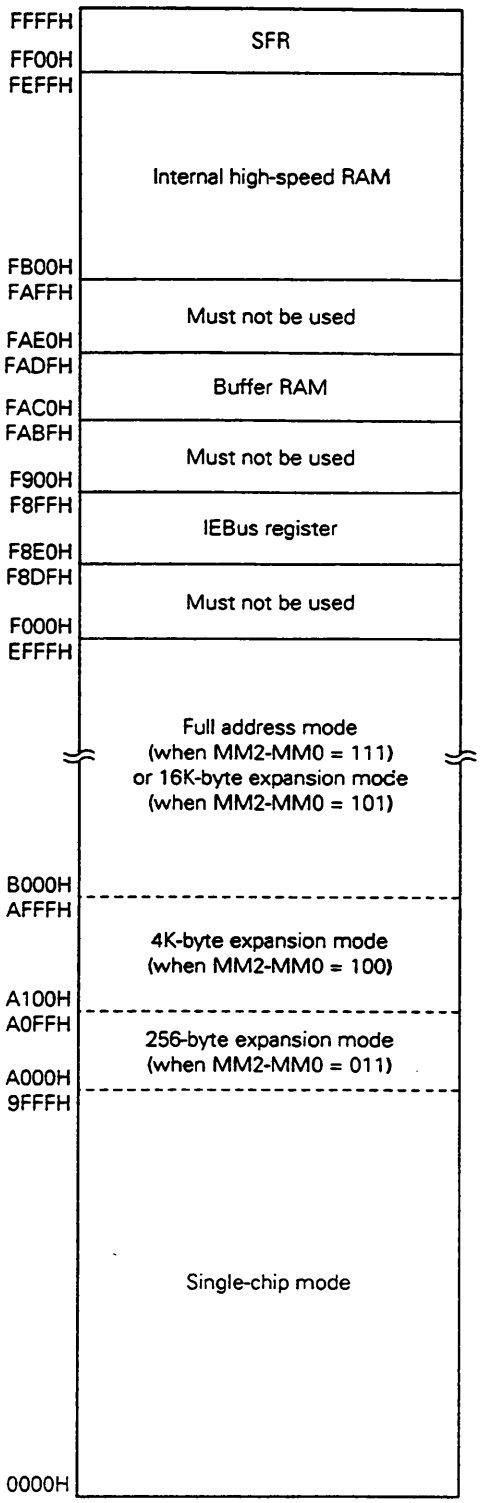


Fig. 19-1 Memory Map when Using External Device Expansion Function (2/3)

(c) μPD78096 Memory map when internal ROM size is 48K bytes

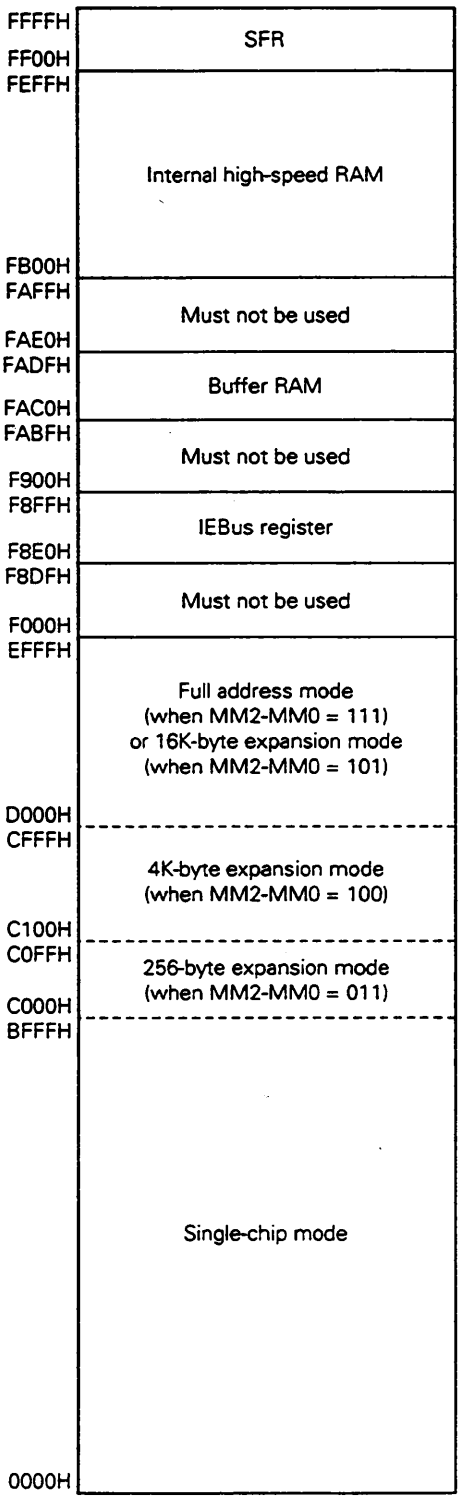
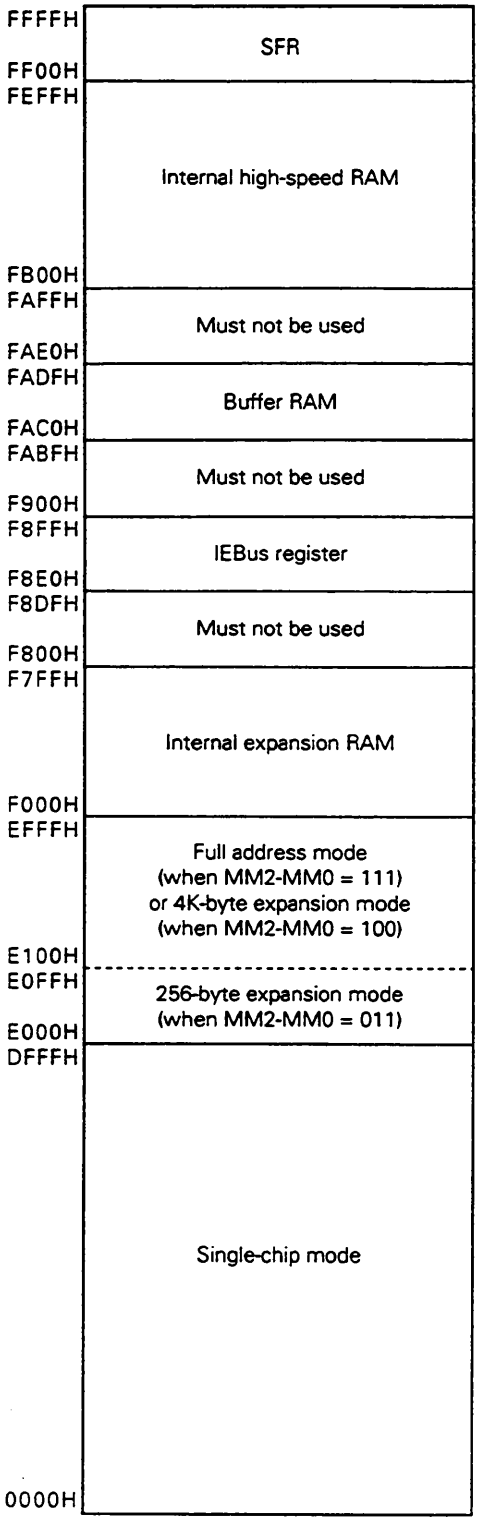
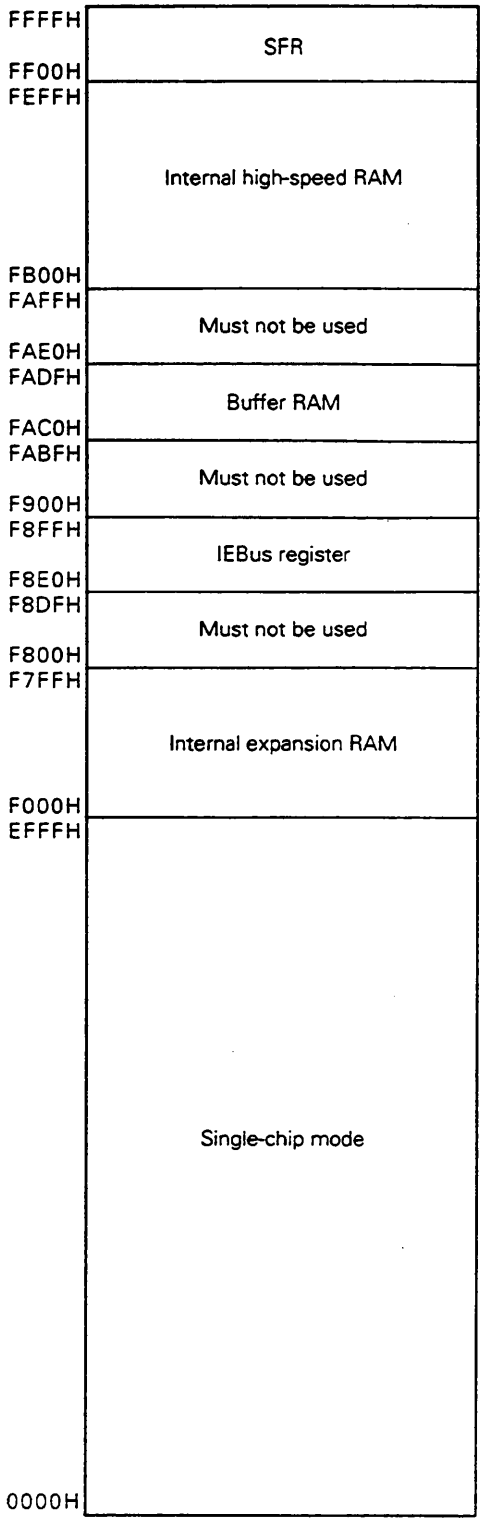


Fig. 19-1 Memory Map when Using External Device Expansion Function (3/3)

(d) μPD78098A and μPD78P098A Memory map when internal ROM size is 56K bytes



(e) μPD78098A and μPD78P098A Memory map when internal ROM size is 60K bytes



Caution: When the internal ROM (PROM) is 60K bytes, the external device expansion function cannot be used. By setting the internal ROM (PROM) to below 56K bytes using the memory size switching register, the area from the last address to address EFFFH of the internal ROM (PROM) can be used as the external memory.

19.2 External Device Expansion Function Control Register

The external device expansion function is controlled by the memory expansion mode register (MM) and memory size switching register (IMS).

(1) Memory expansion mode register (MM)

MM sets the wait count and external expansion area, and also sets the input/output of port 4.

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

$\overline{\text{RESET}}$ input sets this register to 10H.

Fig. 19-2 Memory Expansion Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
MM	0	0	PW1	PW0	0	MM2	MM1	MM0	FFF8H	10H	R/W

MM2	MM1	MM0	Single-chip/Memory Expansion Mode Selection		P40 to P47, P50 to P57, P64 to P67 Pin State				
					P40 to P47	P50 to P53	P54, P55	P56, P57	P64 to P67
0	0	0	Single-chip mode		Port mode: Input	Port mode			
0	0	1			Port mode: Output				
0	1	1	Memory expansion mode	256-byte mode	AD0 to AD7	Port mode			P64=$\overline{\text{RD}}$, P65=$\overline{\text{WR}}$, P66=$\overline{\text{WAIT}}$, P67=ASTB
1	0	0		4K-byte mode		A8 to A11	Port mode		
1	0	1		16K-byte mode			A12, A13	Port mode	
1	1	1		Full address mode[Note]				A14, A15	
Other than above			Setting prohibited						

Note: The full address mode allows external expansion to the entire 64K-byte address space except for the internal ROM, RAM, and SFR areas.

Remark: P60 to P63 enter the port mode without regard to the mode (single-chip mode or memory expansion mode).

PW1	PW0	Wait Control
0	0	No wait
0	1	Wait (one wait state insertion)
1	0	Setting prohibited
1	1	Wait control by external wait pin

Caution: When the 2/3 divider (in divider 1) of the clock generator circuit is used to generate the main system clock, the external device expansion function cannot be used.

(2) Memory size switching register (IMS)

The μPD78098A and μPD78P098A can specify the internal memory size by memory size switching register (IMS).

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

The value of this register is set as shown in Table 19-3 at $\overline{\text{RESET}}$.

★ **Fig. 19-3 Memory Size Switching Register Format**

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	Note 1	W

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
1	0	0	0	32K bytes
1	0	1	0	40K bytes
1	1	0	0	48K bytes
1	1	1	0	56K bytes Note 2
1	1	1	1	60K bytes
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
1	1	0	1024 bytes
Other than above			Setting prohibited

Notes:

(1) According to products, the value of the IMS register is different as shown in the following table at reset:

Table 19-3 Value of Memory Size Switching Register when Reset

Part Number	Value at Reset
μPD78094	C8H
μPD78095	CAH
μPD78096	CCH
μPD78098A	CFH
μPD78P098A	

(2) When using the external device expansion function for the μPD78098A and 78P098A, set the internal PROM capacity to below 56K bytes.

Caution: When using a mask ROM version, do not set values other than reset to IMS (excluding when using the external device expansion function for the μPD78098A).

Remark: Set IMS according to the internal ROM capacity.

19.3 External Device Expansion Function Timing

Timing control signal output pins in the external memory expansion mode are as follows.

(1) $\overline{RD}$ pin (Dual-function : P64)

Read strobe signal output pin. The read strobe signal is output in data accesses and instruction fetches from external memory.

During internal memory access, the read strobe signal is not output (maintains high level).

(2) $\overline{WR}$ pin (Dual-function : P65)

Write strobe signal output pin. The write strobe signal is output in data access to external memory.

During internal memory access, the write strobe signal is not output (maintains high level).

(3) $\overline{WAIT}$ pin (Dual-function : P66)

External wait signal input pin. When the external wait is not used, the $\overline{WAIT}$ pin can be used as an input/output port.

During internal memory access, the external wait signal is ignored.

(4) ASTB pin (Dual-function : P67)

Address strobe signal output pin. Timing signal is output without regard to the data accesses and instruction fetches from external memory.

(5) AD0 to AD7, A8 to A15 pins (Dual-function : P40 to P47, P50 to P57)

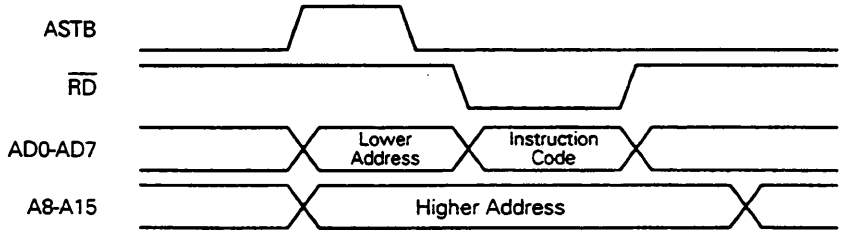
Address/data signal output pin. Valid signal is output or input during data accesses and instruction fetches from external memory.

When manipulating space mapped to the internal RAM area, the internal ROM area, and the SFR area (except the external SFR area), each area of internal RAM, ROM, and SFR is to be manipulated. At this time, the address signal is output externally, but $\overline{RD}$ and $\overline{WR}$ signals are not output (they remain inactive).

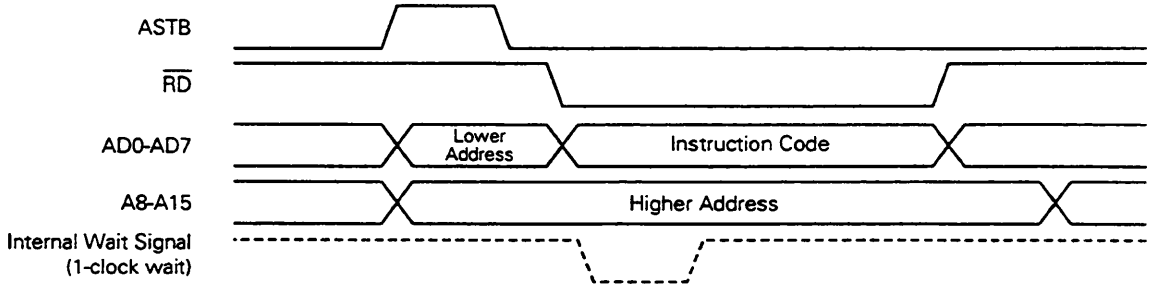
Timing charts are shown in Figs. 19-4 to 19-7.

Fig. 19-4 Instruction Fetch from External Memory

(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



(c) External wait (PW1, PW0 = 1, 1) setting

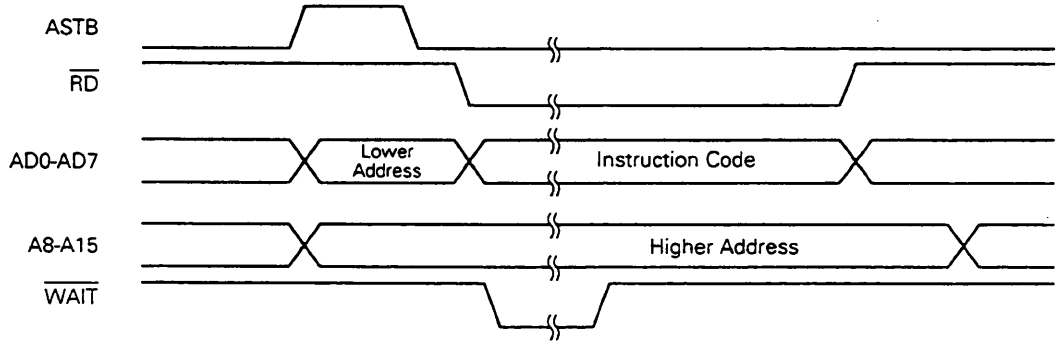
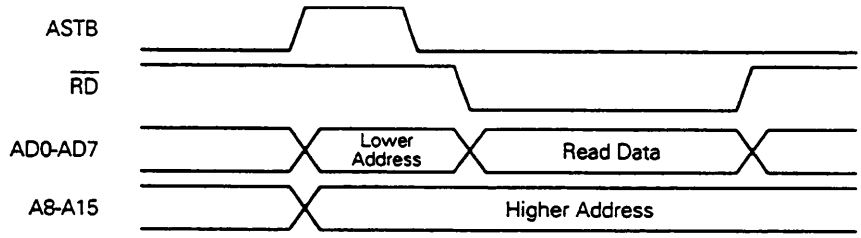
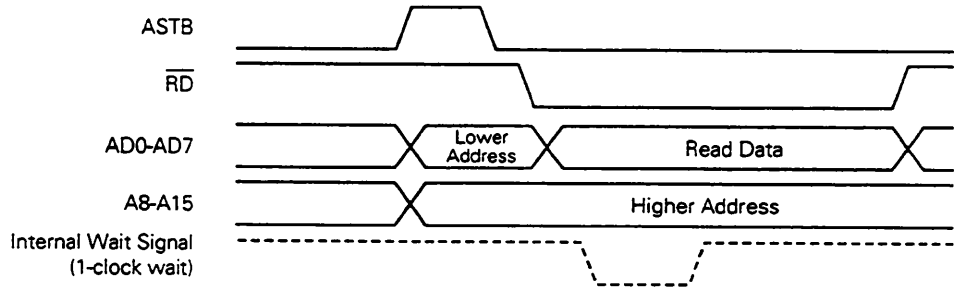


Fig. 19-5 External Memory Read Timing

(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



(c) External wait (PW1, PW0 = 1, 1) setting

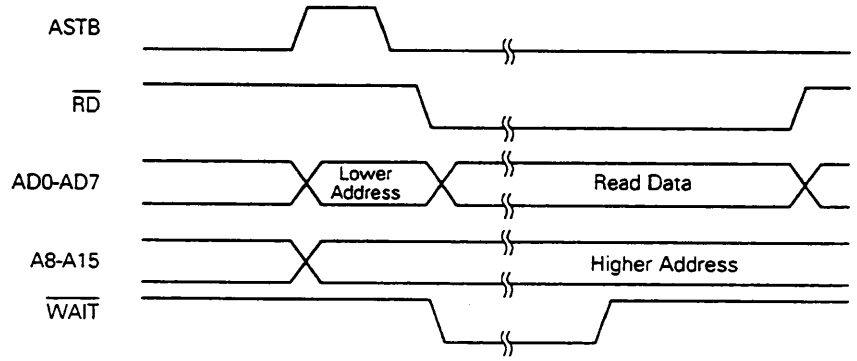
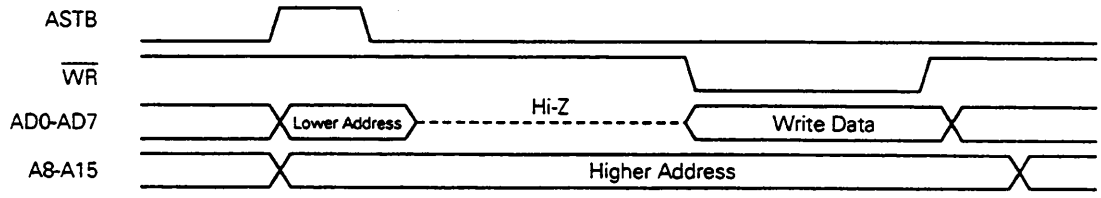
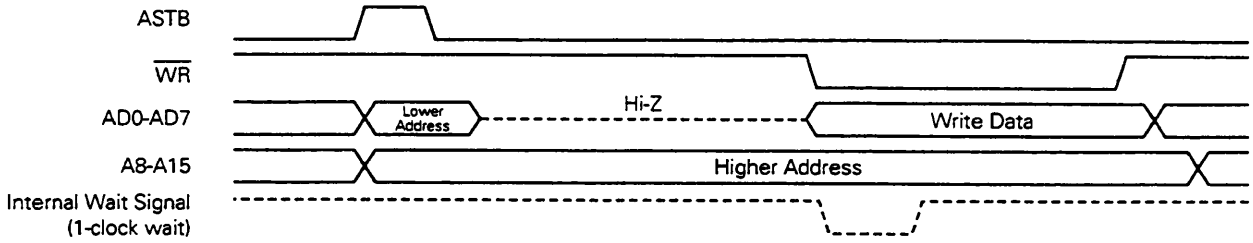


Fig. 19-6 External Memory Write Timing

(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



(c) External wait (PW1, PW0 = 1, 1) setting

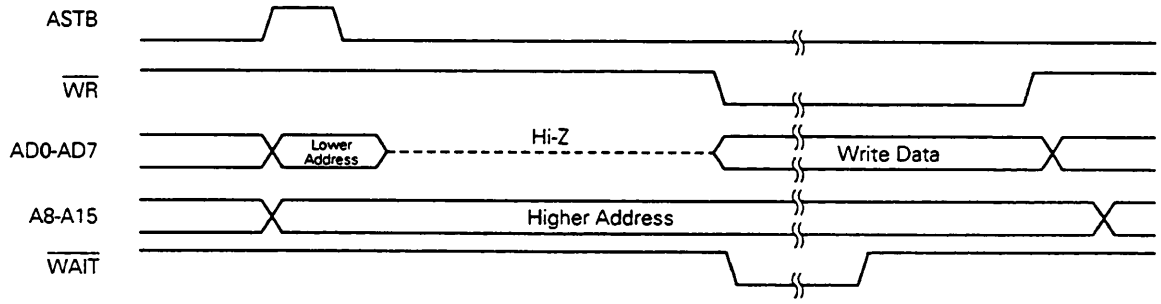
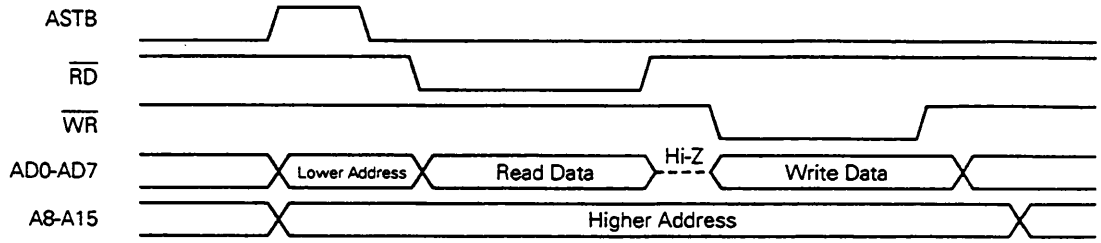
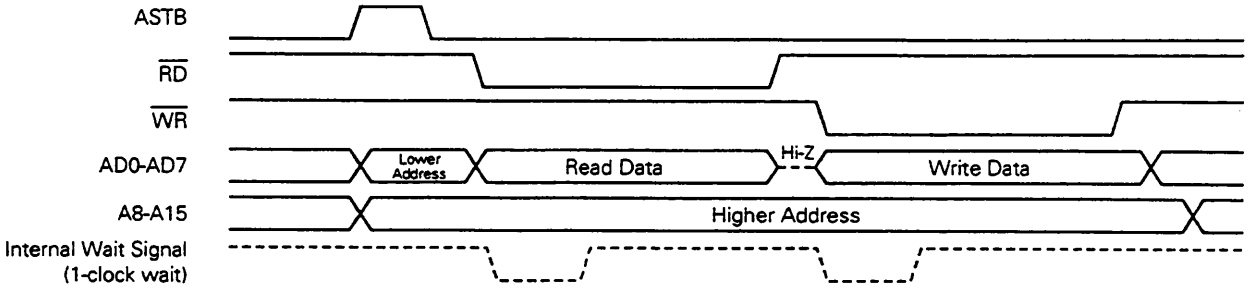


Fig. 19-7 External Memory Read Modify Write Timing

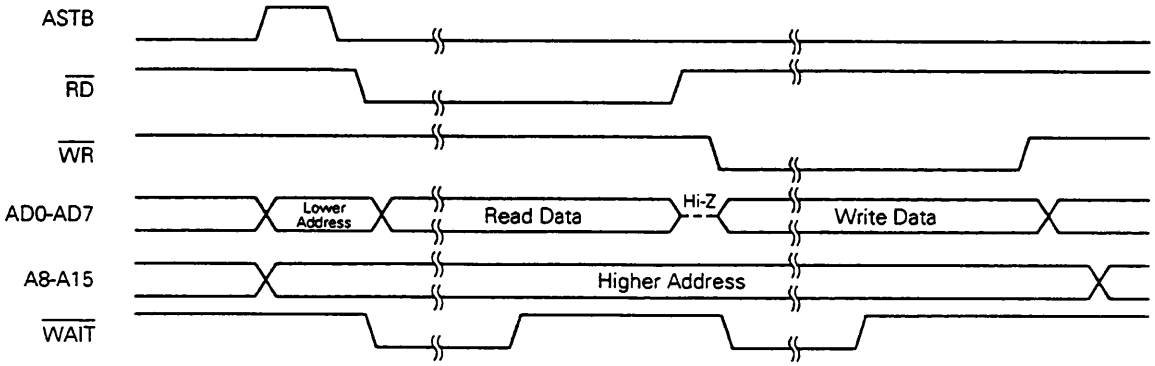
(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



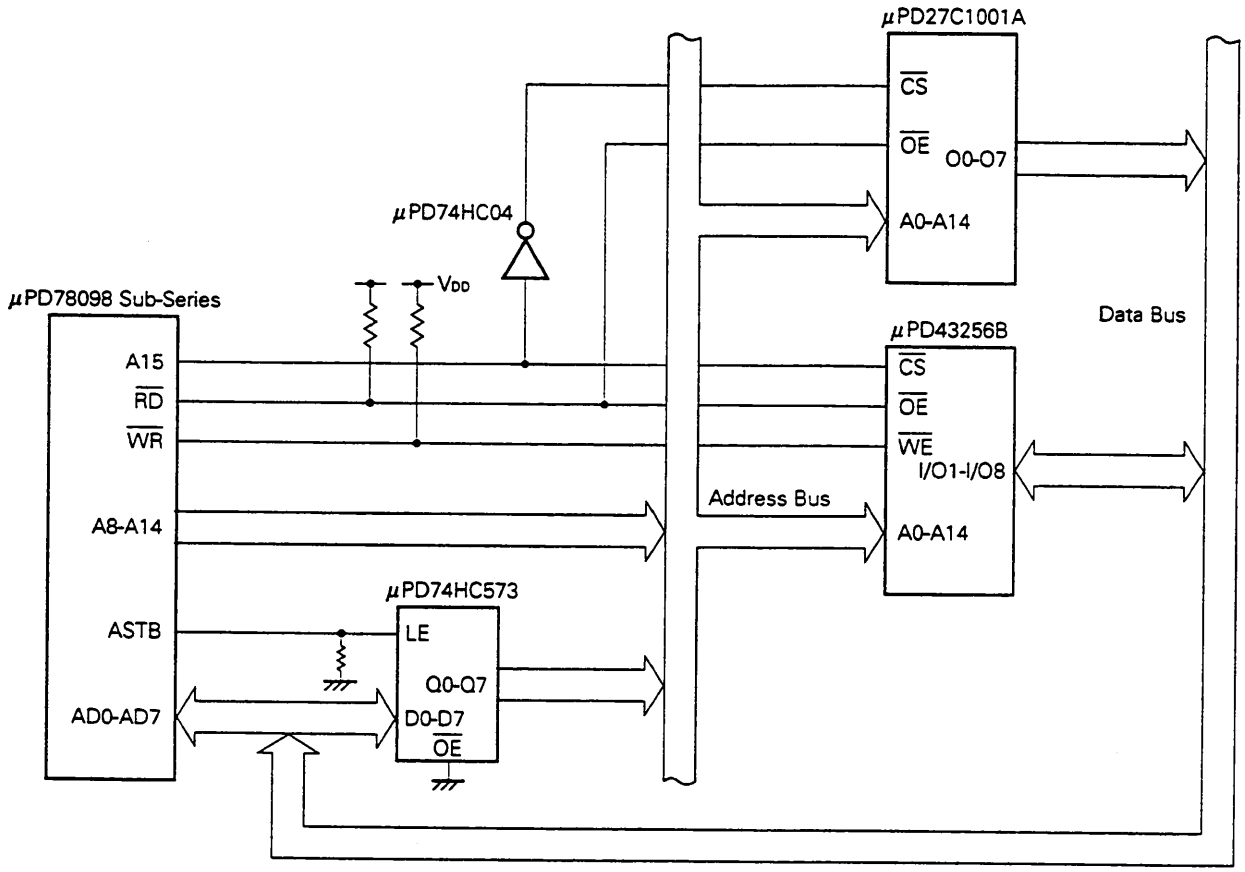
(c) External wait (PW1, PW0 = 1, 1) setting



19.4 Example of Connection with Memory

Fig. 19-8 shows example of connections between the μPD78098 sub-series and external memories. In this application example, a PROM and a SRAM are connected.

Fig. 19-8 Example of Connection of μPD78098 Sub-Series and Memory



Caution: Because the time from the rising of the $\overline{RD}$ signal to write data output is extremely short when the read-modify-write operation is performed on the external memory, the write data may contend with the output data of the external memory (such as SRAM). In this case, data contention can be avoided by not using the following instructions that generate the read-modify-write timing.

XCH	A, !addr16	XCH	A, [HL + C]
XCH	A, [DE]	MOV1	[HL] . bit, CY
XCH	A, [HL]	SET1	[HL] . bit
XCH	A, [HL + byte]	CLR1	[HL] . bit
XCH	A, [HL + B]	BTCLR	[HL] . bit, $addr16

CHAPTER 20 IEBus CONTROLLER

20.1 Function of IEBus Controller

IEBus (Inter Equipment Bus™) is a small-scale digital data transfer system intended to transfer data between equipment and equipment. To implement an IEBus with the μPD78098 sub-series, an IEBus driver/receiver must be externally prepared as they are not provided to the μPD78098.

The IEBus controller which the μPD78098 sub-series contains can select the positive or negative logic through software in accordance with the external IEBus driver/receiver (refer to **20.3 (2)**).

20.1.1 IEBus communication protocol

The communication protocol of the IEBus is as follows:

(1) Multi-master method

All the units connected to the IEBus can transfer data to the other units.

(2) Synchronous communication function

Communication between one unit and two or more units can be executed as follows:

- Group multiaddress communication : multiaddress communication to group unit
- Broadcasting communication : multiaddress communication to all units

(3) Effective transfer rate

The effective transfer rate can be selected from the following three modes:

- Mode 0 : approx. 3.9 k bps
- Mode 1 : approx. 17 k bps
- Mode 2 : approx. 26 k bps

Caution: Different modes must not exist on one IEBus.

(4) Communication method

Data are transferred by means of half duplex asynchronous communication.

(5) Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

The priority according to which the IEBus is occupied is as follows:

① Multiaddress communication takes precedence over individual communication (communication between one unit and another).

② The lower master address takes precedence over the higher address.

(6) Communication scale

The communication sale of the IEBus is as follows:

- Number of units: 50 max.
- Cable length: 150 m max. (when twisted pair cable is used)

Caution: The communication scale of the actual system differs depending on the IEBus driver/receiver and the characteristics of the cable constituting the IEBus.

20.1.2 Determining bus mastership (arbitration)

The equipment connected to the IEBus performs an operation to occupy the bus when it controls another equipment. This operation is called arbitration.

Arbitration is to grant the bus mastership to one of several units that have simultaneously started transmission.

Because only one equipment acquires the bus mastership as a result of arbitration, the following priority according to which equipment acquires the bus mastership is predetermined:

(1) Priority according to type of communication

Multiaddress communication (communication between one unit and several units) takes precedence over normal communication (communication between one unit and another).

(2) Priority according to master address

If the types of communication are the same, the lower master address takes precedence over the higher one.

Example: The master address consists of 12 bits, unit at 000H has the highest priority, and unit at FFFH has the lowest priority.

20.1.3 Communication mode

The IEBus provides three communication modes of which each has a different transfer rate. The transfer rate in each communication mode and the maximum number of transfer bytes in one communication frame are as shown in Table 20-1.

Table 20-1 Transfer Rate and Maximum Number of Transfer Bytes in Each Communication Mode

Communication mode	Maximum number of transfer bytes (bytes/frame)	Effective transfer rate (Kbps) Note
0	16	Approx. 3.9
1	32	Approx. 17
2	128	Approx. 26

Note: Effective transfer rate when the maximum number of transfer bytes has been received

A communication mode is selected for each equipment connected to the IEBus before communication is started. Note that communication is not performed correctly unless the communication modes of the master unit and the other unit (slave unit) are the same.

20.1.4 Communication address

The IEBus assigns a specific 12-bit communication address to each equipment. The communication address consists as follows:

Higher 4 bits : group number (number identifying group to which each equipment belongs)
Lower 8 bits : unit number (number identifying each equipment in a group)

20.1.5 Multiaddress communication

In the normal communication mode, there are only one master unit and one slave unit between which communication is performed on a one-to-one basis. In contrast, multiaddress communication allows the master unit to transmit data to more than one slave unit.

Because two or more slave units exist, no slave unit returns an acknowledge signal during communication.

Whether multiaddress communication or normal communication is to be performed can be selected by a multiaddress bit (for details on the multiaddress bit, refer to **20.1.6 (2) Multiaddress bit**).

Multiaddress communication can be carried out in the following two modes:

(1) Group multiaddress communication

Multiaddress communication is performed to equipment in a group having a group number equal to the one of the higher 4 bits of the communication address.

(2) Broadcasting communication

Multiaddress communication is performed on all the equipment, regardless of the value of the group number.

Whether group multiaddress communication or broadcasting communication is performed is identified by the value of the salve address (for details on the slave address, refer to **20.1.6 (4) Slave address field**).

20.1.6 IEBus transmission format

Fig. 20-1 shows the transmit signal format of the IEBus.

Fig. 20-1 Transmit Signal Format of the IEBus

	Header		Master address field		Slave address field			Control field			Telegraph length field			Data field					
Frame format	Start bit	Multi-address bit	Master address bit	P	Slave address bit	P	A	Control bit	P	A	Tele-graph length bit	P	A	Data bit	P	A	Data bit	P	A

Remarks:

(1) P: parity bit, A: $\overline{\text{ACK}}$/NAK bit

(2) The master unit ignores the acknowledge bit when multiaddress communication is performed.

(1) Start bit

The start bit is a signal by which a unit informs the other units of the start of data transmission.
The unit that is to start data transmission outputs a low-level signal (start bit) for a specific time and then outputs the multiaddress bit.
If another unit has already output a start bit when one unit tries to output the start bit, this unit does not output the start bit but waits for the end of the start bit output by the another unit, and outputs the multiaddress bit in synchronization with the output end timing of the start bit.
The units other than the one that has started transmission detect this start bit and enters the receive status.

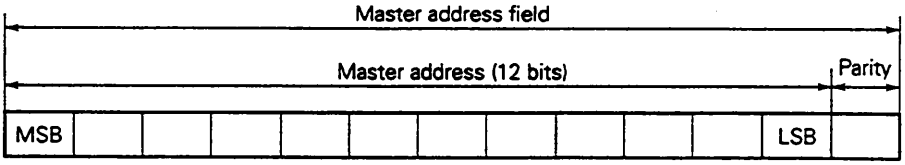
(2) Multiaddress bit

This bit indicates whether the master selects only one slave (individual communication) or two or more slaves (multiaddress communication).
When the multiaddress bit is 0, it indicates that the master selects two or more units; when this bit is 1, the master selects only one unit.
Multiaddress communication is divided into two modes: group multiaddress communication and broadcasting communication. These modes are identified by the value of the slave address (for details on the slave address, refer to **(4) Slave address field**).
Since there are two or more slave units in the case of multiaddress communication, the acknowledge bit for each field following the master address field is not returned.
If two or more units start transmission of a communication frame at the same time, multiaddress communication takes precedence over individual communication and is the winner of arbitration.
If a unit acquires the bus mastership as the master, the value set to the multiaddress bit (bit 7) of the master communication control register (MCR) is output .

(3) Master address field

This field is output by the master to inform the slaves of its unit address.
Fig. 20-2 shows the configuration of the master address field.
If two or more units starts transmitting the multiaddress bit of the same value at the same time, judgment of arbitration is made by the master address field.
The master address field compares the data output by the master with the data on the bus each time the master transmits 1 bit.
If the master address output by the master is found to be different from the data on the bus as a result of the comparison, it is judged that the master is the loser of arbitration, and the master stops transmission and enters receive status.
Since the IEBus is configured as wired AND, the unit having the least master address of the units participating in arbitration (arbitration masters) wins arbitration.
After each unit has output a 12-bit master address, only one unit remains in transmit status as the master unit.
This master unit then outputs the parity bit, and makes the other units determine the master address, and then outputs the slave address field.
If a unit acquires the bus mastership as the master, the address set by the unit address registers 1 and 2 (UAR1 and UAR2) is output.

Fig. 20-2 Master Address Field



(4) Slave address field

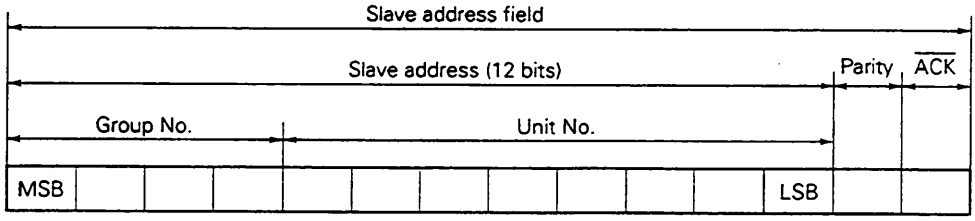
This field outputs the address of the other party with which the master is to communicate.
The slave address field is configured as shown in Fig. 20-3.
After a 12-bit slave address has been transmitted, a parity bit is output so that the slave address is not received by mistake.
Next, the master unit detects an acknowledge signal from the slave unit to confirm that the slave unit exists on the bus.
When the master unit has detected the acknowledge signal, the master unit starts outputting the control field. In multiaddress communication mode, however, the master starts outputting the control field without detecting the acknowledge bit.
The slave unit detects that the slave address has coincided, and that the parities of both the master address and slave address are even, and then outputs an acknowledge signal. If a parity is odd, the slave unit judges that either the master address or slave address has not been received correctly, and does not output the acknowledge signal. At this time, the master unit enters the standby (monitor) status, and communication ends.
In multiaddress communication mode, the slave address is used to identify whether group multiaddress communication or broadcasting communication is to be executed, as follows:

When slave address is FFFH : Broadcasting communication
When slave address is not FFFH : Group multiaddress communication

Remark: The group number for group multiaddress communication is the value of the higher 4 bits of the slave address.

When a unit acquires the bus mastership as the master, the address set by the slave address registers 1 and 2 (SAR1 and SAR2) is output.

Fig. 20-3 Slave Address Field



(5) Control field

Output to this field is the operation the master requested of the slave.

Fig. 20-4 shows the configuration of the control field.

If the parity following the control bits is even and if the slave can execute the function requested by the master, the slave unit outputs an acknowledge signal, and outputs the telegraph length field. Even if the parity is even, however, if the slave unit cannot execute the function requested by the master, or if the parity is odd, the slave unit does not output the acknowledge signal and returns to the standby (monitor) status.

The master unit confirms the acknowledge signal and then proceeds to receiving the telegraph length field. If the master cannot confirm the acknowledge signal, it enters the standby status, and communication ends. In multiaddress communication mode, however, the master unit does not confirm the acknowledge signal, but receives the telegraph length field.

Table 20-2 shows the contents of the control bits.

Table 20-2 Contents of Control Bits

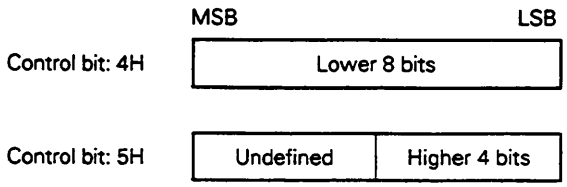
Bit 3 Note1	Bit 2	Bit 1	Bit 0	Function
0	0	0	0	Reads slave status
0	0	0	1	Undefined
0	0	1	0	Undefined
0	0	1	1	Reads and locks data Note2
0	1	0	0	Reads lock address (lower 8 bits) Note3
0	1	0	1	Reads lock address (higher 4 bits) Note3
0	1	1	0	Reads and unlocks slave status Note2
0	1	1	1	Reads data
1	0	0	0	Undefined
1	0	0	1	Undefined
1	0	1	0	Writes and locks command Note2
1	0	1	1	Writes and locks data Note2
1	1	0	0	Undefined
1	1	0	1	Undefined
1	1	1	0	Writes command
1	1	1	1	Writes data

Notes:

(1) The direction in which the telegraph length field, telegraph length bits, and data in the data field are transferred is changed depending on the value of bit 3 (MSB), as follows:
When bit 3 is '1': Transfer from master unit to slave unit
When bit 3 is '0': Transfer from slave unit to master unit

(2) This is a control bit that specifies locking or unlocking (refer to **20.1.7 (4) Locking and unlocking**).

(3) Since the lock address is transmitted in 1-byte (8-bit) units, it is configured as follows:



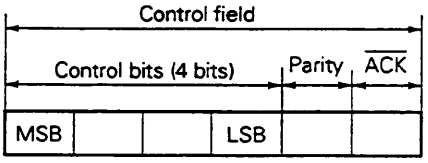
The unit that is locked by the master unit rejects acceptance if the control bits it has received from any unit other than the master unit which has requested for locking are other than as shown in Table 20-3, and does not output the acknowledge bit.

Table 20-3 Control Field for Locked Slave Unit

Bit 3	Bit 2	Bit 1	Bit 0	Function
0	0	0	0	Reads slave status
0	1	0	0	Reads lock address (lower 8 bits)
0	0	0	1	Reads lock address (higher 4 bits)

If a unit acquires the bus mastership as the master, the value set to the control bit of the master communication control register (MCR) is output.

Fig. 20-4 Control Field



(6) Telegraph length field

This field is output by the transmitter to inform the receiver of the number of bytes of the transmit data. The telegraph length field is configured as shown in Fig. 20-5.

Table 20-4 shows the relations between the telegraph length field and the number of transmit data bytes.

Fig. 20-5 Telegraph Length Field

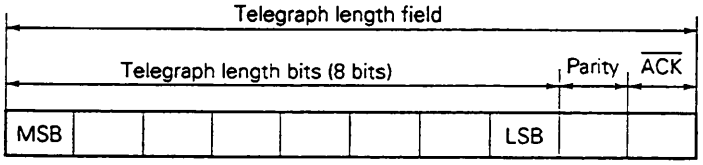


Table 20-4 Contents of Telegraph Length Bits

Telegraph length bits (HEX)	Number of transmit data bytes
01H	1 byte
02H	2 bytes
⋮	⋮
FFH	255 bytes
00H	256 bytes

The operation of the telegraph length field differs depending on whether the master is in the transmit mode (when bit 3 of the control bits is 1) or in the receive mode (when bit 3 of the control bits is 0), as follows:

① **When master is in transmit mode**

The telegraph length bits and parity bit are output by the master unit. The slave unit, if it detects that the parity is even, outputs the acknowledge signal, and then outputs the telegraph length field. In the multiaddress communication mode, however, the slave unit does not output the acknowledge signal.

If the parity is odd, the slave unit judges that the telegraph length bits have not been correctly received, and returns to the standby (monitor) mode without outputting the acknowledge signal.

At this time, the master unit also returns to the standby status, and communication ends.

② **When master is in receive mode**

The slave unit outputs the telegraph length bits and parity bit.

The master unit, if it has detected that the parity is even, outputs the acknowledge signal.

If the parity is odd, the master unit judges that the telegraph length bits have not been correctly received, and returns to the standby status without outputting the acknowledge signal. At this time, the slave unit also returns to the standby status, and communication ends.

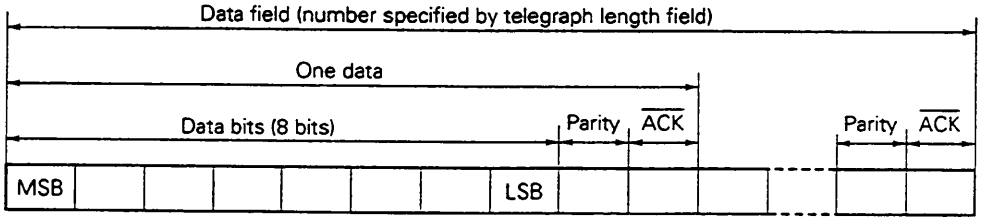
(7) Data field

This field is output by the transmitter.

The master unit transmits data to a slave unit by using the data field, receives data from a slave unit.

Fig. 20-6 shows the configuration of the data field.

Fig. 20-6 Data Field



Following the data bits, a parity bit and acknowledge bit are output by the master and slave units.

Multiaddress communication is performed only when the master unit transmits data. At this time, the acknowledge signal is ignored.

The operations when the master transmits and receives data are as follows:

① **When master transmits data**

When the master unit writes data to a slave unit, the master unit transmits data bit and parity bit to the slave unit. The slave unit receives the data bit and parity bit. If the parity is even, and if the receive buffer is empty, the slave unit outputs an acknowledge signal. If the parity is odd or if the receive buffer is not empty, the slave unit rejects accepting the corresponding data, and does not output the acknowledge signal.

If the slave unit does not output the acknowledge signal, the master unit transmits the same data again. This operation is continued until the master unit detects the acknowledge signal from the slave unit, or the data exceeds the maximum number of transmit bytes.

If the parity is even and if the acknowledge signal is output from the slave unit, the master unit transmits the next data if there is data to be transmitted and if the maximum number of transmit bytes is not exceeded.

In the multiaddress communication mode, the slave unit does not output the acknowledge signal, and the master unit transmits data on a 1-byte-by-1-byte basis.

② **When master receives data**

When the master unit reads data from the slave unit, the master unit outputs a synchronous signal corresponding to all the read bits.

The slave unit outputs the contents of the data and parity bit to the bus in response to the synchronous signal output by the master unit.

The master unit reads the data and parity bit output by the slave unit, to confirm the parity bit.

If the parity is odd, or if the receive buffer is not empty, the master unit rejects accepting the data, and does not output the acknowledge signal. If the data is within the maximum number of bytes that can be transmitted in one communication frame, the master unit repeatedly reads the same data.

If the parity is even and if the receive buffer is empty, the master unit accepts the data, and returns the acknowledge signal.

If the data is within the maximum number of bytes that can be transmitted in one frame, the master unit reads the next data.

(8) Parity bit

The parity bit is used by the communicating units to confirm that the transmit data contains no error.

A parity bit is appended to each data of the master address bits, slave address bits, control bits, telegraph length bits, and data bits.

The parity is an even parity. If the number of bits in the data that are '1' is odd, the parity bit is set to '1'; if the number of bits that is '1' is even, the parity bit is cleared to '0'.

(9) Acknowledge bit

An acknowledge bit is appended to the following location to confirm whether data has been correctly received in the normal communication mode (communication between one unit and another):

- At end of slave address field
- At end of control field
- At end of telegraph length field
- At end of data field

The acknowledge bit is defined as follows:

- '0': The transmit data is recognized (ACK).
- '1': The transmit data is not recognized (NAK).

In the case of multiaddress communication, however, the acknowledge bit is ignored.

① **Acknowledge bit at end of slave field**
The acknowledge bit at the end of the slave field is treated as NAK in any of the following cases, and transmission is aborted:

- If the parity of the master address bit or slave address bit is not correct
- If a timing error (error in bit format) occurs
- If the specified slave unit does not exist

② **Acknowledge bit at end of control field**
The acknowledge bit at the end of the control field is treated as NAK in any of the following cases, and transmission is aborted:

- If the parity of the control bit is not correct
- If bit 3 of the control bits is '1' (write operation) though the slave receive buffer Note is not empty
- If the control bits indicate data read (3H, 7H) through the slave transmit buffer Note is empty
- 3H, 6H, 7H, AH, BH, EH, and FH of the control bits are requested by a unit other than that sets locking through the locking has been set
- If the control bits indicate reading of a lock address (4H) though locking has not been set
- If a timing error occurs
- If the control bits are not defined

Note: Refer to **20.1.7 (1) Slave status**.

③ **Acknowledge bit at end of telegraph length field**
The acknowledge bit at the end of the telegraph length field is treated as NAK in any of the following cases, and transmission is aborted:

- If the parity of the telegraph length bits is not correct
- If a timing error occurs

④ **Acknowledge bit at end of data field**
The acknowledge bit at the end of the data field is treated as NAK in any of the following cases, and transmission is aborted:

- If the parity of the data bits is not correct Note
- If a timing error occurs after the previous acknowledge bit has been transmitted
- If the receive buffer has become full and thus no more data cannot be accepted Note

Note: In this case, the transmitter resumes transmitting the data field if the data is within the maximum number of bytes that can be transmitted in one frame.

20.1.7 Transmit data

(1) Slave status

The master unit can learn the reason why the slave unit has not returned the acknowledge bit (ACK) by reading the slave status.

The slave status is determined in respect to the result of the last communication performed by the slave unit.

All the slave units can supply information of the slave status.

Table 20-5 shows the meaning of the slave status.

Fig. 20-7 Slave Status Bit Configuration

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 20-5 Meaning of Slave Status

Bit	Value	Meaning	
Bit 0 Note1	0	Slave transmit buffer is empty	
	1	Slave transmit buffer is not empty	
Bit 1 Note2	0	Slave receive buffer is empty	
	1	Slave receive buffer is not empty	
Bit 2	0	Unit is not locked	
	1	Unit is locked	
Bit 3	0	Fixed to '0'	
Bit 4 Note3	0	Slave stops transmitting	
	1	Slave is enabled to transmit	
Bit 5	0	Fixed to '0'	
Bits 7 and 6	00	Mode 0	Indicate highest mode supported by unit Note4
	01	Mode 1	
	10	Mode 2	
	11	For future expansion	

Notes:

(1) The slave transmit buffer is the buffer accessed when data is read (control bits: 3H, 7H).
This buffer is the TBF if the μPD78098 sub-series is the slave unit.

(2) The slave receive buffer is the buffer accessed when data is written (control bits: 8H, AH, BH, EH, FH).
This buffer is the RBF if the μPD78098 sub-series is the slave unit.

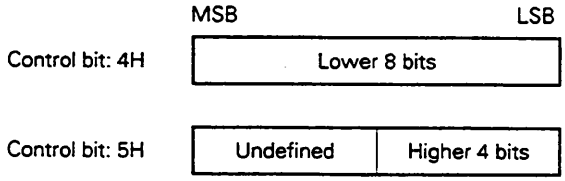
(3) If the μPD78098 sub-series is the slave unit, the status indicated by the bit 4 (SLRE) of the status register 2 (STR2).

(4) If the μPD78098 sub-series is the slave unit, bits 6 and 7 are respectively fixed to '0' and '1'.

(2) Lock address

When the lock address is read (control bits: 4H, 5H), the address (12-bit) of the master unit that has issued the lock instruction is configured in 1-byte units as shown below and is read.

Fig. 20-8 Lock Address Configuration



(3) Data

If the control bits indicate data read (3H, 7H), the data in the data buffer of the slave unit is read to the master unit.

If the control bits indicate data write (BH, FH), the data received by the slave unit is processed in accordance with the operation regulations of that slave unit.

(4) Locking and unlocking

The lock function is used to transmit a message over two or more communication frames.

The unit that has been locked does not receive data from any unit other than the one that has locked it.

A unit is locked or unlocked as follows:

① Locking

If the communication frame ends after the acknowledge bit '0' of the telegraph length field has been transmitted or received by the control bits that have specified locking (3H, AH, BH) without transmission or reception of the number of data bytes specified by the telegraph length bits being successful, the slave unit is locked by the master unit. At this time, the bit related to locking (bit 2) in the byte that indicates the slave status is set to '1'.

② Unlocking

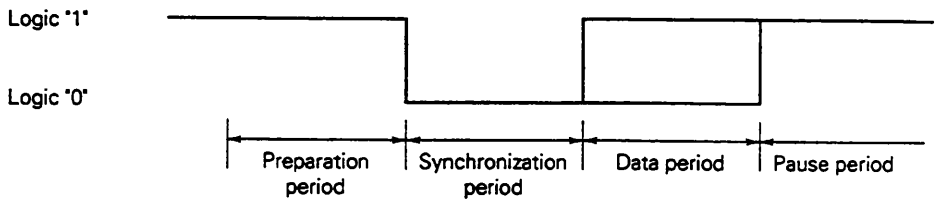
After the number of data bytes specified by the telegraph length bits has been transmitted or received within one communication frame by the control bits that have specified locking (3H, AH, BH), or by the control bit that have specified unlocking (6H), the slave unit is unlocked by the master. At this time, the bit related to locking (bit 2) in the byte that indicates the slave status is reset to '0'.

The slave unit is not locked or unlocked while multiaddress communication is performed.

20.1.8 Bit format

Fig. 20-9 shows the format of the bits constituting an IEBus communication frame.

Fig. 20-9 IEBus Bit Format



Preparation period	:	First low-level period (logic "1")
Synchronization period	:	Next high-level period (logic "0")
Data period	:	Period indicating bit value
Pause period	:	Last low-level period (logic "1")

The lengths of the synchronization period and data period are almost the same.

The IEBus establishes synchronization for each bit. The specifications of the time of the entire bit and the time of the period assigned to the bit differ depending on the type of the transmit bit, and whether the unit is master or slave.

20.2 IEBus Controller Configuration

The IEBus controller consists of the following hardware:

Table 20-6 IEBus Controller Configuration

Item	Configuration
Register	Transmit buffer register (TBF) Receive buffer register (RBF)
Control register	Clock select register 1 (IECL1) IEBus controller mode register (IECM) Control register (CTR) Command register (CMR) Master communication control register (MCR) Status registers 1 and 2 (STR1 and STR2) Receive data number registers 1 and 2 (RDR1 and RDR2) Return code register (RCR) Unit address registers 1 and 2 (UAR1 and UAR2) Slave address registers 1 and 2 (SAR1 and SAR2) Multiaddress registers 1 and 2 (DAR1 and DAR2) Lock address registers 1 and 2 (LOR1 and LOR2)

(1) Transmit buffer register (TBF)

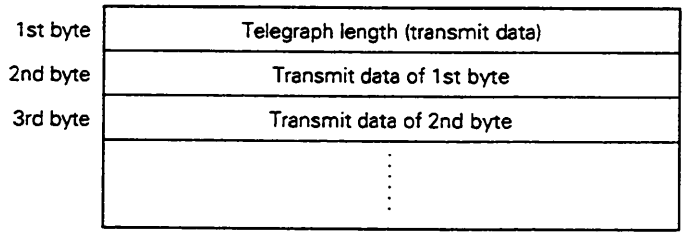
This register transfers the telegraph length and transmit data to the IEBus controller when data is to be transmitted.

Internally, this register is a 33-byte FIFO buffer.

Data can be written to TBF when the TFL flag of status register 1 (STR1) is 0 (when TBF is not full), by an 8-bit memory manipulation instruction.

The value of this register is undefined when the hardware $\overline{\text{RESET}}$ signal is input, or when the IEBus controller is reset via software.

Fig. 20-10 TBF Setting Format



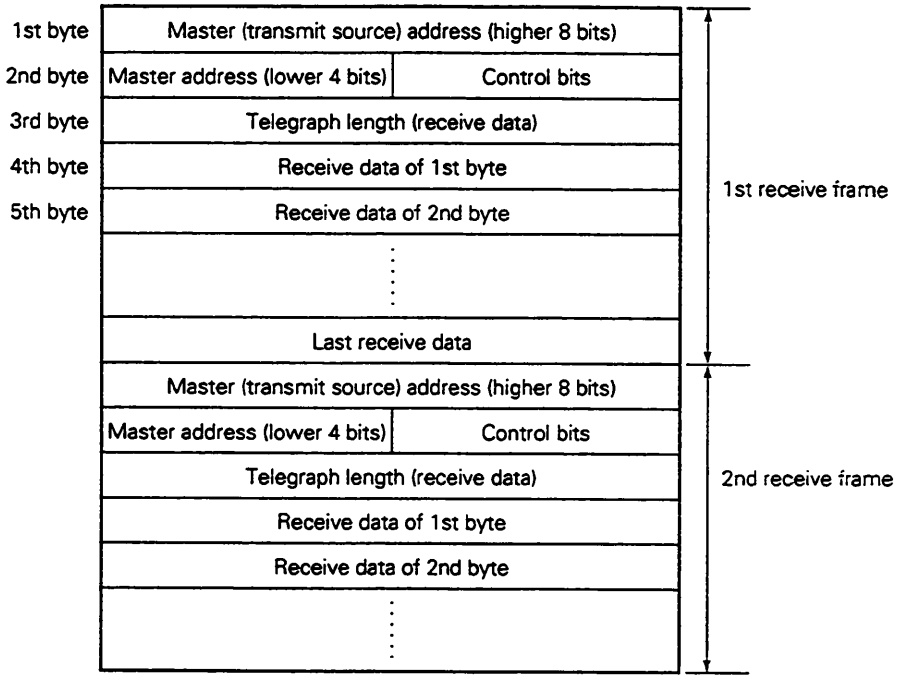
(2) Receive buffer register (RBF)

This register receives information such as the master address (transmit unit address), telegraph length, and receive data from the IEBus controller when data is to be received. Internally, this register is a 40-byte FIFO buffer.

RBF can be read when the REP flag of status register 1 (STR1) is 0 (when RBF is not empty), by an 8-bit manipulation instruction.

The value of this register is undefined when the hardware $\overline{\text{RESET}}$ signal is input or the IEBus controller is reset via software.

Fig. 20-11 RBF Storing Format



20.3 Registers Controlling IEBus Controller

The IEBus controller is controlled by the following 12 registers:

- Clock select register 1 (IECL1)
- IEBus controller mode register (IECM)
- Control register (CTR)
- Command register (CMR)
- Master communication control register (MCR)
- Status registers 1 and 2 (STR1 and STR2)
- Receive data number registers 1 and 2 (RDR1 and RDR2)
- Return code register (RCR)
- Unit address registers 1 and 2 (UAR1 and UAR2)
- Slave address registers 1 and 2 (SAR1 and SAR2)
- Multiaddress registers 1 and 2 (DAR1and DAR2)
- Lock address registers 1 and 2 (LOR1 and LOR2)

(1) Clock select register 1 (IECL1)

This register specifies whether the output of the main system clock oscillator circuit is supplied to the IEBus controller via divider 2 or not.

This register is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IECL1 to 00H.

Fig. 20-12 Clock Select Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
IECL1	0	0	0	0	0	0	IECL11	IECL10	F8E0H	00H	R/W

IECL11	Controls divider of main system clock (divider 2)
0	Uses 1/2 divider in divider 2
1	Does not use 1/2 divider in divider 2

Caution: Be sure to clear bits 2 to 7 to 0.

(2) IEBus controller mode register (IECM)

This register selects a function of a dual-function pin and sets the output logic level when the IEBus function is used.

This register is set with an 8-bit memory manipulation instruction.

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

Fig. 20-13 IEBus Controller Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
IECM	0	0	0	0	0	0	IECM1	IECM0	F8E3H	00H	R/W

IECM0	Selects functions of P124/$\overline{TX}$ and P125/$\overline{RX}$ pins
0	Uses as normal port pins
1	Uses as IEBus controller pins

IECM1	Selects logics of P124/$\overline{TX}$ and P125/$\overline{RX}$ pins
0	Makes output of pin negative logic (logic 1 is low level and logic 0 is high level)
1	Makes output of pin positive logic (logic 1 is high level and logic 0 is low level)

★

Caution: Be sure to clear bits 2 to 7 to 0.

(3) Control register (CTR)

This register controls the operation of the IEBus controller.
This register is set with an 8-bit memory manipulation instruction.
The contents of this register are set to 01H when the $\overline{RESET}$ signal is input or when the IEBus controller is reset via software.

Caution: When using the IEBus controller, be sure to perform reset operation (SRST = 1) first. Reset operation by setting SRST = 1 and releasing the standby mode by resetting STREQ = 0 can be simultaneously performed.

Fig. 20-14 Control Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
CTR	0	0	0	REEN	SRST	0	FAI	STREQ	F8F0H	01H	W

REEN[Note1]	Used to request IEBus controller to enable reception (individual/multiaddress). By setting this bit to 1, IEBus controller is requested to enable reception. This bit is automatically cleared to 0 by IEBus controller; therefore, 0 needs not to be written to this bit after request has been made.

SRST	Used to reset IEBus controller via software. By setting this bit to 1, IEBus controller is requested to be reset. This bit is automatically cleared to 0 by IEBus controller; therefore, 0 needs not to be written after request has been made.

FAI[Note2]	Sets position at which FCH is set to 1 after receive buffer register (RBF) has been read. When this bit is set to 1, position at which FCH is set to 1 is moved after current frame has been read. Each time this bit is set to 1, position is moved 1 frame backward. This bit is automatically cleared to 0 by IEBus controller. Position at which FCH is set can be moved 4 frames at a time (including currently read frame).

STREQ	Controls standby operation [Note3]
0	Requests IEBus controller to release standby mode
1	Requests IEBus controller to set standby mode

Caution: Be sure to clear bits 2 and bits 5 to 7 to 0.

Notes:

(1) The result of the request (to enable/disable reception) is reflected on SLRE (bit 4 of STR2); therefore, check SLRE for the actual status.

(2) This function is valid only when FCHC is set to 1 through optional setting to CMR. Do not set a value greater than the number of completely received frames stored in the receive buffer register (RBF).

(3) The result of the request (setting/releasing standby mode) is reflected on STM (bit 1 of STR2); therefore, check STM for the actual status. If STREQ is set to 1 in the middle of communication, the request to set/release the standby mode is not made until the current communication frame completes communication.

Remark: FCH: bit 3 of status register 1 (STR1)
FCH is a flag that is set to 1 when the number of communication frames set by FAI has been completely read when RBF is read. By reading this flag, the RBF read status can be learned. For further information, refer to **20.3 (6)**.

(4) Command register (CMR)

This register controls communication and transmit/receive buffers, and sets option functions.
The usage of this register is specified by the value of the most significant bit (0: control/1: option setting).
This register is set with an 8-bit memory manipulation instruction.
The contents of this register are cleared to 00H when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Fig. 20-15 Command Register Format (when used for control)

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
CMR	0	LOCK	BUFC1	BUFC0	COMC3	COMC2	COMC1	COMC0	F8F1H	00H	W

LOCK	Controls updating operation of lock address registers 1 and 2 (LOR1 and LOR2)
0	Disables updating LOR1 and LOR2
1	Enables updating LOR1 and LOR2

BUFC1	BUFC0	Controls transmit/receive buffer
0	0	Executes noting
0	1	Clears transmit buffer register (TBF)
1	0	Clears all frames of receive buffer register (RBF)
1	1	Clears 1 frame (latest receive frame) of receive buffer register (RBF)

COMC3	COMC2	COMC1	COMC0	Control related to communication to IEBus controller
0	0	0	0	Executes nothing
0	0	0	1	Requests unlocking (by request from other units)
1	0	0	0	Request communication as master
1	0	0	1	Continues previous master communication status and requests communication as master
1	0	1	0	Requests abortion of master communication. However, frame under communication is output to end
1	0	1	1	Requests slave data transmission
1	1	0	0	Continues previous slave data transmit status and requests slave data transmission
1	1	0	1	Requests abortion of slave data transmission. However, frame under communication is output to end
1	1	1	0	Requests self check communication
1	1	1	1	Requests disabling reception (individual/multiaddress)
Other than above				Setting prohibited

Fig. 20-16 Command Register Format (when used for option function setting)

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
CMR	1	0	0	0	0	0	FCHC	DERC	F8F1H	00H	W

FCHC	Controls read check of received communication frame
0	Does not perform read check of received communication frame
1	Performs read check of received communication frame

DERC	Controls multiaddress reception error
0	Disables multiaddress receive error check
1	Enables multiaddress receive error check

Cautions:

(1) Be sure to clear bits 2 to 6.

(2) Be sure to set option functions after the IEBus controller has been reset via software (SRST ← 1). The IEBus controller does not accept communication at all until the option functions are set.

(3) Write data to CMR when the CEX bit of status register 2 (STR2) is 0.

(5) Master communication control register (MCR)

This register sets the communication conditions under which the master unit communicates.
This register is set with an 8-bit memory manipulation instruction.
Its value becomes undefined when the $\overline{\text{RESET}}$ signal is input or the IEBus controller is reset via software.

Fig. 20-17 Master Communication Control Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
MCR	Multi-address bit	Number of arbitrations			Control bit				F8F6H	Undefined	W

Multi-address bit	Selects multiaddress or individual communication
0	Multiaddress communication (more than one slave)
1	Individual communication (one slave)

Number of arbitrations			Number of retries in case master loses in arbitration during master communication or self check communication
0	0	0	0 time (no retry)
0	0	1	1 time
0	1	0	2 times
0	1	1	3 times
1	0	0	4 times
1	0	1	5 times
1	1	0	6 times
1	1	1	7 times

Control bit				Control operation
0	0	0	0	Slave status read
0	0	1	1	Data read and lock
0	1	0	0	Lock address read (lower 8 bits)
0	1	0	1	Lock address read (higher 4 bits)
0	1	1	0	Slave status read and unlock
0	1	1	1	Data read
1	0	1	0	Command write and lock
1	0	1	1	Data write and lock
1	1	1	0	Command write
1	1	1	1	Data write
Other than above				Setting prohibited

Caution: Write MCR when MARQ bit of status register 2 (STR2) is 0.

(6) Status register 1 (STR1)

This register indicates the statuses of the transmit buffer register (TBF) and receive buffer register (RBF). The contents of this register are read with an 8-bit memory manipulation instruction, and are initialized to 01011xxxB when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Fig. 20-18 Status Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
STR1	TFL	TEP	RFL	REP	FCH	–	–	–	F8F0H	01011xxxB	R

TFL	Indicates whether TBF is full
0	TBF is not full
1	TBF is full

TEP	Indicates whether TBF is empty
0	TBF is not empty
1	TBF is empty

RFL	Indicates whether RBF is full
0	RBF is not full
1	RBF is full

REP	Indicates whether RBF is empty
0	RBF is not empty
1	RBF is empty

FCH	Indicates that number of communication frames set by FAI has been completely read Note
0	Number of communication frames set by FAI has not been read
1	Number of communication frames set by FAI has been completely read. This bit is automatically cleared to 0 when next frame information is read.

Note: The information of FCH is valid only when FCHC = 1 by the setting of the option function of CMR and FAI of CTR is set to 1.

Remark: FAI: bit 1 of control register (CTR)

(7) Status register 2 (STR2)

This register indicates the communication status, command execution status, and interrupt status. The contents of this register are read with an 8-bit memory manipulation instruction, and initialized to 02H when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Fig. 20-19 Status Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
STR2	0	MARQ	STRQ	SLRE	CEX	RAW	STM	IRQ	F8F1H	02H	R

MARQ	Indicates whether μPD78098 sub-series is requesting communication as master unit
0	Not requesting communication as master unit
1	Requesting communication as master unit

STRQ	Indicates whether μPD78098 sub-series is requesting communication as slave unit
0	Not requesting communication as slave unit
1	Requesting communication as slave unit

SLRE	Indicates whether slave reception or multiaddress reception is enabled or disabled Note
0	Disables slave reception or multiaddress reception
1	Enables slave reception or multiaddress reception

CEX	Indicates whether command is being processed
0	Command is not being processed (writing to CMR is enabled)
1	Command is being processed (writing to CMR is disabled)

RAW	Indicates whether IEBus interface hangs up
0	IEBus interface does not hang up
1	IEBus interface hangs up

STM	Indicates whether IEBus controller is in standby status
0	Not standby status
1	Standby status

IRQ	Indicates change of RCR and hang-up of IEBus controller
0	Change of RCR and hang-up of IEBus controller not detected
1	Change of RCR and hang-up of IEBus controller detected

Note: Unless SLRE is set to 1 by the end of control field of a reception field, the communication field cannot be received.

(8) Receive data number registers 1 and 2 (RDR1 and RDR2)

These registers store the number of receive data when reception is performed.

RDR1 stores the number of data received when the μPD78098 sub-series operates as the master. RDR2 receives the number of data when the microcomputer operates as a slave (individual/multiaddress communication).

The contents of both the registers can be read with 8-bit manipulation instructions, and are initialized to 00H when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

(9) Return code register (RCR)

This register stores a return code for transmission/reception.

Its contents can be read with an 8-bit memory manipulation instruction, and is initialized to 4FH when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Fig. 20-20 Return Code Register Format (1/3)

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
RCR	0	TRC2	TRC1	TRC0	REC3	REC2	REC1	REC0	F8F8H	4FH	R

TRC2	TRC1	TRC0	Meanings of return codes (related to transmission)
0	0	0	Indicates start of master transmission or slave data transmission. ① When master transmission is started Indicates that master address field in communication frame has been transmitted, that microcomputer has won in arbitration as master unit, and that master transmission has been started ② When slave transmission is started Indicates that microcomputer has received from master unit control bit (0011B, 0111B) that requests data transmission, and that slave data transmission has been started
0	0	1	Indicates that TBF is empty during master or slave data transmission. Unless following data is set to TBF within following time limit, transmission is aborted: In mode 0: approx. 1570 μs Note In mode 1: approx. 390 μs Note In mode 2: approx. 280 μs Note
0	1	0	Indicates that transmission of number of data specified by telegraph length field has been completed when master or slave data transmission is performed
0	1	1	Indicates that communication has ended without transmission of number of data specified by telegraph length field completed in one communication frame when master or slave data transmission is performed
1	0	0	Undefined
1	0	1	Indicates that microcomputer did not win arbitration and that communication ends without self check communication performed
1	1	0	Indicates that result of self check communication is normal
1	1	1	Indicates that result of self check communication is abnormal

(Cont'd)

Note: fs = 6.0 MHz

Remark: fs: bus controller system clock frequency

Fig. 20-20 Return Code Register Format (2/3)

REC3	REC2	REC1	REC0	Meanings of return codes (related to reception)
0	0	0	0	Indicates that master reception has been started. Three bytes of information including slave (transmitter unit) address, control bit, and telegraph length (number of data to be received) can be read from receive buffer (RBF)
0	0	0	1	Indicates that receive buffer (RBF) for master reception is full. Unless data of RBF is read within following time limit, data cannot be received and NAK is returned: In mode 0: approx. 1570 μs Note In mode 1: approx. 390 μs Note In mode 2: approx. 280 μs Note
0	0	1	0	Indicates that data of telegraph length has been received and that communication has been completed normally as result of master reception. Receive data can be read from receive buffer (RBF). Number of master receive data can be read from receive data number register 1 (RDR1).
0	0	1	1	Indicates that number of data of telegraph length cannot be received and that communication has been aborted as result of master reception. Receive data can be read from receive buffer (RBF). Number of master receive data can be read from receive data number register 1 (RDR1).
0	1	0	0	Indicates that slave individual reception has been started. Three bytes of information including slave (transmitter unit) address, control bit, and telegraph length (number of data to be received) can be read from receive buffer (RBF).
0	1	0	1	Indicates that receive buffer (RBF) for slave individual reception is full. Unless data of RBF is read within following time limit, data cannot be received and NAK is returned: In mode 0: approx. 1570 μs Note In mode 1: approx. 390 μs Note In mode 2: approx. 280 μs Note
0	1	1	0	Indicates that data of telegraph length has been received and that communication has been completed normally as result of slave individual reception. Receive data can be read from receive buffer (RBF). Number of slave receive data can be read from receive data number register 2 (RDR2).
0	1	1	1	Indicates that number of data of telegraph length cannot be received and that communication has been aborted as result of slave individual reception. Receive data can be read from receive buffer (RBF). Number of slave receive data can be read from receive data number register 2 (RDR2).
1	0	0	0	Indicates that multiaddress reception has been started. Three bytes of information including slave (transmitter unit) address, control bit, and telegraph length (number of data to be received) can be read from receive buffer (RBF).

(Cont'd)

Note: fs = 6.0 MHz

Remark: fs: bus controller system clock frequency

Fig. 20-20 Return Code Register Format (3/3)

REC3	REC2	REC1	REC0	Meanings of return codes (related to reception)
1	0	0	1	Indicates that receive buffer (RBF) for multiaddress reception is full. Unless data of RBF is read within following time limit, data cannot be received and NAK is returned: In mode 0: approx. 1570 μs Note1 In mode 1: approx. 390 μs Note1 In mode 2: approx. 280 μs Note1
1	0	1	0	Indicates that data of telegraph length has been received and that communication has been completed normally as result of multiaddress reception. Receive data can be read from receive buffer (RBF). Number of slave receive data can be read from receive data number register 2 (RDR2).
1	0	1	1	Indicates that number of data of telegraph length cannot be received and that communication has been aborted as result of multiaddress reception. Receive data can be read from receive buffer (RBF). Number of slave receive data can be read from receive data number register 2 (RDR2).
1	1	0	0	Indicates that error has occurred as result of multiaddress reception. Address of master that has performed multiaddress transmission can be checked by multiaddress registers (DAR1 and DAR2).Note2
1	1	0	1	Undefined
1	1	1	0	
1	1	1	1	

Notes:

(1) fs = 6.0 MHz

(2) This code is generated if the DERC bit of the command register (CMR) is set to 1 to set the option functions and if an error has occurred.

Remark: fs: bus controller system clock frequency

(10) Unit address registers 1 and 2 (UAR1 and UAR2)

These registers set a unit address and transmission conditions (transmission mode and slave transmission control).

Of the 12 bits of a unit address value, the higher 8 bits are set by UAR2 and the lower 4 bits are set by UAR1. The transmission conditions are set by UAR1.

The contents of these registers is written to these registers with 8-bit memory manipulation instructions.

The values of both the registers become undefined when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Caution: Be sure to set the unit address after resetting the IEBus controller via software (SRST ← 1).

Fig. 20-21 Formats of Unit Address Registers 1 and 2

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
UAR2	Unit address (higher 8 bits)								F8F3H	Undefined	W
	7	6	5	4	3	2	1	0			
UAR1	Unit address (lower 4 bits)				UAR13	UAR12	0	UAR10	F8F2H	Undefined	W

UAR10	Controls slave transmission
0	Disables slave transmission
1	Enables slave transmission

UAR13	UAR12	Selects transmission mode
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Setting prohibited

(11) Slave address registers 1 and 2 (SAR1 and SAR2)

These registers set a slave address (address of the unit to communicate).

Of the 12 bits of the slave address value, the higher 8 bits are set by SAR2 and the lower 4 bits are set by UAR1.

The contents of these registers can be written to these registers with 8-bit memory manipulation instructions.

The values of both the registers become undefined when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Fig. 20-22 Formats of Slave Address Registers 1 and 2

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
SAR2	Slave address (higher 8 bits)								F8F5H	Undefined	W
	7	6	5	4	3	2	1	0			
SAR1	Slave address (lower 4 bits)				0	0	0	0	F8F4H	Undefined	W

Cautions:

(1) Be sure to clear bits 3 to 0 to 0.

(2) Write data to SAR1 and SAR2 when the MARQ bit of status register 2 (STR2) is 0.

(12) Multiaddress calling destination address registers 1 and 2 (DAR1 and DAR2)

These registers store a multiaddress calling destination address (address of the master that performs multiaddress transmission) if a multiaddress reception error has occurred.

The contents of these registers can be read with 8-bit memory manipulation instructions.

The values of both the registers become undefined when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Cautions:

(1) DAR1 and DAR2 are valid only when the DERC bit of the command register (CMR) is set to 1 to set the option functions.

(2) The contents of DAR1 and DAR2 are updated each time a multiaddress error has occurred. Therefore, unless the contents of DAR1 and DAR2 are read within the following time limit, their contents may be updated by a new error:

In mode 0: approx. 5420 μs (at fs = 6.0 MHz)

In mode 1: approx. 1490 μs (at fs = 6.0 MHz)

In mode 2: approx. 1110 μs (at fs = 6.0 MHz)

Remark: fs: bus controller system clock frequency

★ **Fig. 20-23 Formats of Multiaddress Calling Destination Address Registers 1 and 2**

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
DAR1	Multiaddress (lower 4 bits)				–	–	–	–	F8F6H	Undefined	R
	7	6	5	4	3	2	1	0			
DAR2	Multiaddress (higher 8 bits)								F8F7H	Undefined	R

(13) Lock address registers 1 and 2 (LOR1 and LOR2)

These registers store a lock address (the address of the master unit that has requested locking) when a slave unit is locked.

The contents of these registers can be read with 8-bit memory manipulation instructions.

The lock status is initialized to 0000B and the lock address becomes undefined when the $\overline{\text{RESET}}$ signal is input or when the IEBus controller is reset via software.

Caution: The contents of LOR1 and LOR2 can be read only when the CEX bit of status register 2 (STR2) is 0 after the LOCK bit of the command register (CMR) has been set to 1.

Fig. 20-24 Formats of Lock Address Registers 1 and 2

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
LOR1	Lock address (lower 8 bits)								F8F4H	Undefined	R
	7	6	5	4	3	2	1	0			
LOR2	Lock status				Lock address (higher 4 bits)				F8F5H	0xH	R

Higher 4 bits				Lock status
0	0	0	0	Not locked
0	0	0	1	Locked
Other than above				Undefined

20.4 Communication Operations of IEBus

Figs. 20-25 through 20-27 show the timing chart of communication using the IEBus.

Fig. 20-25 Operation Example of Individual Communication (master transmission - slave reception)

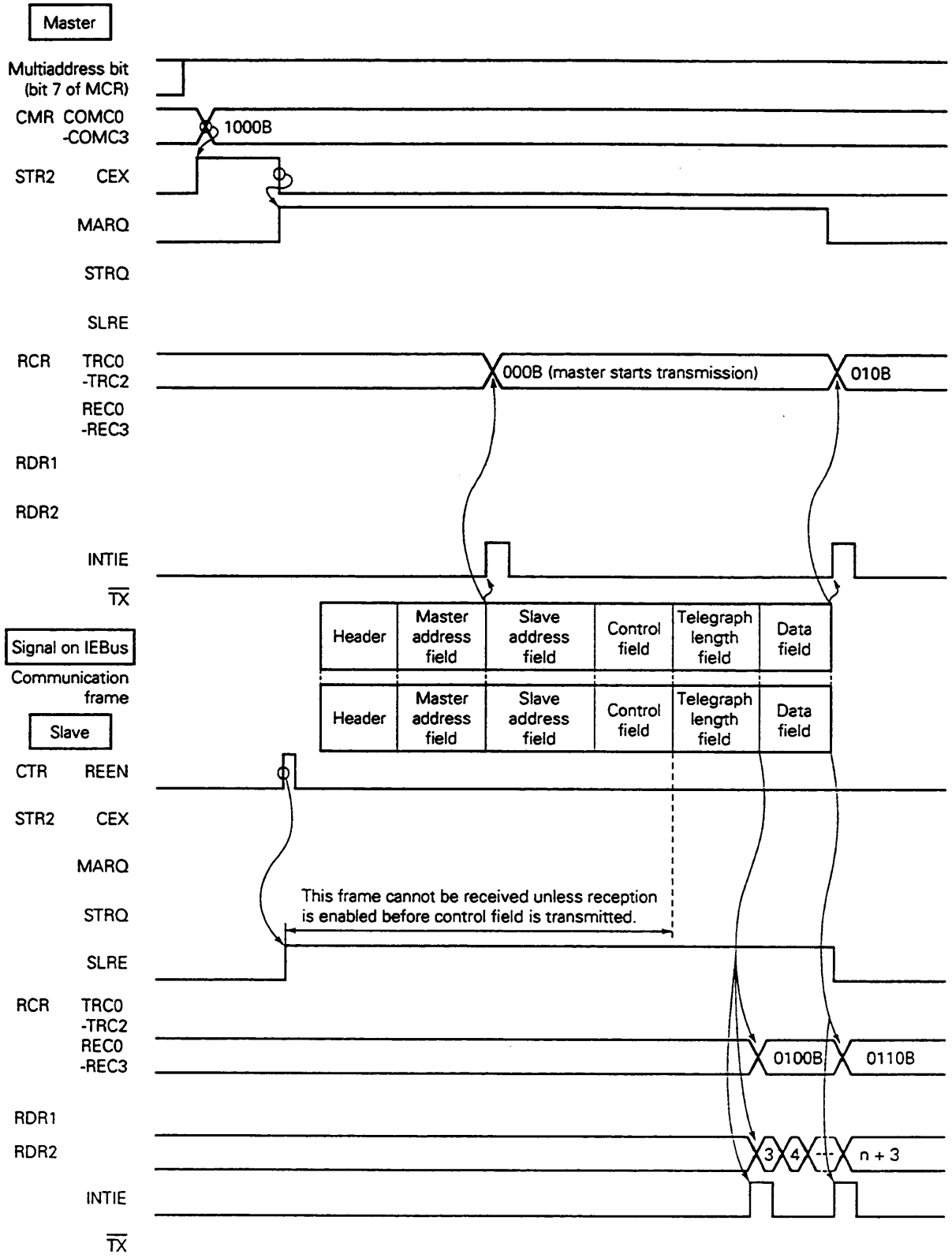


Fig. 20-26 Operation Example of Individual Communication (master reception - slave transmission)

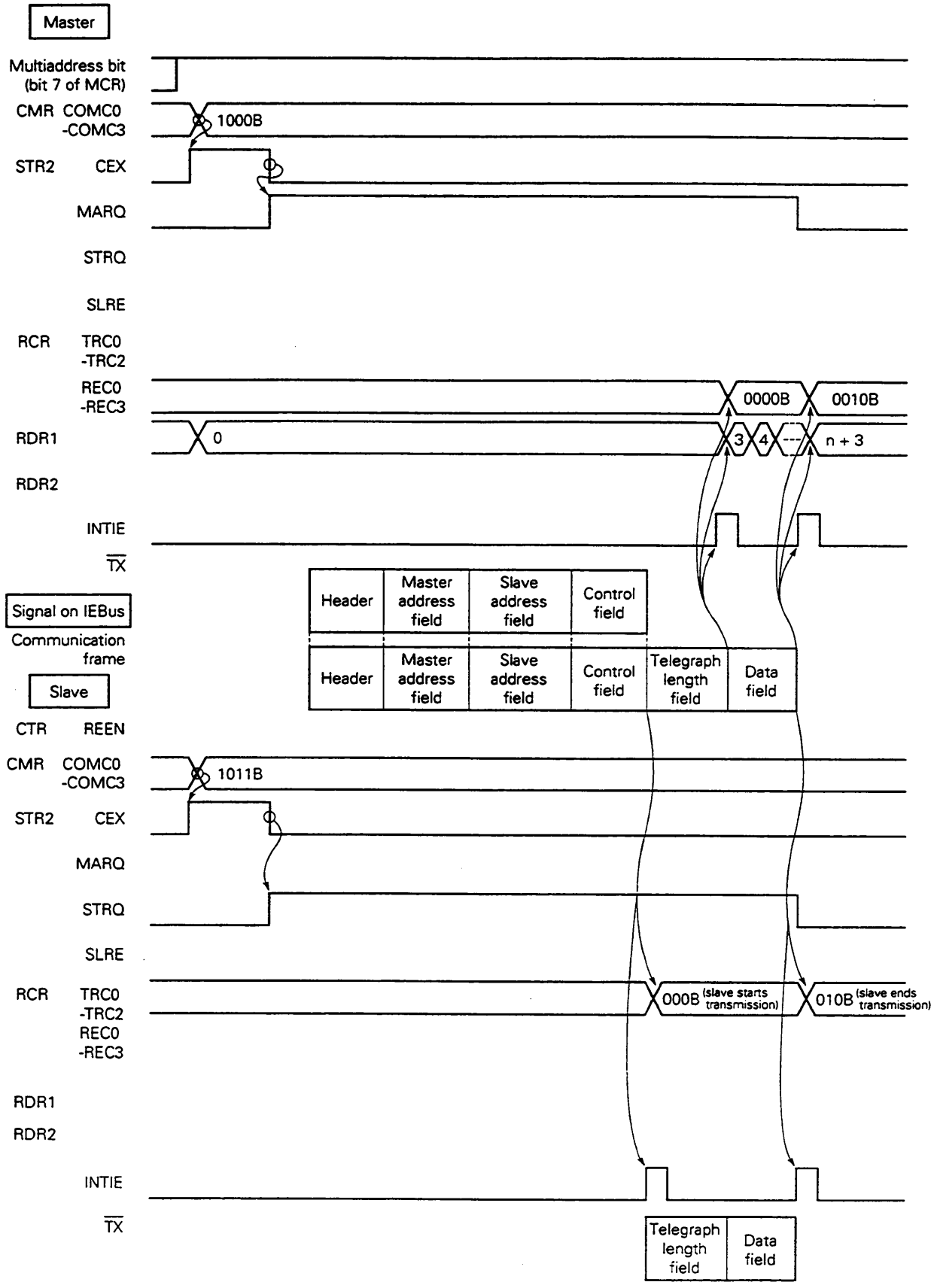
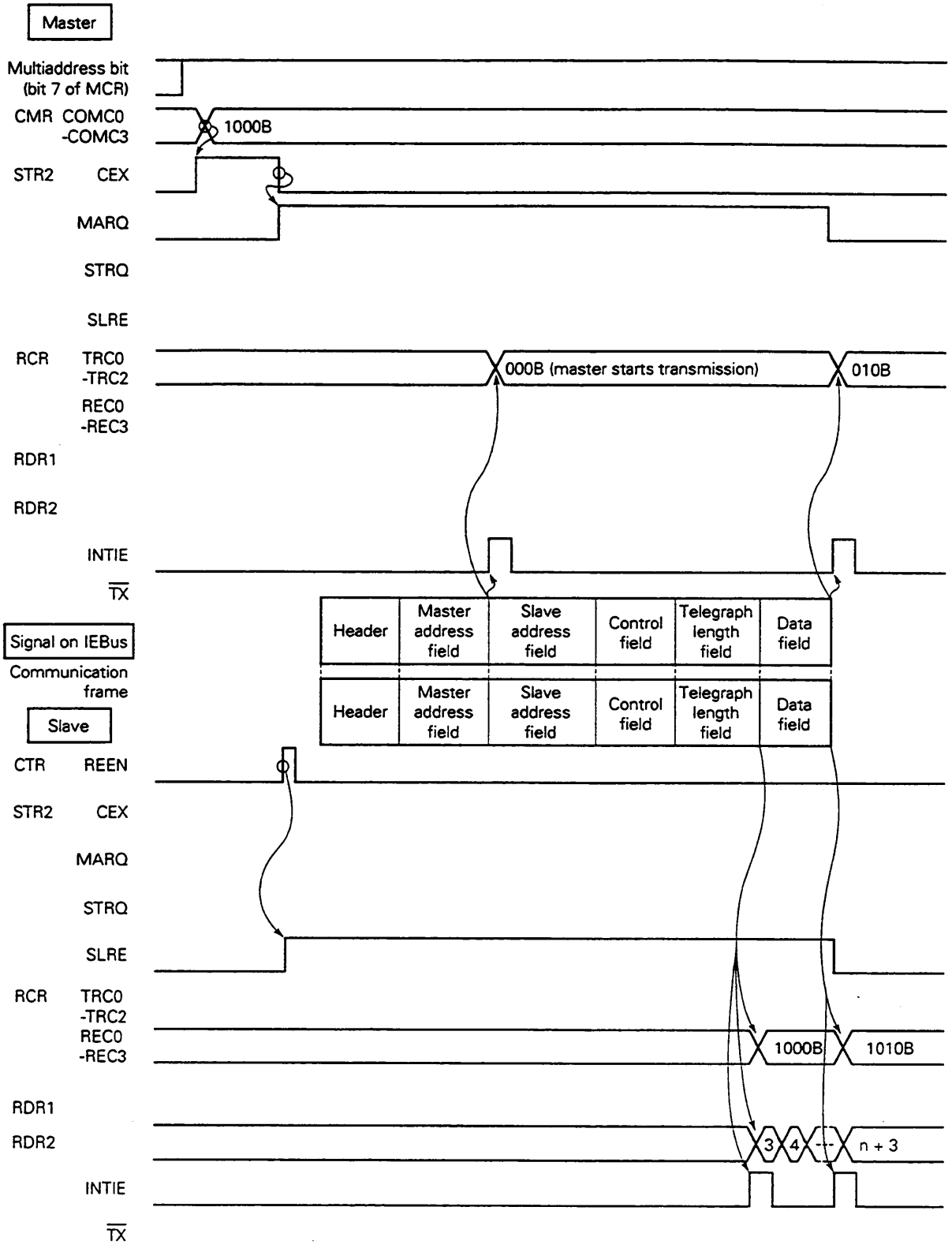


Fig. 20-27 Operation Example of Multiaddress Communication (master transmission - slave reception)



[MEMO]

CHAPTER 21 STANDBY FUNCTION

21.1 Standby Function and Configuration

21.1.1 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. System clock oscillator continues oscillation. 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.

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.

(3) To reduce the power dissipation of the IEBus controller, set bit 0 of CTR (STREQ), confirm that bit 2 of STR2 (STM) has been set, and then execute the HALT or STOP instruction.

21.1.2 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 manipulation instruction.

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

Fig. 21-1 Oscillation Stabilization Time Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

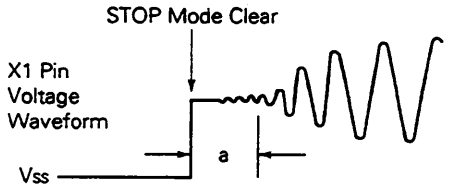
OSTS2	OSTS1	OSTS0	Selection of Oscillation Stabilization Time when STOP Mode is Released
0	0	0	2^{12}/fxx (1.02ms)
0	0	1	2^{14}/fxx (4.10ms)
0	1	0	2^{15}/fxx (8.19ms)
0	1	1	2^{16}/fxx (16.4ms)
1	0	0	2^{17}/fxx (32.8ms)
Other than above			Setting prohibited

Remarks:

(1) fxx: Main system clock frequency

(2) Values in parentheses apply to operating at fxx = 4.0 MHz

Caution: The wait time after STOP mode clear does not include the time (see "a" below) from STOP mode clear to clock oscillation start, regardless of clearance by $\overline{\text{RESET}}$ input or by interrupt generation.



21.2 Standby Function Operations

21.2.1 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 21-1 HALT Mode Operating Status

Clock Generator	Both main system clock and subsystem clock can be oscillated. Clock supply to the CPU stops.
CPU	Operation stop.
Port	Status before HALT mode setting is held.
16-bit timer/event counter	Operable when main system clock oscillation is used or when watch timer output with f_{XT} selected or TI00 is selected as count clock.
8-bit timer/event counter	Operable when main system clock oscillation is used or when TI1 or TI2 is selected as count clock.
Watch timer	Operable when main system clock oscillation is used or when f_{XT} is selected as count clock.
Watchdog timer	Operable when main system clock oscillation is used.
A/D converter	
D/A converter	Operation enabled.
Real-time output port	
Serial Interface	Operable when main system clock oscillation is used or when input clock from off-chip is selected as serial clock except automatic transmit/receive function. Operation of automatic transmit/receive function stops.
IEBus controller	Operable when main system clock is used. Operation can be started or stopped by bit 0 of CTR (STREQ)
External interrupt	INTP0 operation enabled when the clock ($f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$) for the peripheral hardware are selected as sampling clock. INTP1 to INTP6 are operable.

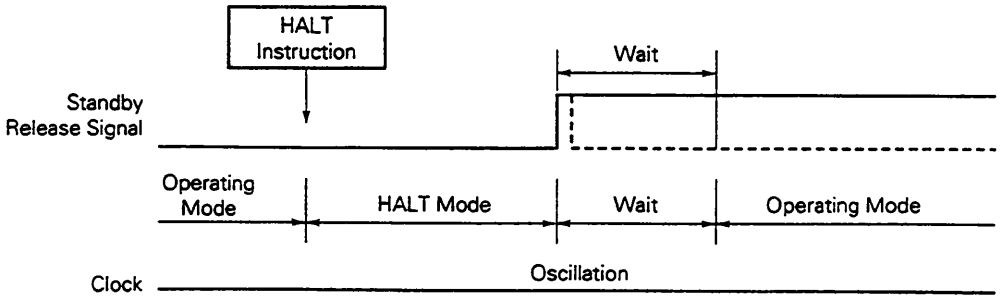
(2) HALT mode clear

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. 21-2 HALT Mode Clear upon Interrupt Generation



Remarks:

(1) The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

(2) Wait time will be as follows:

- When vectored interrupt service is carried out : 8 to 9 clocks
- When vectored interrupt service is not carried out : 2 to 3 clocks

(b) Clear upon non-maskable interrupt request

The HALT mode is cleared and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

(c) Clear upon 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 upon $\overline{\text{RESET}}$ input

As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Fig. 21-3 HALT Mode Release by $\overline{\text{RESET}}$ Input

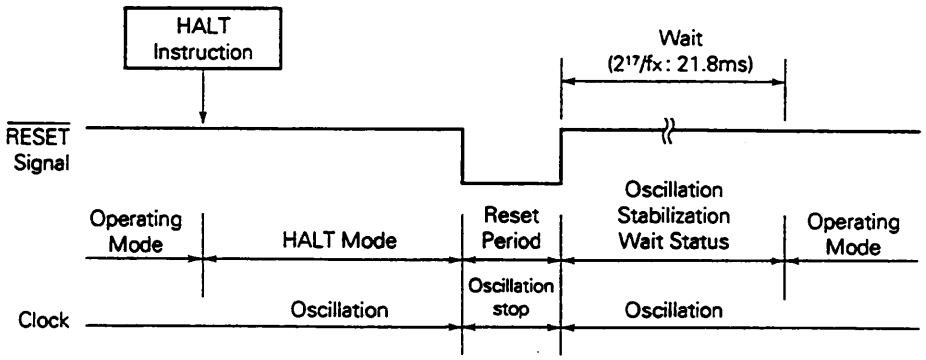


Table 21-2 Operation after HALT Mode Release

Release Source	MKxx	PRxx	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	–	–	×	×	Interrupt service execution
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	HALT mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark: ×: Don't care

21.2.2 Stop mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions:

(1) When the STOP mode is set, X1 input is internally short-circuited to Vss (ground potential) to suppress the leakage at the crystal oscillator.
Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

(2) Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operating mode is set.

The operating status in the STOP mode is described below.

Table 21-3 STOP Mode Operating Status

Clock Generator	Only main system clock stops oscillation.
CPU	Operation stop.
Port	Status before STOP mode setting is held.
16-bit timer/event counter	Operable when either watch timer output with f_{XT} selected is selected as count clock.
8-bit timer/event counter	Operation enabled only when TI1 and TI2 are selected for the count clock.
Watch timer	Operation enabled only when f_{XT} is selected for the count clock.
Watchdog timer	Operation stop.
A/D converter	
D/A converter	Operation enabled.
Real-time output port	Operable when external trigger is used or TI1 and TI2 are selected for the 8-bit timer/event counter count clock.
Serial interface	Operable only when external input clock is selected to serial clock except UART and automatic transmit/receive function. Operation of UART and automatic transmit/receive function stops.
IEBus controller	Stops operation, retaining internal data.
External interrupt	INTP0 operation disabled. INTP1 to INTP6 are operable.

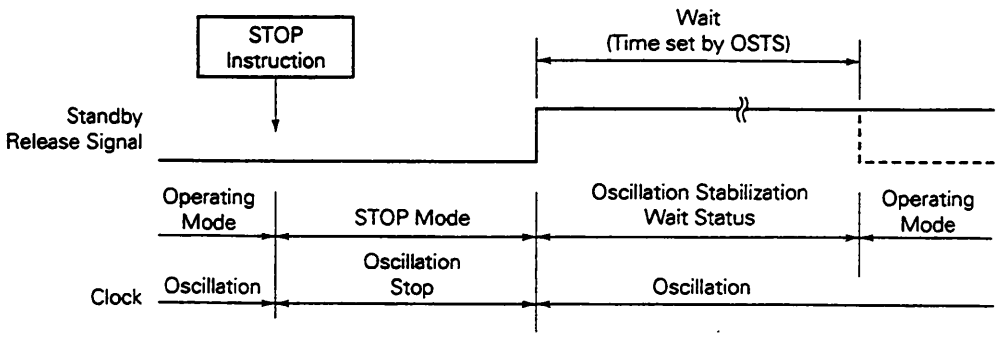
(2) STOP mode release

The STOP mode can be cleared with the following three types of sources.

(a) Release by unmasked interrupt request

An unmasked interrupt request is used to release 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. 21-4 STOP Mode Release by Interrupt Generation



Remark: The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

(b) Release by unmasked test input

The STOP mode is cleared by unmasked test input.

After the lapse of oscillation stabilization time, the instruction at the next address of the STOP instruction is executed.

(c) Release by $\overline{\text{RESET}}$ input

The STOP mode is cleared and after the lapse of oscillation stabilization time, reset operation is carried out.

Fig. 21-5 Release by STOP Mode $\overline{\text{RESET}}$ Input

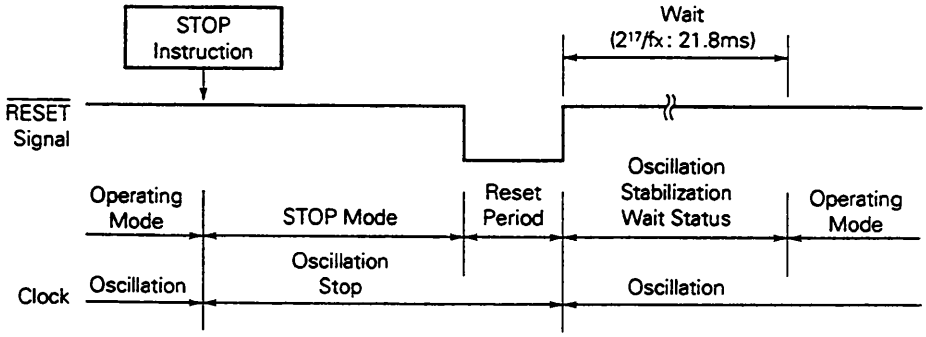


Table 21-4 Operation after STOP Mode Release

Release Source	MKxx	PRxx	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark: ×: Don't care

CHAPTER 22 RESET FUNCTION

22.1 Reset Function

The following two operations are available to generate the reset function.

(1) External reset input with $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer overrun time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status as shown in Table 22-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ 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. 22-2** to **22-4**).

Cautions:

(1) For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
(2) During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.
(3) When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Fig. 22-1 Block Diagram of Reset Function

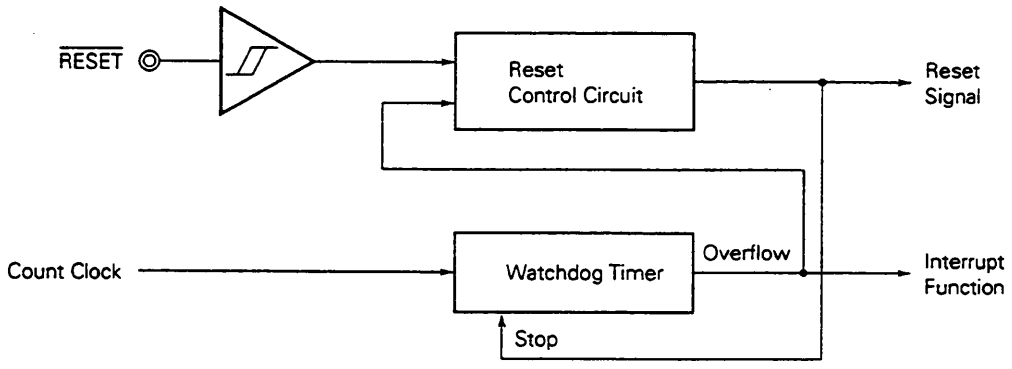


Fig. 22-2 Timing of Reset Input by $\overline{\text{RESET}}$ Input

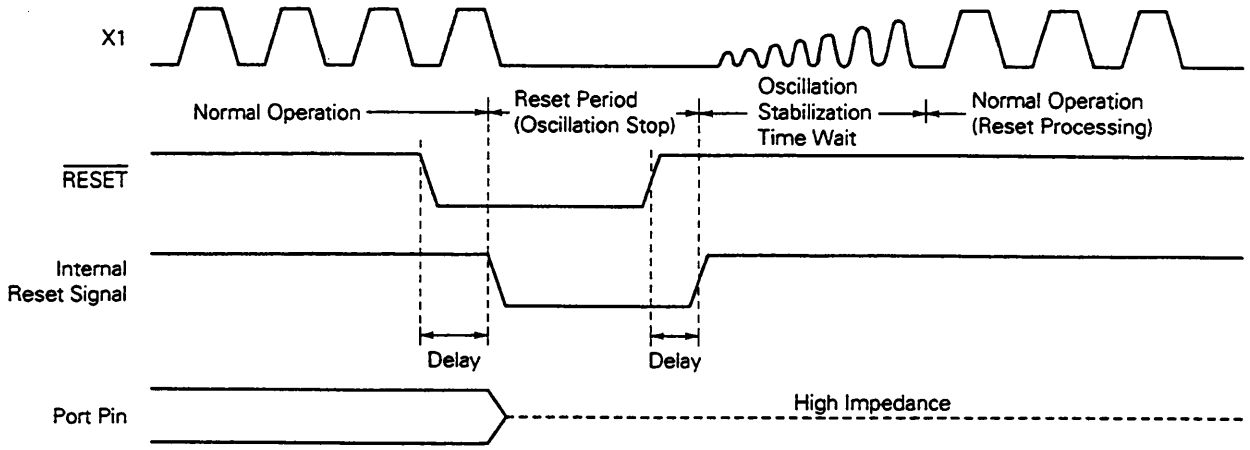


Fig. 22-3 Timing of Reset due to Watchdog Timer Overflow

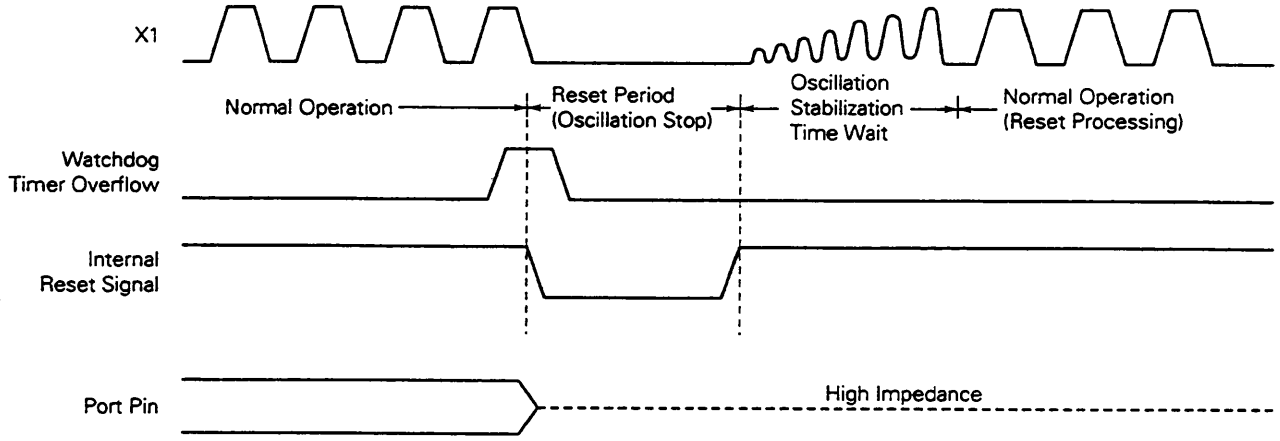


Fig. 22-4 Timing of Reset Input in STOP Mode by $\overline{\text{RESET}}$ Input

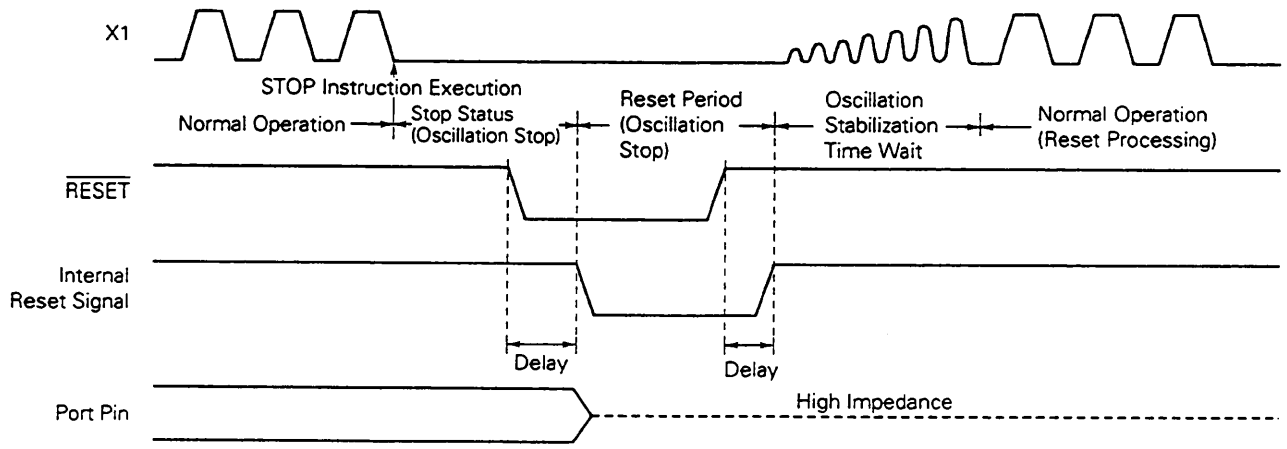


Table 22-1 Hardware Status after Reset (1/3)

Hardware		Status after Reset
Program counter (PC) Note1		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined Note2
	General register	Undefined Note2
Port (Output latch)	Ports 0 to 3, Ports 7, 12, 13 (P0 to P3, P7, P12, P13)	00H
	Port 4 to Port 6 (P4 to P6)	Undefined
Port mode register (PM0 to PM3, PM5 to PM7, PM12, PM13)		FFH
Pull-up resistor option register (PUOH, PUOL)		00H
Processor clock control register (PCC)		04H
Oscillation mode selection register (OSMS)		00H
Clock select registers 1, 2 (IECL1, IECL2)		00H
Memory size switching register (IMS)		**Note 3**
Internal expansion RAM size switching register (IXS, only μPD78098A and 78P098A)		08H ★
Memory expansion mode register (MM)		10H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	00H
	Capture/compare register (CR00, CR01)	Undefined
	Clock selection register (TCL0)	00H
	Mode control register (TMC0)	00H
	Capture/compare control register 0 (CRC0)	04H
	Output control register (TOC0)	00H
8-bit timer/event counter	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

Notes:

(1) During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

(2) The post-reset status is held in the standby mode.

(3) The values after reset depend on products.
μPD78094: C8H, μPD78095: CAH, μPD78096: CCH, μPD78098A: CFH, μPD78P098A: CFH

Table 22-1 Hardware Status after Reset (2/3)

Hardware		Status after Reset
Watch timer Watchdog timer	Clock select register (TCL2)	00H
	Mode register (WDTM)	00H
Serial interface	Clock select register (TCL3)	88H
	Shift registers (SIO0, SIO1)	Undefined
	Mode registers (CSIM0, CSIM1, CSIM2)	00H
	Serial bus interface control register (SBIC)	00H
	Slave address register (SVA)	Undefined
	Automatic data transmit/receive control register (ADTC)	00H
	Automatic data transmit/receive address pointer (ADTP)	00H
	Automatic data transmit/receive interval specify register (ADTI)	00H
	Asynchronous serial interface mode register (ASIM)	00H
	Asynchronous serial interface status register (ASIS)	00H
	Baud rate generator control register (BRGC)	00H
	Transmit shift register (TXS) Receive buffer register (RXB)	FFH
	Interrupt timing specify register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
	A/D current cutting select register (IEAD)	00H
D/A converter	Mode register (DAM)	00H
	Conversion value setting register (DACS0, DACS1)	00H
Real-time output port	Mode register (RTPM)	00H
	Control register (RTPC)	00H
	Buffer register (RTBL, RTBH)	00H

Table 22-1 Hardware Status after Reset (3/3)

Hardware		Status after Reset
IEBus controller	IEBus controller mode register (IECM)	00H
	Control register (CTR)	xxx00x01B
	Status register 1 (STR1)	01011xxxB
	Command register (CMR)	00H
	Status register 2 (STR2)	02H
	Unit address register 1 (UAR1)	Undefined
	Receive data number register 1 (RDR1)	00H
	Unit address register 2 (UAR2)	Undefined
	Receive data number register 2 (RDR2)	00H
	Slave address register 1 (SAR1)	Undefined
	Lock address register 1 (LOR1)	Undefined
	Slave address register 2 (SAR2)	Undefined
	Lock address register 2 (LOR2)	0000xxxxB
	Master communication control register (MCR)	Undefined
	Multiaddress register 1 (DAR1)	Undefined
	Multiaddress register 2 (DAR2)	Undefined
	Return code register (RCR)	4FH
	Transmit buffer register (TBF)	Undefined
	Receive buffer register (RBF)	Undefined
Interrupt	Request flag register (IF0L, IF0H, IF1L)	00H
	Mask flag register (MK0L, MK0H, MK1L)	FFH
	Priority specify flag register (PR0L, PR0H, PR1L)	FFH
	External interrupt mode register (INTM0, INTM1)	00H
	Key return mode register (KRM)	02H
	Sampling clock select register (SCS)	00H

[MEMO]

CHAPTER 23 μPD78P098A ★

The μPD78P098A is a product which incorporates a one-time programmable PROM or an EPROM enabled for program write, erase and rewrite. Table 23-1 lists differences between μPD78P098A and mask ROM version.

Table 23-1 Difference between the μPD78P098A and Mask ROM Version

Item	μPD78P098A	Mask ROM Version
IC pin	None	Available
V_{PP} pin	Available	None
On-chip mask option pull-up resistors for P60 to P63 pins	None	Available

23.1 Memory Size Switching Register

The μPD78P098A can select the internal memory with the memory size switching register (IMS). The same memory mapping as that of the μPD78094, 78095, 78096 and 78098A with a different internal memory is possible by setting IMS.

To make the memory map of the μPD78P098A same as that of the mask ROM version, set the values of the mask ROM version at reset to IMS.

With the mask ROM version, however, IMS needs not to be set.

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

The value of IMS is as shown in Table 23-2 at $\overline{\text{RESET}}$.

Caution: To use a mask ROM version, do not set a value other than those shown in Table 23-2 to IMS.

Fig. 23-1 Memory Size Switching Register Format

Symbol	7	6	5	4	3	2	1	0	Address	When Reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	**Note 1**	W

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
1	0	0	0	32K bytes
1	0	1	0	40K bytes
1	1	0	0	48K bytes
1	1	1	0	56K bytes Note 2
1	1	1	1	60K bytes
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
1	1	0	1024 bytes
Other than above			Setting prohibited

Notes:

(1) The value of the memory size select register at reset differs depending on the model as shown in the table below.

Table 23-2 Value of Memory Size Switching Register when Reset

Part Number	Value at Reset
μPD78094	C8H
μPD78095	CAH
μPD78096	CCH
μPD78098A	CFH
μPD78P098A	

(2) When using the external device expansion function for the μPD78098A and 78P098A, set the internal PROM capacity to below 56K bytes.

Caution: When using a mask ROM version do not set values other than reset to IMS (excluding when using the external device expansion function for the μPD78098A).

Remark: Set IMS according to the internal ROM capacity.

23.2 Internal Expansion RAM Size Switching Register ★

The μPD78P098A can select the internal expansion RAM with the internal expansion RAM size switching register (IXS). The same memory mapping as that of a mask ROM version with a different internal expansion RAM is possible by setting IXS.

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

It is set to 08H by the $\overline{\text{RESET}}$ input.

Cautions:

(1) The internal expansion RAM size switching register is incorporated only in the μPD78098A and 78P098A.

(2) To use a mask ROM version (μPD78094, 78095, 78096, 78098A), do not set values other than those shown in Table 23-3 to IXS.

Fig. 23-2 Internal Expansion RAM Size Switching Register Format

Symbol	7	6	5	4	3	2	1	0	Address	when Reset	R/M
IXS	0	0	0	0	IX RAM3	IX RAM2	IX RAM1	IX RAM0	FFF4H	08H	W

IX RAM3	IX RAM2	IX RAM1	IX RAM0	Internal Expansion RAM Capacity Selection
1	1	0	0	No internal expansion RAM
1	0	0	0	2048bytes (F000H-F7FFH)
Other than above				Setting prohibited

Table 23-3 shows the IXS values for setting the same memory map as the mask ROM version.

Table 23-3 Internal Expansion RAM Size Switching Register Setting Values

Corresponding Mask ROM Version	IXS Setting Values
μPD78094	0CH[Note]
μPD78095	
μPD78096	
μPD78098A	08H

Note: Executing program created for μPD78P098A, in which "MOV IXS, #0CH" is described, on μPD78094, 78095, 78096 does not affect operations.

23.3 PROM Programming

The μPD78P098A incorporates a 60K-byte PROM as program memory. When programming the μPD78P098A, the PROM programming mode is set by means of the V_{PP} pin and the $\overline{RESET}$ pin. For the connection of unused pins, see **1.5 Pin Configuration (Top View), (2) PROM programming mode.**

★ **Caution: Perform program writing in the 0000H to EFFFH addresses (specify the last address EFFFH). No writing can be performed using PROM programmers with which the writing address cannot be specified.**

23.3.1 Operating modes

When +5 V or +12.5 V is applied to the V_{PP} pin and a low-level signal is applied to the $\overline{RESET}$ pin, the μPD78P098A is set to the PROM programming mode. This is one of the operating modes shown in Table 23-4 below according to the setting of the $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins.

The PROM contents can be read by setting the read mode.

Table 23-4 PROM Programming Operating Modes

Operating mode \ Pin	$\overline{RESET}$	V_{PP}	V_{DD}	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	D0 to 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 disabled				L	H	×	High impedance
Standby				H	×	×	High impedance

Remark: ×: L or H

(1) Read mode

Read mode is set by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

(2) Output disable mode

If $\overline{OE}$ is set to H, data output becomes high impedance and the output disable mode is set.

Therefore, if multiple μPD78P098As are connected to the data bus, data can be read from any one device by controlling the $\overline{OE}$ pin.

(3) Standby mode

Setting $\overline{CE}$ to H sets the standby mode.

In this mode, data output becomes high impedance irrespective of the status of $\overline{OE}$.

(4) Page data latch mode

Setting $\overline{CE}$ to H, $\overline{PGM}$ to H, and $\overline{OE}$ to L at the start of the page write mode sets the page data latch mode. In this mode, 1-page 4-byte data is latched in the internal address/data latch circuit.

(5) Page write mode

After a 1-page 4-byte address and data are latched by the page data latch mode, a page write is executed by applying a 0.1 ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$ = H and $\overline{OE}$ = H. After this, program verification can be performed by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times ($X \leq 10$).

(6) Byte write mode

A byte write is executed by applying a 0.1 ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$ = L and $\overline{OE}$ = H. After this, program verification can be performed by setting $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times ($X \leq 10$).

(7) Program verify mode

Setting $\overline{CE}$ to L, $\overline{PGM}$ to H, and $\overline{OE}$ to L sets the program verify mode.

After writing is performed, this mode should be used to check whether the data was written correctly.

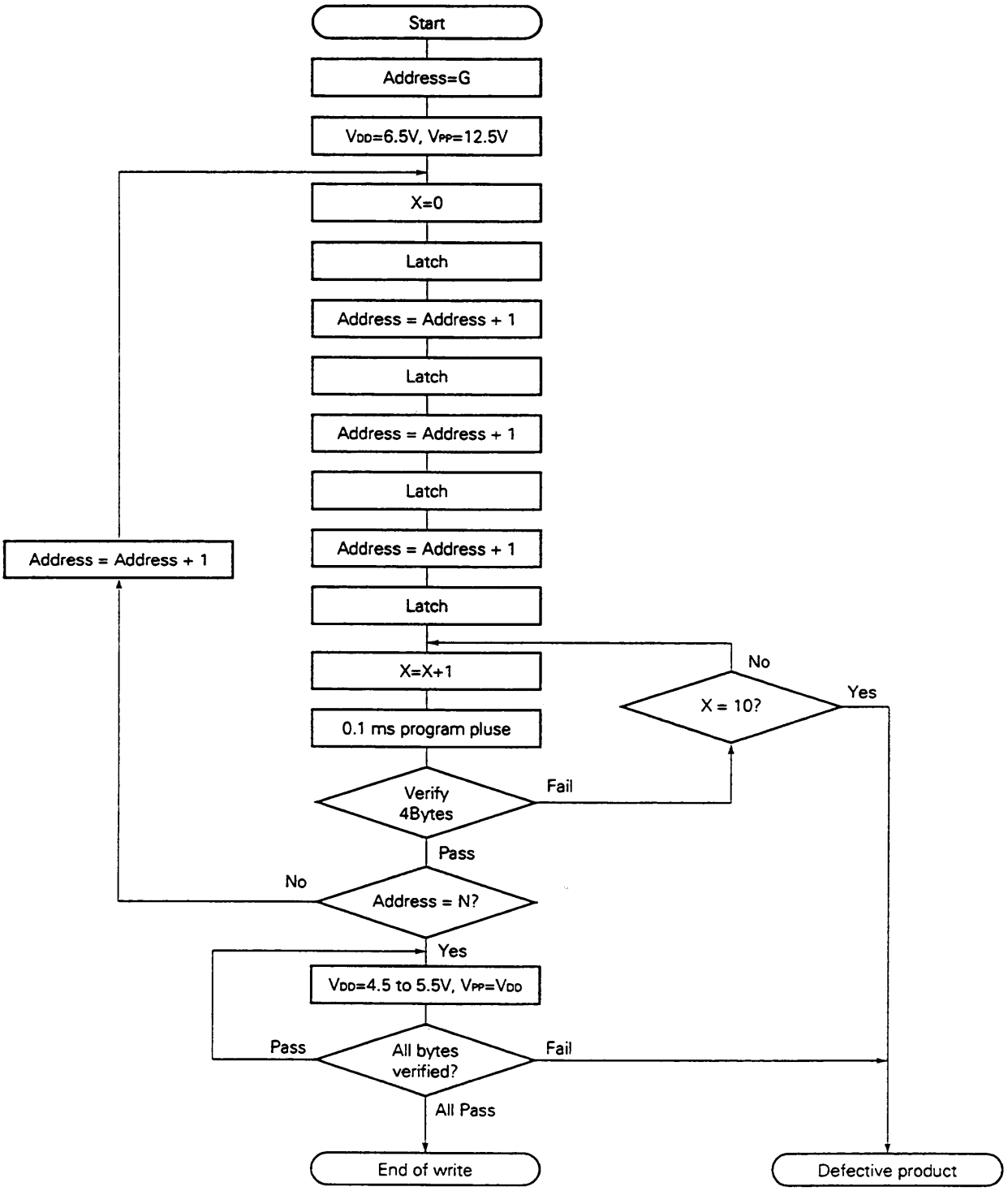
(8) Program inhibit mode

The program inhibit mode is used when the $\overline{OE}$ pins, V_{PP} pins and pins D0 to D7 of multiple μPD78P098As are connected in parallel, and when you wish to write to one of these devices.

The page write mode or byte write mode described above is used to perform a write. At this time, the write is not performed on the device which has the $\overline{PGM}$ pin driven high.

23.3.2 PROM write procedure

Fig. 23-3 Page Program Mode Flowchart



Remarks:

(1) G = Start address

(2) N = Last address of program

Fig. 23-4 Page Program Mode Timing

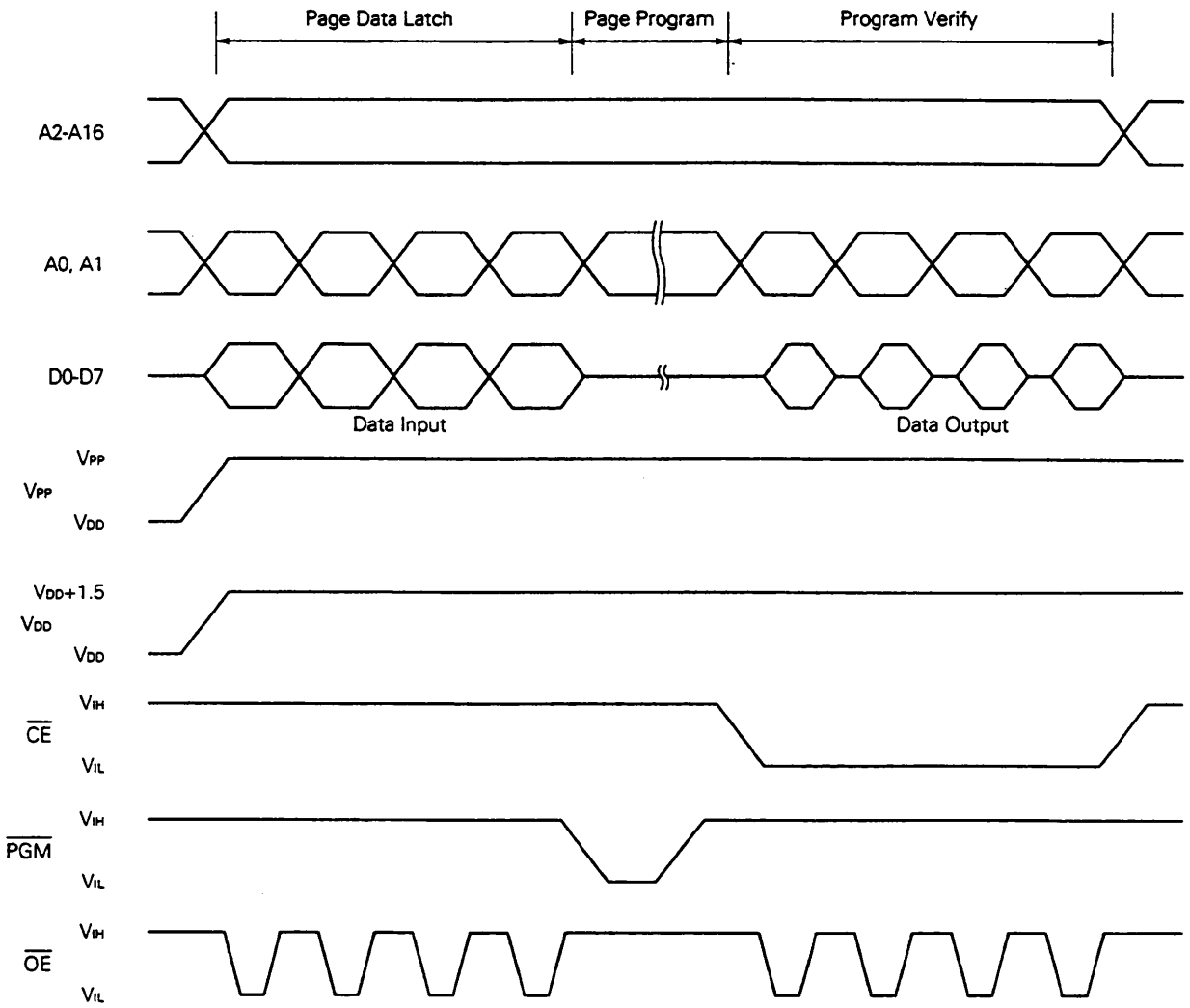
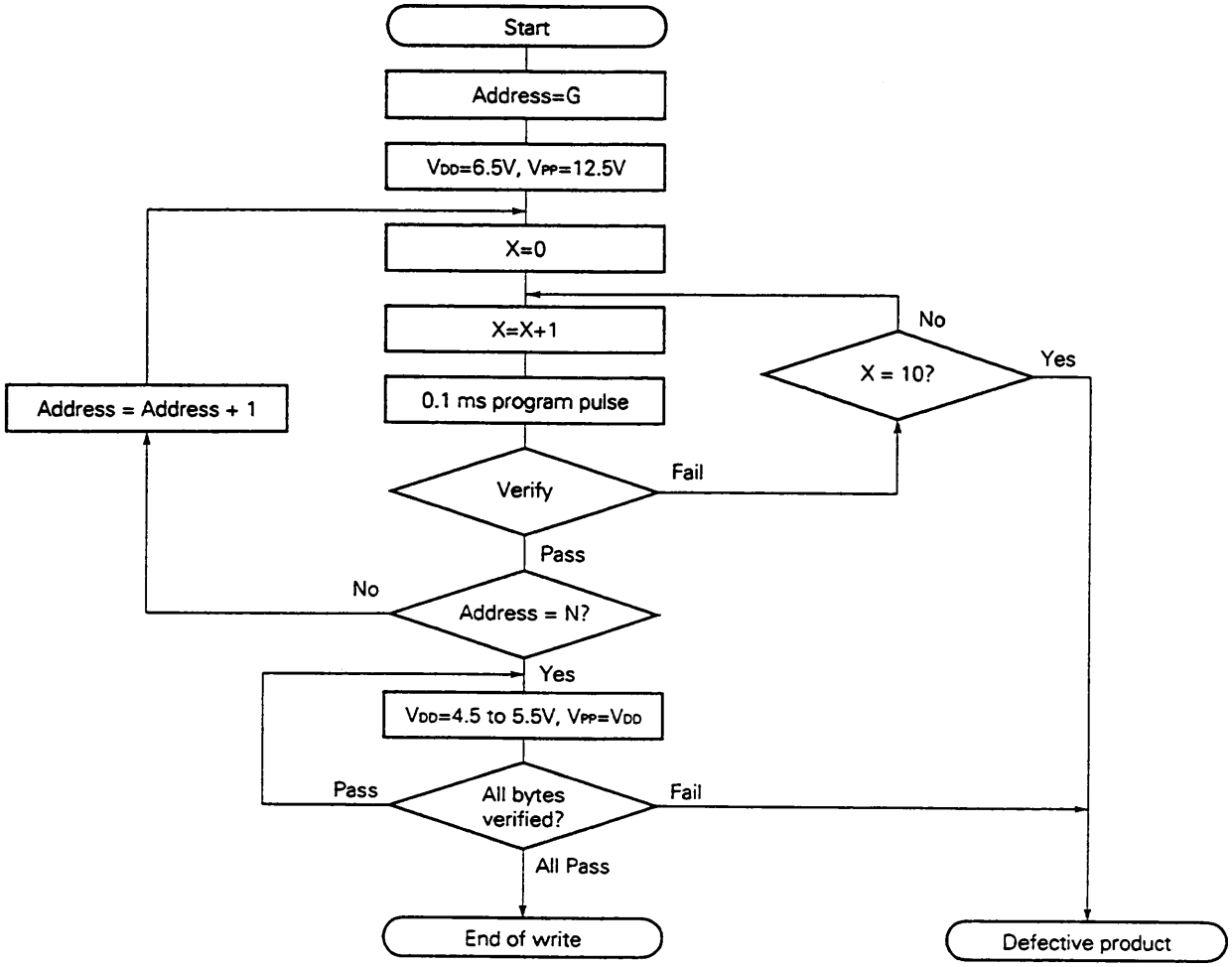


Fig. 23-5 Byte Program Mode Flowchart

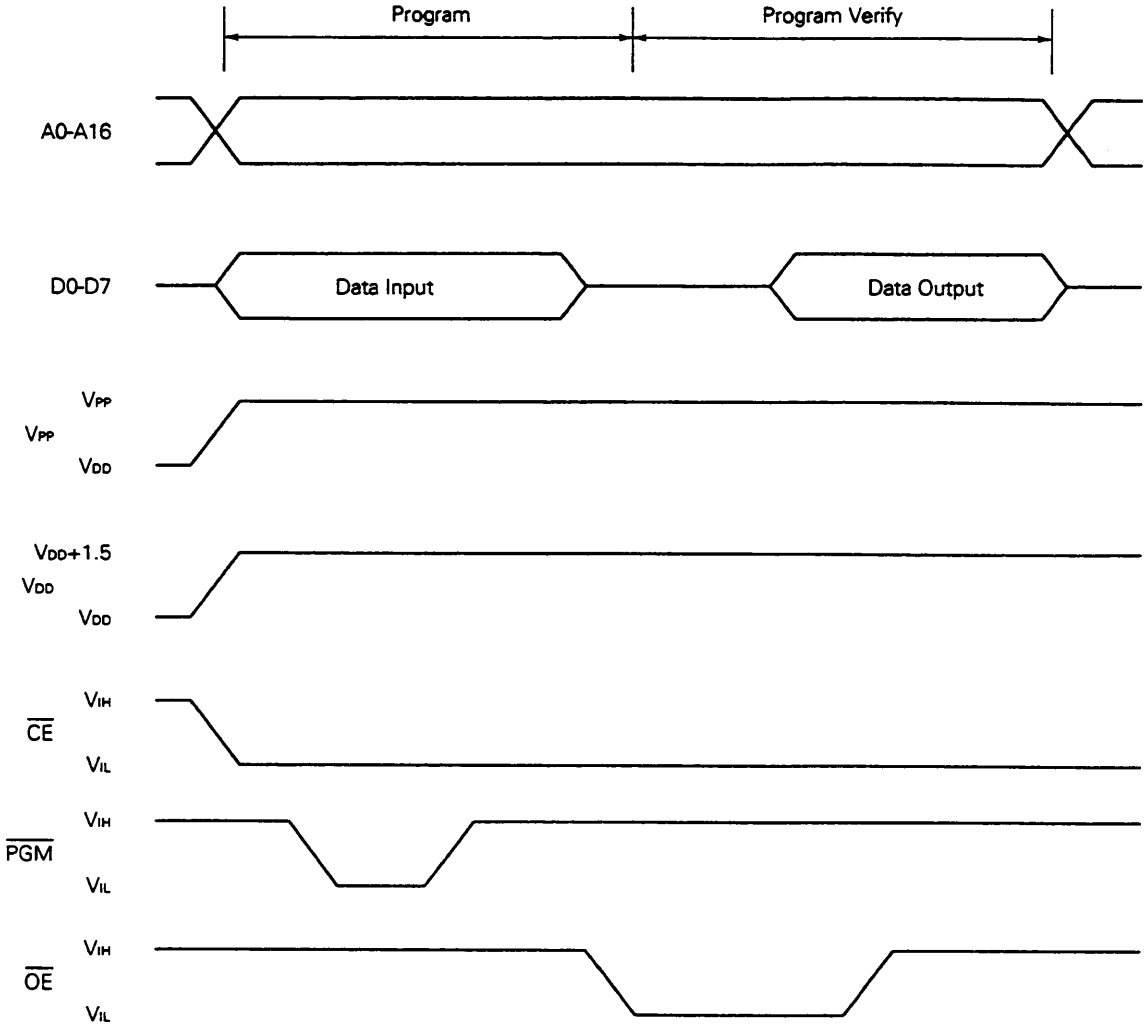


Remarks:

(1) G = Start address

(2) N = Last address of program

Fig. 23-6 Byte Program Mode Timing



Cautions:

(1) Ensure that V_{DD} is applied before V_{PP} and cut after V_{PP}.

(2) Ensure that V_{PP} does not exceed +13.5 V including overshoot.

(3) Removing the device while +12.5 V is being applied to V_{PP} may have an adverse affect on reliability.

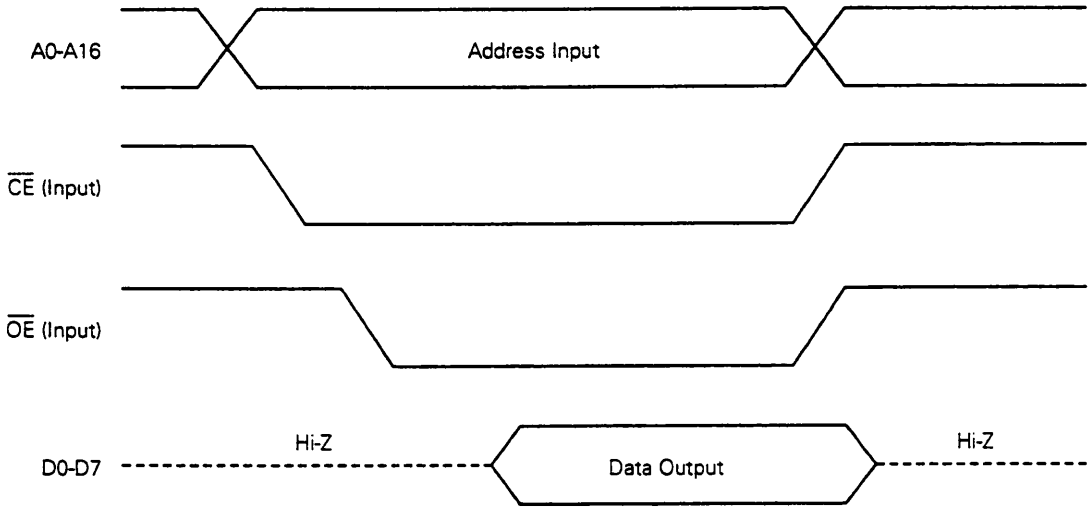
23.3.3 PROM read procedure

PROM contents can be read onto the external data bus (D0 to D7) using the following procedure.

(1) Fix the $\overline{\text{RESET}}$ pin low, and supply +5 V to the V_{PP} pin. Unused pins are handled 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 address of data to be read to pins A0 through A16.
(4) Read mode.
(5) Output data to pins D0 through D7.

The timing for steps (2) through (5) above is shown in Fig. 23-7.

Fig. 23-7 PROM Read Timing



23.4 Screening of One-Time PROM Versions

Because of their construction, one-time PROM versions (μPD78P098AGC-3B9) cannot be fully tested by NEC before shipment. After the necessary data has been written, it is recommended that screening is implemented in which PROM verification is performed after high-temperature storage under the following conditions.

Storage Temperature	Storage Time
125 °C	24 hours

CHAPTER 24 INSTRUCTION SET

The μPD78098 sub-series instruction sets are listed in the following table. For details of the functions of each instruction and machine language (instruction code), refer to **78K/0 SERIES USER'S MANUAL - INSTRUCTION (IEU-1372)**.

24.1 Legend

24.1.1 Operand identifiers and description methods

Operands are described in "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and 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 24-1 Operand Identifiers and Description Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol[Note]
sfrp	Special function register symbols (16-bit manipulatable register even addresses only)[Note]
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even addresses only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note: FFD0H to FFDFH are not addressable.

Remark: For special-function register symbols, refer to **Table 3-2 Special Function Register List**.

24.1.2 Description of "operation" column

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
AX	:	AX register pair; 16-bit accumulator
BC	:	BC register pair
DE	:	DE register pair
HL	:	HL register pair
PC	:	Program counter
SP	:	Stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
RBS	:	Register bank select flag
IE	:	Interrupt request enable flag
NMIS	:	Non-maskable interrupt servicing flag
( )	:	Memory contents indicated by address or register contents in parentheses
X_H, X_L	:	Higher 8 bits and lower 8 bits of 16-bit register
∧	:	Logical product (AND)
∨	:	Logical sum (OR)
∀	:	Exclusive logical sum (exclusive OR)
——	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

24.1.3 Description of "flag operation" column

(Blank)	:	Unchanged
0	:	Cleared to 0
1	:	Set to 1
×	:	Set/cleared according to the result
R	:	Previously saved value is restored

24.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-Bit Data Transfer	MOV	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9+n	A ← (addr16)	
		!addr16, A	3	8	9+m	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5+n	A ← (DE)	
		[DE], A	1	4	5+m	(DE) ← A	
		A, [HL]	1	4	5+n	A ← (HL)	
		[HL], A	1	4	5+m	(HL) ← A	
		A, [HL+byte]	2	8	9+n	A ← (HL+byte)	
		[HL+byte], A	2	8	9+m	(HL+byte) ← A	
		A, [HL+B]	1	6	7+n	A ← (HL+B)	
		[HL+B], A	1	6	7+m	(HL+B) ← A	
		A, [HL+C]	1	6	7+n	A ← (HL+C)	
		[HL+C], A	1	6	7+m	(HL+C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ (sfr)	
		A, !addr16	3	8	10+n+m	A ↔ (addr16)	
		A, [DE]	1	4	6+n+m	A ↔ (DE)	
		A, [HL]	1	4	6+n+m	A ↔ (HL)	
		A, [HL+byte]	2	8	10+n+m	A ↔ (HL+byte)	
		A, [HL+B]	2	8	10+n+m	A ↔ (HL+B)	
		A, [HL+C]	2	8	10+n+m	A ↔ (HL+C)	

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

(3) Except r = A

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

(3) n is the number of waits when external memory expansion area is read from.

(4) m is the number of waits when external memoryexpansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
16-Bit Data Transfer	MOVW	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12+2n	AX ← (addr16)	
		!addr16, AX	3	10	12+2m	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-Bit Operation	ADD	A, #byte	2	4	–	A, CY ← A+byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)+byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A+r	× × ×
		r, A	2	4	–	r, CY ← r+A	× × ×
		A, saddr	2	4	5	A, CY ← A+(saddr)	× × ×
		A, !addr16	3	8	9+n	A, CY ← A+(addr16)	× × ×
		A, [HL]	1	4	5+n	A, CY ← A+(HL)	× × ×
		A, [HL+byte]	2	8	9+n	A, CY ← A+(HL+byte)	× × ×
		A, [HL+B]	2	8	9+n	A, CY ← A+(HL+B)	× × ×
		A, [HL+C]	2	8	9+n	A, CY ← A+(HL+C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A+byte+CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)+byte+CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A+r+CY	× × ×
		r, A	2	4	–	r, CY ← r+A+CY	× × ×
		A, saddr	2	4	5	A, CY ← A+(saddr)+CY	× × ×
		A, !addr16	3	8	9+n	A, CY ← A+(addr16)+CY	× × ×
		A, [HL]	1	4	5+n	A, CY ← A+(HL)+CY	× × ×
		A, [HL+byte]	2	8	9+n	A, CY ← A+(HL+byte)+CY	× × ×
		A, [HL+B]	2	8	9+n	A, CY ← A+(HL+B)+CY	× × ×
		A, [HL+C]	2	8	9+n	A, CY ← A+(HL+C)+CY	× × ×

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

(3) Only when rp = BC, DE or HL

(4) Except r = A

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

(3) n is the number of waits when external memory expansion area is read from.

(4) m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-Bit Operation	SUB	A, #byte	2	4	–	A, CY ← A–byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)–byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A–r	×	×	×
		r, A	2	4	–	r, CY ← r–A	×	×	×
		A, saddr	2	4	5	A, CY ← A–(saddr)	×	×	×
		A, !addr16	3	8	9+n	A, CY ← A–(addr16)	×	×	×
		A, [HL]	1	4	5+n	A, CY ← A–(HL)	×	×	×
		A, [HL+byte]	2	8	9+n	A, CY ← A–(HL+byte)	×	×	×
		A, [HL+B]	2	8	9+n	A, CY ← A–(HL+B)	×	×	×
		A, [HL+C]	2	8	9+n	A, CY ← A–(HL+C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A–byte–CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)–byte–CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A–r–CY	×	×	×
		r, A	2	4	–	r, CY ← r–A–CY	×	×	×
		A, saddr	2	4	5	A, CY ← A–(saddr)–CY	×	×	×
		A, !addr16	3	8	9+n	A, CY ← A–(addr16)–CY	×	×	×
		A, [HL]	1	4	5+n	A, CY ← A–(HL)–CY	×	×	×
		A, [HL+byte]	2	8	9+n	A, CY ← A–(HL+byte)–CY	×	×	×
		A, [HL+B]	2	8	9+n	A, CY ← A–(HL+B)–CY	×	×	×
		A, [HL+C]	2	8	9+n	A, CY ← A–(HL+C)–CY	×	×	×
	AND	A, #byte	2	4	–	A ← A∧byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∧byte	×		
		A, r Note 3	2	4	–	A ← A∧r	×		
		r, A	2	4	–	r ← r∧A	×		
		A, saddr	2	4	5	A ← A∧(saddr)	×		
		A, !addr16	3	8	9+n	A ← A∧(addr16)	×		
		A, [HL]	1	4	5+n	A ← A∧[HL]	×		
		A, [HL+byte]	2	8	9+n	A ← A∧[HL+byte]	×		
		A, [HL+B]	2	8	9+n	A ← A∧[HL+B]	×		
		A, [HL+C]	2	8	9+n	A ← A∧[HL+C]	×		

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

(3) Except r = A

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

(3) n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-Bit Operation	OR	A, #byte	2	4	–	A ← A∨byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨byte	×		
		A, r Note 3	2	4	–	A ← A∨r	×		
		r, A	2	4	–	r ← r∨A	×		
		A, saddr	2	4	5	A ← A∨(saddr)	×		
		A, !addr16	3	8	9+n	A ← A∨(addr16)	×		
		A, [HL]	1	4	5+n	A ← A∨(HL)	×		
		A, [HL+byte]	2	8	9+n	A ← A∨(HL+byte)	×		
		A, [HL+B]	2	8	9+n	A ← A∨(HL+B)	×		
		A, [HL+C]	2	8	9+n	A ← A∨(HL+C)	×		
	XOR	A, #byte	2	4	–	A ← A⊻byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)⊻byte	×		
		A, r Note 3	2	4	–	A ← A⊻r	×		
		r, A	2	4	–	r ← r⊻A	×		
		A, saddr	2	4	5	A ← A⊻(saddr)	×		
		A, !addr16	3	8	9+n	A ← A⊻(addr16)	×		
		A, [HL]	1	4	5+n	A ← A⊻(HL)	×		
		A, [HL+byte]	2	8	9+n	A ← A⊻(HL+byte)	×		
		A, [HL+B]	2	8	9+n	A ← A⊻(HL+B)	×		
		A, [HL+C]	2	8	9+n	A ← A⊻(HL+C)	×		
	CMP	A, #byte	2	4	–	A–byte	×	×	×
		saddr, #byte	3	6	8	(saddr)–byte	×	×	×
		A, r Note 3	2	4	–	A–r	×	×	×
		r, A	2	4	–	r–A	×	×	×
		A, saddr	2	4	5	A–(saddr)	×	×	×
		A, !addr16	3	8	9+n	A–(addr16)	×	×	×
		A, [HL]	1	4	5+n	A–(HL)	×	×	×
		A, [HL+byte]	2	8	9+n	A–(HL+byte)	×	×	×
		A, [HL+B]	2	8	9+n	A–(HL+B)	×	×	×
		A, [HL+C]	2	8	9+n	A–(HL+C)	×	×	×

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

(3) Except r = A

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

(3) n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-Bit Operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX+word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX–word	×	×	×
	CMPW	AX, #word	3	6	–	AX–word	×	×	×
Multiply/ Divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX÷C			
Increase/ Decrease	**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			
Rotation	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
	ROR4	[HL]	2	10	12+n+m	A_{3-0} ← $(HL)_{3-0}$, ← $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12+n+m	A_{3-0} ← $(HL)_{7-4}$, ← $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD Adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit Manipulation	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7+n	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8+n+m	(HL).bit ← CY			

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

(3) n is the number of waits when external memory expansion area is read from.

(4) m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit 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+n	CY ← CY∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY∨ PSW.bit			×
		CY, [HL].bit	2	6	7+n	CY ← CY∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY⊻ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY⊻ PSW.bit			×
		CY, [HL].bit	2	6	7+n	CY ← CY⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8+n+m	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8+n+m	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$			×

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

(3) n is the number of waits when external memory expansion area is read from.

(4) m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call Return	**CALL**	!addr16	3	7	–	(SP–1) ← (PC+3)H, (SP–2) ← (PC+3)L, PC ← addr16, SP ← SP–2			
	CALLF	!addr11	2	5	–	(SP–1) ← (PC+2)H, (SP–2) ← (PC+2)L, PC15–11 ← 00001, PC10–0 ← addr11, SP ← SP–2			
	CALLT	[addr5]	1	6	–	(SP–1) ← (PC+1)H, (SP–2) ← (PC+1)L, PCH ← (00000000, addr5+1), PCL ← (00000000, addr5) SP ← SP–2			
	BRK		1	6	–	(SP–1) ← PSW, (SP–2) ← (PC+1)H, (SP–3) ← (PC+1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP–3, IE ← 0			
	RET		1	6	–	PCH ← (SP+1), PCL ← (SP), SP ← SP+2			
	RETI		1	6	–	PCH ← (SP+1), PCL ← (SP), PSW ← (SP+2), SP ← SP+3, NMIS ← 0	R	R	R
	RETB		1	6	–	PCH ← (SP+1), PCL ← (SP), PSW ← (SP+2), SP ← SP+3	R	R	R
Stack Manipulation	**PUSH**	PSW	1	2	–	(SP–1) ← PSW, SP ← SP–1			
		rp	1	4	–	(SP–1) ← rpH, (SP–2) ← rpL, SP ← SP–2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP+1	R	R	R
		rp	1	4	–	rpH ← (SP+1), rpL ← (SP), SP ← SP+2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional Branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 jdisp8			
		AX	2	8	–	PCH ← A, PCL ← X			
Conditional Branch	**BC**	$addr16	2	6	–	PC ← PC + 2 jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 jdisp8 if Z = 0			

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (fCPU) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional Branch	BT	saddr.bit, $addr16	3	8	9	PC ← PC+3+jdisp8 if(saddr.bit)=1	
		sfr.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if sfr.bit=1	
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit=1	
		PSW.bit, $addr16	3	–	9	PC ← PC+3+jdisp8 if PSW.bit=1	
		[HL].bit, $addr16	3	10	11+n	PC ← PC+3+jdisp8 if (HL).bit=1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC+4+jdisp8 if (saddr.bit)=0	
		sfr.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if sfr.bit=0	
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit=0	
		PSW.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if PSW.bit=0	
		[HL].bit, $addr16	3	10	11+n	PC ← PC+3+jdisp8 if (HL).bit=0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC+4+jdisp8 if(saddr.bit)=1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC+4+jdisp8 if sfr.bit=1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit=1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC+4+jdisp8 if PSW.bit=1 then reset PSW.bit	× × ×
		[HL].bit.$addr16	3	10	12+n+m	PC ← PC+3+jdisp8 if (HL).bit=1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B–1, then PC ← PC+2+jdisp8 if B≠0	
		C, $addr16	2	6	–	C ← C–1, then PC ← PC+2+jdisp8 if C≠0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr)–1, then PC ← PC+3+jdisp8 if (saddr)≠0	
CPU Control	SEL	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes:

(1) When the internal high-speed RAM area is accessed or instruction with no data access

(2) When an area except the internal high-speed RAM area is accessed

Remarks:

(1) One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

(2) This clock cycle applies to internal ROM program.

(3) n is the number of waits when external memory expansion area is read from.

(4) m is the number of waits when external memory expansion area is written to.

24.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte] [HL+B] [HL+C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
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

First Operand \ Second Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note: Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruciton					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD78098 sub-series.

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.
	Part Number: 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.
	Part Number: μSxxxxCC78K0
CC78K/0-L C Complier Library Source File	This is a function source program configurating object library included in CC78K/0 C compiler.
	Part Number: μSxxxxCC78K0-L

Remarks:

(1) When using CC78K/0 C compiler package, RA78K/0 assembler package (option) is necessary.

(2) Purchase the CC78K/0-L C complier library source file when modifying the object library included in the C compiler package to customer's specification.

(3) xxxx of the part number differs depending on the host machine used. Refer to the table below.

Host Machine	OS	Supply Medium	XXXX Part of Part Number
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[Note]	3.5-inch 2HC	7B13
		5-inch 2HC	7B10
HP9000 series 300™	HP-UX™ (rel.7.05B)	Cartridge tape (QIC-24)	3H15
SPARCstation™	Sun OS™ (rel.4.1.1)		3K15
EWS-4800 series (RISC)	EWS-UX/V (rel.4.0)		3M15

Note: The MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0 have a task swap function, but this task swap function is not used in this software.

PROM Programming Tools

<table>
<tr><td rowspan="2">Hardware</td><td>PG-1500</td><td colspan="4">This is a PROM programmer capable of programming the single-chip microcomputer incorporated in the PROM by manipulating from the stand-alone or host machine through connection of a program adapter separately purchasable and attached board. It can also program representative PROMs ranging from 256K bits to 4M bits.</td></tr>
<tr><td>PA-78P054GC
PA-78P054KK-T</td><td colspan="4">PROM programmer adapter that is also used with the μPD78P054 and is connected to the PG-1500.
PA-78P054GC : 80-pin plastic QFP (14 x 14 mm)
PA-78P054KK-T: 80-pin ceramic WQFN</td></tr>
<tr><td rowspan="6">Software</td><td rowspan="6">PG-1500
Controller</td><td colspan="4">The PG-1500 is controlled in the host machine through connection with the host machine and PG-1500 via serial and parallel interfaces.</td></tr>
<tr><td rowspan="2">Host Machine</td><td rowspan="2">OS</td><td rowspan="2">Supply Medium</td><td rowspan="2">Part Number (Product Name)</td></tr>
<tr></tr>
<tr><td rowspan="2">PC-9800 series</td><td rowspan="2">MS-DOS Ver.3.30 to Ver.5.00A Note</td><td>3.5-inch 2HD</td><td>μS5A13PG1500</td></tr>
<tr><td>5-inch 2HD</td><td>μS5A10PG1500</td></tr>
<tr><td rowspan="2">IBM PC/AT</td><td rowspan="2">PC DOS Ver. 3.3 to Ver.5.0 Note</td><td>3.5-inch 2HC</td><td>μS7B13PG1500</td></tr>
<tr><td>5-inch 2HC</td><td>μS7B10PG1500</td></tr>
</table>

★

Note: The MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0 have a task swap function, but this task swap function is not used in this software.

Debugging Tool

An in-circuit emulator (IE-78000-R) is available as μPD78098 sub-series 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 that serves to debug hardware and software when developing application systems using 78K/0 series. 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-78098-R-EM	Emulation board for the μPD78098 sub-series ★
	EP-78230GC-R	Emulation probe used in common with the μPD78234 sub-series. The EP-78230GC-R is for 80-pin plastic QFP (14 x 14 mm). An 80-pin conversion socket EV-9200GC-80 is provided to facilitate user system development.
	EV-9200GC-80	Conversion socket to connect the EP-78230GC-R and user system board created to mount the 80-pin plastic QFP (14 x 14 mm). The μPD78P098AKK-T (ceramic WQFN) can be mounted instead of connecting the EP-78230GC-R.
	EV-9900	A jig used to remove the μPD78P098AKK-T from the EV-9200GC-80.

Remarks:

(1) Above tools can be used for any of the μPD78098 sub-series products.

(2) EV-9200GC-80s are sold in sets of five units.

Debugging tool (2/2)

Software	SD78K/0	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	Part Number (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
	DF78098	Device file for the μPD78098 sub-series. Used in conjunction with SD78K/0.			
		Host Machine	OS	Supply Medium	Part Number (Product Name)
★		PC-9800 series	MS-DOS Ver.3.30 to Ver.5.00A Note	3.5-inch 2HD	μS5A13DF78098
				5-inch 2HD	μS5A10DF78098
★		IBM PC/AT	PC DOS Ver.3.3 to Ver.5.0 Note	3.5-inch 2HC	μS7B13DF78098
				5-inch 2HC	μS7B10DF78098

Note: The MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0 have a task swap function, but this task swap function is not used in this software.

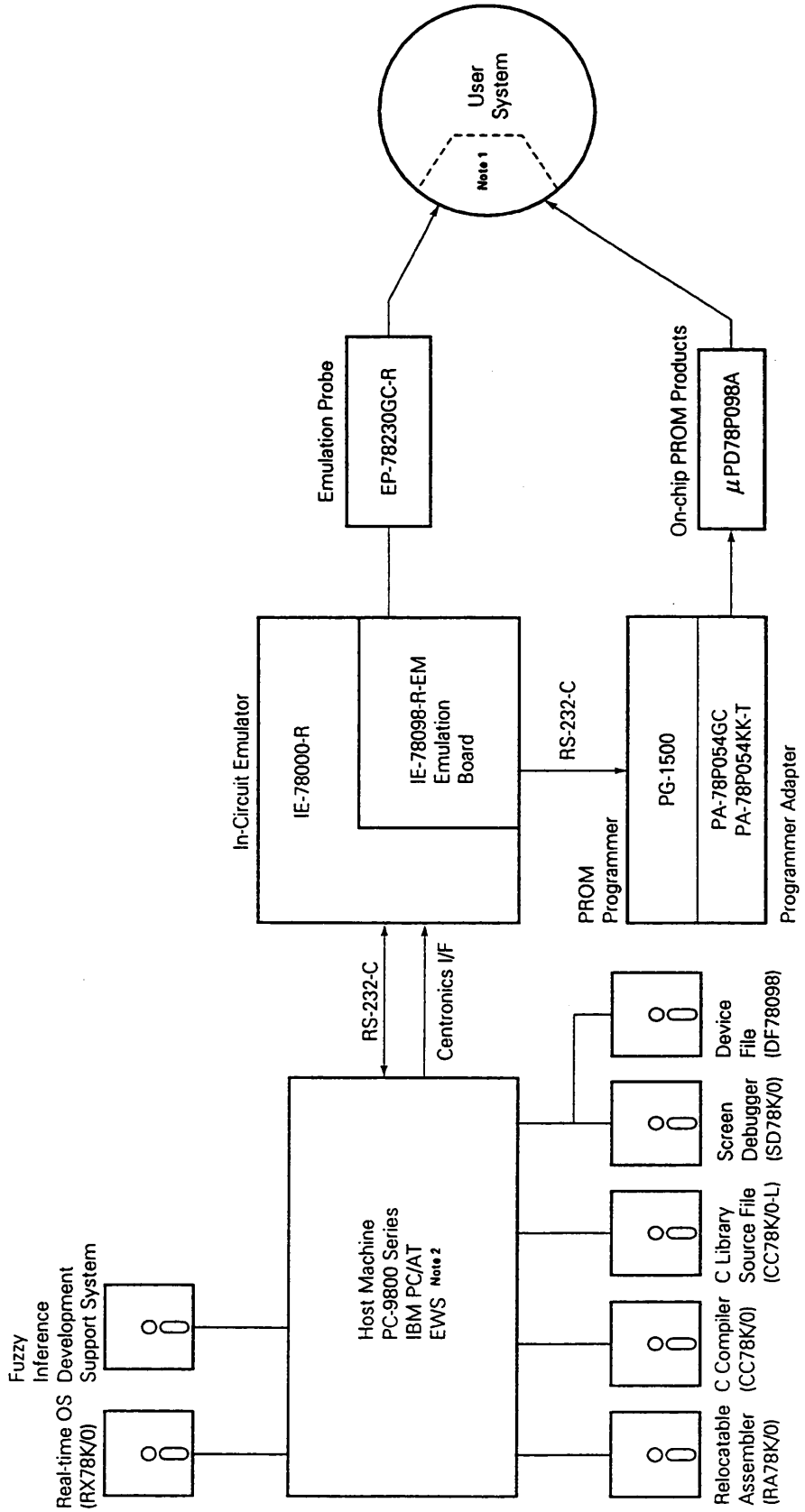
Remark: The above tools can be used for any of the μPD78098 sub-series products.

System-up Method from Other In-Circuit Emulator to IE-78000-R

When you already have an in-circuit emulator for the 78K series or the 75X series, you can use that in-circuit emulator as the equivalent of a 78K/0 in-circuit emulator IE-78000-R by replacing the internal break board with the IE-78000-R-BK.

Series Name	In-circuit Emulator Owned	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



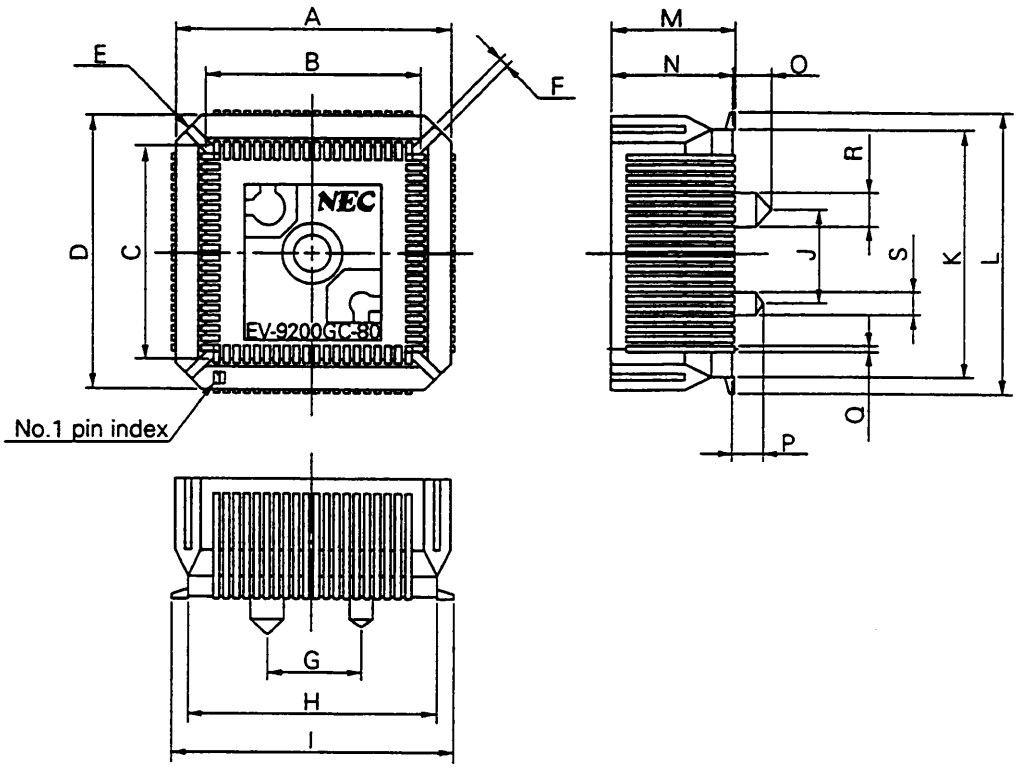
Notes:

(1) EV-9200GC-80 (When EP-78230GC-R or μPD78P098AKK-T is used)

(2) In EWS (HP9000 series 300, SPARCstation, and EWS-4800 series), only language processing software is operated.

Conversion Socket Drawing (EV-9200GC-80) and Footprint

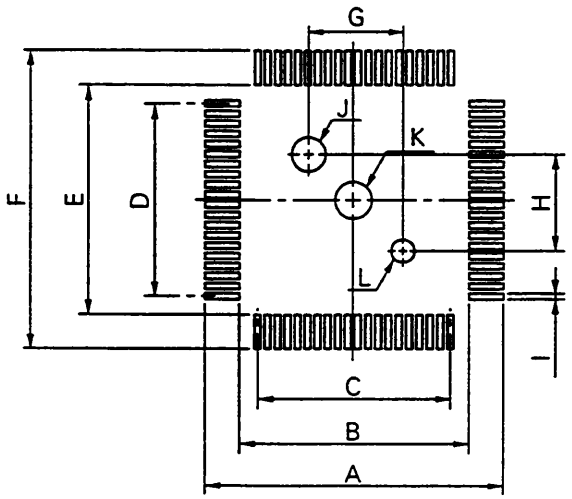
Fig. A-1 EV-9200GC-80 Socket Drawing (Reference)



EV-9200GC-80-G0

ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	14.4	0.567
C	14.4	0.567
D	18.0	0.709
E	4-C 2.0	4-C 0.079
F	0.8	0.031
G	6.0	0.236
H	16.0	0.63
I	18.7	0.736
J	6.0	0.236
K	16.0	0.63
L	18.7	0.736
M	8.2	0.323
O	8.0	0.315
N	2.5	0.098
P	2.0	0.079
Q	0.35	0.014
R	ϕ2.3	ϕ0.091
S	ϕ1.5	ϕ0.059

Fig. A-2 EV-9200GC-80 Footprint (Reference)



EV-9200GC-80-P0

ITEM	MILLIMETERS	INCHES
A	19.7	0.776
B	15.0	0.591
C	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002} \times 0.748 = 0.486^{+0.003}_{-0.002}$
D	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002} \times 0.748 = 0.486^{+0.003}_{-0.002}$
E	15.0	0.591
F	19.7	0.776
G	6.0±0.05	$0.236^{+0.003}_{-0.002}$
H	6.0±0.05	$0.236^{+0.003}_{-0.002}$
I	0.35±0.02	$0.014^{+0.001}_{-0.001}$
J	φ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	φ2.3	φ0.091
L	φ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (IEI-1207).

APPENDIX B EMBEDDED SOFTWARE

Real-time OS

RX78K/0 Real-time OS	RX78K/0 is a real-time OS which is based on the μITRON specification. Supplied with the RX78K/0 nucleus and a tool to prepare multiple information tables (configurator).
	Part Number: μSXXXXRX78013-ΔΔΔΔ

Cautions:

(1) When using the RX78K/0, the RA78K/0 assembler package (option) is necessary.

(2) When purchasing the RX78K/0, fill in the purchase application form in advance, and sign the Use Approval Contract.

Remark: Part number differs depending on host machine and OS, etc.

μSXXXXRX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum number for use in mass production
001	Evaluation object	Do not use for mass-producing product.
100K	Mass production object	100,000
001M		1,000,000
010M		10,000,000
S01	Source program	Source program of mass production object

XXXX	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS Ver.3.30 to Ver.5.00A Note	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT	PC DOS Ver 3.3 to Ver. 5.0 Note	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 series 300	HP-UX (rel.7.05B)	Cartridge tape (QIC-24)
3K15	SPARCstation	Sun OS (rel.4.1.1)	
3M15	EWS-4800 series (RISC)	EWS-UX/V (rel.4.0)	

★

Note: The MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0 have a task swap function, but this task swap function is not used in this software.

★ Fuzzy Inference Development Support System

FE9000/FE9200 Fuzzy Knowledge Data Creation Tool	Program supporting input of fuzzy knowledge data (fuzzy rule and membership function), editing (edit), and evaluation (simulation).
	Part Number: μSXXXXFE9000 (PC-9800 series) μSXXXXFE9200 (IBM PC/AT) Note 1
FT9080/FT9085 Translator	Program converting fuzzy knowledge data obtained by using fuzzy knowledge data creation tool to RA78K/0 assembler source program.
	Part Number: μSXXXXFT9080 (PC-9800 series) μSXXXXFT9085 (IBM PC/AT)
FI78K0 Fuzzy Inference Module	Program executing fuzzy inference. Fuzzy inference is executed by linking fuzzy knowledge data converted by translator.
	Part Number: μSXXXXFI78K0 (PC-9800 series, IBMPC/AT)
FD78K0 Fuzzy Inference Debugger	Support software evaluating and adjusting fuzzy knowledge data at hardware level by using in-circuit emulator.
	Part Number: μSXXXXFD78K0 (PC-9800 series, IBM PC/AT)

Remark: Part number XXXX differs depending on host machine and OS, etc.

μSXXXXFE9000
μSXXXXFT9080
μSXXXXFI78K0
μSXXXXFD78K0

XXXX	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS Ver.3.30 to Ver.5.00A Note 2	3.5-inch 2HD
5A10			5-inch 2HD

μSXXXXFE9200 Note 1
μSXXXXFT9085
μSXXXXFI78K0
μSXXXXFD78K0

XXXX	Host Machine	OS	Supply Medium
7B13	IBM PC/AT	PC DOS Ver.3.3 to Ver.5.0 Note 2	3.5-inch 2HC
7B10			5-inch 2HC

Notes:

(1) To operate FE9200, Windows™ (Ver. 3.0 to Ver. 3.1) is necessary.

(2) The MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0 have a task swap function, but this task swap function is not used in this software.

APPENDIX C REGISTER INDEX

C.1 Register Index (In Alphabetical Order with Respect to the Register Name)

C.2 Register Index (In Alphabetical Order with Respect to the Register Symbol)

APPENDIX D REVISION HISTORY ★

The history of revisions made up to now is shown in the following. The Application column shows in which chapters these revisions were made in.

(1/3)

Edition No.	Revision Made	Application
2nd Edition	μPD78094, 78095, 78096: Under development → Development completed	Throughout
	μPD78P098: Erased from applied models	
	μPD78098A, 78P098A: Added to applied models	
	P60 to P63 pins (for μPD78P098A) input/output circuit type: Corrected to 13 → 13-D	CHAPTER 2 PIN FUNCTION
	The recommended connecting method of unused pins P130/ANO0 and P131/ANO1 was changed	
	Cautions for use of pins P130/ANO0 and P131/ANO1 were added	
	Cautions for **4.2.3 Port 2 and 4.2.8 Port 7** were revised	CHAPTER 4 PORT FUNCTIONS
	Cautions for use of pins P130/ANO0 and P131/ANO1 were added	
	Table 4-3 Port Mode Register and Output Latch Settings when Using Dual Function was added	
	Fig. 5-3 Subsystem Clock Feedback Resistor was added	CHAPTER 5 CLOCK GENERATOR
	Cautions were added to **Fig. 5-5 Oscillation Mode Selection Register Format**	
	Cautions were added to **5.6.2 System clock and CPU clock switching procedure**	
	Cautions were added to format of timer clock selection register 0	CHAPTER 6 16-BIT TIMER/EVENT COUNTER
		CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT
	Cautions were added to **(2) 16-bit timer mode control register (TMC0)** of **6.3 16-Bit Timer/Event Counter Control Registers**	CHAPTER 6 16-BIT TIMER/EVENT COUNTER
	Cautions were added to the following section in **6.4.4 Pulse width measurement operations** **(3) Pulse width measurement with free-running counter and two capture registers** **(4) Pulse width measurement by means of restart**	

(2/3)

Edition No.	Revision Made	Application
2nd Edition	**Fig. 6-32 Timing of One-Shot Pulse Output Operation Using Software Trigger** was revised	CHAPTER 6 16-BIT TIMER/EVENT COUNTER
	Fig. 6-34 Timing of One-Shot Pulse Output Operation Using External Trigger (with Rising Edge Specified) was revised	
	Cautions were added to **Fig. 7-4 Timer Clock Select Register 1 Format**	CHAPTER 7 8-BIT TIMER/EVENT COUNTER
	Table 7-7 8-Bit Timer/Event Counter 2 Interval Time was added	
	Table 7-10 Square-Wave Output Ranges when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter was added	
	Cautions were added to the timer clock select register 2 format	CHAPTER 8 WATCH TIMER
		CHAPTER 9 WATCHDOG TIMER
		CHAPTER 11 BUZZER OUTPUT CONTROL CIRCUIT
	The A/D converter mode register format was changed	CHAPTER 12 A/D CONVERTER
	Cautions were added to the timer clock select register 3 format	CHAPTER 14 SERIAL INTERFACE CHANNEL 0
		CHAPTER 15 SERIAL INTERFACE CHANNEL 1
	The settings of the port mode register and output latch during the use of serial interface were changed	CHAPTER 14 SERIAL INTERFACE CHANNEL 0
		CHAPTER 15 SERIAL INTERFACE CHANNEL 1
		CHAPTER 16 SERIAL INTERFACE CHANNEL 2
	The explanation for the automatic transmit/receive interval time was revised	CHAPTER 15 SERIAL INTERFACE CHANNEL 1
	Table 16-2 Serial Interface Channel 2 Operating Mode Settings was revised	CHAPTER 16 SERIAL INTERFACE CHANNEL 2

(3/3)

Edition No.	Revision Made	Application
2nd Edition	**CHAPTER 18 INTERRUPT FUNCTIONS** was changed to **CHAPTER 18 INTERRUPT FUNCTIONS AND TEST FUNCTIONS**, and **18.5 Test Functions** was added	CHAPTER 18 INTERRUPT FUNCTIONS AND TEST FUNCTIONS
	Accompanying the additions of μPD78098A and 78P098A, changes were made so that 56K bytes can be selected for the internal ROM capacity with the memory size switching register	CHAPTER 19 EXTERNAL DEVICE EXPANSION FUNCTION
		CHAPTER 23 μPD78P098A
	Fig. 20-13 IEBus Controller Mode Register Format was revised	CHAPTER 20 IEBus CONTROLLER
	Fig. 20-23 Formats of Multiaddress Calling Destination Address Registers 1 and 2 was changed	
	23.2 Internal Expansion RAM Size Switching Register was added	CHAPTER 23 μPD78P098A
	Cautions for the specification of write addresses for **23.3 PROM Programming** were added	
	24.2 Instruction Codes and **24.3 Operation List** were erased	CHAPTER 24 INSTRUCTION SET
	IE-78098-R-EM, DF78098: Under development → Development completed	APPENDIX A DEVELOPMENT TOOLS
	APPENDIX C INSTRUCTION INDEX (In Alphabetical Order) was erased	—
	APPENDIX D REVISION HISTORY was added	APPENDIX D REVISION HISTORY