DATA SHEET

NEC

MOS INTEGRATED CIRCUIT

μPD75028(A)

4-BIT SINGLE-CHIP MICROCOMPUTER

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

Document No. IC−3168A
(O. D. No. IC−8669A)
Date Published February 1993 P
Printed in Japan

The uPD75028(A) belongs to the NEC CMOS 4-bit single chip microcomputer 75X series which enables data processing matching an 8-bit microcomputer.

The uPD75028(A) is a powerful and high cost performance product which contains high function such as A/D converter and serial bus interface (SBI) adopting the NEC standard format in addition to 0high-speed operation (minimum instruction execution time by CPU: 0.95 us).

uPD75P036 which contains PROM instead of uPD75028(A) ROM is also provided for evaluation during system development or for small production.

Features

o High reliability as compared with uPD75028

o Fast execution time (@4.19 MHz)
  - High speed cycle: 0.95 us
  - Low voltage cycles: 1.91 us and 15.3 us

o Power-reducing operation
  - With system clock operating at 32.768 kHz (execution time: 122 us)

o A/D converter
  - 8-channel, 8-bit

o Low-voltage operation possible (V_{DD}=2.7 to 6.0V)

o Four timers
  - One of them can be used as PWM output, 16-bit counter for an integrating A/D converter, etc.

o NEC standard serial bus interface
  - SBI mode

o Very low-power clock operation: 5 uA TYP. (at 3V in HALT mode)

o 43 I/O lines
  - With mask-option or software-selectable pull-up/pull-down resistors

o OTP version: uPD75P036 (V_{DD} = 2.7 to 6.0 V)

Applications

Car electronics, etc.

Ordering Information

Ordering Code	Package	Quality Grade
uPD75028CW(A)-xxx	64-pin plastic Shrink DIP	Special
uPD75028GC(A)-xxx-AB8	64-pin plastic QFP (14 x 14 mm)	Special

Remarks: xxx is ROM code number.

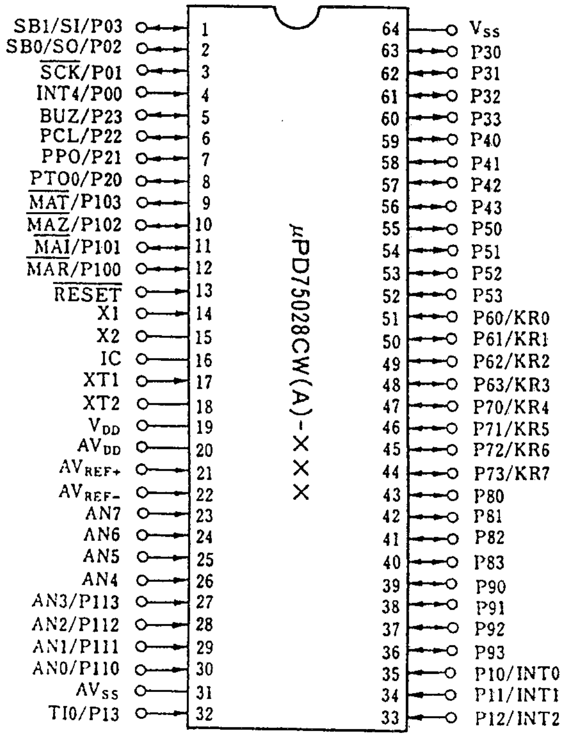
Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

Difference between uPD75028(A) and uPD75028

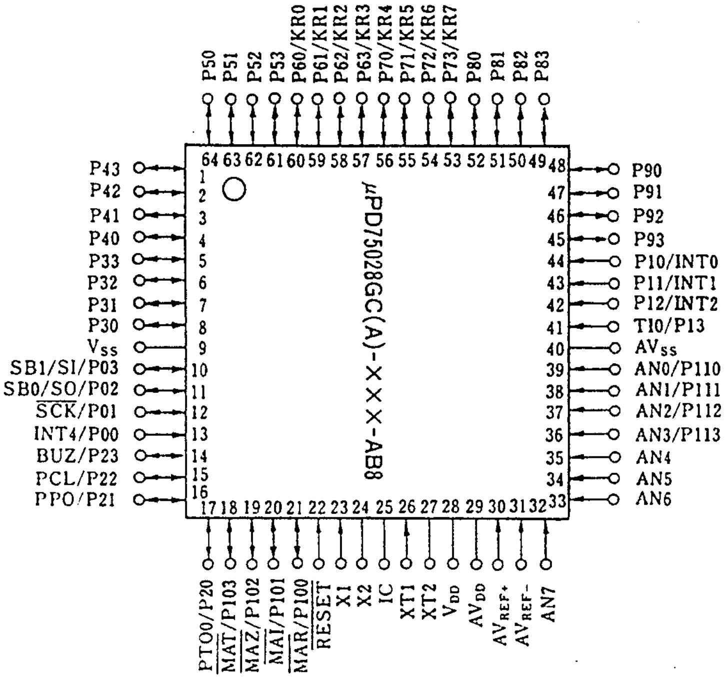
Product Item	uPD75028(A)	uPD75028
Quality grade	Special	Standard
Operation temperature range	-40 to +85°C	-40 to +70°C

Pin Configuration (Top View)

o 64-pin plastic shrink DIP



o 64-pin plastic QFP



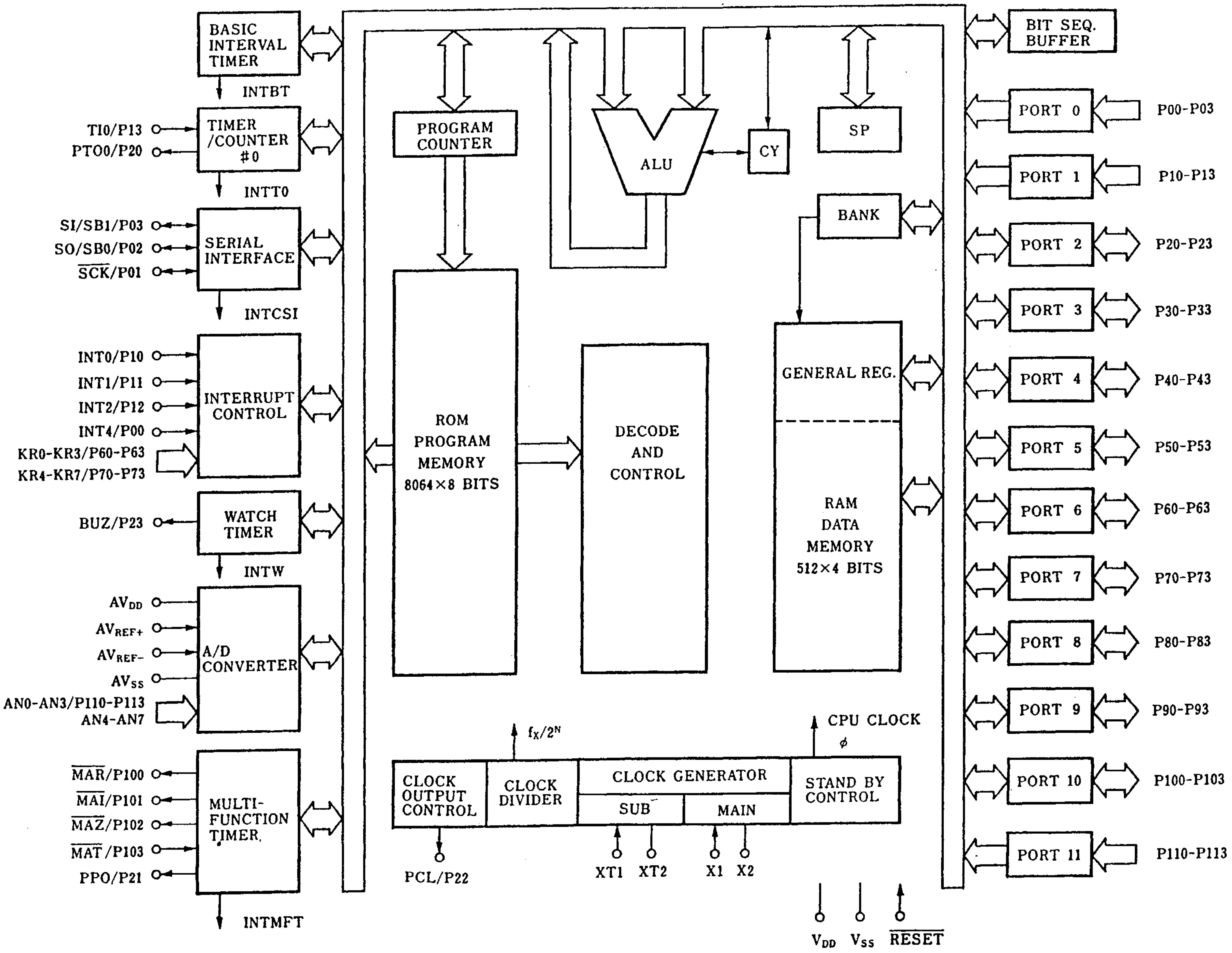
IC: Internally connected.
(Connect the pin to V_{DD})

Pin names

P00-03	: Port0	
P10-13	: Port1	
P20-23	: Port2	
P30-33	: Port3	
P40-43	: Port4	
P50-53	: Port5	
P60-63	: Port6	
P70-73	: Port7	
P80-83	: Port8	
P90-93	: Port9	
P100-103	: Port10	
P110-113	: Port11	
KR0-7	: Key Return	
$\overline{SCK}$	: Serial Clock	
SI	: Serial Input	
SO	: Serial Output	
SB0, 1	: Serial Bus 0, 1	
$\overline{RESET}$	: Reset Input	
TI0	: Timer Input 0	
PTO0	: Programmable Timer Output 0	
BUZ	: Buzzer Clock	
PCL	: Programmable Clock	
INT0, 1, 4	: External Vectored Interrupt 0, 1, 4	
INT2	: External Test Input 2	
X1, 2	: Main System Clock Oscillation 1, 2	
XT1, 2	: Subsystem Clock Oscillation 1, 2	
$\overline{MAR}$	: Reference Integration Control	MFT A/D mode
$\overline{MAI}$	: Integration Control	
$\overline{MAZ}$	: Autozero Control	
$\overline{MAT}$	: External Comparate Timing Input	
PPO	: Programmable Pulse Output...MFT timer mode	
AN0-7	: Analog Input 0-7	
AV_{REF+}	: Analog Reference (+)	
AV_{REF-}	: Analog Reference (-)	
AV_{DD}	: Analog V_{DD}	
AV_{SS}	: Analog V_{SS}	
V_{DD}	: Positive Power Supply	
V_{SS}	: Ground	

Remarks: MFT: Multifunction timer

Block Diagram



Function Outline (1/2)

Item	Function			
Number of instructions	41			
Instruction execution time	• When main system clock is selected: 0.95, 1.91, 15.3us (during 4.19 MHz operation) • When subsystem clock is selected: 122 us (during 32.768 kHz operation)			
On-chip memory	Program memory (ROM): 8064 x 8 bits			
	Data memory (RAM): 512 x 4 bits			
General purpose register	• During 4-bit operation: 8 (X, A, B, C, D, E, H, L) • During 8-bit operation: 4 (XA, BC, DE, HL)			
Accumulator	• Bit accumulator (CY) • 4-bit accumulator (A) • 8-bit accumulator (XA)			
Instruction set	• Various bit manipulation • Efficient 4-bit data handling instruction • 8-bit data transfer instructions • GETI instruction which can convert any 2-byte or 3-byte instruction into a 1-byte instruction for execution			
I/O line	48 lines	12 lines	Input pins	Pull-up resistor can be applied to 27 lines and pull-down resistor can be applied to 4 lines by means of software
		24 lines	CMOS I/O pins (can be LED driving: 4)	
		12 lines	Middle-voltage N-ch open drain input/output (can be LED driving)	Pull-up resistor can be applied to 12 lines by means of mask option
Timer	4 Channels	• 8-bit timer/event counter • Four stages of clock sources • Event count can be made		
		• 8-bit basic interval timer • Reference time generation: 1.95ms, 7.82ms, 31.3ms 250ms (during 4.19MHz operation) • Applicable as a watchdog timer		
		• Watch timer • 0.5-second timer interval generation • Count clock source : Main system clock and subsystem clock can be changed • Watch rapid feed mode (3.9-ms time interval generation) • Buzzer output is enabled (2kHz, 4kHz, 32kHz)		
		• Multifunction timer • 8-bit timer • PWM output • 16-bit free running timer • 16-bit integration A/D converter counter } are applicable		

Function Outline (2/2)

Item	Function
8-bit serial interface	• Can cover the following three modes • 3-line serial I/O mode ... MSB first/LSB first can be selected • 2-line serial I/O mode • SBI mode
Bit sequential buffer	Special bit manipulation memory: 16 bits • Appropriate for remote control application
Clock output function	Timer/event counter output (PTO0): Square wave output of any frequency
	Clock output (PCL): ϕ, $f_X/2^3$, $f_X/2^4$, $f_X/2^6$ (4.19MHz during main system clock operation)
	Buzzer output (BUZ): 2kHz, 4kHz 32kHz (4.19MHz during main system clock operation or 32.768kHz subsystem clock operation)
A/D converter	8-bit resolution A/D converter (successive approximation): 8 channels • Can operate at low voltage: V_{DD}=2.7-6.0V • Reference voltage can be set as desired in the range of AV_{REF+}, AV_{REF-}. 2.5V≤(AV_{REF+})-(AV_{REF-})≤6.0V
Vector interrupt	External: 3 Internal: 4
Test input	External: 1 Internal: 1
System clock oscillator	• Ceramic or crystal oscillator for main system clock oscillation • Crystal oscillator for subsystem clock oscillation
Standby function	• STOP mode: Stops main system clock oscillation. • HALT mode: Continues system clock oscillation. (Stops CPU clock supply)
Package	• 64-pin plastic shrunk DIP (750mil) • 64-pin plastic QFP (14 x 14mm)

CONTENTS

1. PIN FUNCTION

1.1 Port Pins

Pin name	I/O	Dual function pin	Function	8-bit I/O	When reset	Input/output circuit type (Note 1)
P00	I	INT4	4-bit input port (PORT0). Internal pull-up resistor can be specified for P01-P03 in 3-bit units by software.	x	Input	B
P01	I/O	$\overline{\text{SCK}}$				F - A
P02	I/O	SO/SB0				F - B
P03	I/O	SI/SB1				M - C
P10	I	INT0	With noise removing function 4-bit input port (PORT1). Internal pull-up resistor can be specified in 4-bit units by software.	x	Input	B - C
P11		INT1				
P12		INT2				
P13		TI0				
P20	I/O	PTO0	4-bit input/output port (PORT2). Internal pull-up resistor can be specified 4-bit units by software.	x	Input	E - B
P21		PPO				
P22		PCL				
P23		BUZ				
P30(Note 2)	I/O	-	Programmable 4-bit input/output port (PORT3). Input or output mode can be set bit-wise. Internal pull-up resistor can be specified in 4-bit units by software.	x	Input	E - B
P31(Note 2)		-				
P32(Note 2)		-				
P33(Note 2)		-				
P40-P43 (Note 2)	I/O	-	N-ch open drain 4-bit input/output port (PORT4). Internal pull-up resistor can be contained bit-wise (mask option). 10-V voltage 10-V voltage during open drain.	o	High (when pull-up resistor is contained) or high impedance	M
P50-P53 (Note 2)	I/O	-	N-ch open drain 4-bit input/output port (PORT5). Internal pull-up resistor can be contained bit-wise (mask option). 10-V voltage during open drain.		High (when pull-up resistor is contained) or high impedance	M

Note 1: o denotes Schmitt trigger input.

Note 2: LED can be directly driven.

1.1 Port Pins (Continued)

Pin name	I/O	Dual function pin	Function	8-bit I/O	When reset	Input/output circuit type (Note 1)
P60	I/O	KR0	Programmable 4-bit input/output port (PORT6). Input or output mode can be specified bit-wise. Internal pull-up resistor can be specified in 4-bit units by software.	o	Input	Ⓕ - A
P61		KR1				
P62		KR2				
P63		KR3				
P70	I/O	KR4	4-bit input/output port (PORT7). Internal pull-up resistor can be specified 4-bit units by software.		Input	Ⓕ - A
P71		KR5				
P72		KR6				
P73		KR7				
P80-P83	I/O	-	4-bit input/output port (PORT8). Internal pull-up resistor can be specified in 4-bit units by software.	x	Input	E - B
P90-P93	I/O	-	4-bit input/output port (PORT9). Internal pull-down resistor can be specified in 4-bit units by software.		Input	E - D
P100(Note2)	I/O	$\overline{MAR}$	N-ch open drain 4-bit input/output port (PORT10). Internal pull-up resistor can be contained bit-wise (mask option). 10-V voltage during open drain.	x	High (when pull-up resistor is contained) or high impedance	M
P101(Note2)		$\overline{MAI}$				
P102(Note2)		$\overline{MAZ}$				
P103(Note2)		$\overline{MAT}$				
P110	I	AN0	4-bit input port (PORT11).		Input	Y - A
P111		AN1				
P112		AN2				
P113		AN3				

Note 1: o denotes Schmitt trigger input.

Note 2: LED can be directly driven.

1.2 Pins Other Than Port Pins

Pin name	I/O	Dual function pin	Function		When reset	Input/output circuit type (Note 1)
TI0	I	P13	External event pulse input to timer/ event counter.		—	Ⓑ- C
PTO0	I/O	P20	Timer/event counter output.		Input	E - B
PCL	I/O	P22	Clock output.		Input	E - B
BUZ	I/O	P23	Any desired frequency output (for buzzer or system clock trimming).		Input	E - B
$\overline{SCK}$	I/O	P01	Serial clock input/output.		Input	Ⓕ- A
SO/SB0	I/O	P02	Serial data output. Serial bus input/output.		Input	Ⓕ- B
SI/SB1	I/O	P03	Serial data input. Serial bus input/output.		Input	Ⓜ- C
INT4	I	P00	Edge-detected vectored interrupt input (both rising edge detection and falling edge detection are valid).		—	Ⓑ
INT0	I	P10	Edge-detected vectored interrupt input (detection edge can be selected).	Clocked	—	Ⓑ- C
INT1		P11		Asynchronous		
INT2	I	P12	Edge-detected testable input (rising edge detection).	Asynchronous	—	Ⓑ- C
KR0-KR3	I/O	P60-P63	Parallel falling edge detect testable input.		Input	Ⓕ- A
KR4-KR7	I/O	P70-P73	Parallel falling edge detect testable input.		Input	Ⓕ- A
$\overline{MAR}$	I/O	P100	During MFT integration A/D converter mode	Reverse integration signal output.	(Note 2)	-
$\overline{MAI}$	I/O	P101		Integration signal output.	(Note 2)	-
$\overline{MAZ}$	I/O	P102		Auto zero signal output.	(Note 2)	-
$\overline{MAT}$	I/O	P103		Comparator input.	(Note 2)	-

Note 1: ○ denotes Schmitt trigger input

Note 2: High (when pull-up resistor is contained) or high impedance.

Remarks: MFT: Multifunction timer

1.2 Pins Other Than Port Pins (Continued)

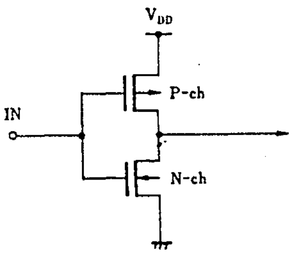
Pin name	I/O	Dual function pin	Function		When reset	Input/output circuit type (Note 1)
PPO	I/O	P21	During MFT timer mode	Timer pulse output.	Input	E - B
AN0-AN3	I	P110-P113	A/D converter dedicated pins	8-bit analog input.	Input	Y - A
AN4-AN7		-				Y
AV_{REF+}	I	-		Reference voltage input (AV_{DD} side).	—	Z - A
AV_{REF-}	I	-		Reference voltage input (AV_{SS} side).	—	Z - A
AV_{DD}	I	-		Operation voltage input	—	–
AV_{SS}		-		Reference GND input.	—	–
X1, X2	I	-	Crystal or ceramic connection pins for main system clock oscillation. To supply external clock, input it to XT1 and its inverted phase to X2. X1 can be used as a 1-bit input (test).		—	–
XT1, XT2	I	-	Crystal connection pins for subsystem clock oscillation. To supply external clock, input it to XT1 and its inverted phase to XT2. XT1 can be used as a 1-bit input (test).		—	–
$\overline{\text{RESET}}$	I	-	System reset input.		—	Ⓑ
IC	-	-	Internally Connected. Connect to V_{DD}		—	–
V_{DD}	-	-	Positive power supply.		—	–
V_{SS}	-	-	GND potential.		—	–

Note 1: o denotes Schmitt trigger input.

1.3 Pin Input/Output Circuits

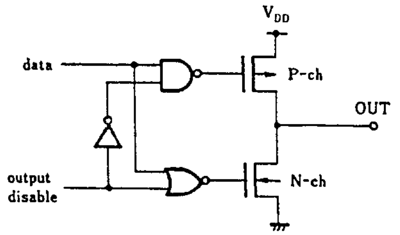
The pin input/output circuits of the uPD75028(A) are shown schematically.

TYPE A (For TYPE E-B)



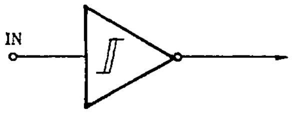
Specification input buffer.

TYPE D (For TYPE E-B and F-A)



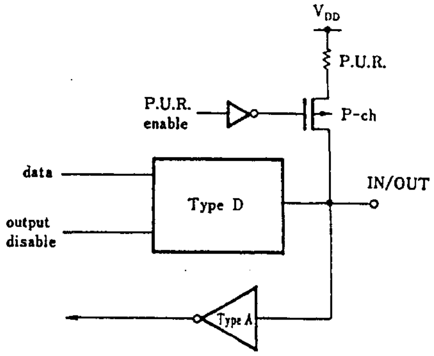
Push-pull output to enable output to became high impedance (both P-ch and N-ch off).

TYPE B



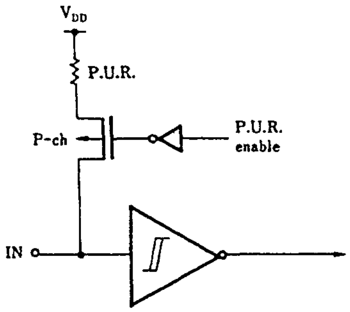
Schmitt trigger input having hysteresis characteristic.

TYPE E-B



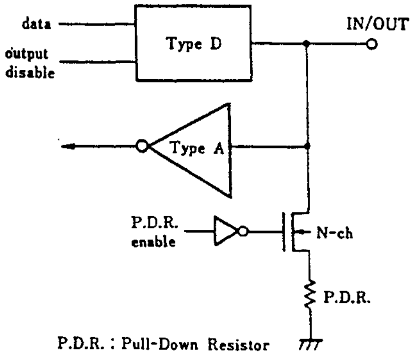
P.U.R. : Pull-Up Resistor

TYPE B-C



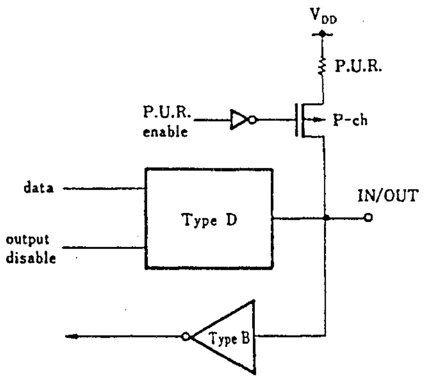
P.U.R. : Pull-Up Resistor

TYPE E-D



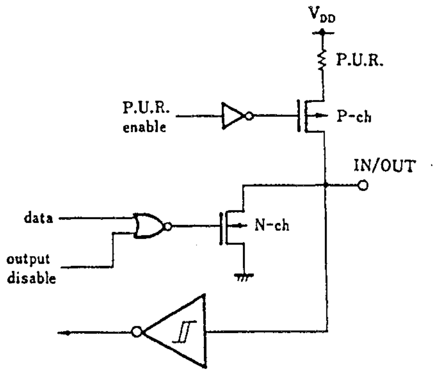
P.D.R. : Pull-Down Resistor

TYPE F-A



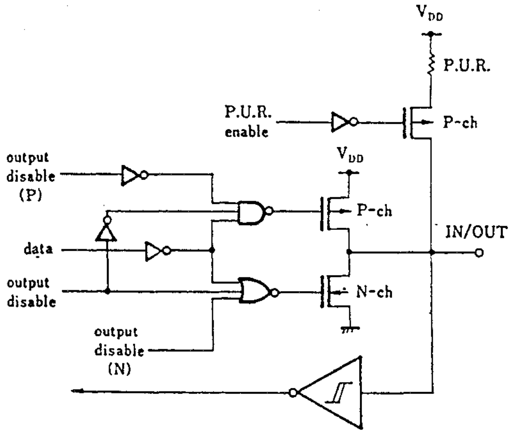
P.U.R. : Pull-Up Resistor

TYPE M-C



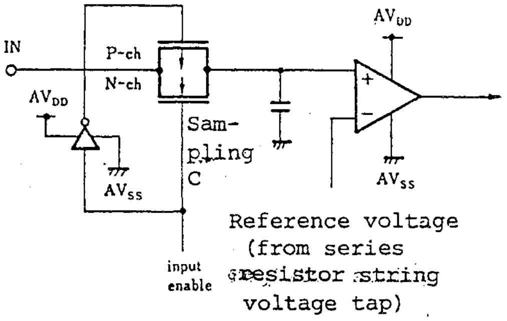
P.U.R. : Pull-Up Resistor

TYPE F-B

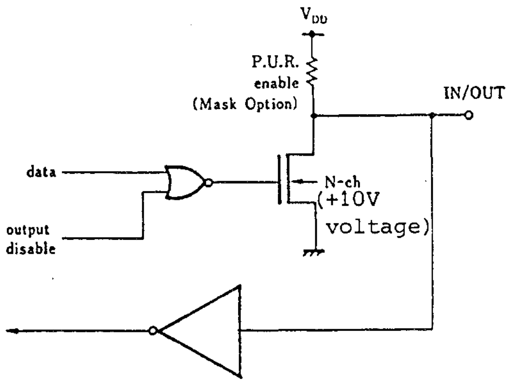


P.U.R. : Pull-Up Resistor

TYPE Y



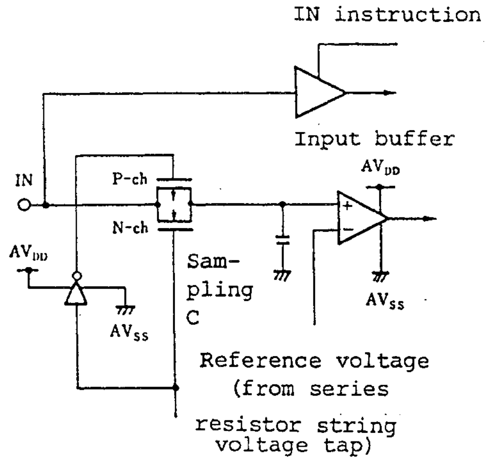
TYPE M

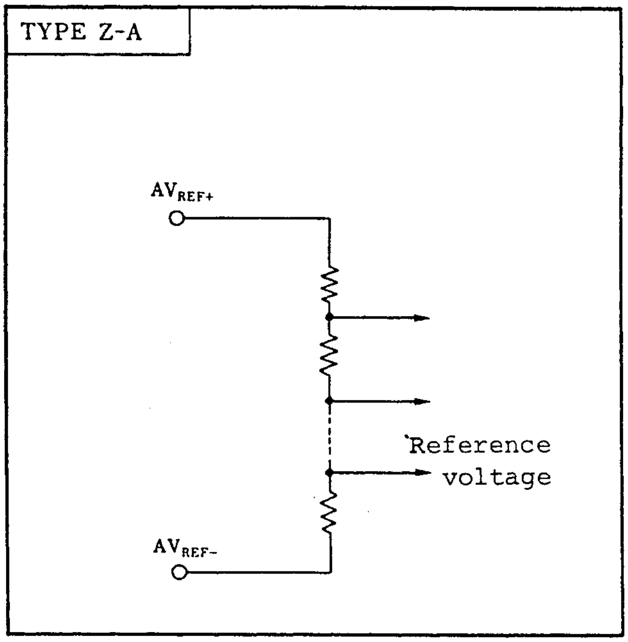


Midium-voltage input buffer
(+10V voltage)

P.U.R. : Pull-Up Resistor

TYPE Y-A





1.4 Mask Option Selection

The pins contain mask option, as listed below:

Pin function	Mask option	
P40-P43, P50-P53, P100-P103	① Pull-up resistor (can be specified bit-wise)	② No pull-up resistor (can be specified bit-wise)
XT1, XT2	① Feedback resistor (when subsystem clock is used)	② No feedback resistor (when subsystem clock is not used)

2. uPD75028(A) ARCHITECTURE AND MEMORY MAP

The uPD75028(A) architecture features are as follows:

(a) Bank configuration of data memory
(b) Memory mapped-I/O

General description of these features is given.

2.1 Data Memory Bank Configuration and Addressing Mode

The uPD75028(A) contains 512-word x 4-bit static RAM at addresses 000H-1FFH of data memory space and peripheral hardware such as input/output ports and timers at addresses F80H-FFFH, as shown in Fig. 2-1. To address the data memory space of 12-bit addresses, the low-order eight bits of an address are specified directly or indirectly by an instruction and the high-order four bits are specified according to memory bank (MB). (Memory bank configuration)

Memory bank enable flag (MBE) and memory bank selection register (MBS) are contained for memory bank (MB) specification and enable addressing as shown in Figs. 2-1 and 2-2 and Table 2-1. (MBS is used to select a memory bank and 0, 1, or 15 can be set. MBE is a flag to determine whether or not the memory bank selected in MBS is validated. MBE is automatically saved and restored during interrupt processing or subroutine processing, thus can be set as desired by the interrupt service routine or subroutine.)

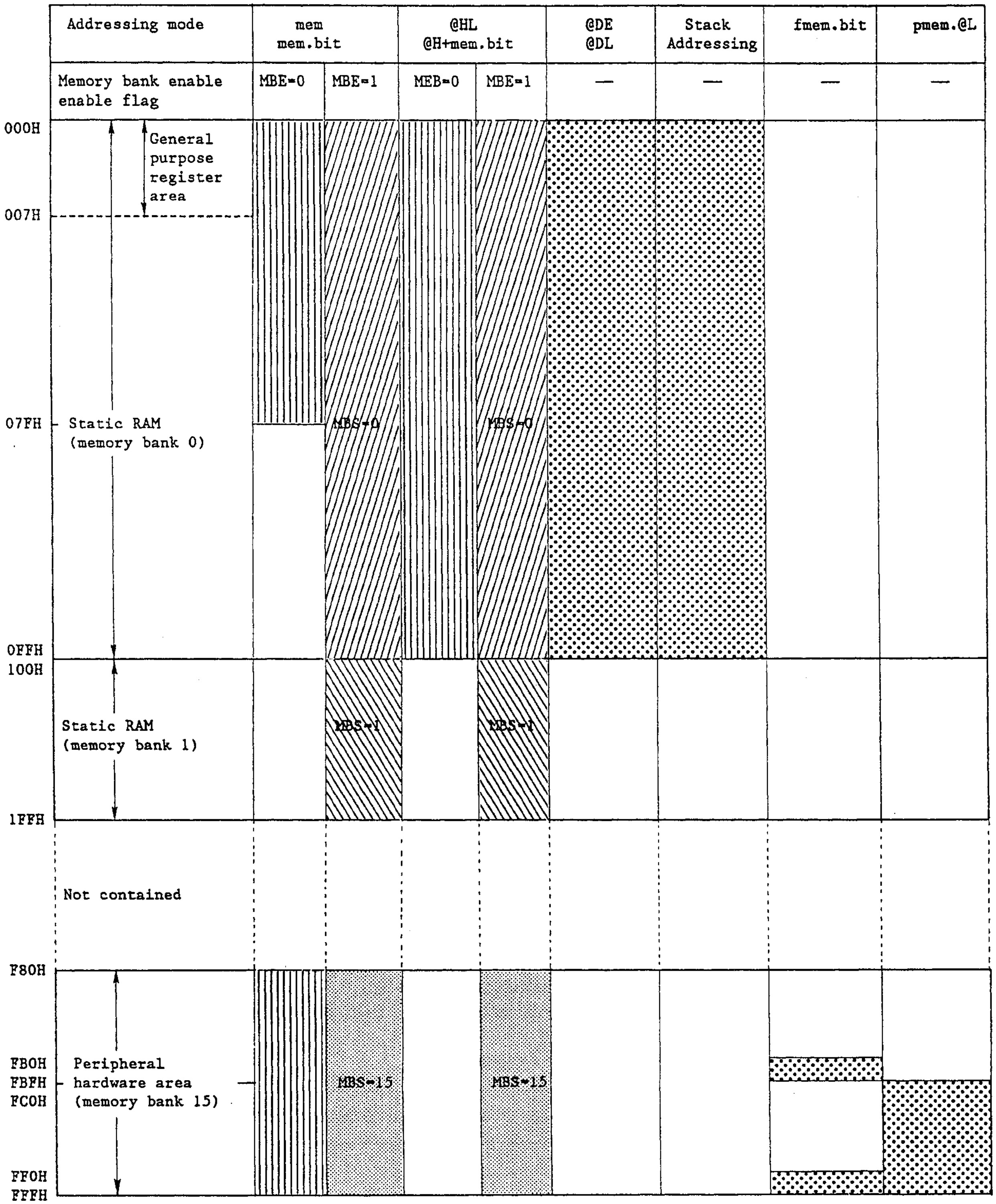
For addressing the data memory space, normally MBE is set to 1 and static RAM of the memory bank specified in MBS is operated; efficient programming is enabled by using MBE = 0 mode and MBE = 1 mode each properly in program processing.

	Applicable program processing
MBE = 0 mode	o Interrupt processing o Repetition of internal hardware operation and static RAM operation o Subroutine processing
MBE = 1 mode	o Normal program processing

MBE and MBS are set as shown below:

```
Example:  SET1 MBE   ; MBE  ←  1
          CLR1 MBE   ; MBE  ←  0
          SEL MB0    ; MBS  ←  0
          SEL MB1    ; MBS  ←  1
          SEL MB15   ; MBS  ←  15
```

Fig. 2-1 Data Memory Configuration and Addressing Range for each Addressing Mode



-: don't care

Table 2-1 Addressing Mode

Addressing mode	Identifier	Specified address
1-bit direct addressing	mem.bit	Bit of address indicated by MB and mem. The bit position is indicated by bit. - When MBE = 0, MB = 0 when mem = 00H-7FH MB = 15 when mem = 80H-FFH - When MBE = 1, MB = MBS
4-bit direct addressing	mem	Address indicated by MB and mem. - When MBE = 0, MB = 0 when mem = 00H-7FH MB = 15 when mem = 80H-FFH - When MBE = 1, MB = MBS
8-bit direct addressing		Address indicated by MB and mem (mem is an even address). - When MBE = 0, MB = 0 when mem = 00H-7FH MB = 15 when mem = 80H-FFH - When MBE = 1, MB = MBS
4-bit register indirect addressing	@HL	Address indicated by MB, HL. MB = MBE·MBS
	@DE	Memory bank 0 address indicated by DE.
	@DL	Memory bank 0 address indicated by DL.
8-bit register indirect addressing	@HL	Address indicated by MB and HL (L register contains an even number). BM = MBE·MBS
Bit manipulation addressing	fmem.bit	Bit of address indicated by fmem. The bit position is indicated by bit. fmem = FB0H=FBFH (hardware related to interrupts) FF0H-FFFH (I/O port)
	pmem. @L	Bit of address indicated by the high-order 10 bits of pmem and the high-order two bits of the L register. The bit position is indicated by the low-order two bits of the L register. pmem = FC0H-FFFH
	@H+mem.bit	Bit indicated by MB, H, and the low-order four bit of mem. The bit position is indicated by bit. MB = MBE·MBS
Stack addressing		Memory bank 0 address indicated by SP.

As listed in Table 2-1, direct or indirect addressing is enabled for each of 1-bit data, 4-bit data, and 8-bit data on uPD75028(A) data memory operation; very efficient and easy-to-understand programs can be prepared.

Example 1: Transfer 8-bit data at port 4, 5 to address 20H, 21H.

```
CLR1 MBE        ; MBE ← 0
IN   XA, PORT4  ; XA ← Port 5, 4
MOV  20H, XA    ; (21H, 20H) ← XA
```

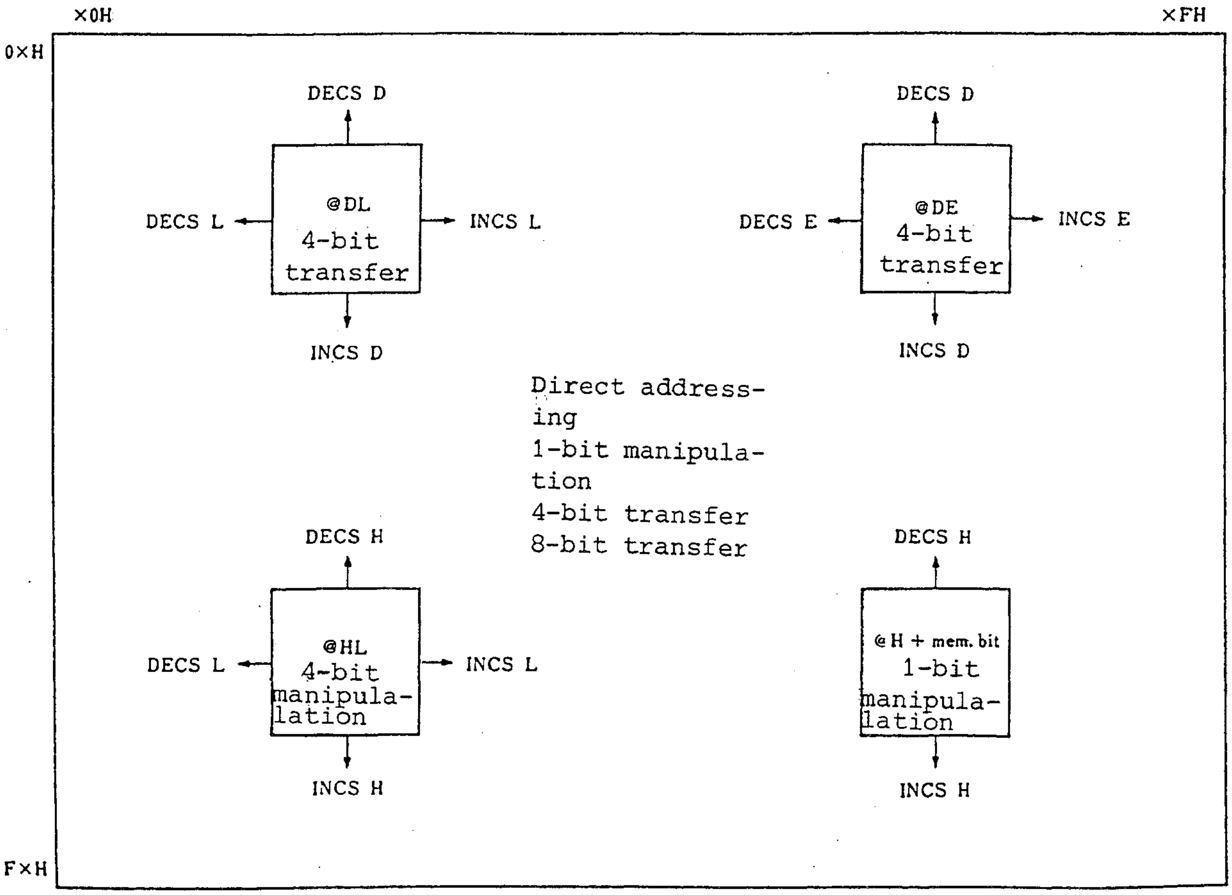
Example 2: Set P33 if P02 is 0.

```
SKT  PORT0.2    ; Skip if port 0 bit 2 is 1
SET1 PORT3.3    ; Set port 3 bit 3
```

Example 3: Output different value to port 6 depending on the P10 state.

```
SKF PORT1.0     ; Skip if port 1 bit 0 is 0
MOV A, #1010B   ; A ← 1010B (string effect)
MOV A, #0101B   ; A ← 0101B (string effect)
SEL MB15        ; or CLR1 MBE
OUT PORT6, A    ; Port 6 ← A
```

Fig. 2-2 Static RAM Address Update Method



2.2 Memory Mapped I/O

The uPD75028(A) adopts memory mapped I/O where peripheral hardware device such as input/output ports and timers are mapped in data memory space addresses F80H-FFFH, as shown in Fig. 2-1. Thus, special instructions to control the peripheral hardware are not provided; the peripheral hardware is controlled by memory operation instructions. (Some hardware control mnemonics for the user easily understand programs are provided.)

Table 2-2 lists the addressing mode that can be used for peripheral hardware operation.

Table 2-2 Addressing Mode Applicable at Peripheral Hardware Operation

	Applicable addressing mode	Applicable hardware
1-bit manipulation	Direct addressing mem.bit by setting MBE = 0 or (MBE = 1 and MBS = 15)	All hardware devices where 1-bit manipulation can be performed
	Direct addressing fmem.bit regardless of MBE or MBS.	IST0, MBE IExxx,IRQxxx,PORTn.x
	Indirect addressing pmem.@L regardless of MBE or MBS.	BSBn.x PORTn.x
4-bit manipulation	Direct addressing mem by setting MBE = 0 or (MBE = 1 and MBS = 15)	All hardware devices where 4-bit manipulation can be performed
	Register indirect addressing @HL by setting (MBE = 1 and MBS = 15)	
8-bit manipulation	Direct addressing mem by setting MBE = 0 or (MBE = 1 and MBS = 15). mem is an even address.	All hardware devices where 8-bit manipulation can be performed
	Register indirect addressing @HL by setting MBE = 1 and MBS = 15. The L register contains an even number.	

Fig. 2-3 shows uPD75028(A) I/O map.

The columns of Fig. 2-3 have the following meanings.

- Symbol: Name indicating the internal hardware address. It can be described in the instruction operand field.
- R/W: Indicates whether or not the hardware device can be read/written.

  R/W: Read and write are enabled

  R: Read only

  W: Write only
- Number of bits that can be manipulated: Indicates the number of bits that can be processed to operate the hardware device.

  o : Operation in the bit units specified under the column (1, 4, or 8 bits) can be performed.

  △ : Only some bits can be manipulated. See Remarks for the bits that can be manipulated.

  - : Operation in the bit units specified under the column (1, 4, or 8 bits) cannot be performed.
- Bit manipulation addressing: Indicates the bit manipulation addressing applicable to the hardware device where 1-bit manipulation is performed.

Fig. 2-3 uPD75028(A) I/O Map (1/4)

Address	Hardware name (symbol)				R/W	Number of bits that can be manipulated			Bit manipulation addressing	Remarks
	b3	b2	b1	b0		1-bit	4-bit	8-bit		
F80H	Stack pointer (SP)				R/W	-	-	○		Bit 0 is fixed to 0.
F85H	Basic interval timer mode register (BTM)				W	△	○	-	mem.bit	Only bit 3 can be manipulated bit-wise.
F86H	Basic interval timer (BT)				R	-	-	○		
F92H	Multifunction timer mode register (MFTM)				W	-	-	○		
F98H	Watch mode register (WM)				R/W	△	-	○	mem.bit	Only bit 3 can be tested bit-wise.
					W	-	-			
F9AH	Multifunction timer control register (MFTC)				R/W	○	○	-	mem.bit	(Note 1)
F9CH	Multifunction timer count register L (MFTL)				R/W	-	-	○		
F9EH	Multifunction timer count register H (MFTH)				R/W	-	-	○		
FA0H	Timer/event counter 0 mode register (TM0)				W	△	-	○	mem.bit	Only bit 3 can be manipulated bit-wise.
						-	-			
FA2H	(Note2) TOE0				W	○	-	-	mem.bit	
FA4H	Timer/event counter 0 count register (T0)				R	-	-	○		
FA6H	Timer/event counter 0 modulo register (TMOD0)				W	-	-	○		

Note 1: Bit 3 (W): In read mode, 0 is read.
Bit 2, 1 (R/W)
Bit 0 (R): In write mode, written data becomes invalid.

2: TOE0: Timer/event counter 0 output enable flag (W)

Fig. 2-3 uPD75028(A) I/O Map (2/4)

Address	Hardware name (symbol)				R/W	Number of bits that can be manipulated			Bit manipulation addressing	Remarks
	b3	b2	b1	b0		1-bit	4-bit	8-bit		
FB0H	0	IST0	MBE	0	R/W	_	o	o	fmem.bit	Only bits 2 and 1 can be manipulated bit-wise.
	Program status word (PSW)				R	-	-			
FB2H	IME				-	-	-	-		Operation by executing EI/DI instruction.
FB3H	Processor clock control register (PCC)				W	-	o	-		
FB4H	INT0 mode register (IM0)				W	-	o	-		Bit 2 is fixed to 0.
FB5H	INT1 mode register (IM1)				W	-	o			Bits 3 to 1 are fixed to 0.
FB6H	INT2 mode register (IM2)				W	-	o			Bits 3 and 2 are fixed to 0.
FB7H	System clock control register (SCC)				W	o	-	-		Bits 2 and 1 are fixed to 0.
FB8H	IE4	IRQ4	IEBT	IRQBT	R/W	o	o	-	fmem.bit	
FBAH			IEW	IRQW	R/W	o	o	-		
FBBH			IEMFT	IRQMFT	R/W	o	o			
FBCH			IET0	IRQT0	R/W	o	o	-		
FBDH			IECSI	IRQCSI	R/W	o	o			
FBEH	IE1	IRQ1	IE0	IRQ0	R/W	o	o	-		
FBFH			IE2	IRQ2	R/W	o	o			
FC0H	Bit sequential buffer 0 (BSB0)				R/W	o	o	o	mem.bit pmem.@L	
FC1H	Bit sequential buffer 1 (BSB1)				R/W	o	o			
FC2H	Bit sequential buffer 2 (BSB2)				R/W	o	o	o		
FC3H	Bit sequential buffer 3 (BSB3)				R/W	o	o			

Remarks 1: IExxx is an interrupt enable flag.
2: IRQxxx is an interrupt request flag.

Fig. 2-3 uPD75028(A) I/O Map (3/4)

Address	Hardware name (symbol)				R/W	Number of bits that can be manipulated			Bit manipulation addressing	Remarks
	b3	b2	b1	b0		1-bit	4-bit	8-bit		
FD0H	Clock output mode register (CLOM)				W	-	○	-		
FD6H	Pull-down resistor specification register group B (PDGB)				W	-	-	○		
FD8H	SOC	EOC			R/W	△	-	○	mem.bit	EOC ... R SOC ... W
	A/D conversion mode register (ADM)				W	-	-			
FDAH	SA register (SA)				R	-	-	○		
FDCH	Pull-up resistor specification register group A (POGA)				W	-	-	○		
FDEH	Pull-up resistor specification register group B (POGB)				W	-	-	○		
FE0H	Serial operation mode register (CSIM)				W	-	-	○		
					R/W	△	○		mem.bit	Only bits 3 to 1 can be manipulated bit-wise.
FE2H	SBI control register (SBIC)				R/W	○	-	-	mem.bit	All bits can be manipulated bit-wise.
FE4H	Serial I/O shift register (SIO)				R/W	-	-	○		
FE6H	Slave address register (SVA)				W	-	-	○		
FE8H	PM33	PM32	PM31	PM30	W	-	-	○		
	Port mode register group A (PMGA)									
	PM63	PM62	PM61	PM60						
FECH	--	PM2	--	--	W	-	-	○		
	Port mode register group B (PMGB)									
	PM7	--	PM5	PM4						
FEEH	__	PM10	PM9	PM8	W	-	-	○		
	Port mode register group C (PMGC)									
	--	--	--	--						

Fig. 2-3 uPD75028(A) I/O Map (4/4)

Address	Hardware name (symbol)				R/W	Number of bits that can be manipulated			Bit manipulation addressing	Remarks
	b3	b2	b1	b0		1-bit	4-bit	8-bit		
FF0H	Port 0 (PORT0)				R	o	o	-	fmem.bit pmem.@L	
FF1H	Port 1 (PORT1)				R	o	o			
FF2H	Port 2 (PORT2)				R/W	o	o	-		
FF3H	Port 3 (PORT3)				R/W	o	o			
FF4H	Port 4 (PORT4)				R/W	o	o	o		
FF5H	Port 5 (PORT5)				R/W	o	o			
(Note) FF6H	KR3	KR2	KR1	KR0	R/W	o	o	o		
	Port 6 (PORT6)									
(Note) FF7H	KR7	KR6	KR5	KR4	R/W	o	o			
	Port 7 (PORT7)									
FF8H	Port 8 (PORT8)				R/W	o	o	-		
FF9H	Port 9 (PORT9)					o	o			
FFAH	Port 10 (PORT10)				R/W	o	o	-		
FFBH	Port 11 (PORT11)				R	o	o			

Note: KR0-KR7 can be read only. Specify PORT6 or PORT7 during 4-bit parallel input.

3. INTERNAL CPU FUNCTION

3.1 Program Counter (PC) ••• 13 bits

The program counter (PC) is a 13-bit binary counter which retains program memory address information.

Fig. 3-1 Program Counter Format

PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PC

When the $\overline{\text{RESET}}$ signal is generated, the low-order five bits of program memory address 0000H are set in PC12-PC8 and the contents of address 0001H are set in PC7-PC0 for initialization.

3.2 Program Memory (ROM) ••• 8064 words x 8 bits

The program memory is 8064-word x 8-bit mask programmable ROM which stores programs, table data, etc.

The program memory is addressed by the program counter. Table data can be referenced by executing a table look-up instruction (MOVT).

Fig. 3-2 shows the branch-possible address range when a branch instruction or subroutine call instruction is executed. Relative branch instruction (BR $addr) enables a branch to address [PC contents -15 to -1, +2 to +16] independently of block boundaries.

Program memory addresses are 0000H-1F7FH; the following addresses are specially assigned: (Every area except for 0000H or 0001H can be used as normal program memory.)

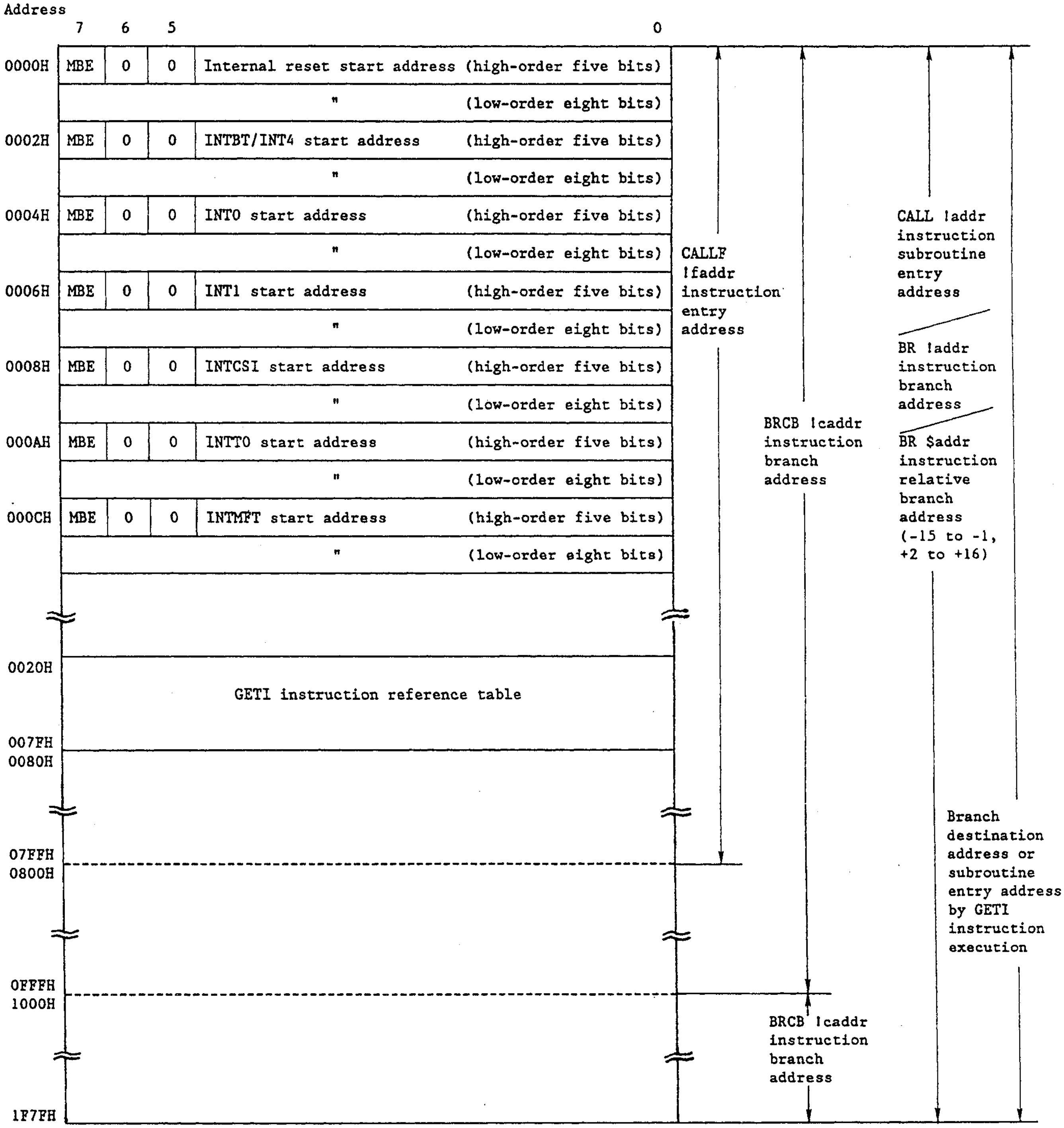
- Addresses 0000H-0001H
  Vector table into which the program start address and MBE setup value are written when the $\overline{\text{RESET}}$ signal is generated. Reset start is enabled at any desired address.

- Addresses 0002H-000DH
  Vector table into which the program start address and MBE setup value are written when a vectored interrupt occurs. Interrupt processing can be started at any desired address.

- Addresses 0020H-007FH
  Table area referenced by executing GETI instruction$^{\text{(Note)}}$.

Note: The GETI instruction enables execution of any 2-byte or 3-byte instruction or two 1-byte instruction as 1-byte instruction; it is used to reduce the number of program steps. (See 8.1.)

Fig. 3-2 Program Memory Map



Address	7	6	5	0
0000H	MBE	0	0	Internal reset start address (high-order five bits)
				" (low-order eight bits)
0002H	MBE	0	0	INTBT/INT4 start address (high-order five bits)
				" (low-order eight bits)
0004H	MBE	0	0	INT0 start address (high-order five bits)
				" (low-order eight bits)
0006H	MBE	0	0	INT1 start address (high-order five bits)
				" (low-order eight bits)
0008H	MBE	0	0	INTCSI start address (high-order five bits)
				" (low-order eight bits)
000AH	MBE	0	0	INTT0 start address (high-order five bits)
				" (low-order eight bits)
000CH	MBE	0	0	INTMPT start address (high-order five bits)
				" (low-order eight bits)
0020H 007FH				GETI instruction reference table
0080H 07FFH				
0800H 0FFFH				
1000H 1F7FH				

CALLF !faddr instruction entry address
BRCB !caddr instruction branch address
CALL !addr instruction subroutine entry address
BR !addr instruction branch address
BR $addr instruction relative branch address (-15 to -1, +2 to +16)
Branch destination address or subroutine entry address by GETI instruction execution
BRCB !caddr instruction branch address

3.3 Data Memory (RAM)

(1) Data area

The uPD75028(A) data area consists of 512-word x 4-bit static RAM and is used to store process data. It is handled by executing memory handling instructions.

The static RAM is mapped each 256 x 4 bits in memory banks 0 and 1. Bank 0 is mapped as a data area; it can also be used as a general purpose register area (000H-007H) and a stack area (000H-0FFH). (Bank 1 can only be used as a data area.)

One address of the static RAM consists of four bits; however it can be handled in 8-bit units by executing 8-bit memory handling instructions and bit-wise by executing bit manipulation instructions. To execute an 8-bit memory handling instruction, specify an even address.

- General purpose register area
  Can be handled by executing general purpose register and memory handling instructions.
  A maximum of eight 4-bit registers can be used. The portions of the eight general purpose registers, not used by a program, can be used as a data area or stack area. (See 3.4)

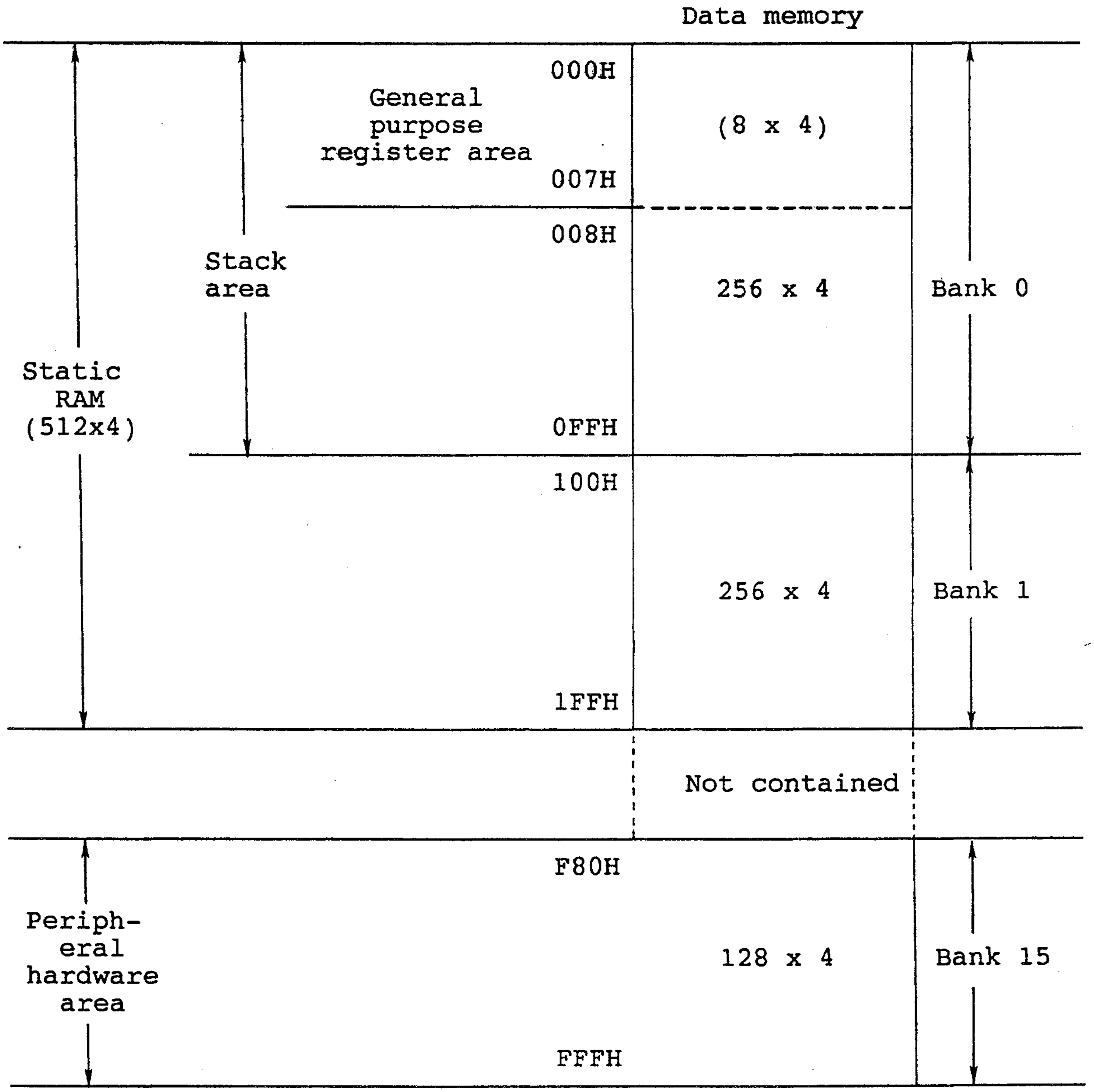
- Stack area
  Is set by an instruction and can be used as a save area when a subroutine is executed or interrupt processing is performed. (See 3.6.)

(2) Peripheral hardware area

The peripheral hardware area is mapped in addresses F80H-FFFH of memory bank 15.

Like the static memory, the peripheral hardware area is handled by executing memory handling instructions. However, the bit units in which the peripheral hardware can be handled vary depending on the address. Addresses in which the peripheral hardware is not mapped do not contain data memory and cannot be accessed. (See Fig. 2-3.)

Fig. 3-3 Data Memory Map



3.4 General Purpose Registers ••• 8 x 4 bits

The general purpose registers are mapped in specific addresses of the data memory. They are eight 4-bit registers (B, C, D, E, H, L, X, and A).

The general purpose registers are handled each in 4-bit units; register pairs BC, DE, HL, and XA are used for 8-bit handling. In addition to DE and HL, register pair DL can also be used; these three register pairs can be used as data pointers.

The general purpose register area can be addressed as normal RAM for an access regardless of whether or not it is used as register.

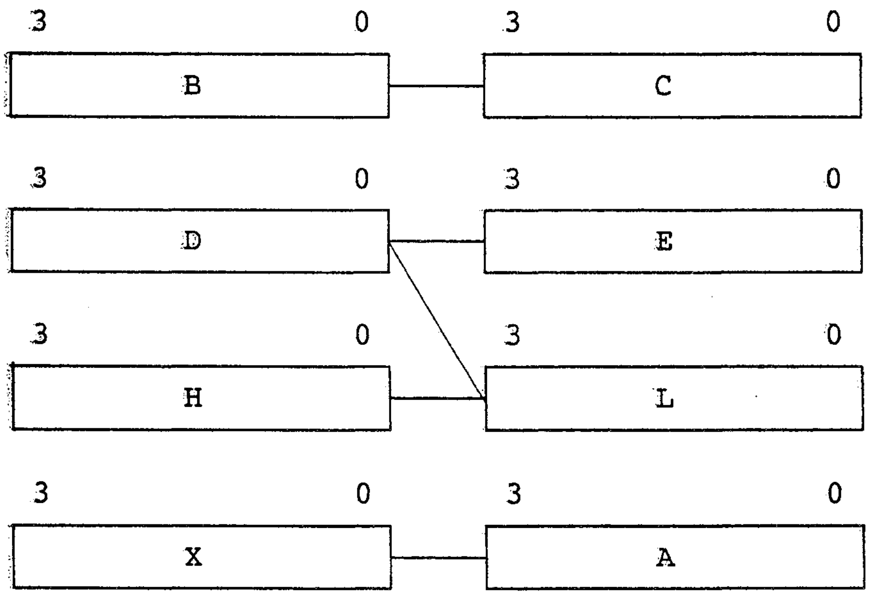
Fig. 3-4 General Purpose Register Configuration
(for 4-bit processing)

X	01H	A	00H
H	03H	L	02H
D	05H	E	04H
B	07H	C	06H

Fig. 3-5 General Purpose Register Configuration
(for 8-bit processing)

XA	00H
HL	02H
DE	04H
BC	06H

Fig. 3-6 Register Pair Configuration

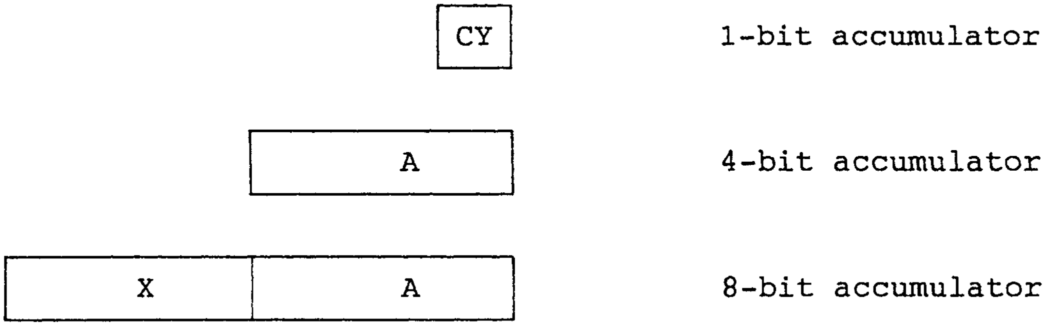


3.5 Accumulators

On the uPD75028(A), the A register and XA register pair function as accumulators. A 4-bit data processing instruction is executed centering around the A register; an 8-bit data processing instruction is executed centering around the XA register pair.

In bit manipulation instruction execution, the carry flag (CY) functions as a 1-bit accumulator.

Fig. 3-7 Accumulators



3.6 Stack Pointer (SP) ••• 8-bits

On the uPD75028(A), static RAM is used as stack memory (LIFO). The stack pointer (SP) is an 8-bit register which retains the top address of the stack area.

The stack area is addresses 000H-0FFH of memory bank 0 independently of how MBE and MBS are set.

SP is decremented before write (save) operation into (in) the stack memory and is incremented after read (restore) operation from the stack memory.

Figs. 3-9 and 3-10 show how data is saved in and restored from the stack memory.

An initial value is set in SP by executing an 8-bit memory handling instruction to determine the stack area. The contents can also be read.

"0" is always written into SP0.

The initial value of SP should be set to 00H for use as stack area from the highest address of data memory bank 0 (0FFH).

When the $\overline{\text{RESET}}$ signal is generated, the SP content become undefined. Be sure to initialize SP to any desired value at the beginning of a program.

```
Example:  Initialize SP
          SEL MB15        ; or CLR1 MBE
          MOV XA, #00H
          MOV SP, XA      ; SP ← 00H
```

Fig. 3-8 Stack Pointer Format

Address	7	6	5	4	3	2	1	0	Symbol
F80H	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SP

Fixed to 0

Fig. 3-9 Data Saved in Stack Memory

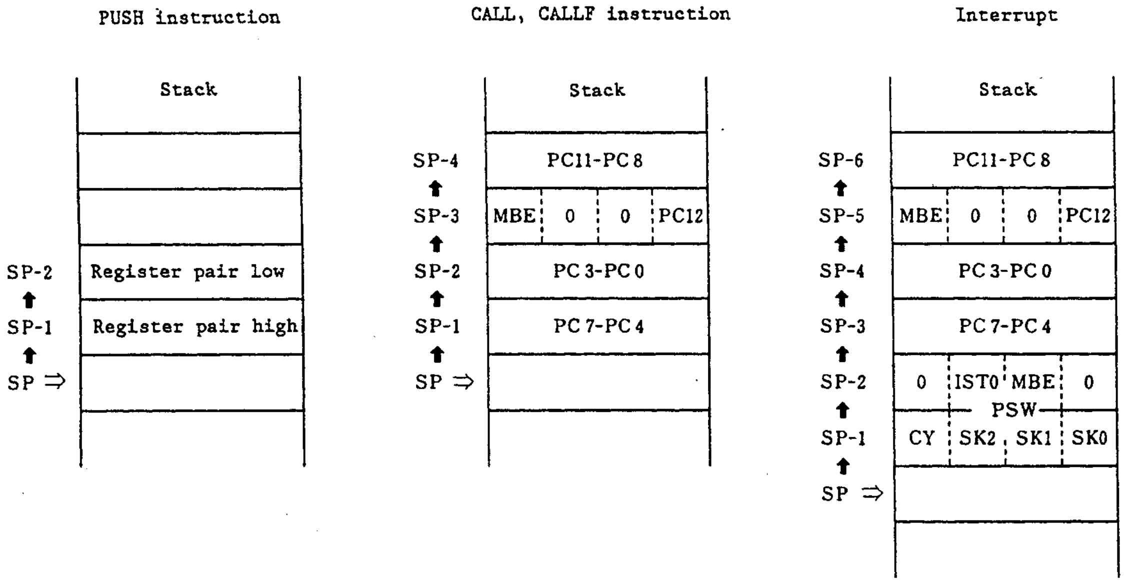
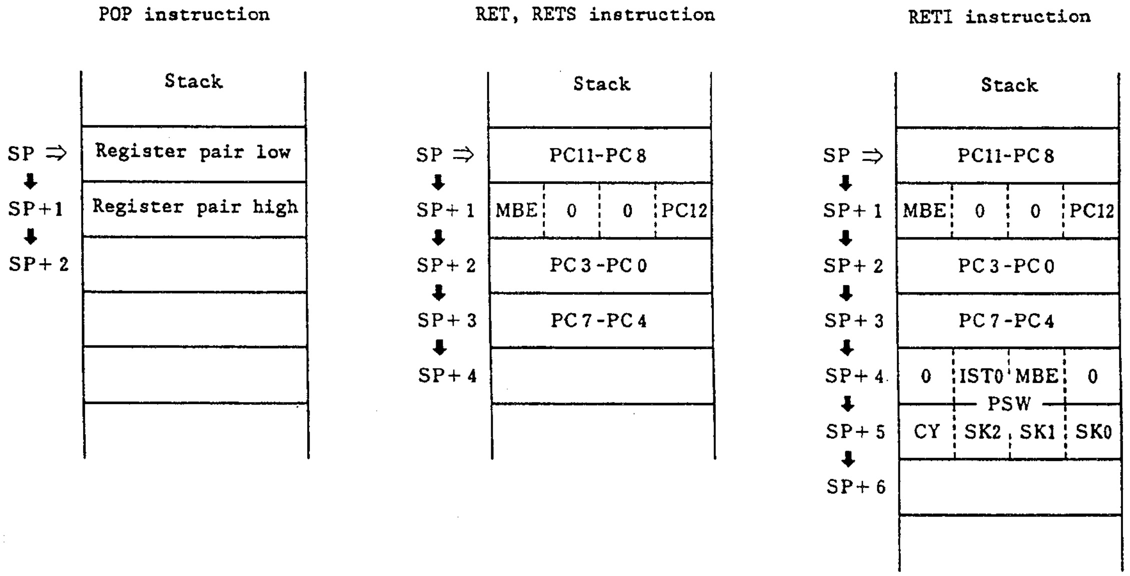


Fig. 3-10 Data Restored from Stack Memory



3.7 Program Status Word (PSW) ••• 8 bits

The program status word (PSW) consists of flags related closely to processor operation.

PSW is mapped in data memory space addresses FB0H and FB1H and two bits of address FB0H can be handled by executing a memory handling instruction.

Fig. 3-11 Program Status Word Format

Address	← (FB1H) →				← FB0H →				Symbol
FB0H	CY	SK2	SK1	SK0	0	IST0	MBE	0	PSW

CY: Can be handled by executing a dedicated instruction

SK2, SK1, SK0: Cannot be handled

IST0, MBE: Can be handled

Table 3-1 Saved or Restored PSW Flags during Stack Operation

		Flag to be saved or restored
Save	When CALL or CALLF instruction is executed	Save MBE
	When a hardware interrupt is executed	Save all PSW bits
Restore	When RET or RETS instruction is executed	Restore MBE
	When RETI instruction is executed	Restore all PSW bits

(1) Carry flag (CY)

The carry flag is a 1-bit flag which stores overflow or underflow information during execution of an instruction with carry (ADDC or SUBC).

The carry flag also serves as a 1-bit accumulator; Boolean algebra operation between the 1-bit accumulator and data memory bit specified by a bit address can be performed and the result can be stored in the 1-bit accumulator.

The carry flag is handled by executing a dedicated instruction independently of other PSW bits.

When the $\overline{\text{RESET}}$ signal is generated, the carry flag become undefined.

Table 3-2 Carry Flag Handling Instructions

	Instruction (mnemonics)	Carry flag operation or processing
Carry flag handling dedicated instructions	SET1 CY CLR1 CY NOT1 CY SKT CY	Set CY to 1 Clear CY Invert the CY contents Skip if CY contains 1
Bit Boolean instructions	AND1 CY, mem*.bit OR1 CY, mem*.bit XOR1 CY, mem*.bit	AND, OR, exclusive-OR the contents of the specified bit and the CY contents together and set the result in CY
Interrupt processing	When an interrupt is executed	Save CY and other PSW bits in parallel in stack memory
	RETI	Restore CY and other PSW bits in parallel from stack memory

Remarks: mem*.bit denotes any of the following three bit manipulation addressing modes:

. fmem.bit

. pmem.@L

. @H+mem.bit

Example: AND bit 3 of address 3FH and P33 together and set the result in CY.

```
SET1 CY            ; CY  ←  1
CLR1 MBE           ; or SEL MB15
SKT  3FH. 3        ; Skip if Bit3 of Address 3FH is 1
CLR1 CY            ; CY  ←  0
AND1 CY, PORT3.3   ; CY  ←  CY ∧ P33
```

(2) Skip flags (SK2, SK1, and SK0)

The skip flags store the skip state and are automatically set or reset by the CPU which executes an instruction.

(3) Interrupt status flag (IST0)

The interrupt status flag stores the status of the current processing being performed. (For details, see Table 3-3.)

Table 3-3 Interrupt Status Flag Indication

IST0	Status of processing being performed	Processing and interrupt control
0	Status 0	During normal program processing. Every interrupt can be acknowledged.
1	Status 1	During interrupt processing. Acknowledgement of every interrupt is disabled.

If an interrupt is acknowledged, the IST0 contents are saved in the stack memory as a part of PSW, then automatically set to 1. When the RETI instruction is executed, IST0 is reset to 0. The interrupt status flag can be handled by executing a memory handling instruction, and the status of processing being executed can also be changed under the program control.

Caution: To handle the interrupt status flag, be sure to execute DI instruction to disable interrupts before handling the flag, and execute EI instruction to enable interrupts after handling the flag.

(4) Memory bank enable flag (MBE)

The memory bank enable flag is a 1-bit flag to specify the address information generation mode of the high-order four bits of a 12-bit data memory address.

When the flag is set to 1, the data memory address space is expanded and every data memory space can be addressed.

When the flag is reset to 0, the data memory address space is fixed regardless of how MBS is set. (See Fig. 2-1.)

When the $\overline{\text{RESET}}$ signal is generated, the bit 7 contents of program memory address 0 is set for automatic initialization.

When a vectored interrupt is processed, the bit 7 contents of the corresponding vector address table are set and the MBE state during the interrupt service is automatically set.

Normally, static RAM of memory bank 0 is used by setting MBE to 0 in interrupt processing.

3.8 Bank Selection Register (BS)

The memory bank selection register (MBS) to specify a memory bank is mapped in the bank selection register (BS). The low-order four bits are fixed to 0.

MBS is set by executing SEL MBn instruction.

BS can be saved in and restored from a stack area in 8-bit units by executing PUSH BS and POP BS instructions.

Fig. 3-12 Bank Selection Register Format

MBS								Symbol
MBS3	MBS2	MBS1	MBS0	0	0	0	0	BS

(1) Memory bank selection register (MBS)

The memory bank selection register is a 4-bit register which stores high-order 4-bit address information of a 12-bit data memory address. The memory bank to be accessed is specified by the register contents. However, only banks 0, 1, and 15 can be specified on the uPD75028(A).

MBS is set by executing SEL MBn instruction (n = 0, 1, or 15).

The address ranges that can be specified according to how MBE and MBS are set are as shown in Fig. 2-1.

When the $\overline{\text{RESET}}$ signal is generated, MBS is initialized to 0.

4. PERIPHERAL HARDWARE FUNCTION

4.1 Digital Input/Output Ports

The uPD75028(A) adopts memory mapped I/O where all input/output ports are mapped in data memory space.

Fig. 4-1 Data Memory Addresses of Digital Ports

Address	3	2	1	0	
FF0H	P03	P02	P01	P00	PORT 0
FF1H	P13	P12	P11	P10	PORT 1
FF2H	P23	P22	P21	P20	PORT 2
FF3H	P33	P32	P31	P30	PORT 3
FF4H	P43	P42	P41	P40	PORT 4
FF5H	P53	P52	P51	P50	PORT 5
FF6H	P63	P62	P61	P60	PORT 6
FF7H	P73	P72	P71	P70	PORT 7
FF8H	P83	P82	P81	P80	PORT 8
FF9H	P93	P92	P91	P90	PORT 9
FFAH	P103	P102	P101	P100	PORT 10
FFBH	P113	P112	P111	P110	PORT 11

Table 4-1 lists the input/output port handling instructions. In addition to 4-bit input/output instructions, 8-bit input/output and 1-bit manipulation instructions can be used for ports 4-7 for control in very various manners.

Example 1: Test the P13 state and output a different value to port 4, 5 depending on the result.

```
    SKT PORT1. 3    ; Skip if port 1 bit 3 is 1
    MOV XA, #18H    ; XA  ←  18H ⎫ String effect
    MOV XA, #14H    ; XA  ←  14H ⎭
    SEL MB15        ; Or CLR1 MBE
    OUT PORT4, XA   ; Port 5, 4  ←  XA
2:  SET1 PORT4, @L  ; Set the bit of PORT4-7 addressed
                      by the L register to 1
```

Table 4-1 Input/Output Pin Handling Instruction List

<table>
<tr><th></th><th>PORT
0</th><th>PORT
1</th><th>PORT
2</th><th>PORT
3</th><th>PORT
4</th><th>PORT
5</th><th>PORT
6</th><th>PORT
7</th><th>PORT
8</th><th>PORT
9</th><th>PORT
10</th><th>PORT
11</th></tr>
<tr><td>IN A, PORTn (Note 1)</td><td colspan="12">o</td></tr>
<tr><td>IN XA, PORTn (Note 1)</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">o</td><td colspan="2">o</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>OUT PORTn, A (Note 1)</td><td>—</td><td>—</td><td colspan="9">o</td><td>—</td></tr>
<tr><td>OUT PORTn. XA (Note 1)</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">o</td><td colspan="2">o</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>SET1 PORTn. bit</td><td>—</td><td>—</td><td colspan="9">o</td><td>—</td></tr>
<tr><td>SET1 PORTn. @L (Note 2)</td><td>—</td><td>—</td><td colspan="9">o</td><td>—</td></tr>
<tr><td>CLR1 PORTn. bit</td><td>—</td><td>—</td><td colspan="9">o</td><td>—</td></tr>
<tr><td>CLR1 PORTn. @L (Note 2)</td><td>—</td><td>—</td><td colspan="9">o</td><td>—</td></tr>
<tr><td>SKT PORTn. bit</td><td colspan="12">o</td></tr>
<tr><td>SKT PORTn. @L (Note 2)</td><td colspan="12">o</td></tr>
<tr><td>SKF PORTn. bit</td><td colspan="12">o</td></tr>
<tr><td>SKF PORTn. @L (Note 2)</td><td colspan="12">o</td></tr>
<tr><td>AND1 CY, PORTn. bit</td><td colspan="12">o</td></tr>
<tr><td>AND1 CY, PORTn. @L (Note 2)</td><td colspan="12">o</td></tr>
<tr><td>OR1 CY, PORTn. bit</td><td colspan="12">o</td></tr>
<tr><td>OR1 CY, PORTn. @L (Note 2)</td><td colspan="12">o</td></tr>
<tr><td>XOR1 CY, PORTn. bit</td><td colspan="12">o</td></tr>
<tr><td>XOR1 CY, PORTn. @L (Note 2)</td><td colspan="12">o</td></tr>
</table>

Note 1: Preset MBE = 0 or (MBE = 1 and MBS = 15) before executing the instruction.

Note 2: The low-order two bits of address and bit address are indirectly specified in the L register.

(1) Types, features, and structures of digital input/output ports

Table 4-2 lists the digital input/output port types. Fig. 4-2 to 4-8 show the structures of the ports.

Table 4-2 Digital Port Types and Features

Port (symbol)	Function	Operation and features	Remarks
PORT0 PORT1	4-bit input	Can always be read or tested regardless of the dual function pin operation mode.	Also used for SO/SB0, SI/SB1, $\overline{SCK}$, INT0-2, 4 and TI0. (See Chapter 1.)
(Note) PORT3 PORT6	4-bit input/output	Input or output mode can be set bit-wise.	Port 6 pins are also used for KR0-3.
PORT2 PORT7		Input or output mode can be set in 4-bit units. Ports 6 and 7 can be paired for data input/output in 8-bit units.	Port 2 pins are also used for PTO0, PPO, PCL, and BUZ.
			Also used for KR4-7.
(Note) PORT4 (Note) PORT5 (Note) PORT10	4-bit input/output (N-ch open drain 10-V voltage)	Input or output mode can be set in 4-bit units. Ports 4 and 5 can be paired for data input/output in 8-bit units.	Internal pull-up resistor can be specified bit-wise by mask option.
			Port 10 pins are also used for $\overline{MAR}$, $\overline{MAI}$, $\overline{MAZ}$, and $\overline{MAT}$.
PORT8 PORT9	4-bit input/output	Input or output mode can be set in 4-bit units.	
PORT11	4-bit input	4-bit input-only port.	Port 11 pins are also used for AN0-3.

Note: LED can be directly driven.

P10 is also used for an external vectored interrupt input pin. It is an input pin with noise remover. (For details, see 5.2.)

(2) Input/output mode setting

The input or output mode of each input/output port is set in the port mode register as shown in Fig. 4-9.

When a port mode register bit is set to 0, its corresponding port serves as an input port; when 1, its corresponding port serves as an output port.

Port mode register groups A, B, and C are set by executing 8-bit memory handling instructions. When the $\overline{\text{RESET}}$ signal is generated, all bits of the port mode registers are reset to 0, turning off the output buffers and placing all ports in the input mode.

Fig. 4-2 Port 0, 1 Structure

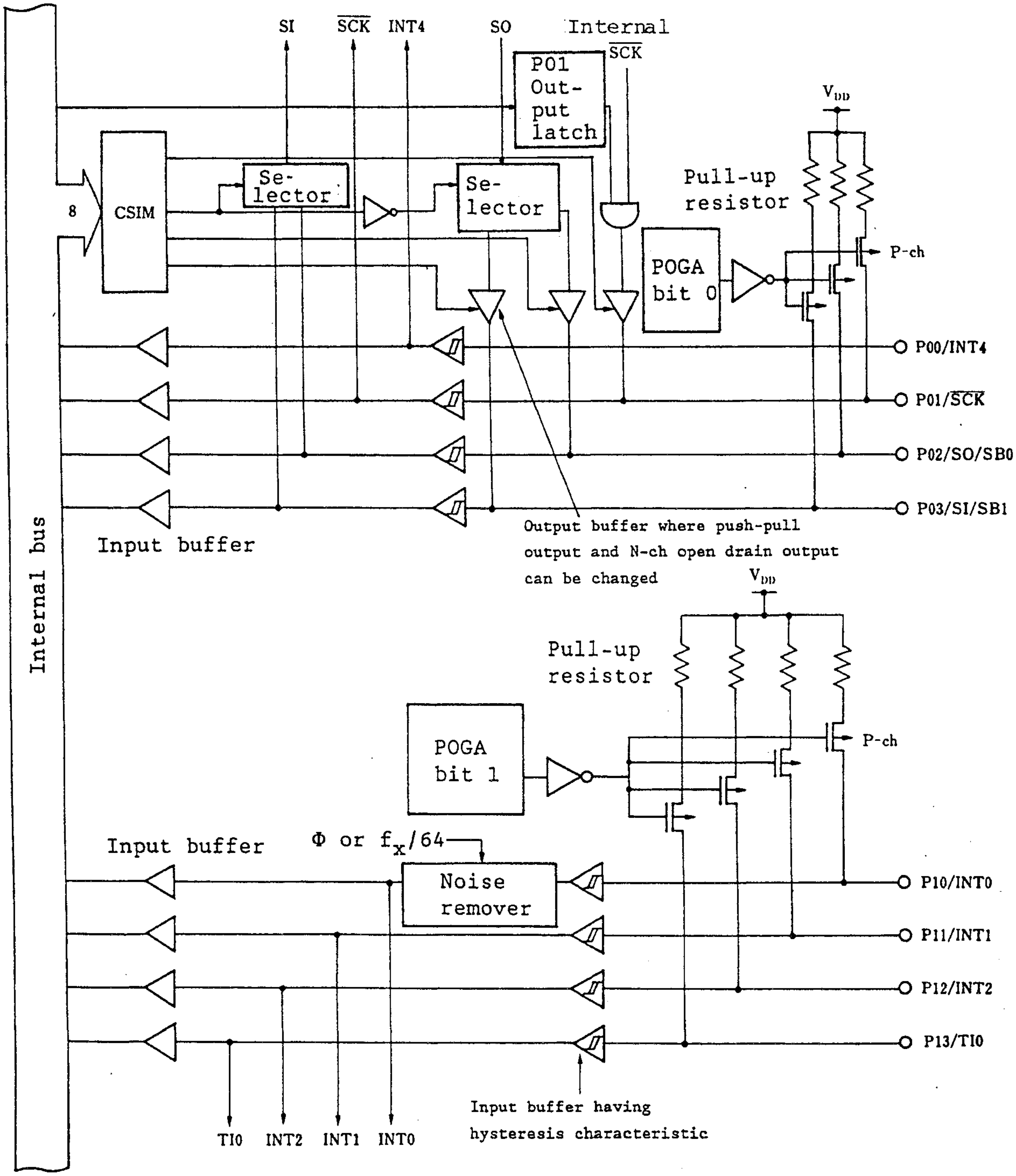


Fig. 4-3 Port 3n, Port 6n Structure (n = 0-3)

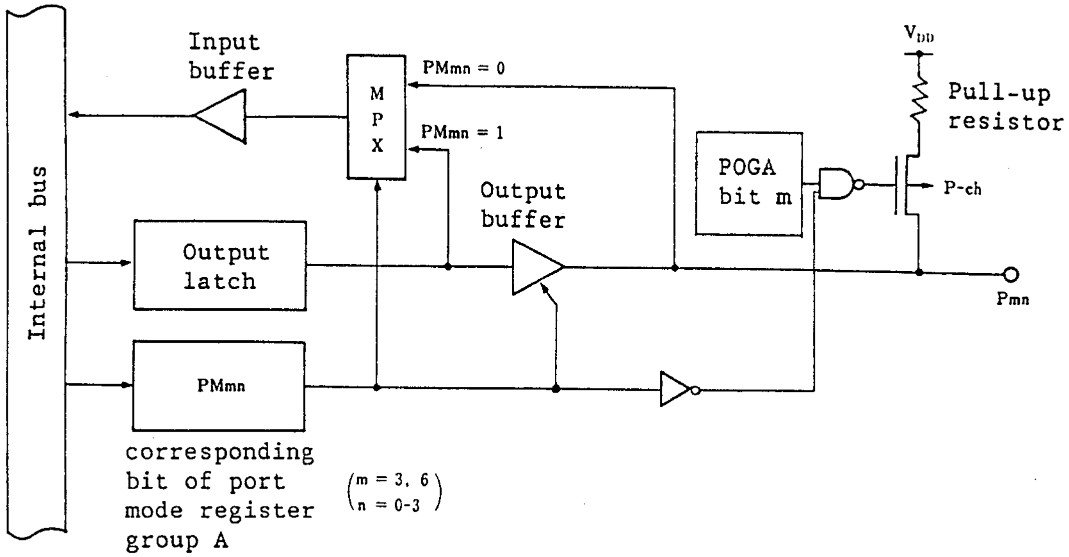


Fig. 4-4 Port 2, 7, 8 Structure

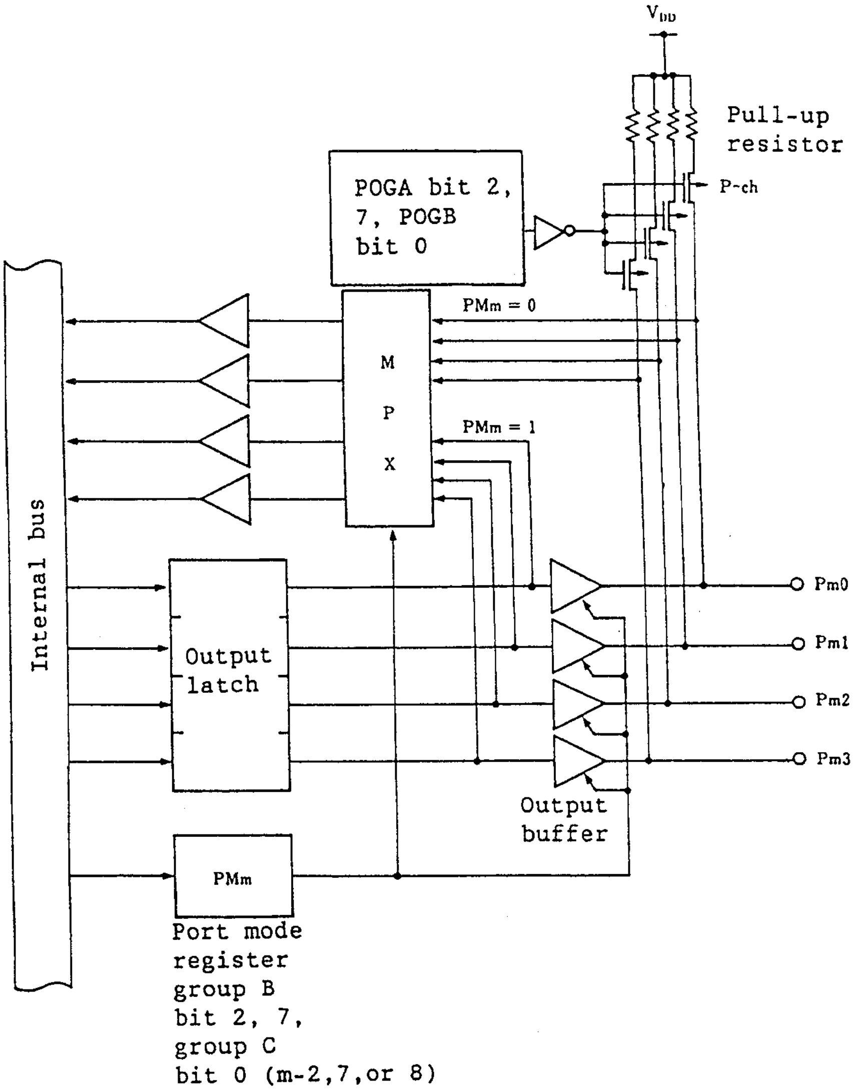


Fig. 4-5 Port 4, 5 Structure

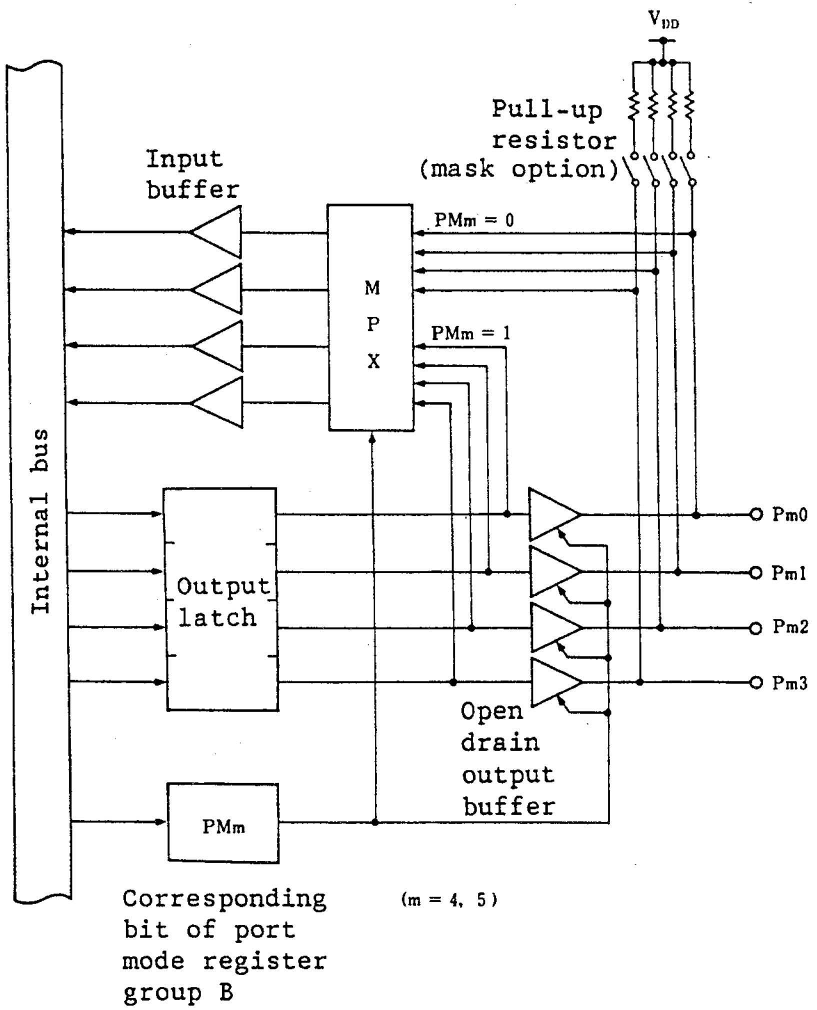


Fig. 4-6 Port 9 Structure

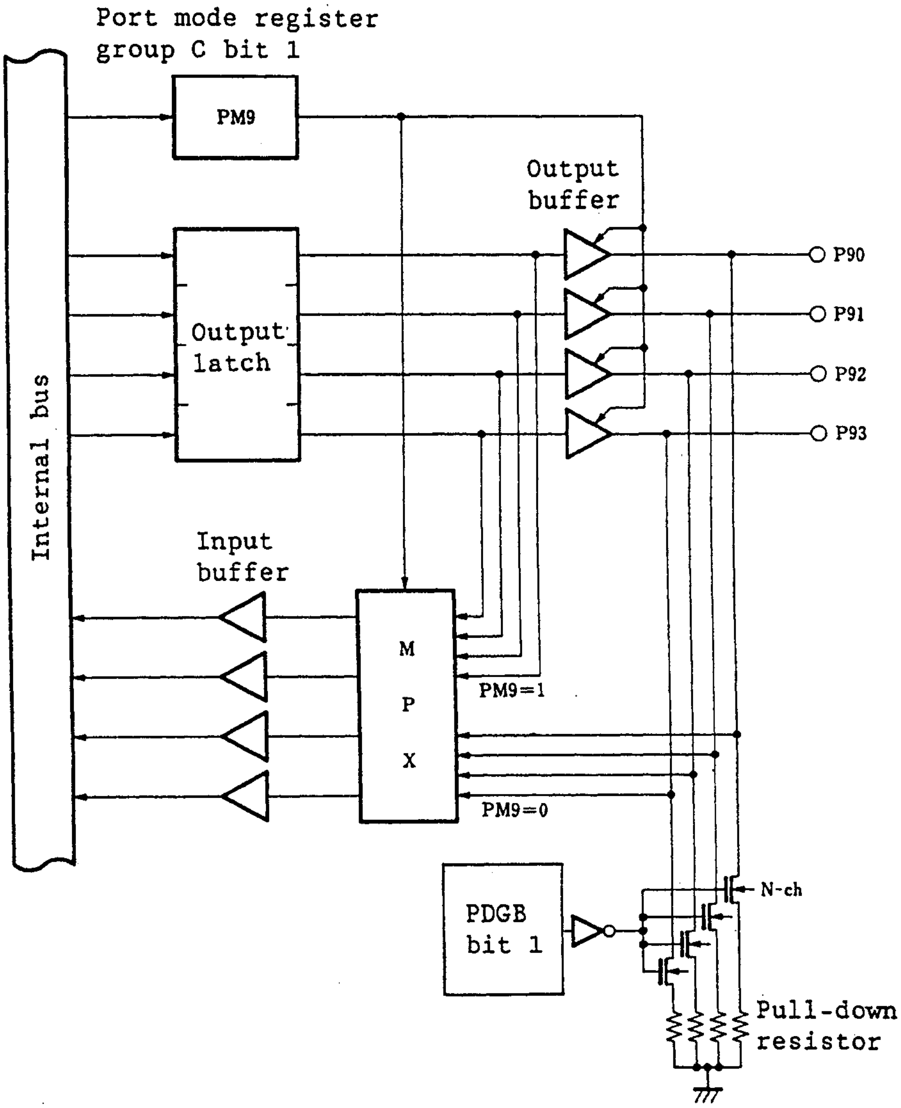


Fig. 4-7 Port 10 Structure

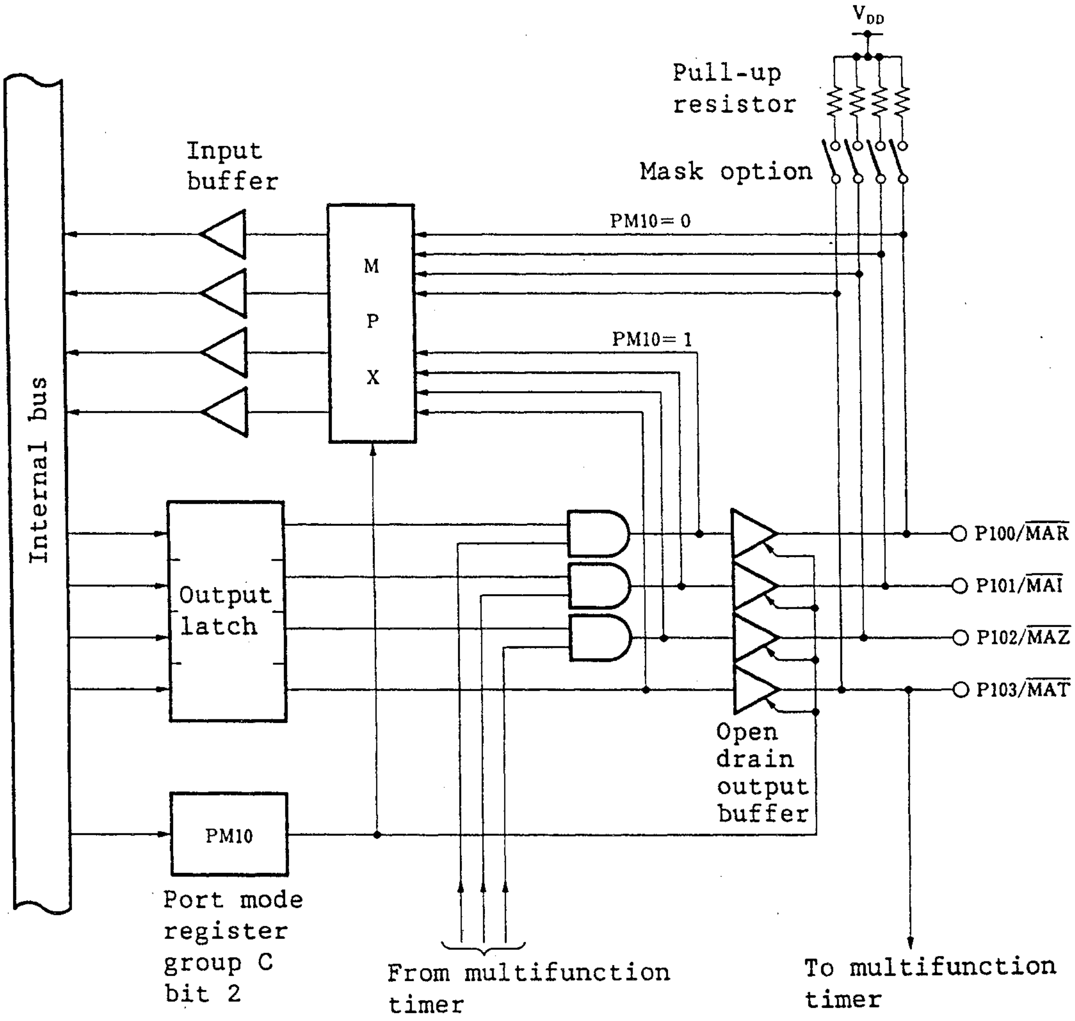


Fig. 4-8 Port 11 Structure

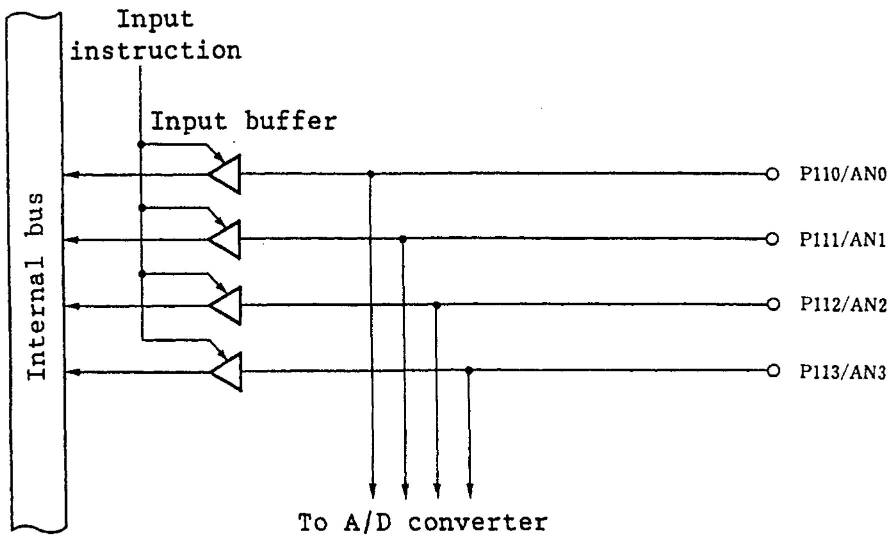


Fig. 4-9 Port Mode Register Formats

Port mode register group A

Address	7	6	5	4	3	2	1	0	Symbol
FE8H	PM63	PM62	PM61	PM60	PM33	PM32	PM31	PM30	PMGA

- I/O mode specification for P30
- I/O mode specification for P31
- I/O mode specification for P32
- I/O mode specification for P33
- I/O mode specification for P60
- I/O mode specification for P61
- I/O mode specification for P62
- I/O mode specification for P63

Port mode register group B

Address	7	6	5	4	3	2	1	0	Symbol
FECH	PM7	—	PM5	PM4	—	PM2	—	—	PMGB

- I/O mode specification for Port 2 (P20-P23)
- I/O mode specification for Port 4 (P40-P43)
- I/O mode specification for Port 5 (P50-P53)
- I/O mode specification for Port 7 (P70-P73)

Port mode register group C

Address	7	6	5	4	3	2	1	0	Sybmol
FEEH	—	—	—	—	—	PM10	PM9	PM8	PMGC

- I/O mode specification for Port 8 (P80-P83)
- I/O mode specification for Port 9 (P90-P93)
- I/O mode specification for Port 10 (P100-P103)

	Specification
0	Input mode (output buffer off)
1	Output mode (output buffer on)

(3) Digital input/output port operation

Port pin operation during instruction execution varies depending on which mode of input and output is set, as listed in Table 4-3.

Table 4-3 Input/Output Port Operation when Input/Output Instruction is Executed

	Input mode [corresponding bit of mode register is set to 0] [output buffer off]	Output mode [corresponding bit of mode register is set to 1] [output buffer on]
When a 1-bit test, 1-bit input, or 4/8 bit input instruction is executed	Pin data is input	Output latch contents are input
When a 4/8 bit output instruction is executed	Accumulator data is transferred to output latch	Accumulator data is output to output pin
(Note) When a 1-bit output instruction is executed	Output latch contents become undefined	Output pin state is changed according to instruction

Note: Instruction such as SET1/CLR1, PORTn.bit, or CY

(4) Internal pull-up or pull-down resistor specification

Each uPD75028(A) port pin except P00, port 11 can contain a pull-up or pull-down resistor. An internal pull-up resistor is specified by software or mask option depending on the pin.

Port 9 can contain pull-down resistor.

For ports 3 and 6, internal pull-up resistor can be specified only for the pins set to the input mode. The pins set to the output mode do not contain pull-up resistor regardless of how POGA is set.

Fig. 4-10 Pull-up and Pull-down Resistor Specification Register Formats (1/2)

Pull-up resistor register group A

Address	7	6	5	4	3	2	1	0	Symbol
FDCH	PO7	PO6	——	——	PO3	PO2	PO1	PO0	POGA

Port 0 (P01-P03)
Port 1 (P10-P13)
Port 2 (P20-P23)
Port 3 (P30-P33)

Port 6 (P60-P63)
Port 7 (P70-P73)

Pull-up resistor register group B

Address	7	6	5	4	3	2	1	0	Symbol
FDEH	——	——	——	——	——	——	——	PO8	POGB

Port 8 (P80-P83)

	Specification
0	Pull-down resistor is not contained
1	Pull-down resistor is contained

Fig. 4-10 Pull-up and Pull-down Resistor Specification Register Formats (2/2)

Pull-down resistor register group B

Address	7	6	5	4	3	2	1	0	Symbol
FD6H	—	—	—	—	—	—	PD9	—	PDGB

PD9 — Port 9 (P90-P93)

	Specification
0	Pull-down resistor is not contained
1	Pull-down resistor is contained

Table 4-4 Specification of Internal Pull-up or Pull-down Resistor

Port (pin names)	Specification of internal pull-up or pull-down resistor	Specification bit
(Note 1) Port 0 (P01-P03)	Internal pull-up resistor specification in 3-bit units by software	POGA. 0
Port 1 (P10-P13)	Internal pull-up resistor specification in 4-bit units by software.	POGA. 1
Port 2 (P20-P23)		POGA. 2
Port 3 (P30-P33)		POGA. 3
Port 6 (P60-P63)		POGA. 6
Port 7 (P70-P73)		POGA. 7
Port 8 (P80-P83)		POGB. 0
Port 9 (P90-P93)	Internal pull-down resistor specification in 4-bit units by software	PDGB. 1
Port 4 (P40-P43)	Internal pull-up resistor specification bit-wise by mask option	—
Port 5 (P50-P53)		
Port 10 (P100-P103)		

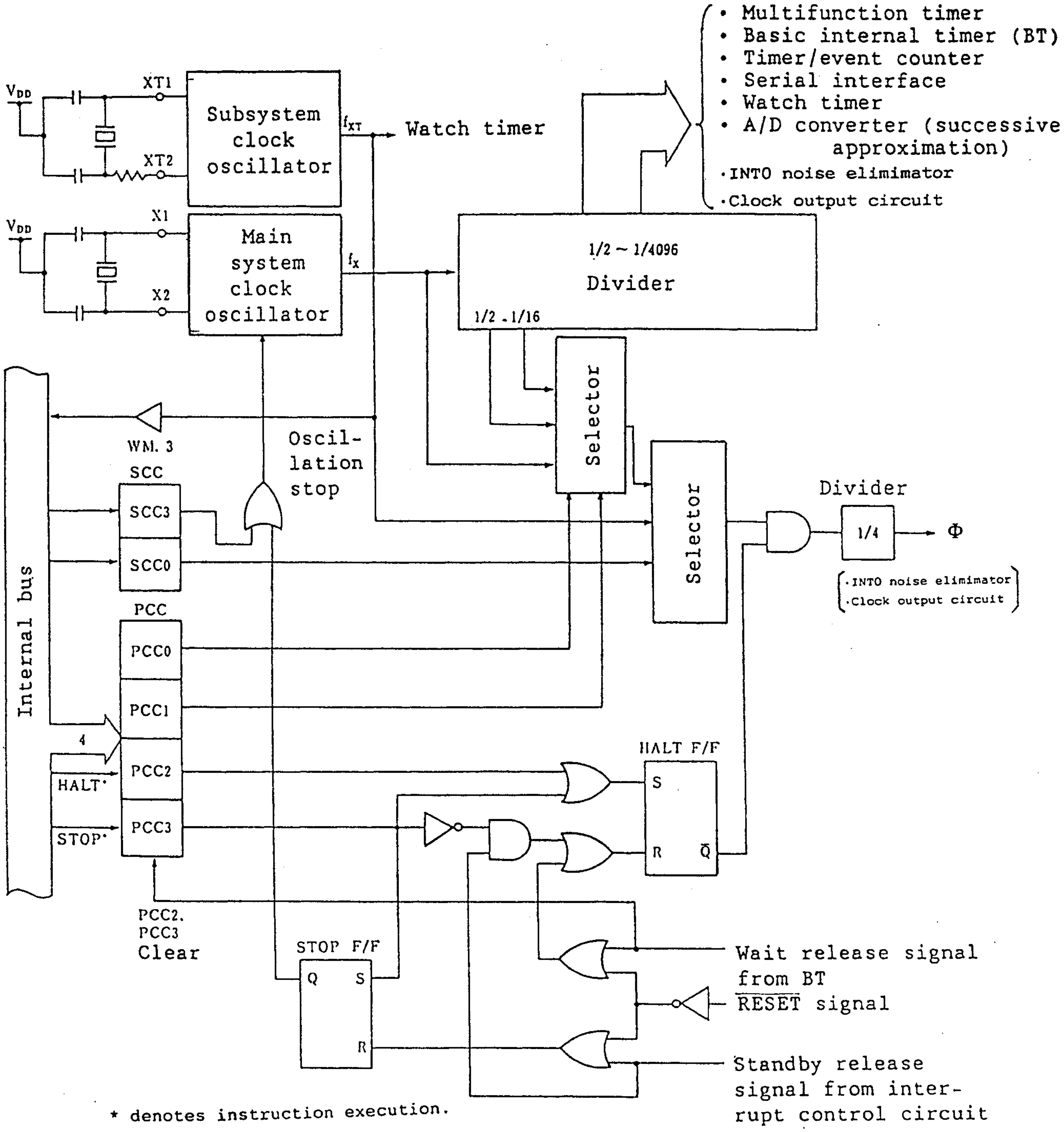
Note: The P00 pin cannot contain a pull-up resistor.

4.2 Clock Generator

(1) Clock generator configuration

The clock generator generates clocks supplied to the CPU and peripheral hardware. Fig. 4-11 shows the clock generator block diagram.

Fig. 4-11 . Clock Generator Block Diagram



* denotes instruction execution.

Remarks 1: f_X = main system clock frequency

2: f_{XT} = subsystem clock frequency

3: Φ = CPU clock

4: PCC: Processor clock control register

5: SCC: System clock control register

6: One clock cycle of (t_{CY})is equal to 1 machine cycle of an instruction. For the t_{CY}, refer to AC characteristics in 9. ELECTRICAL SPECIFICATIONS.

(2) Clock generator function

The clock generator generates the following clocks and controls the CPU operation mode such as the standby mode:

. main system clock f_X
. subsystem clock f_{XT}
. CPU clock Φ
. clock to peripheral hardware

Clock generator function and operation are determined by the processor clock control register (PCC) and system clock control register (SCC), as described below:

(a) When the $\overline{\text{RESET}}$ signal is generated, the lowest speed mode of the main system clock (15.3 us at 4.19 MHz) is selected. (PCC = 0 and SCC = 0)

(b) When the main system clock is selected, one of three CPU clock speeds can be selected by setting PCC. (0.95, 1.91, or 15.3 us at 4.19 MHz)

(c) When the main system clock is selected, the standby mode STOP or HALT can be used.

(d) Subsystem clock can be selected by setting SCC for operation at very low speed and low consumption current (122 us at 32.768 kHz). In this case, the value set in PCC does not affect CPU clock.

(e) When the subsystem clock is selected, main system clock oscillation can be stopped by setting SCC. The HALT mode can also be used. However, the STOP mode cannot be used (subsystem clock oscillation cannot be stopped).

(f) Clock is supplied to the peripheral hardware by dividing the main system clock, but the subsystem clock can be directly supplied only to the watch timer. Thus, the watch function and buzzer output function can be continued even in the standby mode.

(g) When the subsystem clock is selected, normal operation of the watch timer can be continued. However, other hardware devices operate based on the main system clock, thus cannot be used when the main system clock is stopped.

(3) Processor clock control register (PCC)

The processor clock control register (PCC) is a 4-bit register whose low-order two bits are used for CPU clock Φ selection and high-order two bits are used for CPU operation mode control. (See Fig. 4-12.)

When bit 3 or 2 is set to 1, the standby mode is set. If the standby mode is released by the standby release signal, automatically bits 3 and 2 are cleared and the normal operation mode is set. (For details, see Chapter 6.)

The low-order two bits of PCC are set by executing a 4-bit memory handling instruction (the high-order two bits are set to 0.)

Bits 3 and 2 are set to 1 by executing the STOP and HALT instructions respectively.

The STOP and HALT instructions can always be executed independently of the MBE contents.

CPU clock can be selected only during main system clock operation. In subsystem clock operation, the low-order two bits of PCC become invalid and the CPU clock is fixed to $f_{XT}/4$. The STOP instruction can also be executed only in main system clock operation.

Example 1: Set machine cycle to 0.95 us (f_x=4.19 MHz)

```
SEL MB15
MOV A, #0011B
MOV PCC, A
```

2: Set machine cycle to 1.63us (f_x=4.91 MHz)

```
SEL MB15
MOV A, #0010B
MOV PCC, A
```

3: Set STOP mode (be sure to write an NOP instruction following the STOP or HALT instruction).

```
STOP
NOP
```

When the $\overline{\text{RESET}}$ signal is generated, PCC is reset to 0.

Fig. 4-12 Processor Clock Control Register Format

Address	3	2	1	0	Symbol
FB3H	PCC3	PCC2	PCC1	PCC0	PCC

CPU clock selection bit
When $f_X \leq 4.19$ MHz

		SCC3, SCC0=00 The value enclosed in parentheses is applied when f_X = 4.19 MHz		SCC3, SCC0 = 01 or 11 The value enclosed in parentheses is applied when f_{XT} = 32.768 kHz	
		CPU clock frequency	One machine cycle	CPU clock frequency	One machine cycle
0	0	ϕ output = $f_X/64$ (65.5 kHz)	15.3 us	ϕ = $f_{XT}/4$ (8.192 kHz)	122 us
0	1	Undefined	—	Undefined	—
1	0	ϕ = $f_X/8$ (524 kHz)	1.91 us	ϕ = $f_{XT}/4$ (8.192 kHz)	122 us
1	1	ϕ = $f_X/4$ (1.05 MHz)	0.95 us		

When 4.19 MHz < $f_X \leq 5.0$ MHz

		SCC3, SCC0=00 The value enclosed in parentheses is applied when f_X = 4.19 MHz		SCC3, SCC0 = 01 or 11 The value enclosed in parentheses is applied when f_{XT} = 32.768 kHz	
		CPU clock frequency	One machine cycle	CPU clock frequency	One machine cycle
0	0	ϕ output = $f_X/64$ (76.7 kHz)	13 us	ϕ = $f_{XT}/4$ (8.192 kHz)	122 us
0	1	Undefined	—	Undefined	—
1	0	ϕ = $f_X/8$ (614 kHz)	1.63 us	ϕ = $f_{XT}/4$ (8.192 kHz)	122 us
1	1	Undefined	—	Undefined	—

f_X : Main system clock oscillator output frequency
f_{XT}: Subsystem clock oscillator output frequency

CPU operation mode control bit

0	0	Normal operation mode
0	1	HALT mode
1	0	STOP mode
1	1	Undefined

Caution: To use the f_X value in the range of 4.19 MHz < $f_X \leq$ 5.0 MHz, if you set the CPU clock frequency to the highest speed mode: Φ=f_X4 (PCC1, PCC0=11), one machine cycle becomes less than 0.95 us, which does not follow the specified MIN value 0.95 us. In this case, PCC1, PCC0=11 cannot be set. Set PCC1, PCC0=00 or 10 for the CPU clock frequency.
Resultantly, the highest speed mode of the CPU clock

(one machine cycle=0.95 us) is selected from a combination of f_X=4.19 MHz and PCC1, PCC0=11.

(4) System clock control register (SCC)

The system clock control register (SCC) is a 4-bit register whose least significant bit is used for CPU clock Φ selection and most significant bit is used for main system clock oscillation stop control. (See Fig. 4-13.)

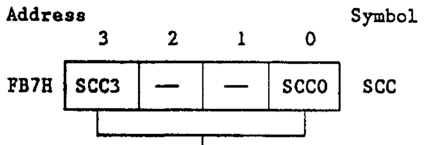
The SCC.0 and SCC.3 bits exist at the same data memory address, but cannot be changed at the same time. Thus, SCC.0 and SCC.3 are set by executing 1-bit manipulation instructions.

SCC.0 and SCC.3 can always be manipulated bit-wise independently of the MBE contents.

Main system clock oscillation can be stopped by setting SCC.3 to 1 only during subsystem clock operation. In main system clock operation, oscillation is stopped by executing the STOP instruction.

When the $\overline{\text{RESET}}$ signal is generated, SCC is reset to 0.

Fig. 4-13 System Clock Control Register Format



SCC3	SCC0	CPU clock selection	Main system clock oscillation
0	0	Main system clock	Oscillation is enabled
0	1	Subsystem clock	
1	0	Undefined	
1	1	Subsystem clock	Oscillation stop

Caution 1: System clock change requires a maximum of the 1/f_{XT} time. To stop main system clock oscillation, set SCC.3 to 1 in the machine cycle time listed in Table 4-5 after changing to subsystem clock.

Caution 2: Even if SCC.3 is set to stop oscillation during main system clock operation, the CPU does not enter the normal STOP mode.

Caution 3: If SCC.3 is set to 1, X1, input is internally short-circuited to V_{SS} (ground potential) to suppress crystal oscillator leakage.

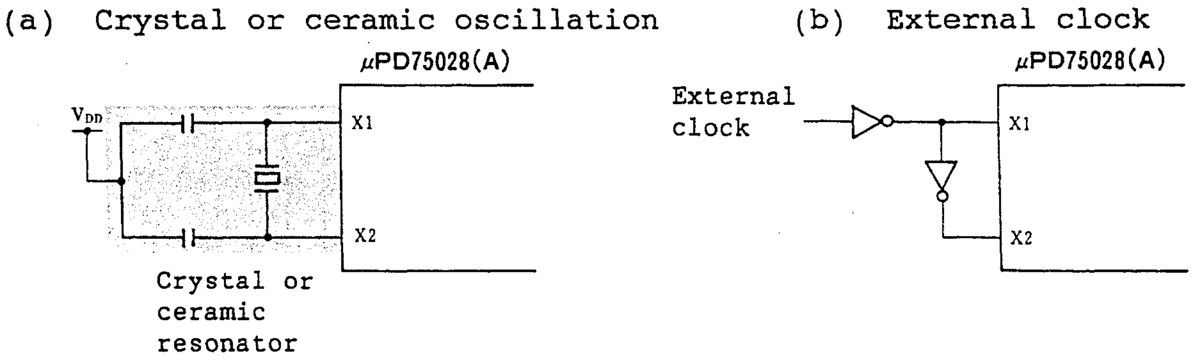
Caution 4: To select a sybsystm clock for the CPU clock, do not exceed the +70°C temperature limit.

(5) System clock oscillator

The main system clock oscillator oscillates by a crystal resonator (standard: 4.194304 MHz) or ceramic resonator connected to the X1 and X2 pins.

External clock can also be input.

Fig. 4-14 Main System Clock Oscillator External Circuit



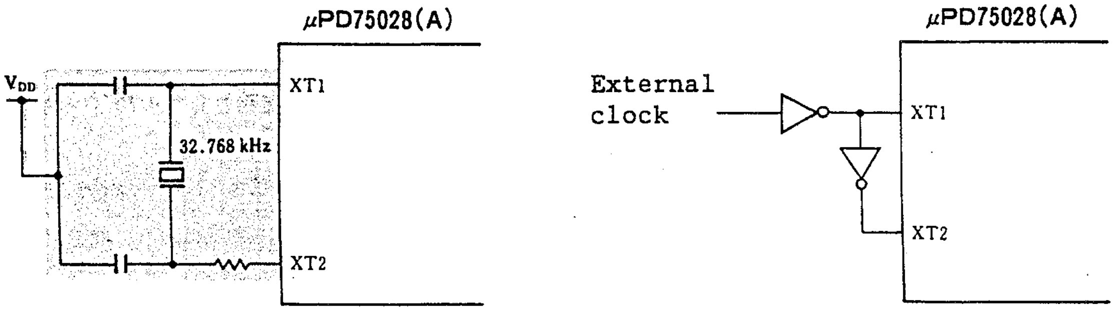
(a) Crystal or ceramic oscillation

(b) External clock

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

External clock can also be input.

Fig. 4-15 Subsystem Clock Oscillator External Circuit



(a) Crystal oscillation (b) External clock

Caution: To use the main system clock and subsystem clock oscillators, wire the shaded portions in the Fig. 4-14, 4-15 to avoid wiring capacitance affection, etc.:

- Make wiring as extremely short as possible.
- Do not cross the oscillator and any other signal line.

  Do not put the oscillator near any line where high current which changes flows.
- Be sure to place the oscillator capacitor ground point in the same potential as the V_{DD} pin. Do not connect to any power pattern where high current flows.
- Do not take out any signal from the oscillator.

The subsystem clock oscillator is low in amplification degree to consume low current, and a noise error tends to occur in the subsystem clock oscillator more than in the main system clock oscillator. To use the subsystem clock, pay extreme attention to the wiring method.

(6) Time required for system clock and CPU clock change

The system clock and CPU clock can be changed by setting the least significant bit of SCC and the low-order two bits of PCC. This change is not made just after register rewrite. The CPU operates based on the clock selected before the change during given machine cycles. To stop main system clock oscillation, the STOP instruction must be executed or SCC.3 must be set to 1 after the change time has elapsed.

Table 4-5 Maximum Time Required for System Clock, CPU Clock Change

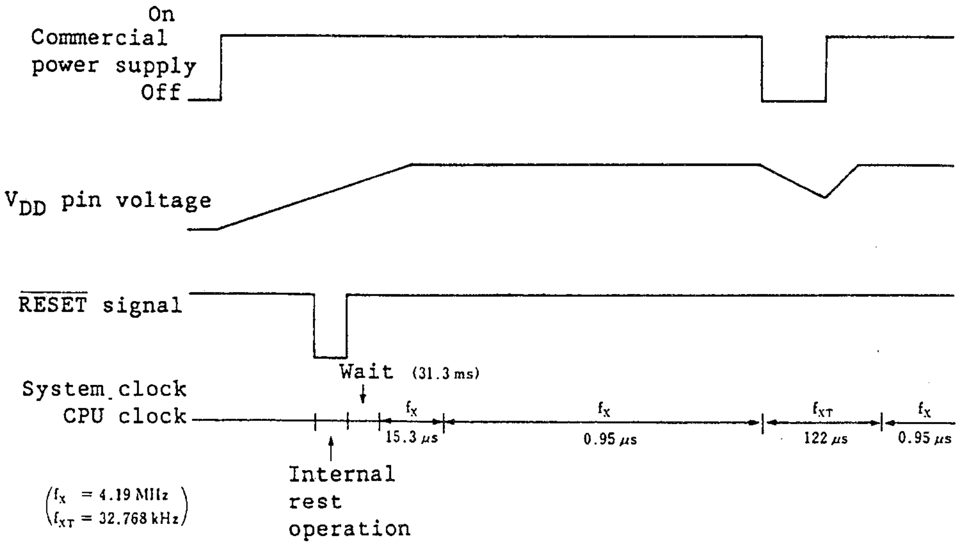
Setup value before change			Setup value after change											
SCC 0	PCC 1	PCC 0	SCC0	PCC1	PCC0	SCC0	PCC1	PCC0	SCC0	PCC1	PCC0	SCC0	PCC1	PCC0
			0	0	0	0	1	0	0	1	1	1	x	x
0	0	0				One machine cycle			One machine cycle			$\frac{f_X}{64f_{XT}}$ machine cycle (two machine cycle)		
	1	0	Eight machine cycle						Eight machine cycle			$\frac{f_X}{8f_{XT}}$ machine cycle (16 machine cycle)		
	1	1	16 machine cycle			16 machine cycle						$\frac{f_X}{4f_{XT}}$ machine cycle (32 machine cycle)		
0	x	x	One machine cycle			One machine cycle			One machine cycle					

The value enclosed in parentheses is applied when f_X = 4.19 MHz and f_{XT} = 32.768 kHz.
x : Don't care

(7) System clock and CPU clock change procedure

System clock and CPU clock change is explained by using Fig. 4-16.

Fig. 4-16 System Clock and CPU Clock Change



① When the $\overline{\text{RESET}}$ signal is generated, the CPU starts operation at the lowest speed of the main system clock (15.3 us at 4.19 MHz) in the wait time to secure the oscillation stable time (31.3 ms at 4.19 MHz).

② The CPU rewrites PCC and operates at the highest speed in the sufficient time for the V_{DD} pin voltage to rise to the voltage at which the CPU can operate at the highest speed.

③ When detecting commercial power supply being turned off from interrupt input (INT4 should be used), etc., the CPU sets SCC.0 to 1 and operates on subsystem clock. (At the time, a check must be previously made to ensure that subsystem clock oscillation starts.) In the time required to change to the subsystem clock (32 machine cycles), the CPU sets SCC.3 to 1 and stops main system clock oscillation.

④ After detecting commercial power supply being turned on from an interrupt, etc., the CPU resets SCC.3 to 0 and starts main system clock oscillation. In the time required for oscillation to become stable, the CPU resets SCC.0 to 0 and operates at the highest speed.

4.3 Clock Output Circuit

(1) Clock output circuit configuration

Fig. 4-17 shows the clock output circuit configuration.

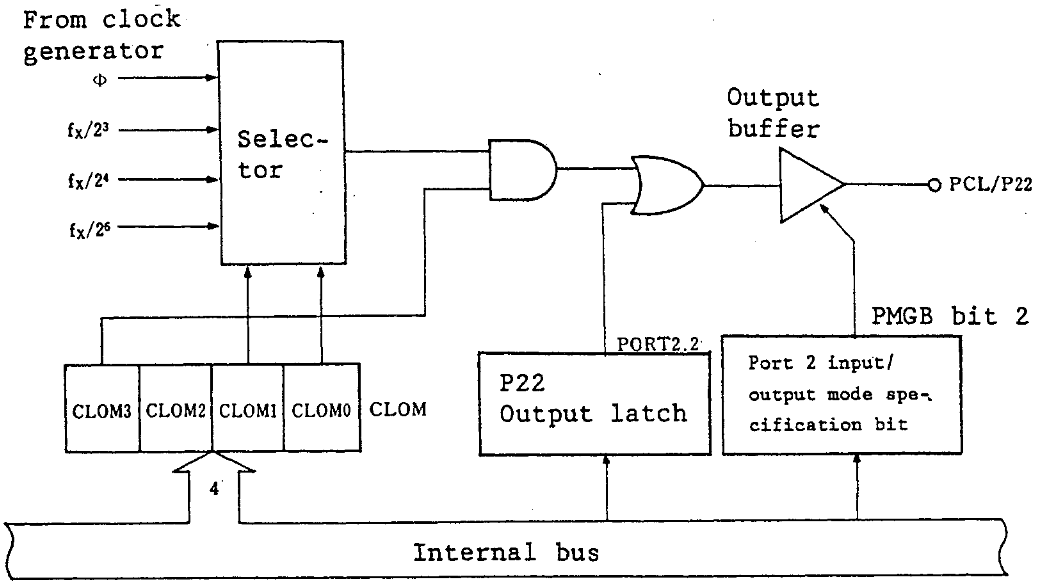
(2) Clock output circuit function

The clock output circuit outputs clock pulses from the P22/ PCL pin. It is used to supply clock pulses to remote control output and peripheral LSI.

The clock pulse output sequency is as follows:

(a) Select clock output frequency. Disable clock output.
(b) Write 0 into the P22 output latch.
(c) Set the output mode for port 2.
(d) Enable clock output.

Fig. 4-17 Clock Output Circuit Configuration



Remarks: The circuit is designed to prevent any pulse whose width is short from being output when clock output enable and disable are changed.

(3) Clock output mode register (CLOM)

The clock output mode register (CLOM) is a 4-bit register to control clock output.

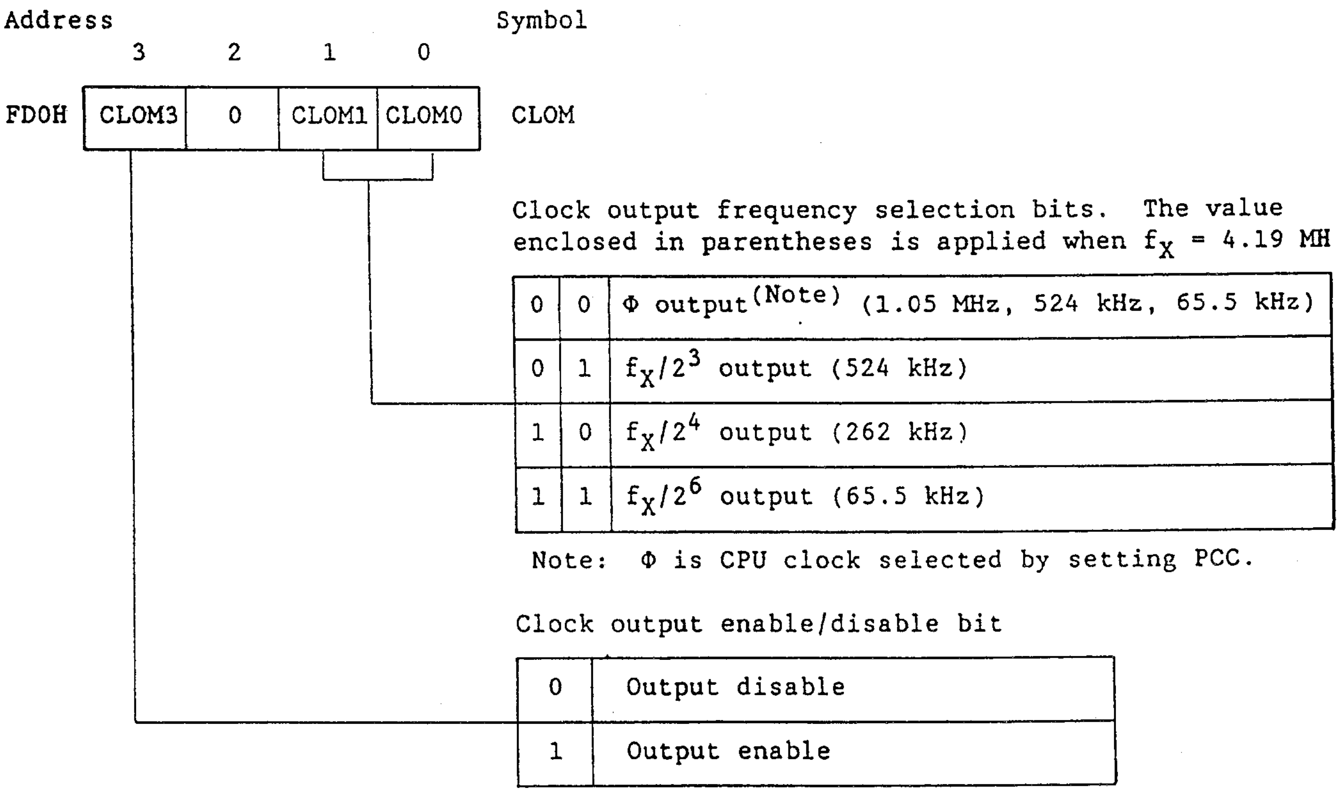
CLOM is set by executing a 4-bit memory handling instruction. It cannot be read.

Example: Output CPU clock Φ from PCL/P22 pin..

```
SEL MB15       ; or CLR1 MBE
MOV A, #1000B
MOV CLOM, A
```

When the RESET signal is generated, CLOM is reset to 0, disabling clock output.

Fig. 4-18 Clock Output Mode Register Format



Address	3	2	1	0	Symbol
FD0H	CLOM3	0	CLOM1	CLOM0	CLOM

Clock output frequency selection bits. The value enclosed in parentheses is applied when f_X = 4.19 MH

0	0	Φ output(Note) (1.05 MHz, 524 kHz, 65.5 kHz)
0	1	$f_X/2^3$ output (524 kHz)
1	0	$f_X/2^4$ output (262 kHz)
1	1	$f_X/2^6$ output (65.5 kHz)

Note: Φ is CPU clock selected by setting PCC.

Clock output enable/disable bit

0	Output disable
1	Output enable

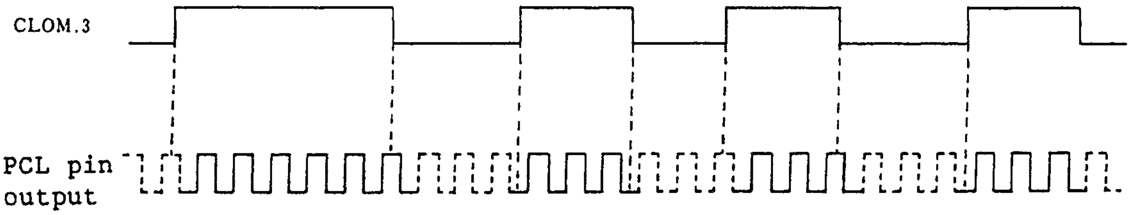
Caution: Be sure to write 0 into CLOM bit 2.

(4) Application example to remote control output

The uPD75028(A) clock output function can be applied to remote control output. The remote control output carrier frequency is selected by setting the clock frequency selection bit of the clock output mode register. Pulse output is enabled or disabled by setting the clock output enable/disable bit to 1 or 0 under software control.

When clock output enable and disable are changed, any pulse whose width is short is not output.

Fig. 4-19 Remote Control Output Application Example



4.4 Basic Interval Timer

(1) Basic interval timer configuration

Fig. 4-20 shows the basic interval timer configuration.

(2) Basic interval timer function

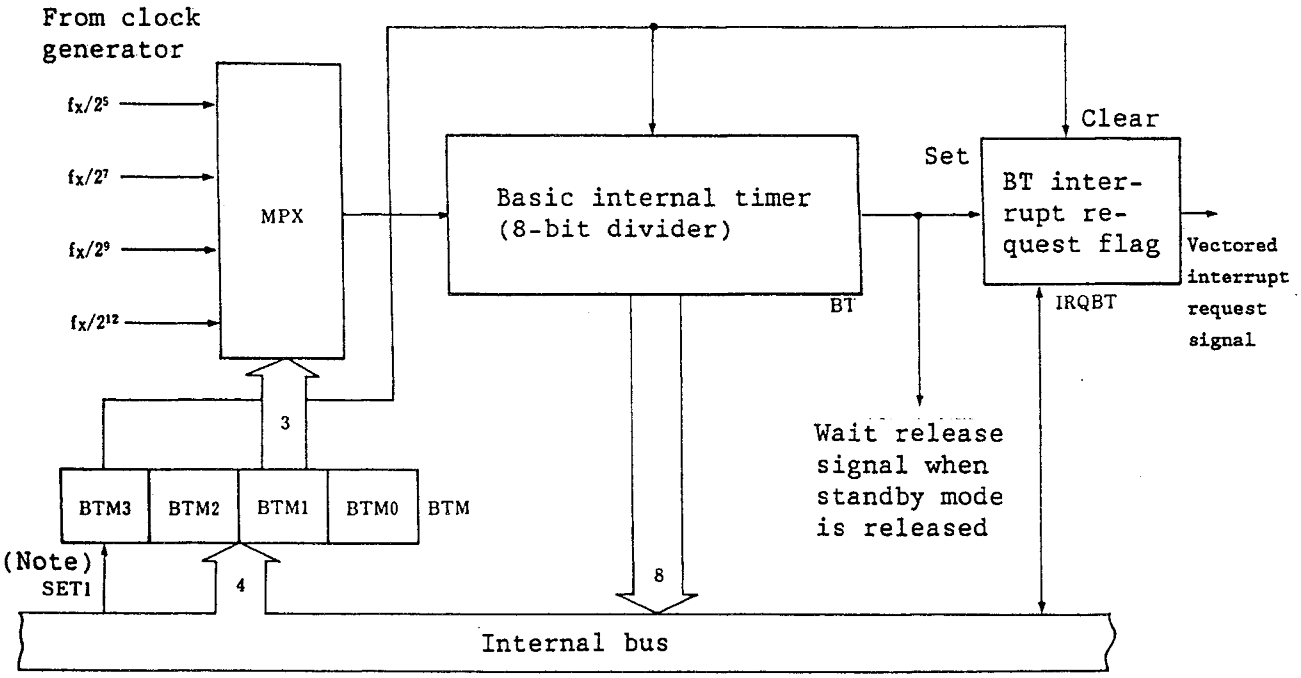
(a) interval timer operation to cause a reference time interrupt

(b) watchdog timer application to detect software upset

(c) wait time selection and count when the standby mode is released

(d) read of the count contents

Fig. 4-20 Basic Interval Timer Configuration



Note: Instruction execution

(3) Basic interval timer mode register (BTM)

The basic interval timer mode register (BTM) is a 4-bit register to control basic interval timer operation.

BTM is set by executing a 4-bit memory handling instruction.

Bit 3 can be set solely by executing a 1-bit manipulation instruction.

Example 1: Set interrupt generation interval to 1.95 ms at 4.19 MHz.

```
SEL MB15        ; or CLR1 MBE
MOV A, #1111B
MOV BTM, A      ; BTM ← 1111B
```

Example 2: Clear BT and IRQBT (watchdog timer application)

```
SEL  MB15       ; or CLR1 MBE
SET1 BTM.3      ; Set BTM bit 3 to 1
```

When bit 3 is set to 1, the basic interval timer contents and the basic interval timer interrupt request flag (IRQBT) are cleared at the same time (basic interval timer start).

When the $\overline{\text{RESET}}$ signal is generated, BTM is reset to 0, setting the interrupt request signal generation interval time to the longest value.

Fig. 4-21 Basic Interval Timer Mode Register Format

Address	3	2	1	0	Symbol
F85H	BTM3	BTM2	BTM1	BTM0	BTM

			Input clock specification	Interrupt interval time wait time when standby mode is released)
0	0	0	$f_X/2^{12}$ (1.02 kHz)	$2^{20}/f_X$ (250 ms)
0	1	1	$f_X/2^9$ (8.18 kHz)	$2^{17}/f_X$ (31.3 ms)
1	0	1	$f_X/2^7$ (32.768 kHz)	$2^{15}/f_X$ (7.82 ms)
1	1	1	$f_X/2^5$ (131 kHz)	$2^{13}/f_X$ (1.95 ms)
Other than the above			Undefined	———

Remarks: The value enclosed in parentheses is applied when f_X = 4.19 MHz

Basic interval timer start control bit

When 1 is written into the bit, the basic interval timer is started (the counter and interrupt request flag are cleared).
When operation is started, automatically the bit is reset to 0.

(4) Basic interval timer operation

The basic interval timer (BT) is always incremented when a clock is input from the clock generator. When BT overflows, the interrupt request flag (IRQBT) is set. BT count operation cannot be stopped.

One of the four types of interrupt generation intervals can be selected by setting BTM. (See Fig. 4-21.)

The basic interval timer and interrupt request flag can be cleared by setting BTM bit 3 to 1 (interval timer start indication).

The basic interval timer (BT) count state can be read by executing an 8-bit handling instruction. Data cannot be written into BT.

Caution: To read the basic interval timer count contents, execute read instruction twice and compare the former read contents with the latter read contents to prevent unstable data during count update from being read. If they are equipment to each other, the latter value is used as the read result; if they differ entirely, again read the count contents from the beginning.

Example: Read the BT count contents.

```
              SEL MB15
              MOV HL, #BT ; Set BT address in HL
       LOOP:  MOV XA, @HL ; First read
              XCH XA, BC  ; Second read
              MOV XA, @HL
              SKE A, C
              BR  LOOP
              XCH A, X
              SKE A, B
              BR   LOOP
```

The wait function, which stops CPU operation until the basic interval timer overflows, can be used to provide the oscillation stable time until system clock oscillation becomes stable when the STOP mode is released.

When wait time after $\overline{\text{RESET}}$ signal generation is fixed; when the STOP mode is released because of interrupt occurrence, the wait time can be selected by setting BTM. This wait time becomes the same as the interval time shown in Fig. 4-21. Set BTM before setting the STOP mode. (For details, see Chapter 6.)

4.5 Watch Timer

(1) Watch timer

The uPD75028(A) contains one channel of watch timer. Fig. 4-22 shows the watch timer block diagram.

(2) Watch timer function

(a) The test flag (IRQW) is set at 0.5-second intervals. The standby mode can be released by setting IRQW.

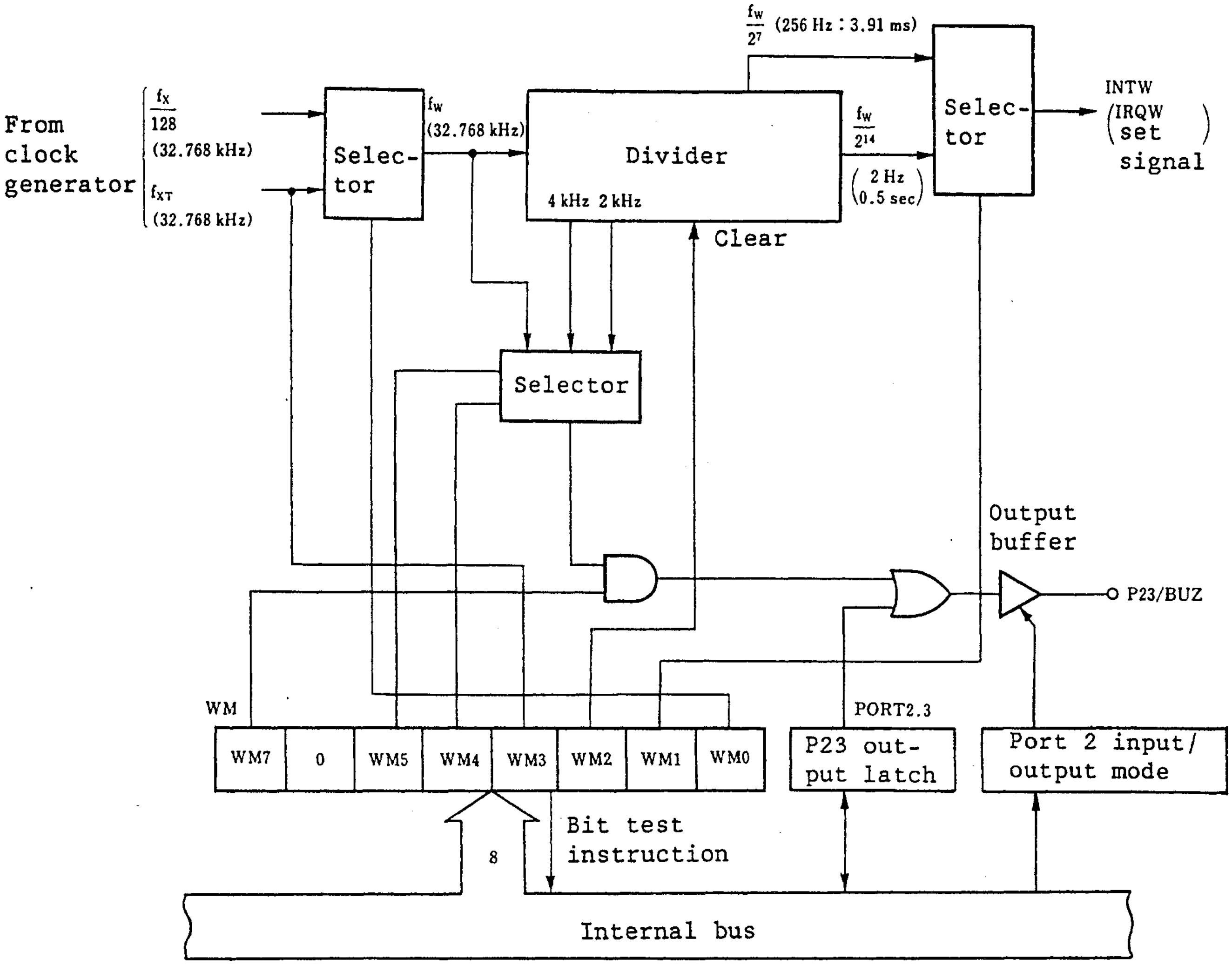
(b) 0.5-second intervals can be generated regardless of whether the clock is the main system clock or subsystem clock.

(c) In the rapid mode, IRQW is set at 0.5-second interval x 128 (3.91 ms). It is useful for programming debugging, and testing.

(d) Any desired frequency 2.048, 4.096, or 32.768 kHz can be output to the P23/BUZ pin. It can be used for buzzer sound generation or system clock oscillation frequency trimming.

(e) Since the divider can be cleared, the watch can be started at zero second.

Fig. 4-22 Watch Timer Block Diagram



Remarks: The value enclosed in parentheses is applied when f_X = 4.194304 kHz and f_{XT} = 32.768 kHz.

(3) Watch mode register (WM)

The watch mode register (WM) is an 8-bit register to control the watch timer. Fig. 4-23 shows the WM format.

The watch mode register except bit 3 is set by executing an 8-bit handling instruction. Bit 3 is used to test the XT1 pin input level. The input level to the XT1 pin can be tested by testing bit 3. Data cannot be written.

When the $\overline{\text{RESET}}$ signal is generated, all bits except bit 3 are reset to 0.

Example: Generate time intervals by using main system clock (4.19 MHz). Enable buzzer output.

```
CLR1 MBE
MOV XA, #84H
MOV WM, XA   ; Set WM
```

Fig. 4-23 Watch Mode Register Format

Address	7	6	5	4	3	2	1	0	Symbol
F98H	WM7	0	WM5	WM4	WM3	WM2	WM1	WM0	WM

Count clock (f_W) selection bit

WM0	0	System clock dividing output $\frac{f_X}{128}$ selection
	1	Subsystem clock f_{XT} selection

Operation mode selection bit

WM1	0	Normal watch mode ($\frac{f_W}{2^{14}}$: IRQW is set at 0.5-second intervals)
	1	Rapid watch mode ($\frac{f_W}{2^7}$: IRQW is set at 3.91-ms intervals)

Watch operation enable/disable bit

WM2	0	Stop watch operation (clear divider)
	1	Enable watch operation

Input level to XT1 pin (bit tested only is enabled)

WM3	0	Input to XT1 pin is low
	1	Input to XT1 pin is high

BUZ output frequency selection bit

WM5	WM4	BUZ output frequency
0	0	$\frac{f_W}{2^4}$ (2.048 kHz)
0	1	$\frac{f_W}{2^3}$ (4.096 kHz)
1	0	Undefined
1	1	f_W (32.768 kHz)

BUZ output enable/disable bit

WM7	0	Disable BUZ output
	1	Enable BUZ output

Remarks: The value enclosed in parentheses is applied when f_W = 32.768 kHz.

4.6 Timer/Event Counter

(1) Timer/event counter configuration

The uPD75028(A) contains one channel of timer/event counter. Fig. 4-24 shows the timer/event counter configuration.

(2) Timer/event counter function

The timer/event counter has the following functions:

(a) programmable interval timer operation
(b) square wave output of any frequency to PTO0 pin
(c) event counter operation
(d) TI0 pin input is divided by N for output to PTO0 pin (divider operation)
(e) serial shift clock supply to serial interface circuit
(f) count state read function

(3) Timer/event counter mode register (TM0) and timer/event counter output enable flag (TOE0)

The mode register (TM0) is an 8-bit register to control the timer/event counter. Fig. 4-25 shows the TM0 format.

The timer mode register is set by executing an 8-bit memory handling instruction

Bit 3 is a timer start bit and can be set solely. When timer operation is started, automatically the bit is reset to 0.

Example 1: Start the timer in the interval timer mode (CP = 4.09 kHz).

```
SEL MB15          ; or CLR1 MBE
MOV XA, #01001100B
MOV TM0, XA       ; TM0  ←  4CH
```

Example 2: Restart the timer according to setting the timer mode register.

```
SEL  MB15         ; or CLR1 MBE
SET1 TM0.3 ; TM0.bit3  ←  1
```

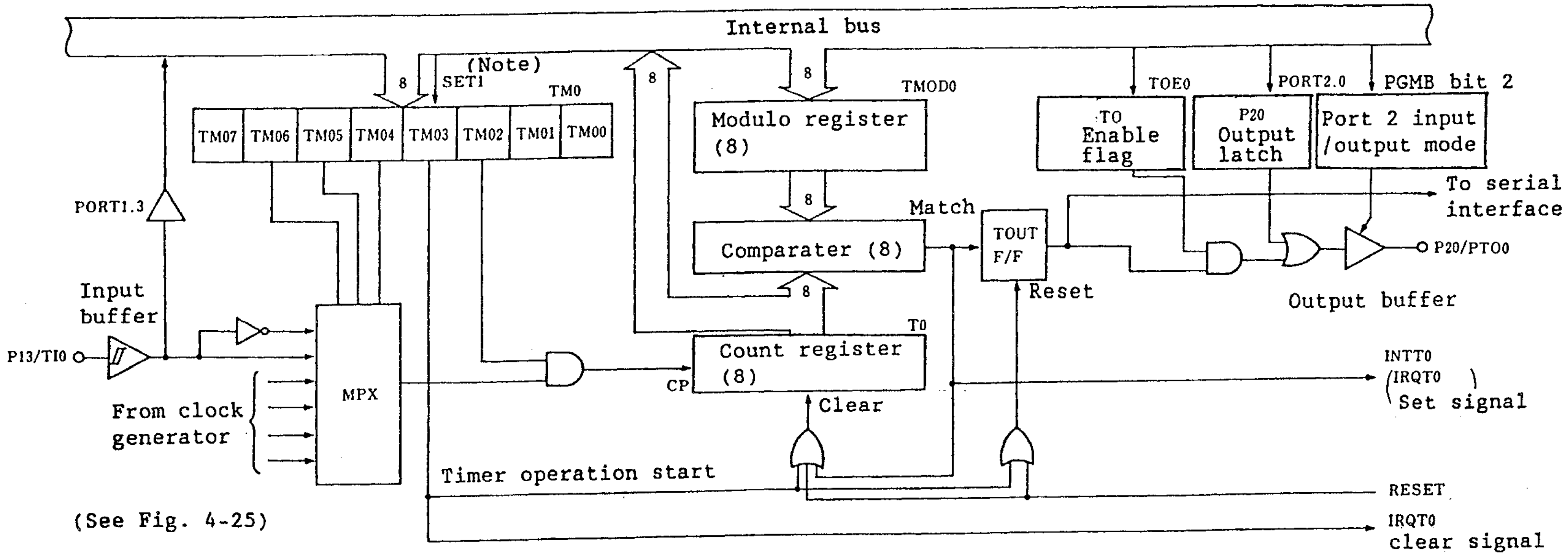
When the $\overline{\text{RESET}}$ signal.is generated, all the timer mode register bits are reset to 0

The timer/event counter output enable flag (TOE0) is a flag to control enabling/disabling output of the timer-out flip-flop (TOUT F/F) state to the PTO0 pin.

The timer-out flip-flop (TOUT F/F) is inverted when a match signal is input from the comparator. The timer-out flip-flop is reset to 0 by executing an instruction to set timer mode register (TM0) bit 3 to 1.

When the $\overline{\text{RESET}}$ signal is generated, TOE0 and TOUT F/F are reset to 0.

Fig. 4-24 Timer/Event Counter Block Diagram



Note: Instruction execution

Fig. 4-25 Timer/Event Counter Mode Register Format

Address	7	6	5	4	3	2	1	0	Symbol
FA0H	—	TM06	TM05	TM04	TM03	TM02	TM01	TM00	TM0

Operation Mode

	Count operation
0	Stop (count contents are retained)
1	Count operation

Timer start indication bit

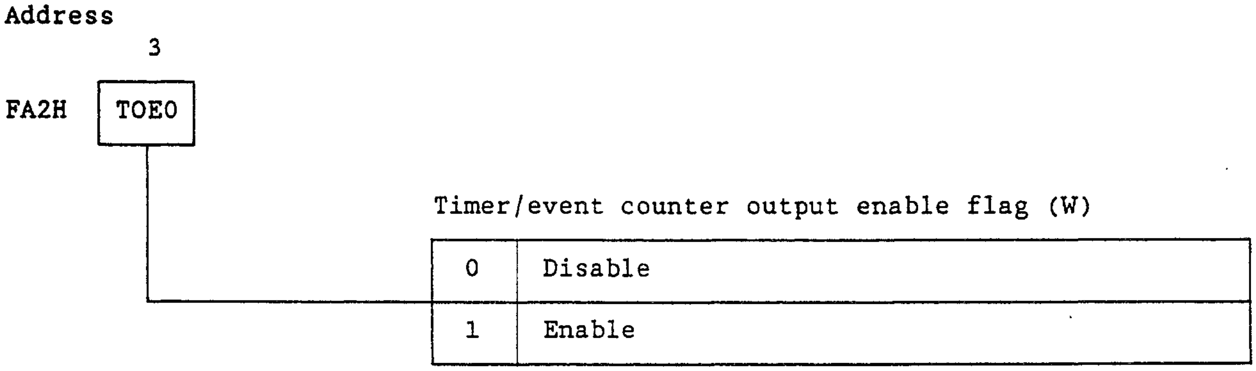
When 1 is written into the bit, the counter and IRQT0 flag are reset to 0. If bit 2 is set to 1, count operation is started.

Count pulse (CP) selection bit

TM06	TM05	TM04	Count pulse (CP)
0	0	0	TI0 input rising edge
0	0	1	TI0 input falling edge
0	1	0	Undefined
0	1	1	
1	0	0	$f_X/2^{10}$ (4.09 kHz)
1	0	1	$f_X/2^8$ (16.4 kHz)
1	1	0	$f_X/2^6$ (65.5 kHz)
1	1	1	$f_X/2^4$ (262 kHz)

Remarks: The value enclosed in parentheses is applied when f_X = 4.19 MHz.

Fig. 4-26 Timer/Event Counter Output Enable Flag Format



Timer/event counter output enable flag (W)

0	Disable
1	Enable

(4) Timer/event counter operation

The timer/event counter operates in the count operation stop mode or count operation mode depending on how the mode register is set.

The timer/event counter can always operate as follows regardless of how the mode register is set:

1 input and test the TI0 pin signal (can test P13 pin input);
2 output the timer-out flip-flop state to PTO0;
3 set the modulo register (TMOD0);
4 read the count register (T0);
5 set, clear, or test the interrupt request flag (IRQT0).

(a) Count operation stop mode

The count operation stop mode is set when TM0 bit 2 is 0. Since count pulse (CP) supply to the count register is stopped, count operation is not performed.

(b) Count operation mode

The count operation mode is set when TM0 bit 2 is 1. Count pulses selected by setting bits 4-6 are supplied to the count register, and count operation is performed as shown in Fig. 4-28.

Normally, count operation is started by

① setting count value in the modulo register (TMOD0)

② setting the operation mode, count clock, and start indication in the mode register (TM0).

The modulo register is set by executing an 8-bit data transfer instruction.

Caution: Set any value other than 0 in the modulo register.

Example: Set 3FH in the modulo register of channel 0.

```
  SEL MB15          ; or CLR1 MBE
  MOV XA, #3FH
  MOV TMOD0, XA
```

When the value set in the modulo register matches the count register contents, a match signal is generated, inverting TOUT F/F and clearing the count register.

The match signal generation interval time is (modulo register setup value + 1) x resolution

The resolution is $\frac{1}{\text{count pulse frequency}}$ Table 4-6 lists the resolution and maximum setup time determined by the selected count pulse (when FFH is set in the modulo register).

Table 4-6 Resolution and Maximum Setup Time (when f_X = 4.19 MHz)

Mode register			Timer channel 0	
TM06	TM05	TM04	Resolution	Maximum setup time
1	0	0	244 us	62.5 ms
1	0	1	61.1 us	15.6 ms
1	1	0	15.3 us	3.91 ms
1	1	1	3.81 us	977 us

Fig. 4-27 Operation in Count Operation Mode

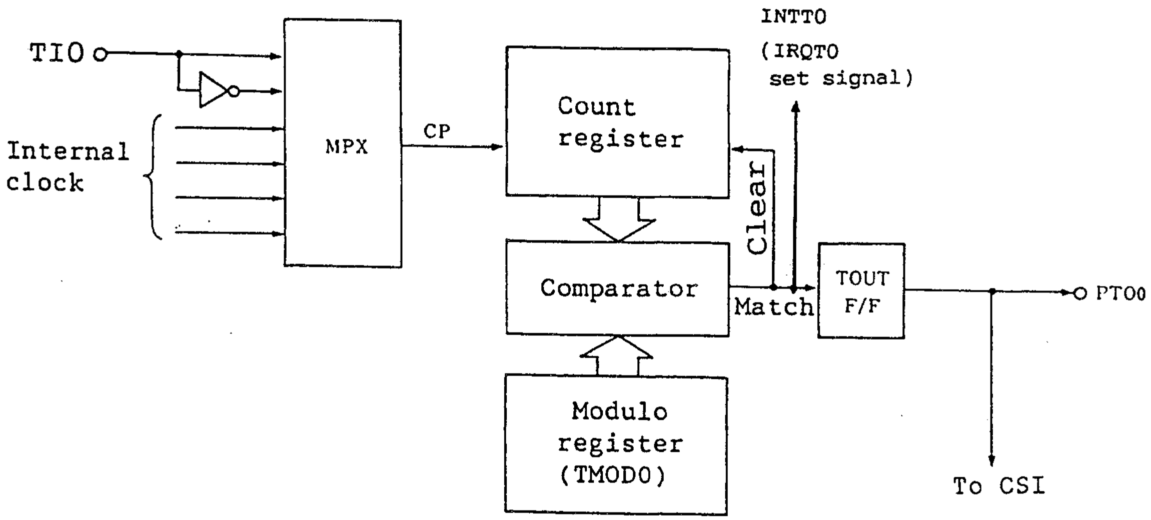
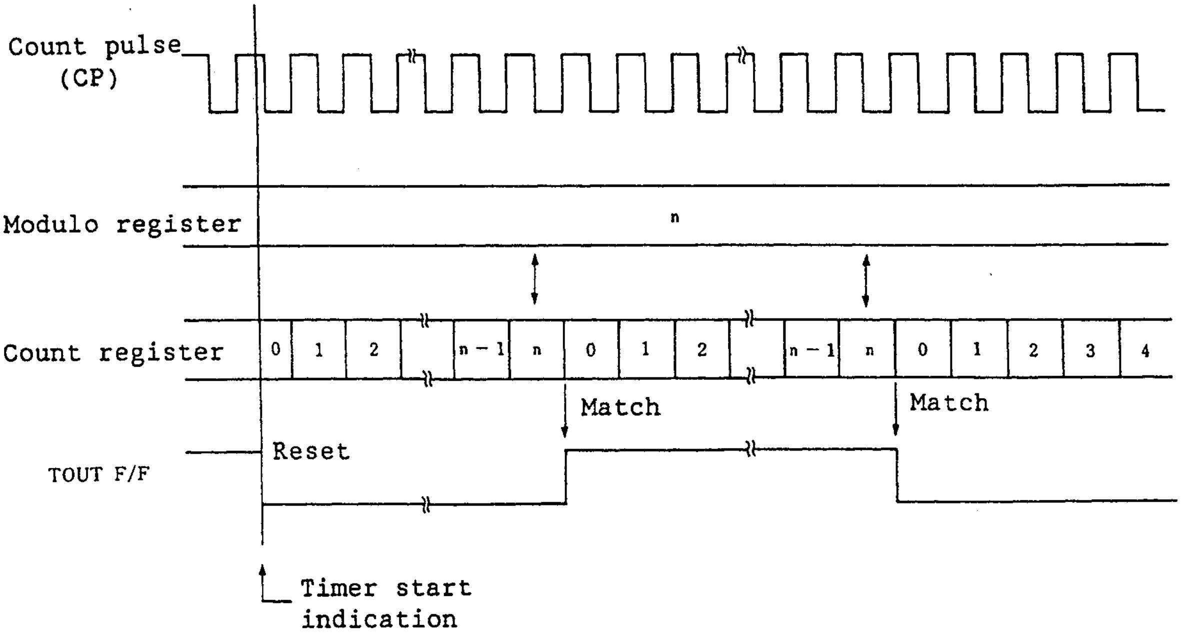


Fig. 4-28 Count Operation Timing



4.7 Serial Interface

(1) Serial interface function

The uPD75028(A) contains a clocked 8-bit serial interface which operates in any of the following four modes:

The functions of the modes are outlined below:

- Operation stop mode
  The operation stop mode can be used to reduce power consumption when serial transfer is not executed.

- 3-line serial I/O mode
  8-bit data is transferred by using the three lines of serial clock ($\overline{\text{SCK}}$), serial output (SO), and serial input (SI).
  The 3-line serial I/O mode enables simultaneous transmission and reception, thus the data transfer processing time can be shortened.
  Since the most significant bit (MSB) or least significant bit (LSB) can be selected for the top bit of 8-bit data to be transferred in series, the serial interface can be connected to any devices, regardless of whether the top bit is the MSB or LSB.
  The 3-line serial I/O mode enables the uPD75028(A) to be connected to the 75X series, 78K series, and peripheral I/O devices.

- 2-line serial I/O mode
  8-bit data is transferred by using the two lines of serial clock ($\overline{\text{SCK}}$) and serial data bus (SB0 or SB1).
  The 2-line serial I/O mode enables the uPD75028(A) to communicate with a number of devices if the output levels to the two lines are handled by the software.
  Since the $\overline{\text{SCK}}$ and SB0 (or SB1) output levels can be handled by the software, any desired transfer format

can be covered. thus, the former handshaking lines required for connection of a number of devices can be reduced and efficient use of the input/output ports can be made.

- SBI mode (serial bus interface mode)
  The SBI mode enables the uPD75028(A) to communicate with a number of devices by using the two lines of serial clock ($\overline{\text{SCK}}$) and serial data bus (SB0 or SB1). The SBI mode is compliant with the NEC serial bus format.
  In the SBI mode, the transmitting party can output "address" to select a device with which the uPD75028(A) is to communicate, "command" given to the device, and actual "data" to the serial data bus. The receiving party can judge the received data to be "address", "command", or "data" by the hardware. This function can make efficient use of the input/output ports as with the 2-line serial I/O mode, and can also simplify the serial interface control part of an application program.

(2) Serial interface configuration

Fig. 4-29 shows the serial interface clock diagram.

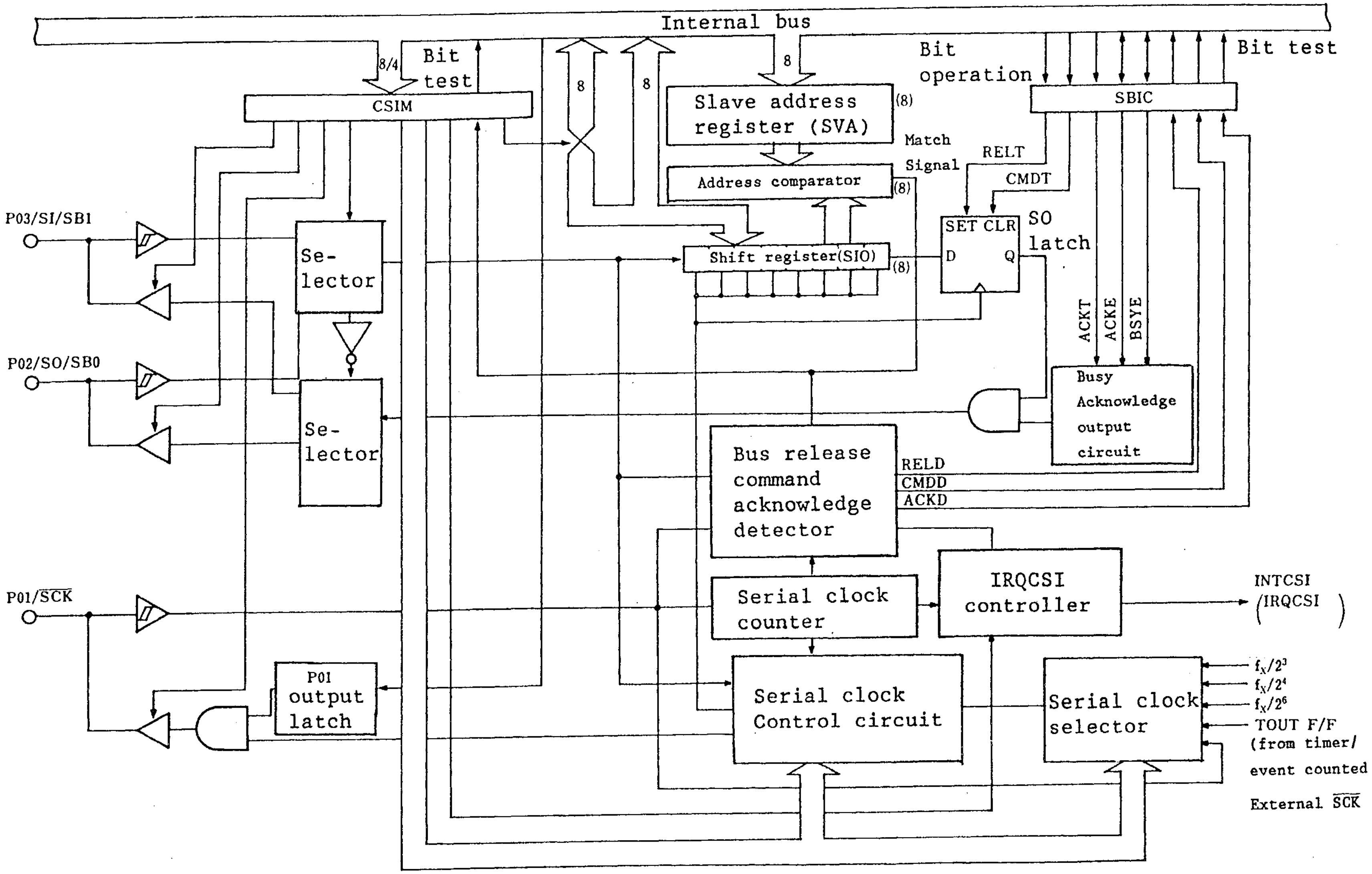


Fig. 4-29 Serial Interface Block Diagram

(3) Register function

(a) Serial operation mode register (CSIM)

Fig. 4-30 shows the serial operation mode register (CSIM) format.

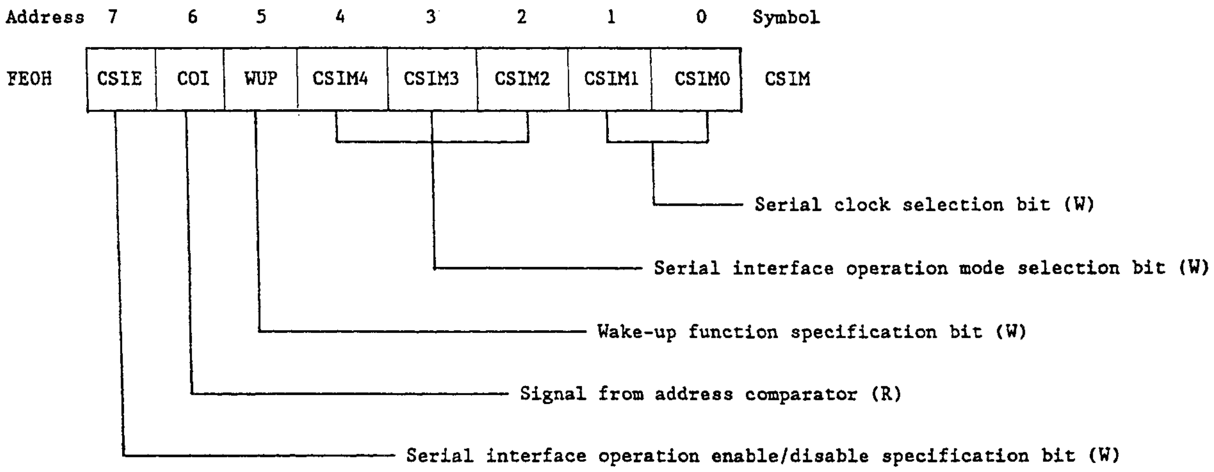
The CSIM register is an 8-bit register to specify the serial interface operation mode, serial clock, wake-up function, etc.

The CSIM register is handled by executing an 8-bit memory handling instruction. The high-order three bits can also be handled bit-wise.

To handle the bits bit-wise, use the bit names. Whether or not read/write operation can be performed depends on the bit. (See Fig.4-30.) Bit 6 can only be tested bit-wise, and if data is written into the bit, it becomes invalid.

When reset is input, the CSIM register is reset to 00H.

Fig. 4-30 Serial Operation Mode Register (CSIM) Format (1/3)


Address
7
6
5
4
3
2
1
0
Symbol
FEOH
CSIE
COI
WUP
CSIM4
CSIM3
CSIM2
CSIM1
CSIMO
CSIM
Serial clock selection bit (W)
Serial interface operation mode selection bit (W)
Wake-up function specification bit (W)
Signal from address comparator (R)
Serial interface operation enable/disable specification bit (W)

Remarks: (R): Read only is enabled
(W): Write only is enabled

Fig. 4-30 Serial Operation Mode Register (CSIM) Format (2/3)

Serial clock selection bit (W)

CSIM1	CSIM0	Serial clock			SCK pin mode
		3-line serial I/O mode	SBI mode	2-line serial I/O mode	
0	0	External input clock to $\overline{\text{SCK}}$ pin			Input
0	1	Timer/event counter output (TO)			Output
1	0	$f_X/2^4$(262 kHz)		$f_X/2^6$(65.5 kHz)	
1	1	$f_X/2^3$(524 kHz)			

Remarks: The values enclosed in parentheses apply when the frequency is 4.19 MHz.

Serial interface operation mode selection bit (W)

CSIM4	CSIM3	CSIM2	Operation mode	Shift register bit order	SO pin function	SI pin function
x	0	0	3-line serial I/O mode	SIO_{7-0} ↔ XA (MSB is first transferred)	SO/P02 (CMOS output)	SI/P03 (input)
		1		SIO_{0-7} ↔ XA (LSB is first transferred)		
0	1	0	SBI mode	SIO_{7-0} ↔ XA (MSB is first transferred)	SB0/P02 (N-ch open drain input /output)	P03 input
1					P02 input	SB1/P03 (N-ch open drain input /output)
0	1	1	2-line serial I/O mode	SIO_{7-0} ↔ XA (MSB is first transferred)	SB0/P02 (N-ch open drain input /output)	P03 input
1					P02 input	SB1/P03 (N-ch open drain input /output)

Remarks: x: don't care

Wake-up function specification bit (W)

WUP	0	IRQCSI is set each time serial transfer in each mode terminates.
	1	The function is used only in the SBI mode. IRQCSI is set only when the address received after bus release matches data in the slave address register (wake-up state). SB0/SB1 is high impedance.

Caution: If WUP is set to 1 when $\overline{\text{BUSY}}$ signal is being output, $\overline{\text{BUSY}}$ is not released. In the SBI mode, $\overline{\text{BUSY}}$ signal is output until the next serial clock ($\overline{\text{SCK}}$) falling edge after $\overline{\text{BUSY}}$ release indication is given. To set WUP to 1, be sure to check that the SB0 (or SB1) pin goes high after $\overline{\text{BUSY}}$ is released.

Fig. 4-30 Serial Operation Mode Register (CSIM) Format (3/3)

Signal from address comparator (R)

(Note) COI	Clear condition (COI=0)	Set condition (COI=1)
	When data in slave address register (SVA) does not match data in shift register	When data in slave address register (SVA) matches data in shift register

Note: COI read is valid only before serial transfer starts or after it completes. An undefined value is read during the serial transfer. COI data written by executing an 8-bit manipulation instruction is ignored.

Serial interface operation enable/disable specification bit (W)

		Shift register operation	Serial clock counter	IRQCSI flag	SO/SB0, SI/SB1 pin
CSIE	0	Shift operation disable	Clear	Retention	Port0 function only
	1	Shift operation enable	Count operation	Can be set	For both function in each mode and port 0

Remarks 1: The operation mode can be selected setting CSIE, CSIM3, and CSIM2, as listed below:

CSIE	CSIM3	CSIM2	Operation mode
0	x	x	Operation stop mode
1	0	x	3-line serial I/O mode
1	1	0	SBI mode
1	1	1	2-line serial I/O mode

Remarks 2: The P01/$\overline{\text{SCK}}$ pin state is set by setting CSIE, CSIM1, and CSIM0, as listed below:

CSIE	CSIM1	CSIM0	P01/SCK pin state
0	0	0	Input port
1	0	0	High impedance
0	1	0	High level output
0	0	1	
0	1	1	
1	1	0	Serial clock output (high level output)
1	0	1	
1	1	1	

Remarks 3: To clear CSIE during serial transfer, execute the following sequence:

① Clear interrupt enable flag to disable interrupts;

② clear CSIE; and

③ clear interrupt request flag

Example 1: Select $f_X/2^4$ for serial clock, generate serial interrupt IRQCSI at the termination of each serial transfer, and perform serial transfer in the SBI mode with the SB0 pin as the serial data bus.

```
SEL  MB15           ;or CLR1 MBE
MOV  XA,#10001010B
MOV  CSIM,XA        ;CSIM ← 10001010B
```

Example 2: Enable serial transfer conforming to the CSIM contents.

```
SEL    MB15          ; or CLR1 MEB
SET1   CSIE
```

(b) Serial bus interface control register (SBIC)

Fig. 4-31 shows the serial bus interface control register (SBIC) format.

The SBIC register is an 8-bit register which consists of the serial bus control bits and the flags indicating the input data state from the serial bus.

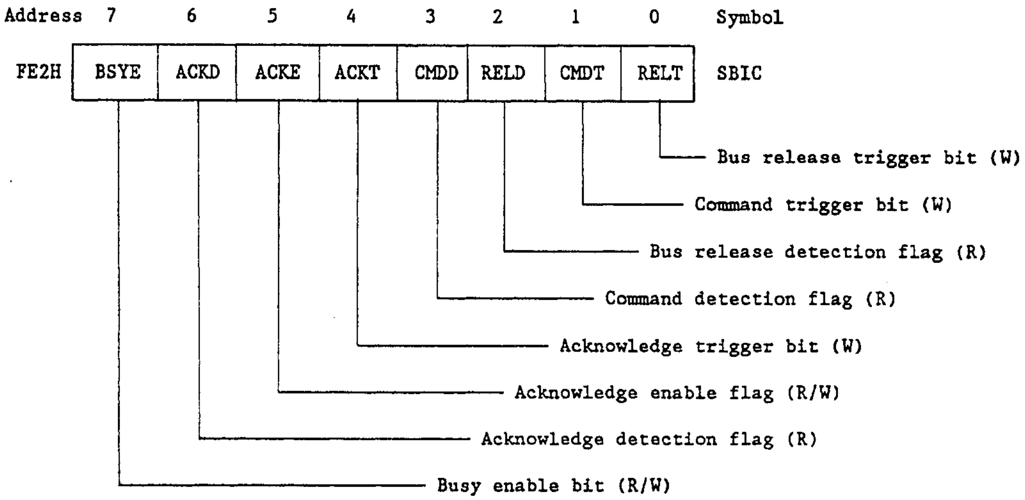
The register is mainly used with the SBI mode.

The register is handled by executing bit manipulation instructions; it cannot be handled by executing a 4-bit or 8-bit memory handling instruction. Whether or not read/write operation can be performed depends on the bit. (See Fig. 4-31.) When reset is input, the SBIC register is reset to 00H.

Caution: In the 3-line serial I/O mode and 2-line serial I/O mode, only the following bits can be used:

- Bus release trigger bit (RELT) ... Set SO latch
- Command trigger bit (CMDT) ... Clear SO latch

Fig. 4-31 Serial Bus Interface Control Register (SBIC) Format (1/3)


Address
7
6
5
4
3
2
1
0
Symbol
FE2H
BSYE
ACKD
ACKE
ACKT
CMDD
RELD
CMDT
RELT
SBIC
Bus release trigger bit (W)
Command trigger bit (W)
Bus release detection flag (R)
Command detection flag (R)
Acknowledge trigger bit (W)
Acknowledge enable flag (R/W)
Acknowledge detection flag (R)
Busy enable bit (R/W)

Remarks: (R): Read only is enabled

(W): Write only is enabled

(R/W): Both read and write are enabled

Bus release trigger bit (W)

RELT	Bus release signal (REL) trigger output control bit. The SO latch is set to 1 by setting the RELT bit to 1. After this, the RELT bit is automatically reset to 0.

Caution: Do not clear SB0 (or SB1) during serial transfer. Be sure to clear it before the transfer starts or after the transfer terminates.

Command trigger bit (W)

CMDT	Command signal (CMD) trigger output control bit. The SO latch is reset to 0 by setting the CMDT bit to 1. After this, the CMDT bit is automatically reset to 0.

Caution: Do not clear SB0 (or SB1) during serial transfer. Be sure to clear it before the transfer starts or after the transfer terminates.

Fig. 4-31 Serial Bus Interface Control Register (SBIC) Format (2/3)

Bus release detection flag (R)

	Clear condition (RELD=0)	Set condition (RELD=1)
RELD	① When transfer start instruction is executed ② When reset is input ③ When CSIE=0 (see Fig. 4-30) ④ When SVA and SIO values do not match when address is received	When bus release signal (REL) is detected

Command detection flag (R)

	Clear condition (CMDD=0)	Set condition (CMDD=1)
CMDD	① When transfer start instruction is executed ② When bus release signal (REL) is detected ③ When reset is input ④ When CSIE=0 (see Fig. 4-30)	When command signal (CMD) is detected

Acknowledge trigger bit (W)

ACKT	If the ACKT bit is set to 1 after transfer terminates, $\overline{\text{ACK}}$ is output in synchronization with the next $\overline{\text{SCK}}$. After the $\overline{\text{ACK}}$ signal is output, the ACKT bit is automatically reset to 0.

Caution 1: Do not set ACKT to 1 before serial transfer terminates or during transfer.

2: ACKT cannot be reset to 0 by software.

3: To set ACKT to 1, set ACKE to 0.

Fig. 4-31 Serial Bus Interface Control Register (SBIC) Format (3/3)

Acknowledge enable bit (R/W)

ACKE	0	Automatic output of acknowledge signal (ACK) is disabled. (ACK can be output by setting ACKT to 1.)	
	1	When ACKE is set before transfer terminates	$\overline{\text{ACK}}$ is output in synchronization with the ninth $\overline{\text{SCK}}$ clock
		When ACKE is set after transfer terminates	$\overline{\text{ACK}}$ is output is synchronization with $\overline{\text{SCK}}$ just after a set instruction is executed

Acknowledge detection flag (R)

	Clear condition (ACKD=0)	Set condition (ACKD=1)
ACKD	① When transfer start instruction is executed ② When reset is input	When acknowledge signal ($\overline{\text{ACK}}$) is detected (in synchronization with the $\overline{\text{SCK}}$ rising edge)

Busy enable bit (R/W)

BYSE	0	① Automatic output of busy signal is disabled ② Busy signal output is stopped in synchronization with the $\overline{\text{SCK}}$ falling edge just after a clear instruction is executed
	1	Busy signal is output in synchronization with the $\overline{\text{SCK}}$ falling edge following acknowledge signal

Example 1: Output command signal.

```
    SEL   MB15    ;or CLR1 MBE
    SET1  CMDT
```

Example 2: Test RELD and CMDD and judge the receive data type to separate processing. Execute the interrupt service routine only when address match is found with WUP=1.

```
    SEL  MB15
    SKF  RELD       ;RELD test
    BR   !ADRS
    SKT  CMDD       ;CMDD test
    BR   !DATA
CMD :  .......    ; Command interpretation
DATA:  .......    ; Data processing
ADRS:  .......    ; Address decode
```

(c) Shift register (SIO)

Fig. 4-32 shows the configuration of the shift register (SIO) and its peripherals. The SIO is an 8-bit register for parallel-serial and serial-parallel conversion and serial transfer (shift operation) in synchronization with a given serial clock.

Serial transfer is started by writing data into the SIO register.

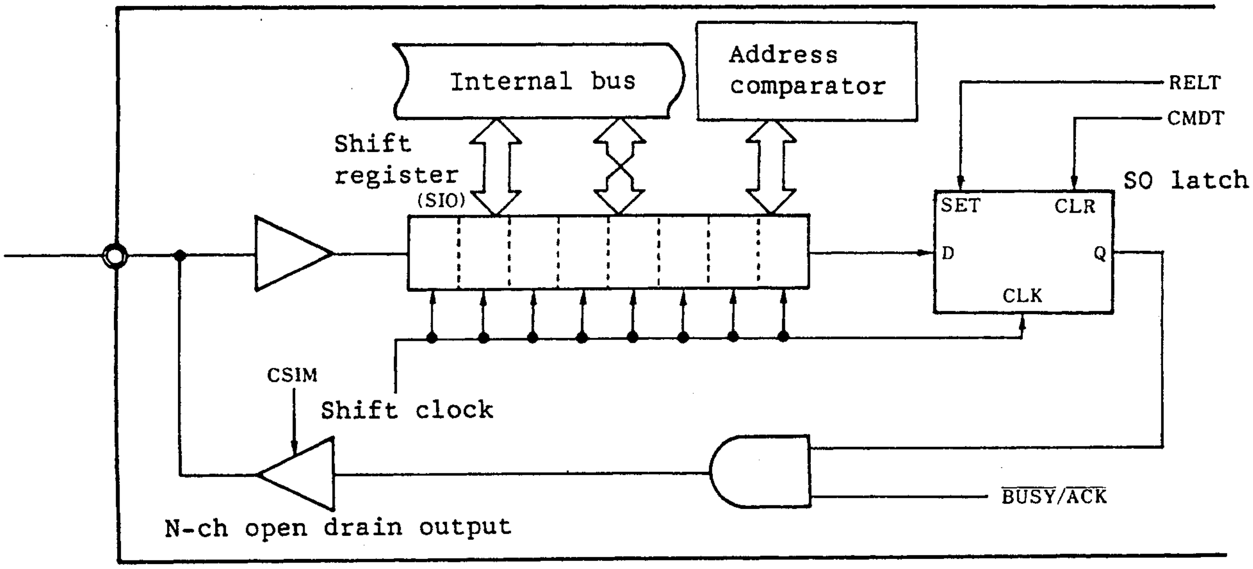
When data is transmitted, the data written into the SIO register is output to serial output (SO) or serial data bus (SB0 or SB1). When data is received, data is read into the SIO register from serial input (SI) or SB0 or SB1.

The SIO register can be read and written by executing 8-bit manipulation instructions.

When reset is input during operation, the SIO register value becomes undefined. When reset is input in the standby mode, the SIO register value is retained.

After 8-bit data has been transmitted or received, shift operation stops.

Fig. 4-32 Configuration of Shift Register and its Peripherals



The SIO register can be read and written (serial transfer can be started) when

- serial interface operation enable/disable bit (CSIE) is set to 1, except when it is set to 1 after data is written into the SIO register
- serial clock is masked after 8-bit serial data transfer; or
- $\overline{\text{SCK}}$ is high

(d) Slave address register (SVA)

The slave address register (SVA) has the two functions described below.

The SVA register is handled by executing an 8-bit manipulation instruction; it can only be written to. When reset signal is input, the SVA value becomes undefined, however if reset signal is input in the standby mode, the SVA value is retained.

- Slave address detection

[SBI mode]

The SVA register is used when the uPD75028(A) is connected to the serial bus as a slave device. The SVA register is an 8-bit register used for the slave to set the slave address (its own specification number). The master outputs a slave address to select a specific slave among slaves connected to the master. The slave address output by the master is compared with the SVA value by the address comparator. If they match, the slave is selected.

At the time, serial operation mode register (SCIM) bit 6 (COI) is set to 1.

Caution 1: Whether or not a slave is selected is decided by detecting a match between the slave address and the slave address received after bus release (when RELD is 1). Address match interrupt (IRQCSI) occurring normally when WUP is 1 is used for address match detection. Detect that the slave is selected or unselected according to the slave address when WUP is 1.

Caution 2: To detected that a slave is selected or unselected with no interrupt when WUP is 0, do not use the address match detection method and transfer previously programmed commands.

- Error detection

  [2-line serial I/O mode, SBI mode]

  The SVA register is used for error detection when the uPD75028(A) transmits address, command, or data as the master device or transmits data as a slave device.

(4) Signals

Table 4-7 lists the signals. Figs. 4-33 to 4-38 show the signals and flag operation.

Table 4-7 Signal in SBI Mode (1)

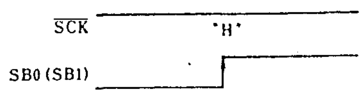
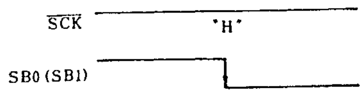
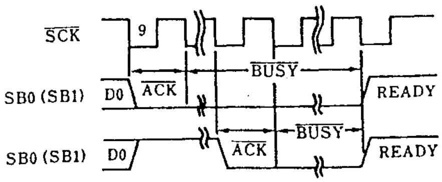
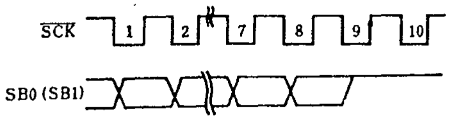
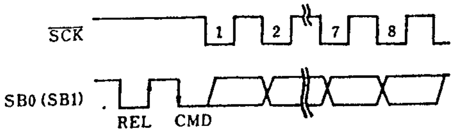
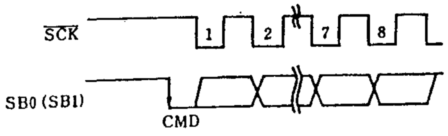
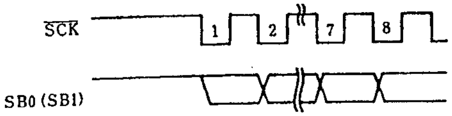
Signal name	Output device	Definition	Timing chart	Output conditions	Flag affection	Signal meaning
Bus release signal (REL)	Master	SB0 (or SB1) rising edge when $\overline{SCK}$ = 1		. RELT is set	. RELD is set . CMDD is clear	When this signal is followed by CMD signal output, it indicates that transmit data is address.
Command signal (CMD)	Master	SB0 (or SB1) falling edge when $\overline{SCK}$ = 1		. CMDT set	. CMDD is set	(1) After REL signal is output, transmit data is address. (2) When REL signal is not output, transmit data is command.
Acknowledge signal ($\overline{ACK}$)	Master/ Slave	Low signal output to SB0 (or SB1) during the period of one $\overline{SCK}$ clock after serial reception is complete	 $\overline{SCK}$ 9 SB0(SB1) D0 $\overline{ACK}$ $\overline{BUSY}$ READY SB0(SB1) D0 $\overline{ACK}$ $\overline{BUSY}$ READY	1 ACKE = 1 2 ACKT set	. ACKD is set	Reception completion
Busy signal ($\overline{BUSY}$)	Slave	Low signal output to SB0 (or SB1) following acknowledge signal		. BSYE = 1	——	Serial transfer is disabled because processing is being performed.
Ready signal (READY)	Slave	High signal output to SB0 (or SB1) before serial transfer start and after it is complete		1) BSYE = 0 2) Execution of data instruction into SIO (transfer start indication)	——	Serial transfer is enabled.

Table 4-7 Signal in SBI Mode (II)

Signal name	Output device	Definition	Timing chart	Output condition	Flag affection	Signal meaning
Serial clock (SCK)	Master	Synchronizing clock for output of address/command/ data, ACK signal, synchronous BUSY signal, etc. Address/command/ data is transferred on the first eight serial clocks.		Execution of data write instruction into SIO when CSIE = 1 (serial transfer start indication) (Note 2)	IRQCSI is set on the rising edge of the ninth clock (Note 1)	Signal output timing to serial data bus
Address (A7-A0)	Master	8-bit data transferred in synchronization with SCK after REL signal and CMD signal are output.				Slave device address on serial bus
Command (C7-C0)	Master	8-bit data transferred in synchronization with SCK after only CMD signal is output without REL signal output.				Command, message, etc., to slave device
Data (D7-D0)	Master/ Slave	8-bit data transferred in synchronization with SCK when neither REL nor CMD sinal is output.				Data processed by master or slave

Note 1: When WUP = 0, IRQCSI is always set on the rising edge of the ninth SCK clock.
When WUP = 1, IRQCSI is set on the rising edge of the ninth SCK clock only when address is received and matches the value in the slave address register (SVA).

Note 2: In the BUSY state for data transmission and reception (transfer), transfer is started after the READY state is entered.

Fig. 4-33 RELT, CMDT, RELD, CMDD Operation (master)

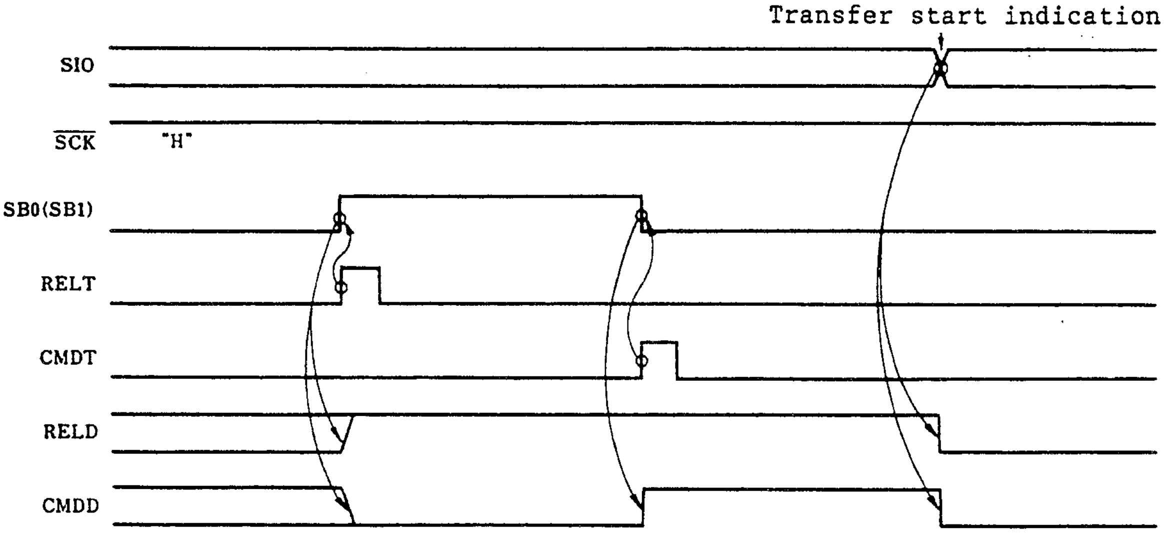


Fig. 4-34 RELT, CMDT, RELD, CMDD Operation (slave)

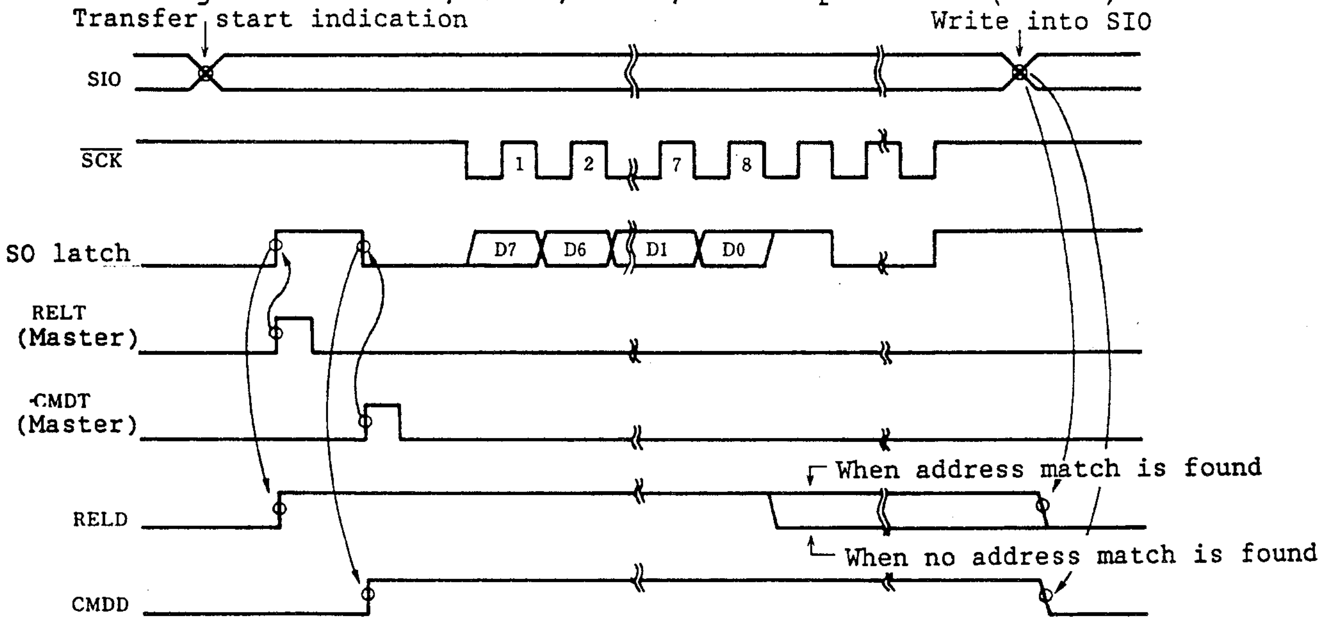
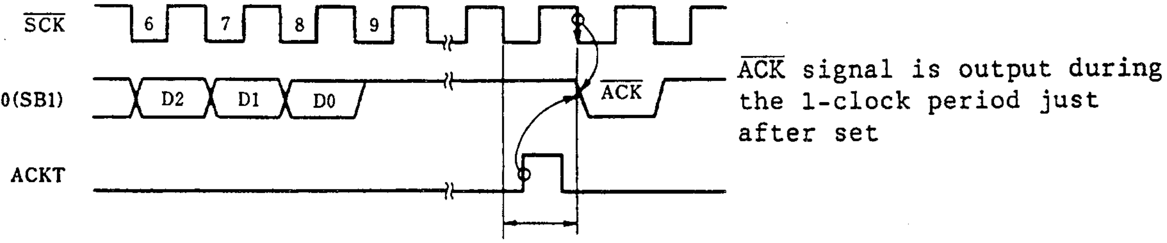

Transfer start indication
Write into SIO
SIO
SCK
1
2
7
8
SO latch
D7
D6
D1
D0
RELT
(Master)
CMDT
(Master)
RELD
When address match is found
When no address match is found
CMDD

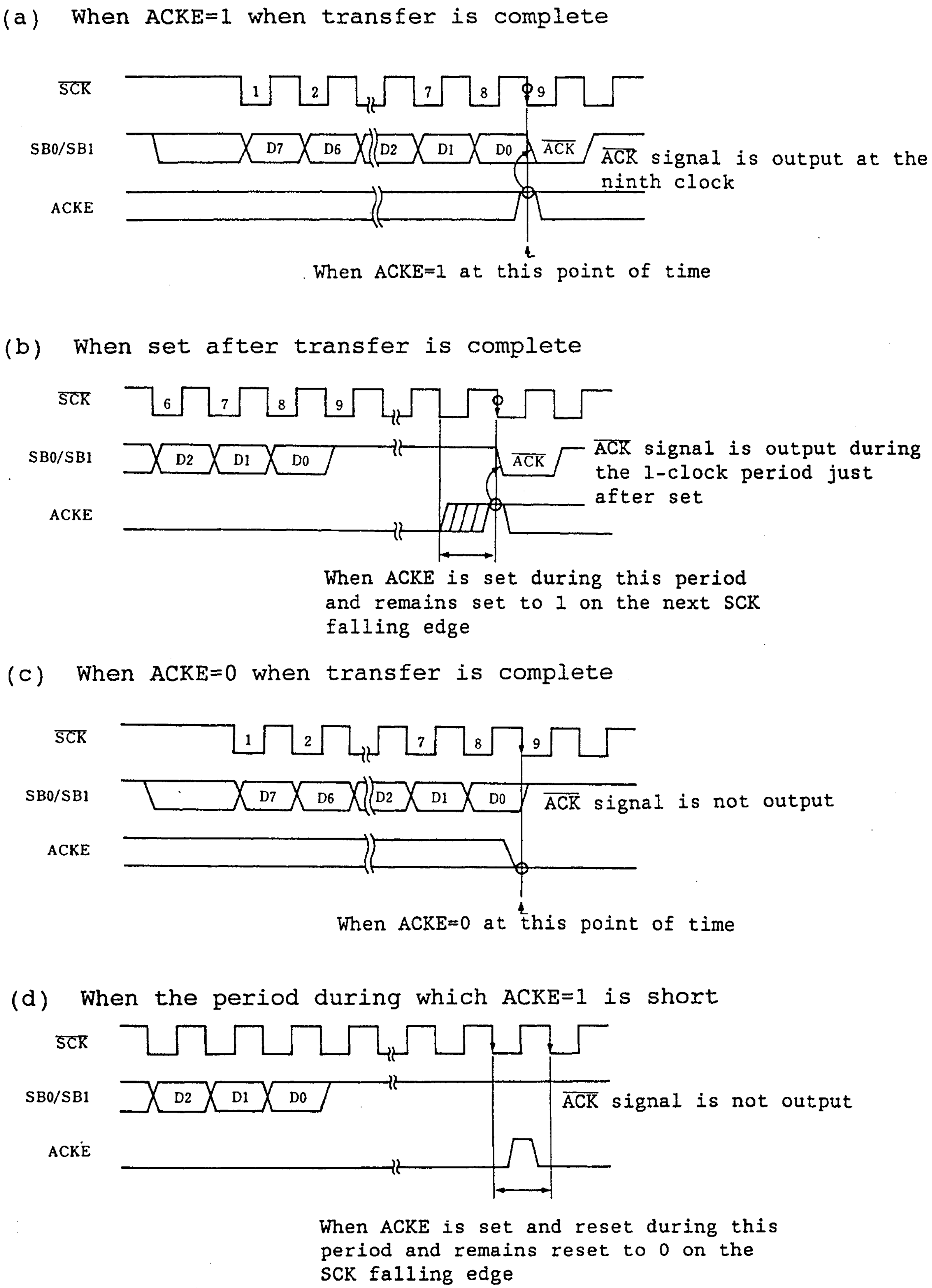
Fig. 4-35 ACKT Operation


SCK
6
7
8
9
SB0(SB1)
D2
D1
D0
ACK
ACKT
ACK signal is output during the 1-clock period just after set

When ACKT is set during this period

Caution: Do not set ACKT before transfer terminates.

Fig. 4-36 ACKE Operation



(a) When ACKE=1 when transfer is complete

$\overline{\text{ACK}}$ signal is output at the ninth clock

When ACKE=1 at this point of time

(b) When set after transfer is complete

$\overline{\text{ACK}}$ signal is output during the 1-clock period just after set

When ACKE is set during this period and remains set to 1 on the next SCK falling edge

(c) When ACKE=0 when transfer is complete

$\overline{\text{ACK}}$ signal is not output

When ACKE=0 at this point of time

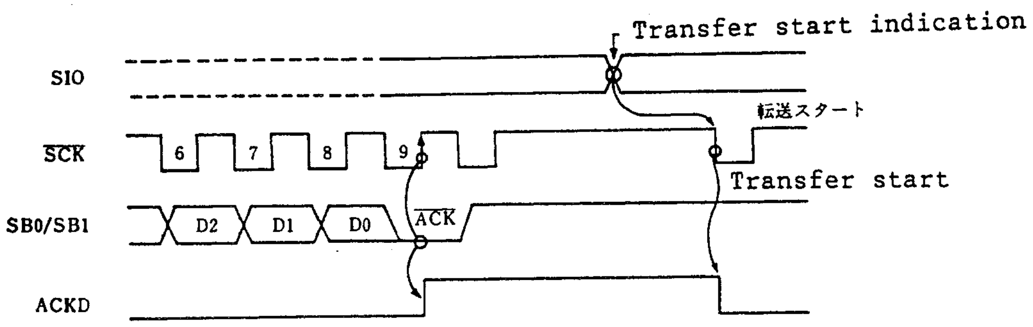
(d) When the period during which ACKE=1 is short

$\overline{\text{ACK}}$ signal is not output

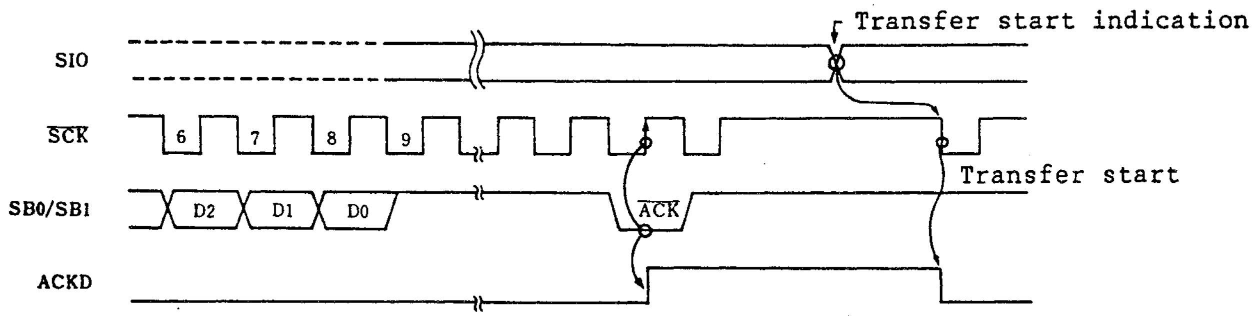
When ACKE is set and reset during this period and remains reset to 0 on the SCK falling edge

Fig. 4-37 ACKD Operation

(a) When $\overline{\text{ACK}}$ signal is output during the period of the ninth $\overline{\text{SCK}}$ clock



(b) When $\overline{\text{ACK}}$ signal is output after the ninth $\overline{\text{SCK}}$ clock



(c) Reset timing when transfer start indication is given during BUSY

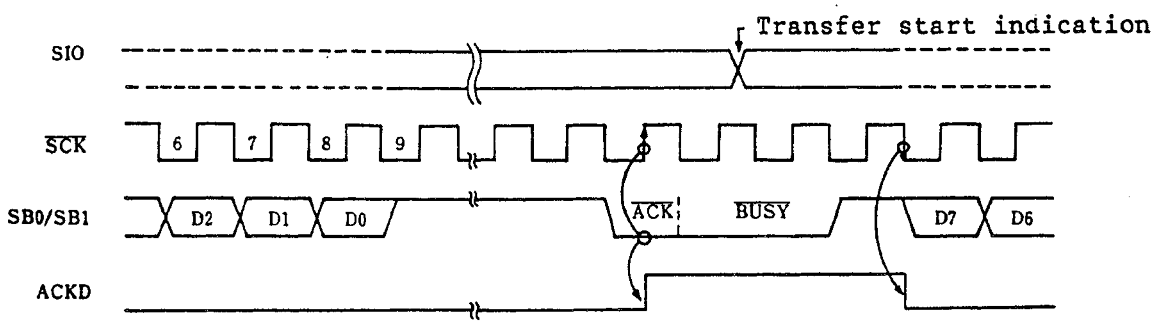
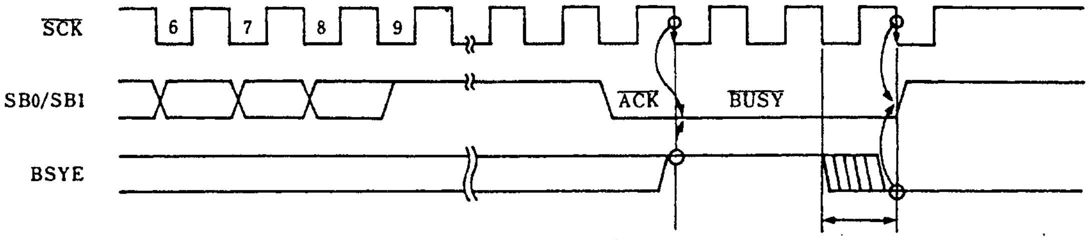


Fig. 4-38 BSYE Operation



When BSYE=1 at this point of time

When BSYE is reset during this period and remains reset to 0 on the $\overline{SCK}$ rising edge

(5) Serial interface operation

(a) Operation stop mode

The operation stop mode is used to reduce power consumption when serial transfer is not executed. In the mode, the shift register does not perform shift operation and can be used as a normal 8-bit register. When the reset signal is input, the operation stop mode is entered.

The P02/SO/SB0 and P03/SI SB1 pins are fixed to the input port mode. The P01/$\overline{\text{SCK}}$ pin can be used as an input port pin by setting the serial operation mode register.

(b) 3-line serial I/O mode operation

The 3-line serial I/O mode enables the serial interface to be connected to the 75X series, 78K series, various peripheral I/O devices, etc.

The three lines of serial clock ($\overline{\text{SCK}}$), serial output (SO), and serial input (SI) are used for communication.

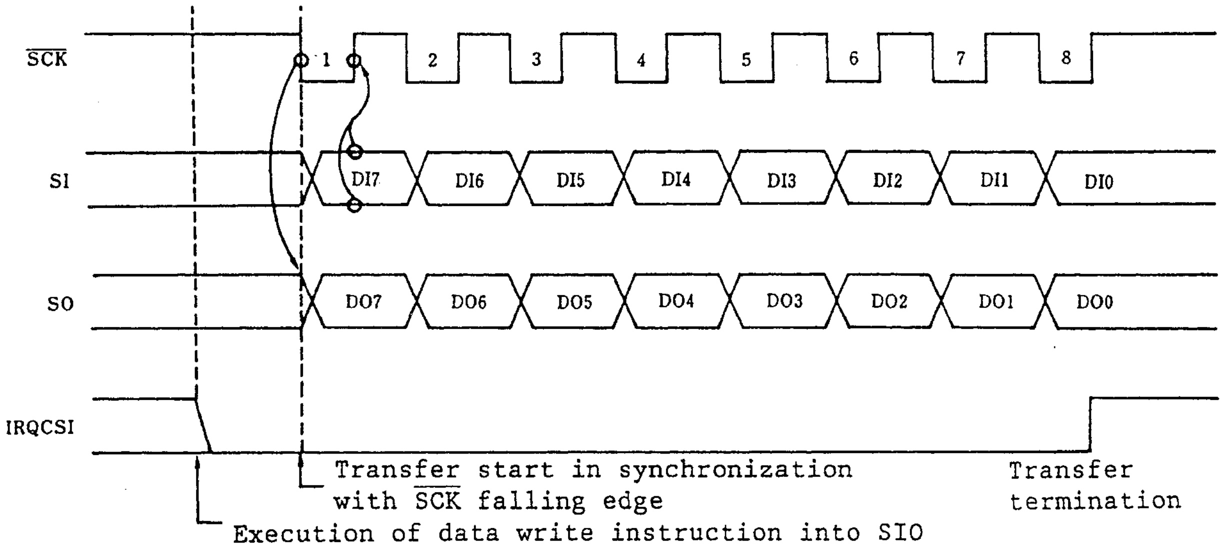
(i) Communication operation

The 3-line serial I/O mode transfers data in 8-bit units. Data is transferred one bit at a time in synchronization with a given serial clock.

Shift register shift operation is performed in synchronization with the serial clock ($\overline{\text{SCK}}$) falling edge. Transmit data is retained in the SO latch and output from the SO pin. On the $\overline{\text{SCK}}$ rising edge, receive data input to the SI pin is latched in the shift register.

When 8-bit transfer terminates, shift register operation automatically stops and the interrupt request flag (IRQSIO) is set.

Fig. 4-39 3-line Serial I/O Mode Timing



Transfer start in synchronization with $\overline{\text{SCK}}$ falling edge

Transfer termination

Execution of data write instruction into SIO (transfer start indication)

Since the SO pin becomes CMOS output and outputs the SO latch state, the SO pin output state can be handled by setting the RELT and CMDT bits.

Do not perform this operation during serial transfer.

In the output mode (internal system clock mode), the $\overline{\text{SCK}}$ pin output state can be controlled if the P01 output latch is handled. (See (7)).

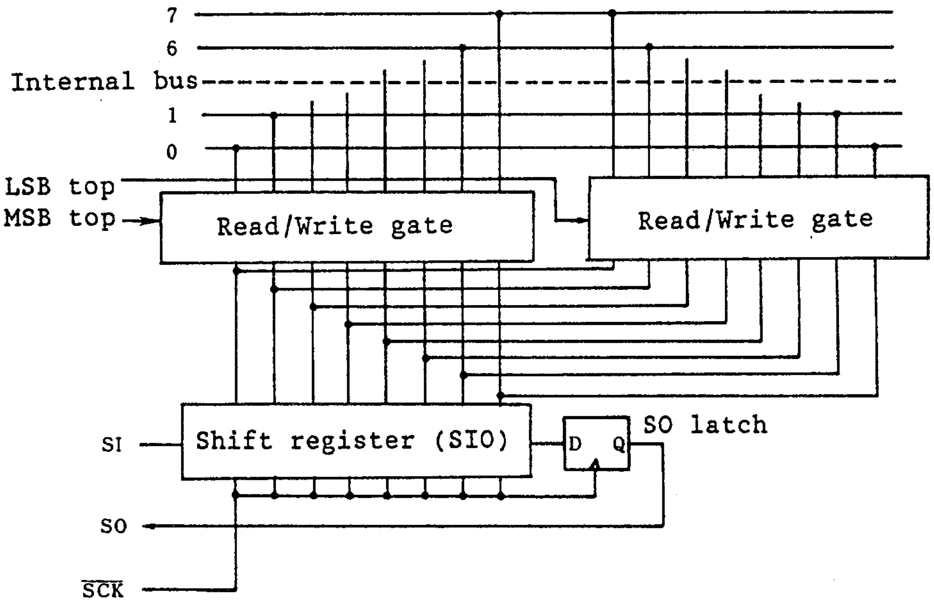
(ii) Transfer top bit change between MSB and LSB

The 3-line serial I/O mode enables selection of the most significant bit (MSB) or least significant bit (LSB) for the transfer top bit.

Fig. 4-40 shows the shift register (SIO) and internal bus configuration. As shown here, MSB and LSB can be reversed for read/write.

MSB or LSB can be specified as the transfer top bit by setting serial operation mode register (CSIM) bit 2 to 0 or 1.

Fig. 4-40 Transfer Bit Top Change Circuit



The transfer top bit is changed by changing the bit order of data write into the shift register (SIO). The SIO shift order is always the same. Change the transfer top bit between MSB and LSB before writing data into the shift register.

(c) 2-line serial I/O mode operation

The 2-line serial I/O mode can cover any desired communication format by a program.

Basically, the two lines of serial clock ($\overline{\text{SCK}}$) and serial data output/input SB0 (or SB1) are used for communication.

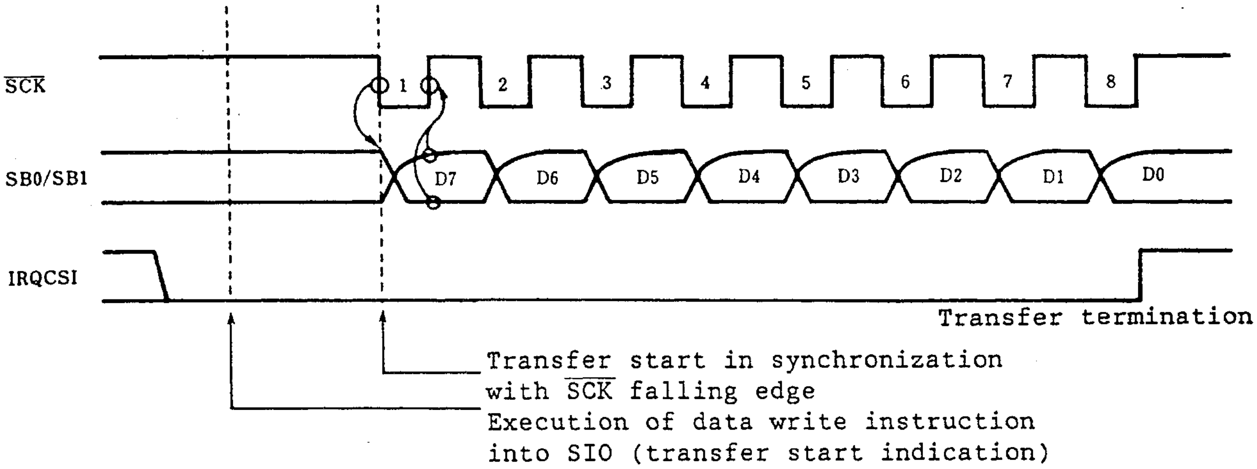
(i) Communication operation

The 2-line serial I/O mode transfers data in 8-bit units. Data is transferred one bit at a time in synchronization with a given serial clock.

Shift register shift operation is performed in synchronization with the serial clock ($\overline{\text{SCK}}$) falling edge. Transmit data is retained in the SO latch and output starting at the MSB from the SB0/P02 (or SB1/P03) pin. On the $\overline{\text{SCK}}$ rising edge, receive data input from the SB0 (or SB1) pin is latched in the shift register.

When 8-bit transfer terminates, shift register operation automatically stops and the interrupt request flag (IRQSIO) is set.

Fig. 4-41 2-Line Serial I/O Mode Timing


$\overline{\text{SCK}}$
SB0/SB1
IRQCSI
1
2
3
4
5
6
7
8
D7
D6
D5
D4
D3
D2
D1
D0
Transfer termination
Transfer start in synchronization with $\overline{\text{SCK}}$ falling edge
Execution of data write instruction into SIO (transfer start indication)

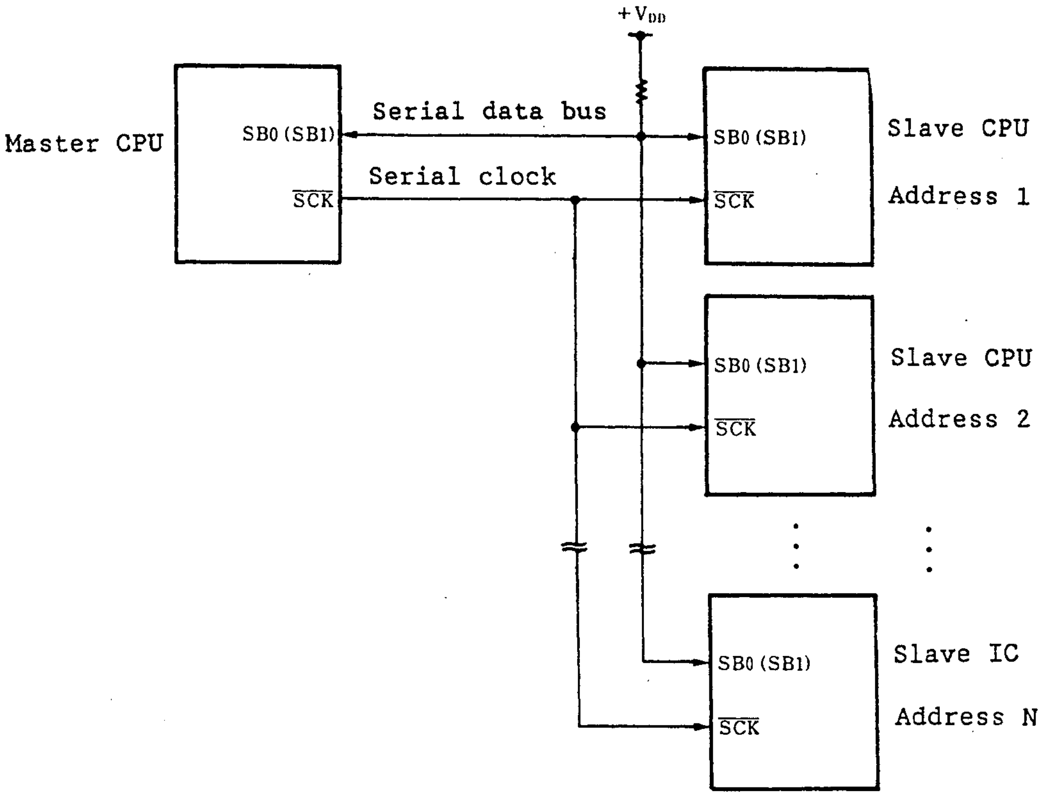
The SB0 or SB1 pin specified for the serial data bus becomes N-ch open drain input/output, thus must be pulled high with an external pull-up resistor.

Since the SB0 (or SB1) pin outputs the SO latch state, the SB0 (or SB1) pin output state can be handled by setting the RELT and CMDT bits.
Do not perform this operation during serial transfer.
In the output mode (internal system clock mode), the $\overline{\text{SCK}}$ pin output state can be controlled if the P01 output latch is handled. (See (7)).

(d) SBI mode operation

The SBI (serial bus interface) is a high-speed serial interface system which is compliant with the NEC serial bus format.
The SBI is a high-speed serial bus of a single master; bus configuration function is added to the clocked serial I/O system so that the master can communicate with a number of devices by using two signal lines. When microcomputers and peripheral ICs make up a serial bus, the number of ports and wiring on the printed circuit boards can be reduced.
Fig. 4-42 shows an SBI system configuration example.

Fig. 4-42 SBI System Configuration Example



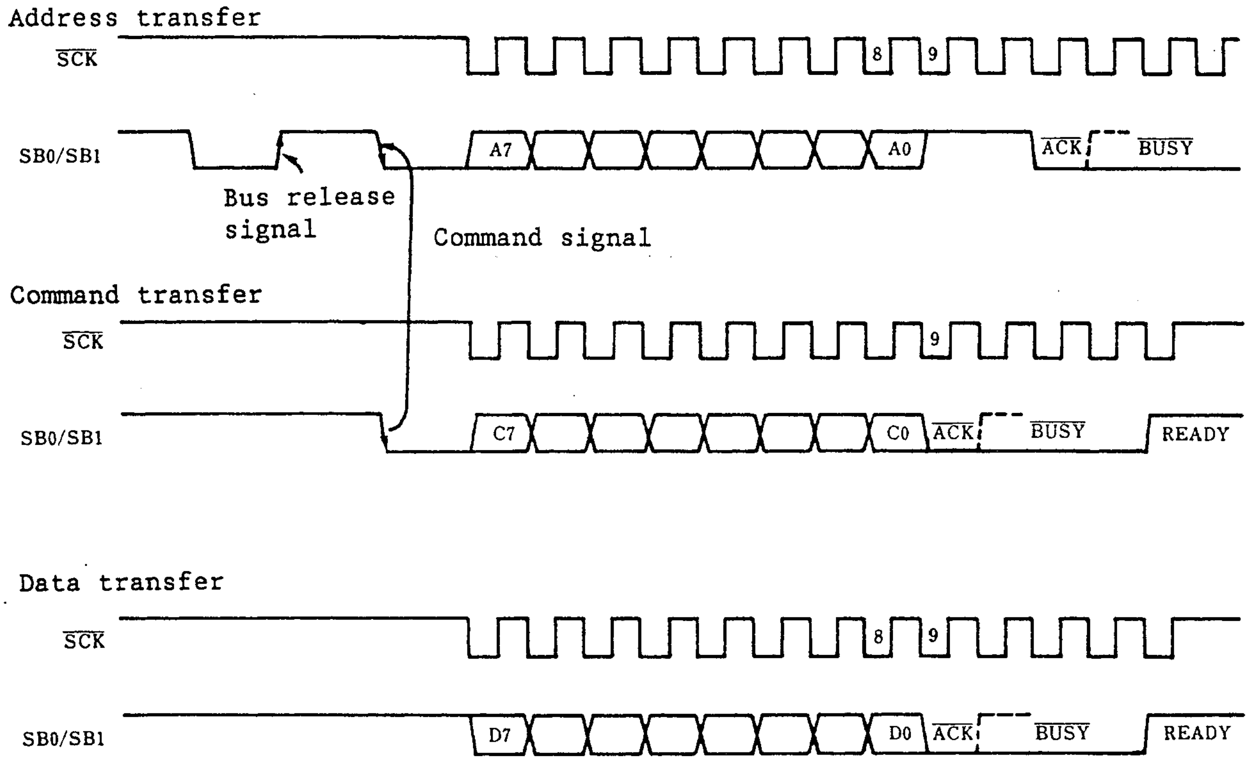
Caution 1: In the SBI mode, the serial data bus pin SB0 (or SB1) becomes open drain output, thus the serial data bus line is placed in the wired-OR state. The serial data bus line must be pulled high with a pull-up resistor.

2: For master and slave exchange processing, the serial clock line ($\overline{\text{SCK}}$) must also be pulled high with a pull-up resistor because serial clock line ($\overline{\text{SCK}}$) input and output are changed asynchronously between the master and slave.

(i) SBI function

- Address, command, and data judgment function
  Serial data is classified into address, command, and data.
- Chip select function by address
  The master selects a slave chip by transmitting the slave address.
- Wake-up function
  Each slave can easily judge that its address is received (its chip is selected) by the wake-up function (that can be set and released by software). If the wake-up function is set, an interrupt (IRQCSI) occurs when the received address matches the slave address. Thus, the CPUs other than the selected slave can operate independently of serial communication if the master communicates with a number of devices.
- Acknowledge signal ($\overline{\text{ACK}}$) control function
  The acknowledge signal to acknowledge that serial data is received is controlled.
- Busy signal ($\overline{\text{BUSY}}$) control function
  The busy signal to inform the master that the selected slave is busy is controlled.

Fig. 4-43 SBI Transfer Timings



(ii) Communication operation

In the SBI mode, the master normally selects one slave device with which the master is to communicate among devices by outputting the slave address to the serial bus.

After the device with which the master is to communicate is determined, commands and data are transferred between the master and slave for serial communication. Fig. 4-44 to 4-47 show data communication timing charts.

In the SBI mode, shift register shift operation is performed in synchronization with the serial clock $(\overline{SCK})$ falling edge. Transmit data is output to the SO latch starting at the MSB from the SB0/P02 or SB1/P03 pin. On the $\overline{SCK}$ rising edge, receive data input to the SB0 (or SB1) pin is latched in the shift register.

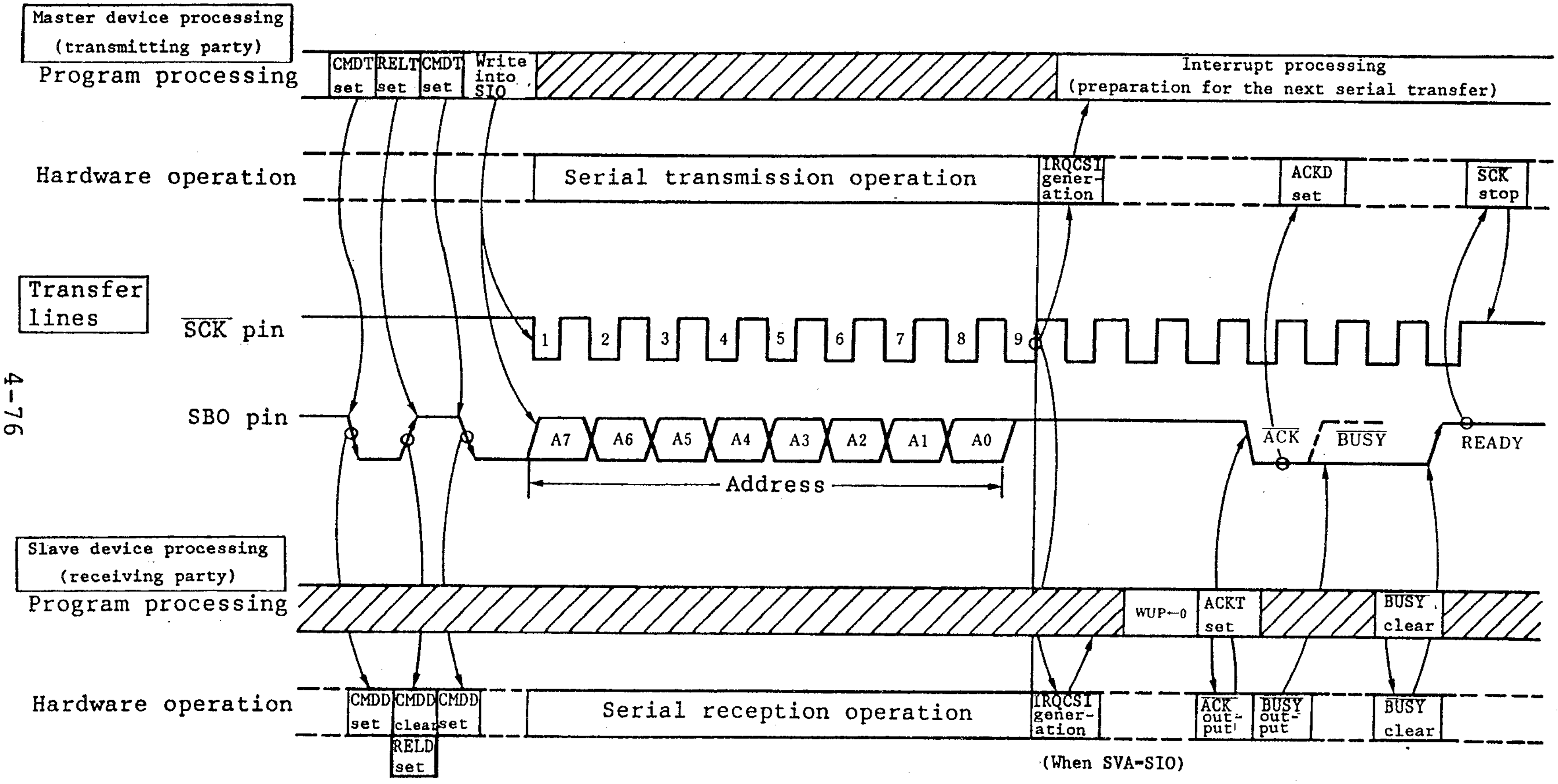


Fig. 4-44 Address Transmission Operation from Master Device to Slave Device (when WUP=1)

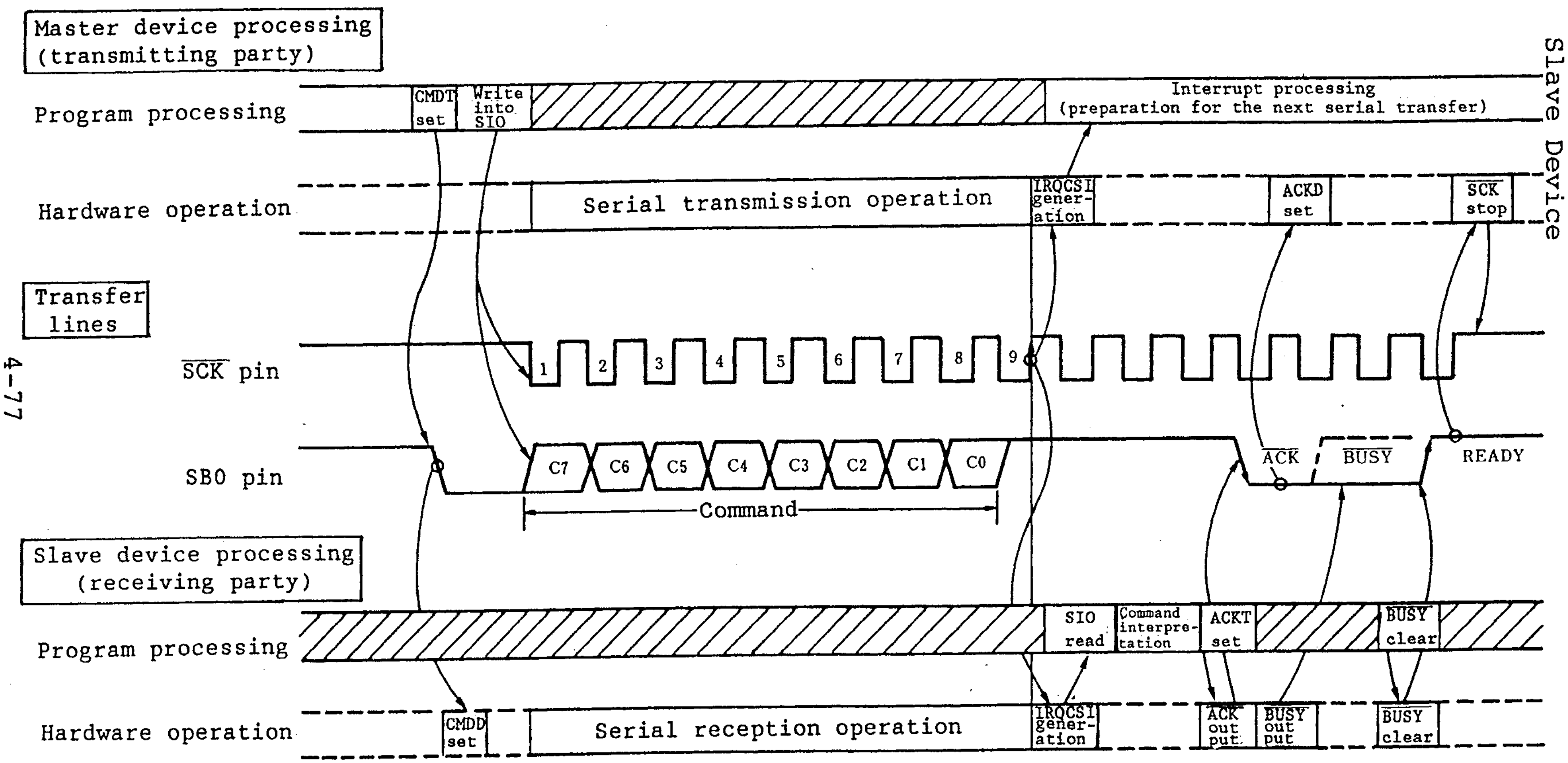


Fig. 4-45 Command Transmission Operation from Master Device to Slave Device

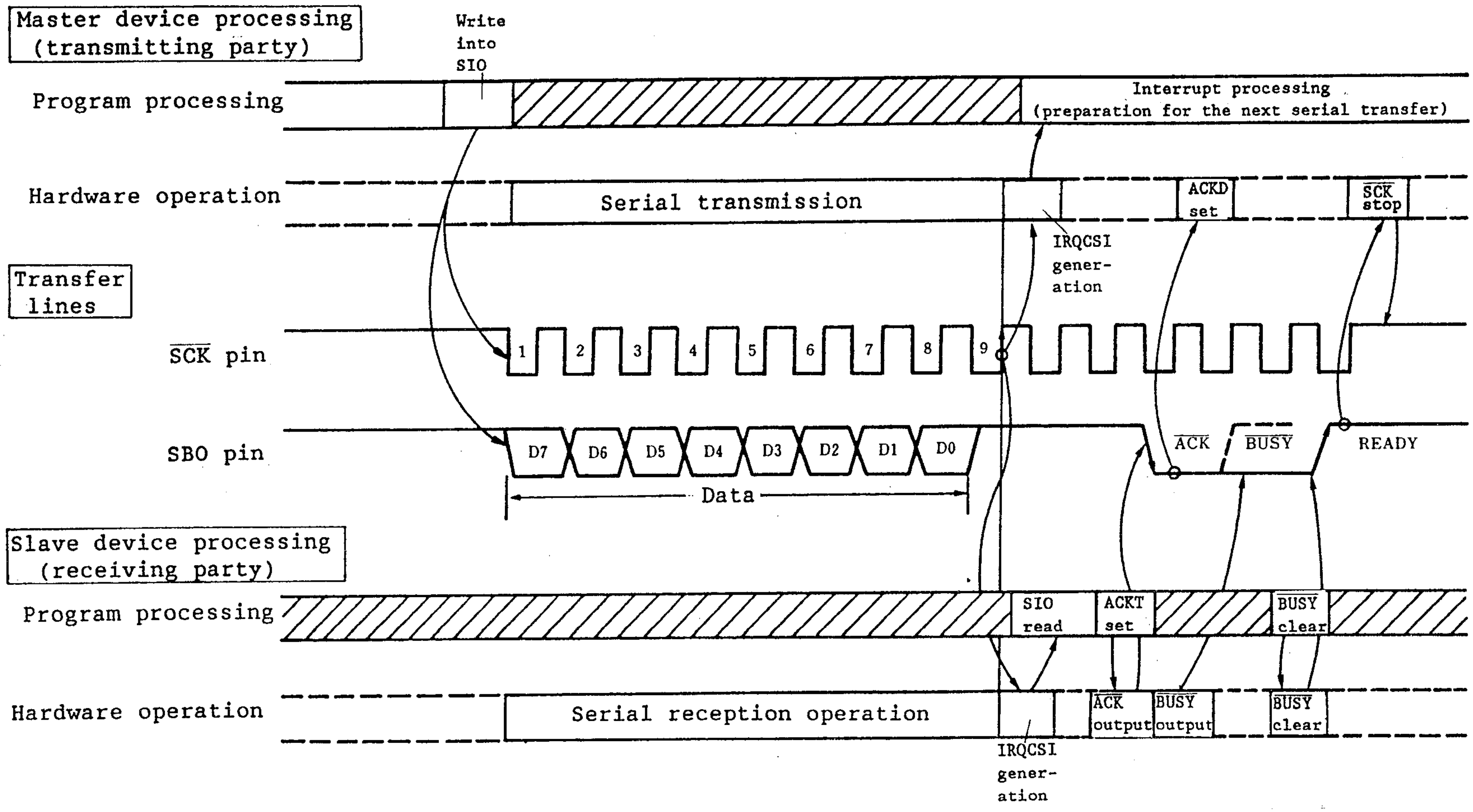


Fig. 4-46 Data Transmission Operation from Master Device to Slave Device

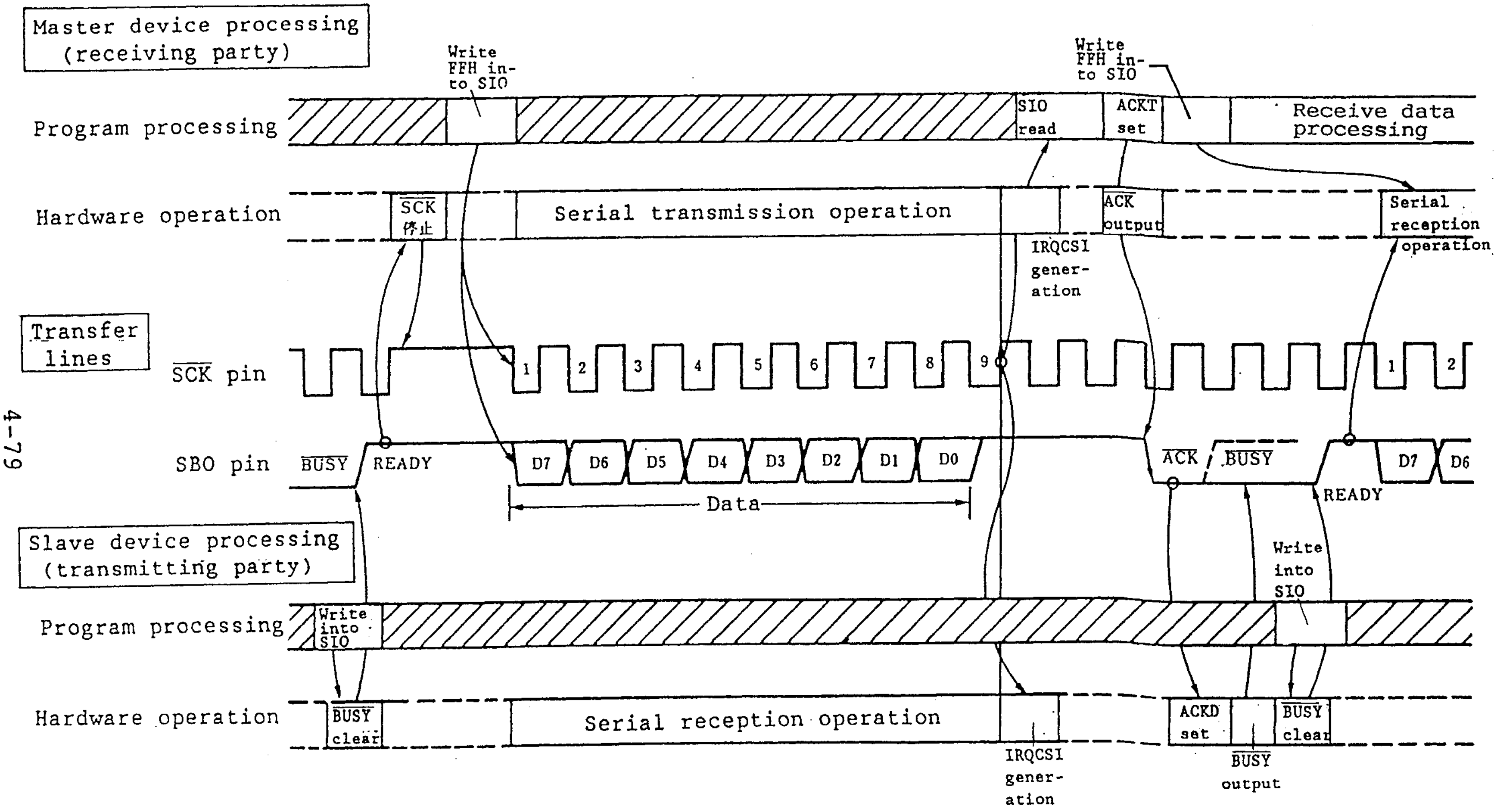


Fig. 4-47 Data Transmission Operation from Slave Device to Master Device

(6) Transfer start in each mode

In any of the 3-line and 2-line serial I/O modes and the SBI mode, serial transfer is started by setting transfer data in the shift register only if the following two conditions are satisfied:

- serial interface operation enable/disable bit (CSIE)=1; and
- after serial transfer of 8-bit data, internal serial clock stops, or $\overline{\text{SCK}}$ is high

Caution: Although CSIE is set to 1 after data is written into the shift register, transfer does not start.

When 8-bit data transfer terminates, serial transfer automatically stops and the interrupt request flag (IRQCSI) is set.

[Caution when 2-line serial I/O mode transfer is started]

Caution: When data is received, N-ch transistor must be turned off. Prewrite FFH into the SIO register.

[Cautions when SBI mode transfer is started]

Caution 1: When data is received, N-ch transistor must be turned off. Prewrite FFH into the SIO register.
However, since the N-ch transistor is always off when the wake-up function specification bit (WUP) is set to 1, FFH need not be written into the SIO register before data is received.

2: If data is written into the SIO register when a given slave is busy, the data is

not lost.
When the busy state is released and SB0 (or SB1) input becomes high (ready state), transfer starts.

Example: At the same time that data in RAM specified in the HL register pair is transferred to the SIO register, data in the SIO register is read into the accumulator and serial transfer is started.

```
MOV  XA, @HL  ; Take out transmit data from
                RAM
REL  MB15     ; Or CLR1 MBE
XCH  XA, SIO  ; Exchange  transmit  data  and
                receive data and start trans-
                fer.
```

(7) $\overline{\text{SCK}}$ pin output handling

Since the $\overline{\text{SCK}}$/P01 pin contains an output latch, static output is also enabled by software operation in addition to normal serial clock.

The $\overline{\text{SCK}}$ count can be set by software as desired by handling the P01 output latch. (The SB0/SB1 pin is controlled by setting the RELT and CMDT bits.)

The $\overline{\text{SCK}}$/P01 pin output handling method is described below:

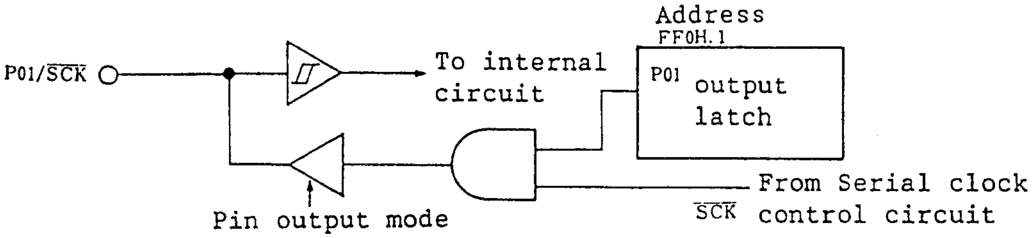
① Set the serial operation mode register (CSIM) ($\overline{\text{SCK}}$ pin: Output mode). When serial transfer stops, $\overline{\text{SCK}}$ from the serial clock control circuit is set to 1.

② Handle the P01 output latch by a bit manipulation instruction.

Example: Output one clock to the P01/$\overline{\text{SCK}}$ pin by software.

```
SEL   MB15              ; or CLR1  MBE
MOV   XA, #10000011B    ; SCK(fX/2^3), output
                          mode
MOV   CSIM, XA
CLR1   0FF0H, 1         ; SCK/P01 ←— 0
SET1   0FF0H, 1         ; SCK/P01 ←— 1
```

Fig. 4-48 $\overline{\text{SCK}}$/P01 Pin Configuration



The P01 output latch is mapped in bit 1 of address FF0H. When the $\overline{\text{RESET}}$ signal is generated, the P01 output latch is set to 1.

Caution 1: During normal serial transfer, the P01 output latch must be set to 1.

2: The P01 output latch address cannot be specified in PORT0.1 as shown below. Describe the address (FF0H.1) directly in the operand. However, when the instruction executed, MBE=1 or (MBE=1 and MBS=15) must have been set.

```
CLR1   PORT0.1  }  Cannot be used
SET1   PORT0.1  }
CLR1   0FF0H.1  }  Can be used
SET1   0FF0H.1  }
```

4.8 A/D Converter

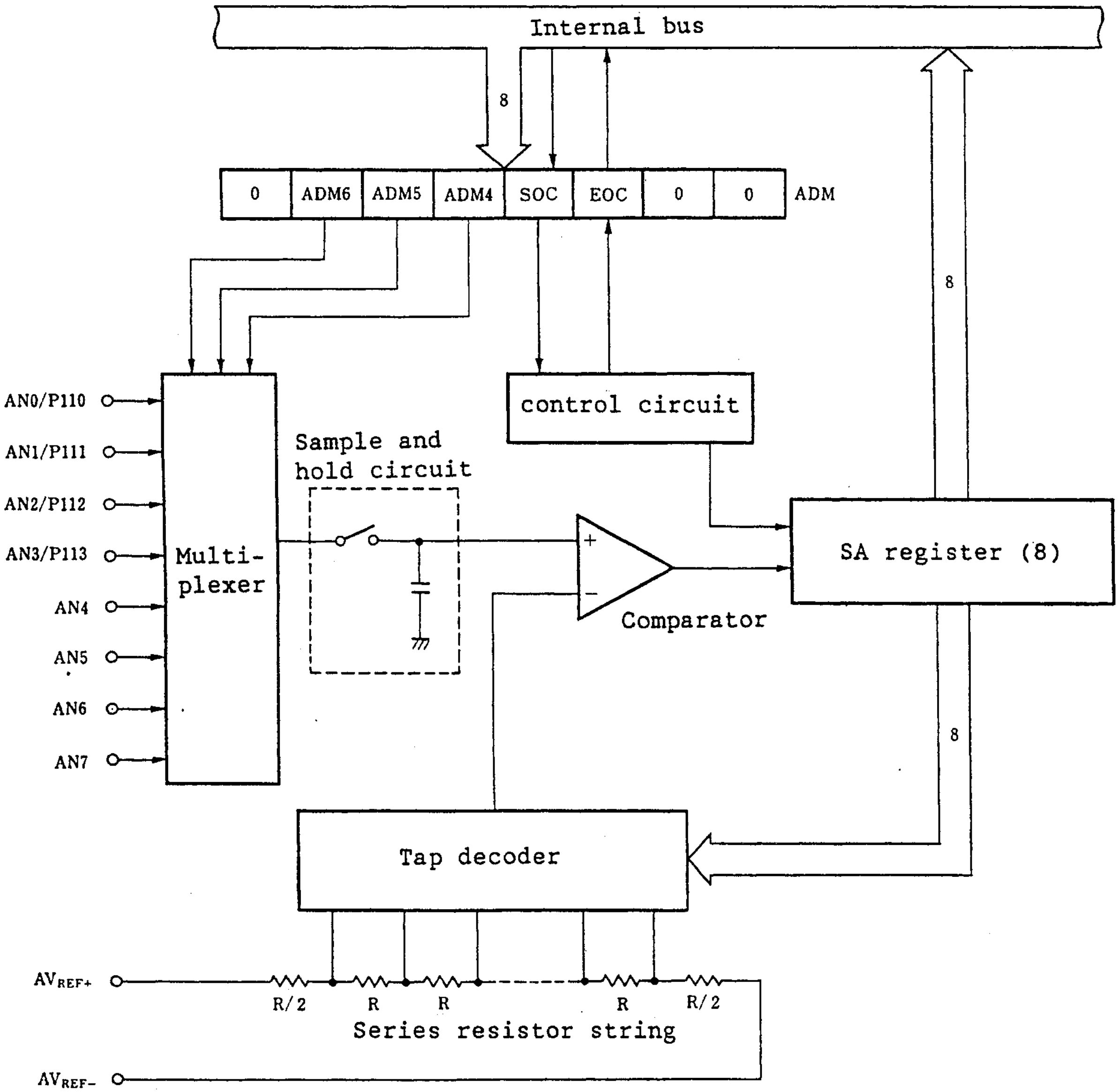
The uPD75028(A) contains an 8-bit resolution analog/digital (A/D) converter which has eight analog input channels (AN0-AN7).

The A/D converter adopts successive approximation.

(1) A/D converter configuration

Fig. 4-49 shows the A/D converter block diagram.

Fig. 4-49 A/D Converter Block Diagram



(a) A/D converter pins

(i) AN0-AN7

AN0-AN7 are 8-channel analog signal input pins to the A/D converter. The analog signals to be converted into digital form are input. AN0/P110-AN3/P113 are also used for port 11 pins.

The A/D converter contains a sample and hold circuit. Analog input voltage is held internally during A/D conversion.

Caution: Use the AN0-AN7 input voltages within the specified limits. Particularly, if voltage more than V_{DD}, less than V_{SS} (even within the absolute maximum range) is input, the conversion value of the channel becomes undefined and the conversion values of other channels may be adversely affected.

(ii) AV_{REF+} and AV_{REF-}

AV_{REF+} and AV_{REF-} are A/D converter reference voltage input pins.

Based on the voltage applied between AV_{REF+} and AV_{REF-}, the signals input to AN0-AN7 are converted into digital signals.

(iii) AV_{DD}

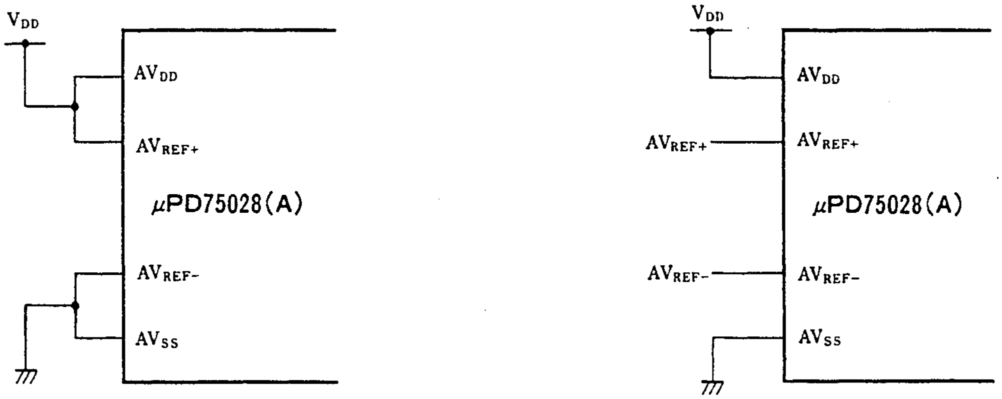
AV_{DD} is an A/D converter power supply pin. Input V_{DD} level.

(iv) AV_{SS}

AV_{SS} is an A/D converter GND pin. Input V_{SS} level.

Fig. 4-50 shows a pin connection example.

Fig. 4-50 Pin Connection Example



(1) $AV_{DD} = AV_{REF+} = V_{DD}$
$AV_{SS} = AV_{REF-} = V_{SS}$

(2) $AV_{REF+} \neq AV_{DD}$ $(AV_{DD} > AV_{REF+})$
$AV_{REF-} \neq AV_{SS}$ $(AV_{SS} < AV_{REF-})$

(b) A/D conversion mode register (ADM)

The A/D conversion mode register (ADM) is an 8-bit register for analog input channel selection, conversion start indication, and conversion end detection. ADM is set by executing an 8-bit handling instruction.

Bit 2 (EOC) and bit 3 (SOC) can be handled bit-wise. When the $\overline{\text{RESET}}$ signal is generated, ADM is initialized to 04H. (Only EOC is set to 1 and other bits are reset to 0.)

Fig. 4-51 A/D Conversion Mode Register Format

Address	7	6	5	4	3	2	1	0	Symbol
FD8H	0	ADM6	ADM5	ADM4	SOC	EOC	0	0	ADM

Conversion end detection flag (R)

EOC	0	Converting
	1	Conversion end

Conversion start indication bit (W)

SOC	When this bit is set to 1, A/D conversion is started. After conversion starts, automatically the bit is reset to 0.

Analog channel selection bits (W)

ADM6	ADM5	ADM4	Analog channel
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

Caution: A/D conversion starts a maximum of $2^4/f_X$ seconds (3.81 us when f_X = 4.19 MHz) late after SOC is set. (See (2) below.)

(c) SA (successive approximation) register

The SA register is an 8-bit register which stores the A/D conversion result.

SA is read by executing an 8-bit handling instruction. SA is a read-only register where write operation and bit manipulation cannot be performed.

When the $\overline{\text{RESET}}$ signal is generated, the SA contents become as follows:

- retained during standby mode; or
- undefined during operation

Caution: If ADM register bit 3 (SOC) is set to 1 to start A/D conversion, the conversion result in SA is destroyed and SA becomes undefined until new conversion result is stored.

(2) A/D converter operation

The analog input signal to be converted into digital form is specified by setting A/D conversion mode register bits 6-4 (ADM6-ADM4).

A/D conversion is started by setting ADM bit 3 (SOC) to 1. After SOC is set, automatically it is reset to 0. A/D conversion is performed by using successive approximation by the hardware and 8-bit data of the conversion result is stored in the SA register. When the conversion ends, ADM bit 2 (EOC) is set to 1.

Fig. 4-52 shows the A/D conversion timing chart.

Use of the A/D converter is as follows:

① Select analog input channel by setting ADM6-ADM4.

② Give A/D conversion start indication by setting SOC to 1.

③ Wait for the A/D conversion to end. (Wait for EOC to be set to 1 or wait on software timer.)

④ Read the A/D conversion result by reading the SA register.

Caution 1: Steps ① and ② can also be performed at the same time.

Caution 2: It takes a maximum of $2^4/f_X$ seconds (3.81 us when f_X = 4.19 MHz) until A/D conversion starts and EOC is reset to 0 after SOC is set to 1. Thus, test EOC in the time listed in Table 4-8 after SOC is set. The A/D conversion time is also listed in Table 4-8.

Table 4-8 SCC and PCC Setting

Value set in SCC, PCC				A/D conversion time	Wait time until EOC test after SOC is set	Wait time until A/D conversion end after SOC is set
SCC3	SCC0	PCC1	PCC0			
0	0	0	0	$168/f_X$ (40.1 us/f_X = 4.19 MHz)	Wait not required	Three machine cycles
		1	0		Two machine cycles	21 machine cycles
		1	1		Four machine cycles	42 machine cycles
0	1	x	x		Wait not required	Wait not required
1	x	x	x	Conversion operation stop	——	——

Remarks: x: don't care

Fig. 4-52 A/D Conversion Timing Chart

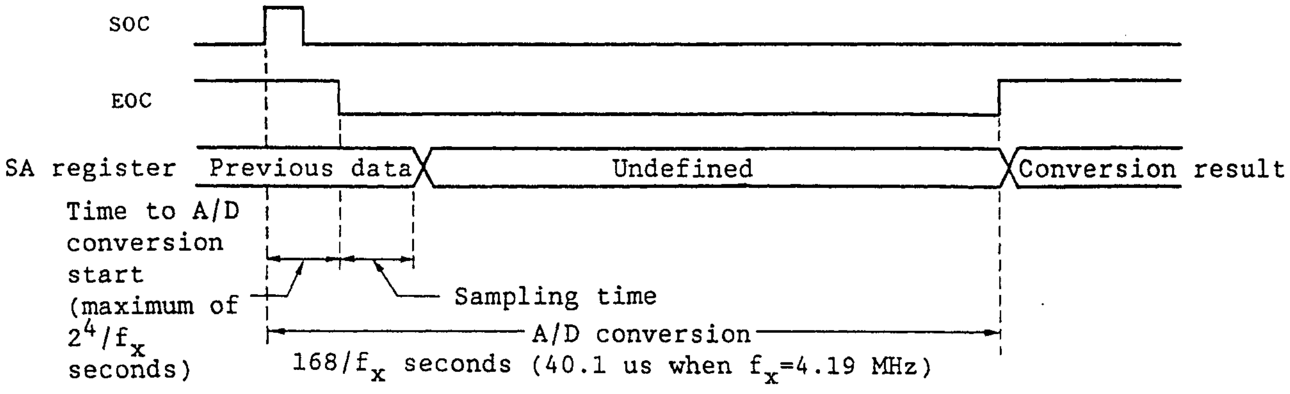
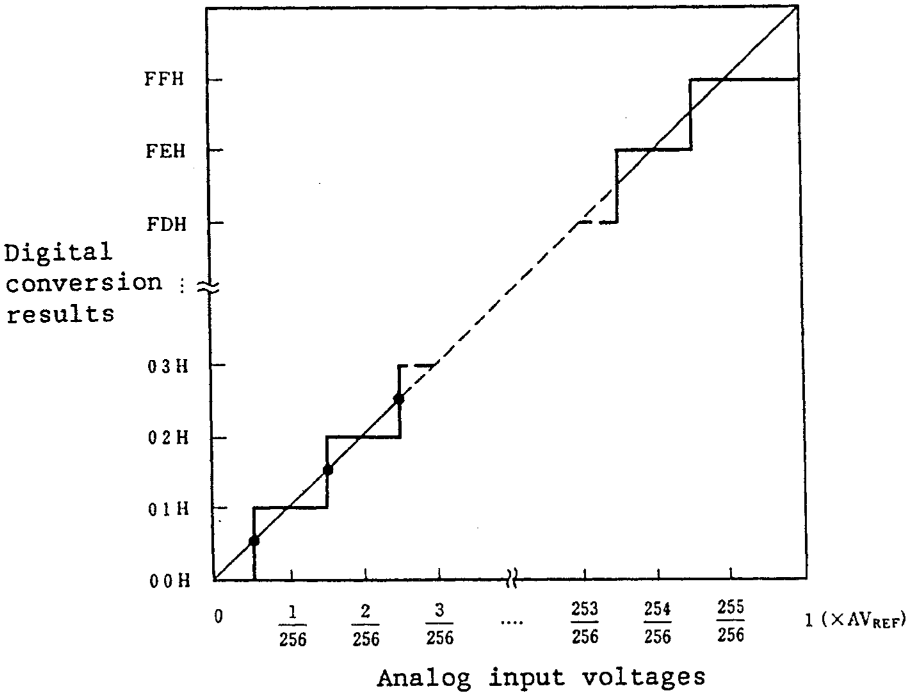

SOC
EOC
SA register
Previous data
Undefined
Conversion result
Time to A/D conversion start (maximum of $2^4/f_x$ seconds)
Sampling time
A/D conversion
$168/f_x$ seconds (40.1 us when f_x=4.19 MHz)

Fig. 4-53 shows the correspondence between analog input voltages and 8-bit digital data after A/D conversion.

Fig. 4-53 Relationship between Analog Input Voltages and A/D Conversion Results (ideal instance)



(3) Caution on standby mode

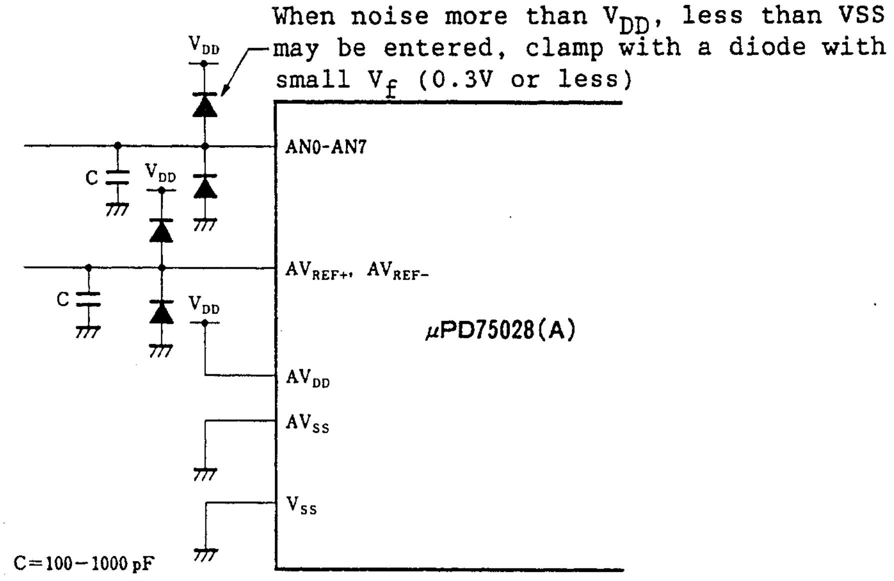
The A/D converter operates on main system clock. Thus, A/D converter operation stops in the STOP mode or the HALT mode with subsystem clock. At the time, current also flows into the AV_{REF+} pin, thus must be cut to lessen power consumption in the entire system.

(4) Cautions on use

(a) AN0-AN7 input range
Use the AN0-AN7 input voltages within the specified limits. Particularly, if voltage more than V_{DD}, less than V_{SS} (even within the absolute maximum range) is input, the conversion value of the channel becomes undefined and the conversion values of other channels may be adversely affected.

(b) Noise countermeasure
To maintain 8-bit resolution, use care to avoid noise to the AV_{REF+}, AV_{REF-}, and AN0-AN7 pins. The higher the analog input source output impedance, the greater is the affection. Connect external resistor and capacitor as shown in Fig. 4-54.

Fig. 4-54 Analog Input Pin Treatment


When noise more than V_{DD}, less than VSS may be entered, clamp with a diode with small V_f (0.3V or less)
V_{DD}
V_{DD}
V_{DD}
C
C
AN0-AN7
AV_{REF+}, AV_{REF-}
μPD75028(A)
AV_{DD}
AV_{SS}
V_{SS}
C=100−1000pF

(c) AN0/P110-AN3/P113 pins

The analog input pins AN0-AN3 are also used for input port (PORT11) pins.

To select any of AN0-AN3 for A/D conversion, do not execute any PORT11 input instruction during the conversion. The conversion resolution may lower.

If a digital pulse is applied to pins contiguous to the pin under A/D conversion, expected A/D conversion value may not be obtained because of a coupling noise. Do not apply any pulse to the pins contiguous to the pin under A/D conversion.

Example: For A/D conversion at AN3
Do not apply any digital pulse to AN2 or AN4 during A/D conversion.

4.9 Multifunction Timer (MFT)

(1) Multifunction timer configuration

The uPD75028(A) contains one channel of the multifunction timer (MFT). Fig. 4-55 shows the multifunction timer block diagram.

(2) Multifunction timer function

MFT contains four modes as described below:

(a) 8-bit timer mode
- programmable interval timer operation
- square wave output of any frequency to PPO pin

(b) PWM output mode
- output of 6-, 7-, or 8-bit precision PWM signal to PPO pin

(c) 16-bit free-running timer mode
- interval timer operation to cause an interrupt to occur at given time intervals
- applicable as a one-shot timer

(d) Integration A/D converter mode
- output of 16-bit integration A/D converter control signal
- 13-, 14-, 15-, or 16-bit resolution can be selected

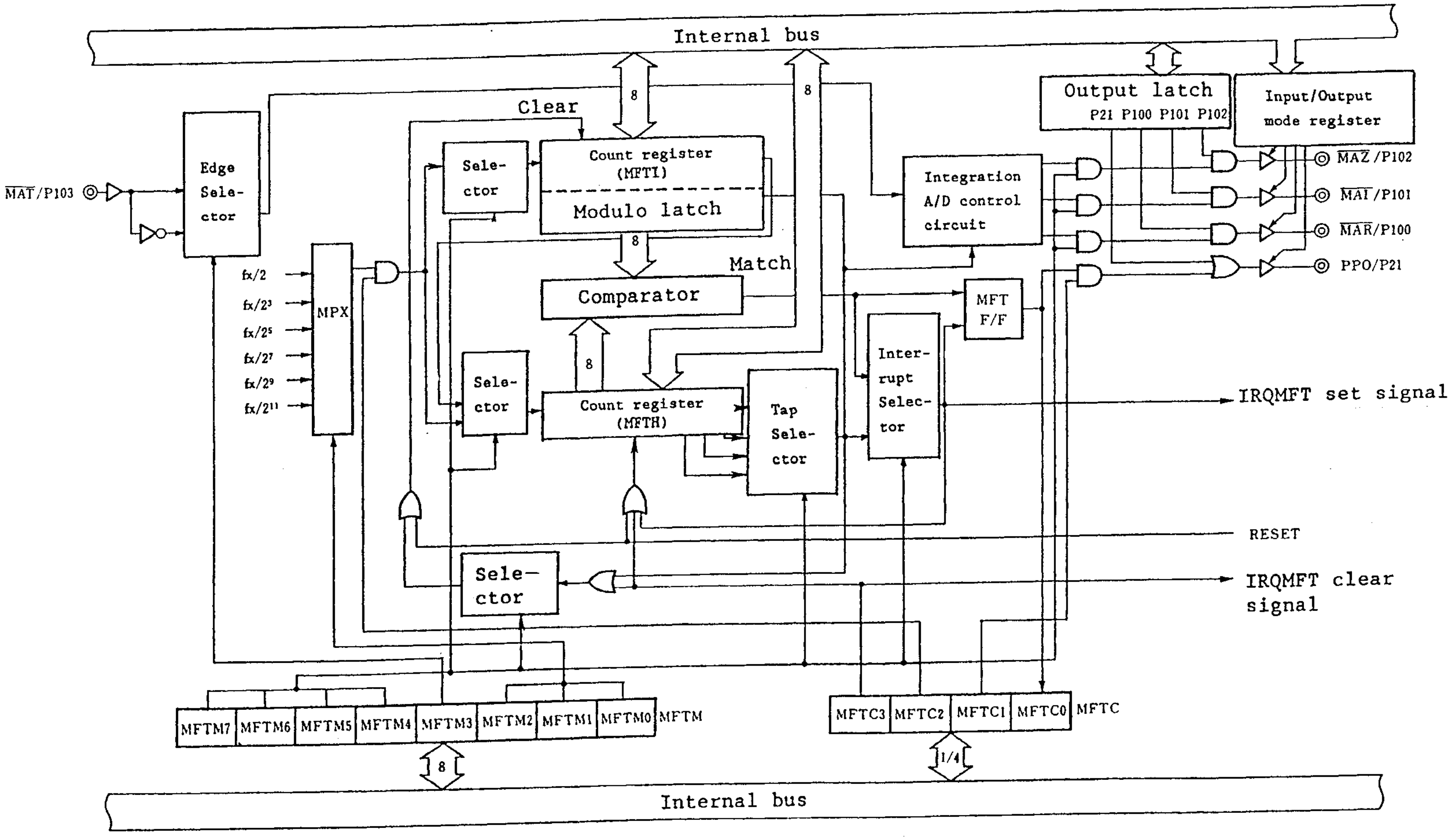


Fig. 4-55 Multifunction Timer Block Diagram

(3) Functions of main registers

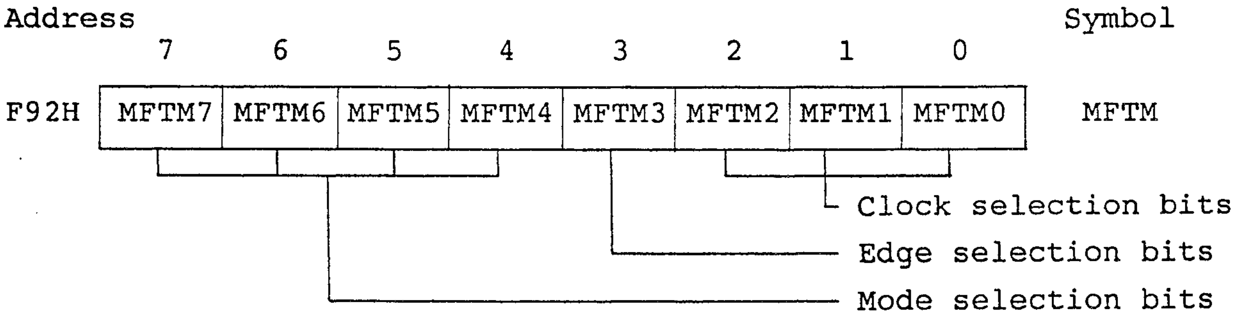
(a) MFT mode register (MFTM)

The MFT mode register (MFTM) is an 8-bit register to select the operation mode and count clock.

MFTM can be written by executing an 8-bit memory handling instruction.

When the RESET signal is generated, MFTM is initialized to 0.

Fig. 4-56 MFT Mode Register Format



Clock selection bits

MFTM2	MFTM1	MFTM0	Selected clock	Resolution
0	0	0	Undefined	——
0	0	1	$f_X/2^{11}$ (2.0 kHz)	488 us
0	1	0	$f_X/2^9$ (8.2 kHz)	122 us
0	1	1	$f_X/2^7$ (32.768 kHz)	30.5 us
1	0	0	$f_X/2^5$ (131 kHz)	7.6 us
1	0	1	$f_X/2^3$ (524 kHz)	1.9 us
1	1	0	$f_X/2$ (2.10 MHz)	0.48 us

Remarks: The value enclosed in parentheses is applied when f_X = 4.19 MHz.

Edge selection bits

MFTM3	0	Rising edge selection
	1	Falling edge selection

Caution: Select edge input to the $\overline{\text{MAT}}$/P103 pin. This enables external comparator input edge specification in the integration A/D converter mode.

Mode selection bits

MFTM7	MFTM6	MFTM5	MFTM4	Mode	Tap selection bit
0	0	0	0	Timer mode	——
			1	PWM output mode	6 bit
		1	0		7 bit
			1		8 bit
	Other than above			Undefined	——
1	0	0	0	Free-running timer mode	13 bit
			1		14 bit
		1	0		15 bit
			1		16 bit
	1	0	0	Integration A/D converter mode	13 bit
			1		14 bit
		1	0		15 bit
			1		16 bit

(b) MFT control register (MFTC)

The MFT control register (MFTC) is a 4-bit register which consists of timer output bit, timer output enable/disable bit, timer operation control bit, and initialization bit. The timer output bit (MFTC.0) indicates the MFT flip-flop contents.

The MFTC can be read and written by executing 1/4-bit memory handling instructions. (However, bit 0 can be read only and the data written into the bit is ignored).

When the $\overline{\text{RESET}}$ signal is generated, the MFTC is initialized to 0.

When the timer operation control bit (MFTC.2) is set to 1, timer operation starts; when the bit is set to 0, timer operation stops.

To clear the count register contents, perform either of the following:

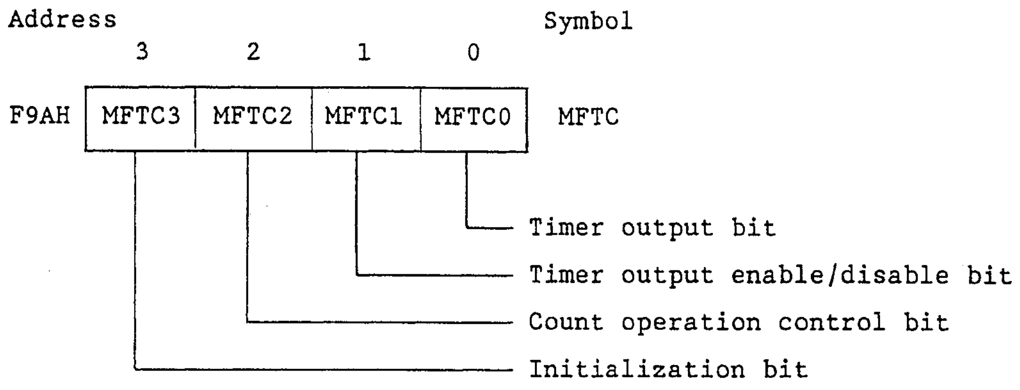
(i) Set MFTC.3 = 1 (initialization) when timer operation stops (MFTC.2 = 0).

(ii) Set MFTC.2 = 1 (timer operation enable) and MFTC.3 = 1 (initialization) at the same time.

Caution 1: To stop timer operation, the timer output enable/ disable bit (MFTC.1) must also be set to 0 (timer output disable).

Caution 2: When timer operation is enabled (MFTC.2 = 1), do not set MFTC.3 to 1 (initialization).

Fig. 4-57 MFT Control Register Format



Timer output bit (R)

MFTC0	Timer mode	Inversion when a match signal is output from the comparator.
	PWM output mode	Clear when a match signal is output from the comparator. Set when counter (MFTH) overflows.
	Free-running timer mode	Inversion when counter (MFTH) overflows.
	Integration A/D converter mode	Inversion when counter (MFTH) overflows.

Caution: When MFTC.3 (initialization bit) is set to 1, MFTC.0 (MFT F/F) is set to 1 regardless of the operation mode.

Timer output enable/disable bit (R/W)

MFTC1	0	Output of the MFT flip-flop contents is disabled.
	1	Output of the MFT flip-flop contents is enabled.

Caution: To output signal from the PPO pin, write 0 into the P21 output latch and set port 2 to the output mode.

Timer operation control bit (R/W)

MFTC2	0	Mode operation is stopped.
	1	Mode operation is enabled.

Caution: Clock supply to the count register is controlled. The counter is not cleared. Even if MFTC.3 is not set, timer operation starts by setting MFTC.2 to 1.

Initialization bit (W)

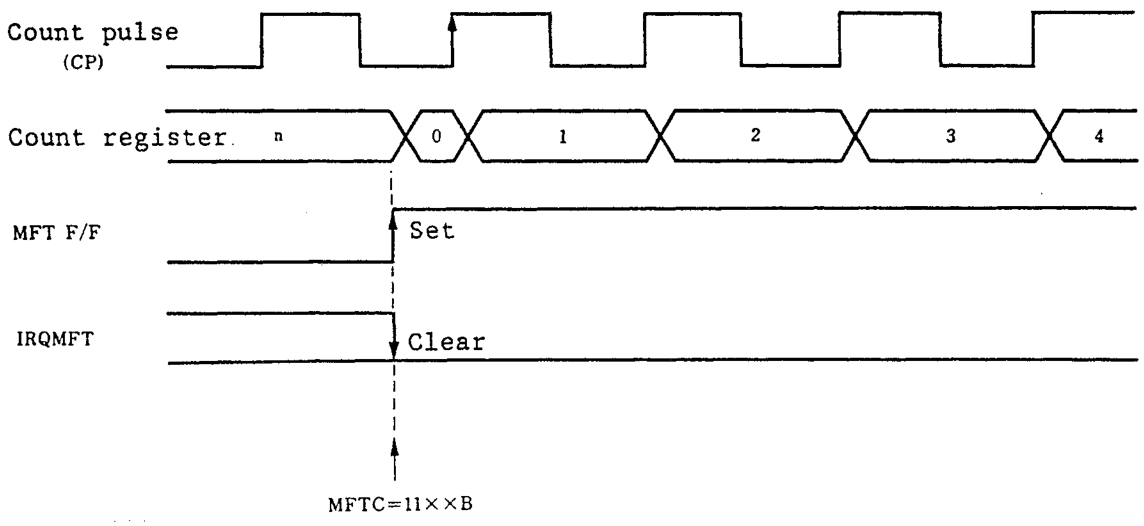
MFTC3	When "1" is written into this bit, IRQMFT and the count register are cleared and the MFT flip-flop is set. When timer operation starts, automatically the bit is reset to 0.

Caution 1: Even if "0" is written, no operation is performed.

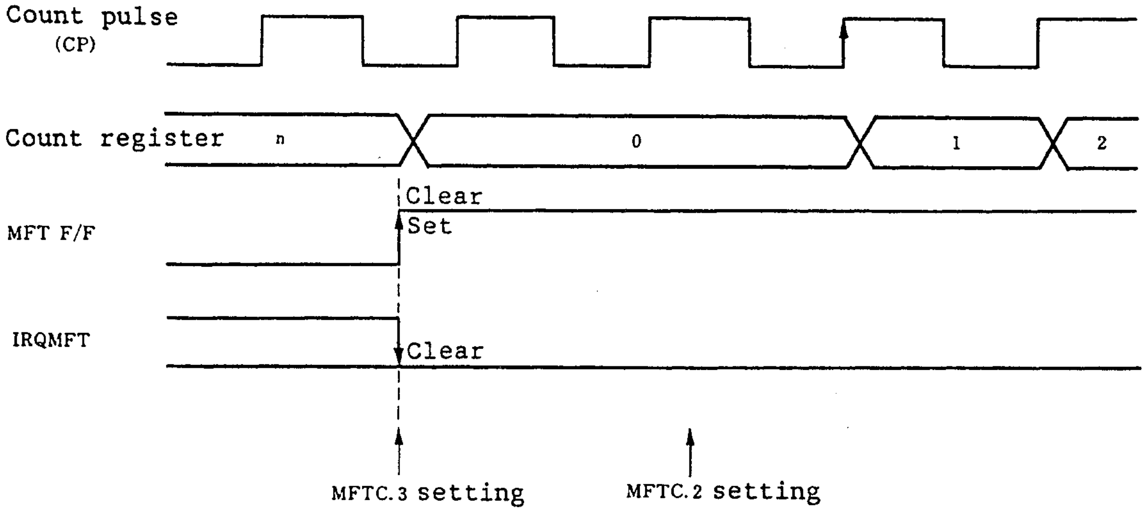
Caution 2: Timer operation starts when MFTC.2 (timer operation control bit) is set to 1.

Fig. 4-58 Start Operation Timings Depending on How MFTC Bits 2 and 3 are Set

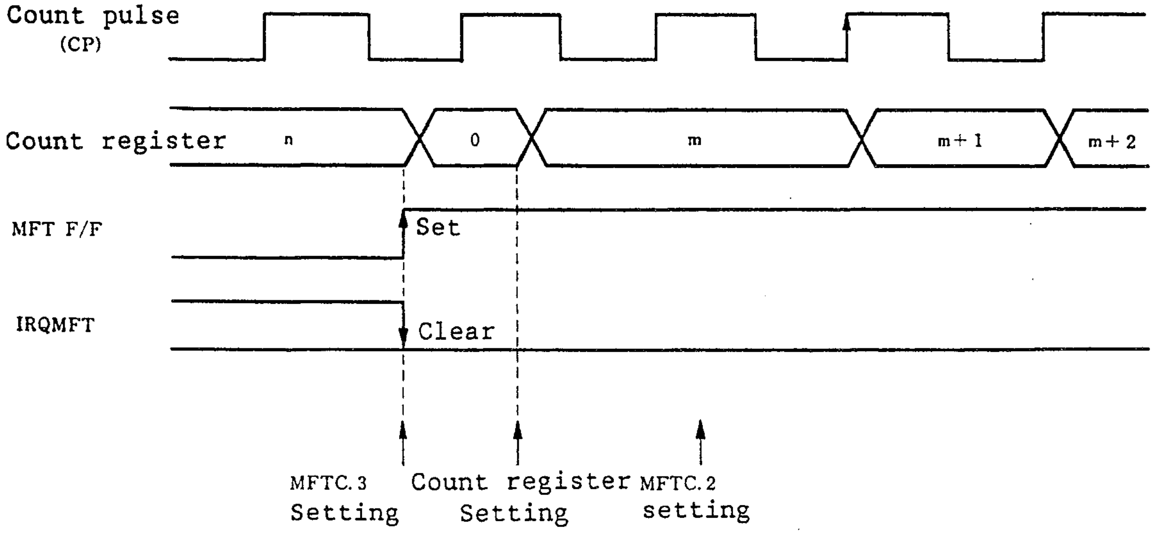
(1) Setting by executing 4-bit write instruction (MFTC = 11xxB)



(2) Setting by executing bit manipulation instruction (when the initial value of the current register is 0)



(3) Setting by executing bit manipulation instruction (when the initial value of the current register is m)



(c) MFT count register (MFTL and MFTH)

The MFT count register consists of two 8-bit counters (MFTL and MFTH) which can be read and written by executing 8-bit memory handling instructions. When the $\overline{\text{RESET}}$ signal is generated, MFTL is set to FFH and MFTH is set to "0".
To clear the count register contents, perform either of the following:

(i) Set MFTC.3 = 1 (initialization) when timer operation stops (MFTC.2 = 0).
(ii) Set MFTC.2 = 1 (timer operation enable) and MFTC.3 = 1 (initialization) at the same time.

Caution: When timer operation is enabled (MFTC.2 = 1), do not set MFTC.3 to 1 (initialization).

The MFTL function varies depending on mode selection as described below:

(i) In timer mode or PWM output mode

MFTL: Modulo register(Note 1)

MFTH: 8-bit counter(Note 2)

(ii) In free-running timer mode or integration A/D con verter mode

$$\left.\begin{array}{l}\text{MFTL: Low-order 8-bit counter}\\ \text{MFTH: High-order 8-bit counter}\end{array}\right\}\rightarrow\text{16-bit counter}$$

Note 1: If the MFTL contents are read in the PWM output mode, the modulo register contents are read.

2: In the PWM output mode, MFTL operates as a 6-, 7-, or 8-bit counter depending on tap selection.

Table 4-9 MFTL and MFTH Clear Conditions for each Mode

Clear condition	MFTL	MFTH
Timer mode	Cannot be cleared. (Modulo register)	Match signal from comparator. MFTC.3 set to 1.
PWM output mode	Cannot be cleared. (Modulo register)	Counter (MFTH) overflow. MFTC.3 set to 1.
Free-running timer mode	Counter (MFTH) overflow. MFTC.3 set to 1.	Counter (MFTH) overflow. MFTC.3 set to 1.
Integration A/D converter mode	Counter (MFTH) overflow. MFTC.3 set to 1.	Counter (MFTH) overflow. MFTC.3 set to 1.

Caution: When an overflow from counter (MFTH) specification tap occurs, the clear signal is generated.

(d) Interrupt request flag (IRQMFT)

The interrupt request flag (IRQMFT) is set to "1" when the IRQMFT occurrence condition in each mode listed in Table 4-10 is satisfied. IRQMFT can be reset to "0" by inputting the $\overline{\text{RESET}}$ signal or initialization bit (MFTC.3) to 1.

Table 4-10 IRQMFT Occurrence Conditions

Timer mode	Match signal from comparator
PWM output mode	Counter (MFTH) specification bit overflow
Free-running timer mode	Counter (MFTH) specification bit overflow
Integration A/D converter mode	External comparator inversion signal input or counter (MFTH) specification bit overflow

(4) Dual function pin function

The MFT signal input/output pins are also used for P21 and P100-P103; they function as shown in (a) to (e) depending on the mode.

When the $\overline{\text{RESET}}$ signal is input, P21 is set to the input mode (output high impedance) and P100-P103 become high impedance (during N-ch open drain) or go high (when internal pull-up resistor is contained).

(a) P21/PPO pin

Timer mode	Timer pulse output (square wave)(Note 1)
PWM output mode	Timer pulse output (PWM)(Note 1)
Free running timer mode	Timer pulse output (square wave)(Note 1)
Integration A/D converter mode	Port (P21)(Note 2)

Note 1: Timer pulse is output from the PPO pin as described below:

① Write "0" into the P21 output latch.

② Set port 2 to the output mode.

③ Set MFTC.1 to 1 (timer output enable mode).

Note 2: In the integration A/D converter mode, set MFTC.1 to 0 (timer output disable mode).

(b) P100/$\overline{\text{MAR}}$ pin

Timer mode	Port (P100)
PWM output mode	
Free running timer mode	
Integration A/D converter mode (Note)	Reverse integration signal output ($\overline{\text{MAR}}$)

(c) P101/$\overline{\text{MAI}}$ pin

Timer mode	Port (P101)
PWM output mode	
Free running timer mode	
Integration A/D converter mode (Note)	Integration signal output ($\overline{\text{MAI}}$)

(d) P102/$\overline{\text{MAZ}}$ pin

Timer mode	Port (P102)
PWM output mode	
Free running timer mode	
Integration A/D converter mode (Note)	Auto zero signal output ($\overline{\text{MAZ}}$)

(e) P103/$\overline{\text{MAT}}$ pin

Timer mode	Port (P103)
PWM output mode	
Free running timer mode	
Integration A/D converter mode (Note)	External comparator signal input ($\overline{\text{MAT}}$)

Note: To use P100-P103 in the integration A/D converter mode, perform the following steps:

① Write "1111" into the P100-P103 output latch.

② Set port 10 to the output mode.

③ Set the integration A/D converter mode.

Caution: P103/$\overline{\text{MAT}}$ functions as an external comparator signal input pin although it is set to the output mode.

(5) MFT operation

The following four MFT operation modes can be used:

. 8-bit timer mode
. PWM output mode
. 16-bit free running timer mode
. integration A/D converter mode

(a) 8-bit timer mode

(i) Function

- programmable interval timer operation
- square wave output of any frequency to PPO pin

(ii) Operation

Fig. 4-59 shows the MFT block diagram in the 8-bit timer mode. MFTL operates as a modulo register and MFTH operates as a count register. Timer operation is performed as described below:

① Set the operation mode and count pulse (PC) in the mode register (MFTM).

② Set count value in the modulo register (MFTL).

③ Set the control register (MFTC) to 11xxB to initialize the count register (MFTH) and enable timer operation.

When the value set in the modulo register (MFTL) matches the count register (MFTH) contents, a match signal is generated. The MFT F/F contents are inverted and the interrupt request flag (IRQMFT) is set. Fig. 4-60 shows this timer operation timing. The interrupt occurrence interval timer can be found from the following expression:

Interrupt occurrence interval time = (modulo register setup value + 1) x resolution

Caution: Do not set 0 in MFTL.

Fig. 4-59 Block Diagram in Timer Mode

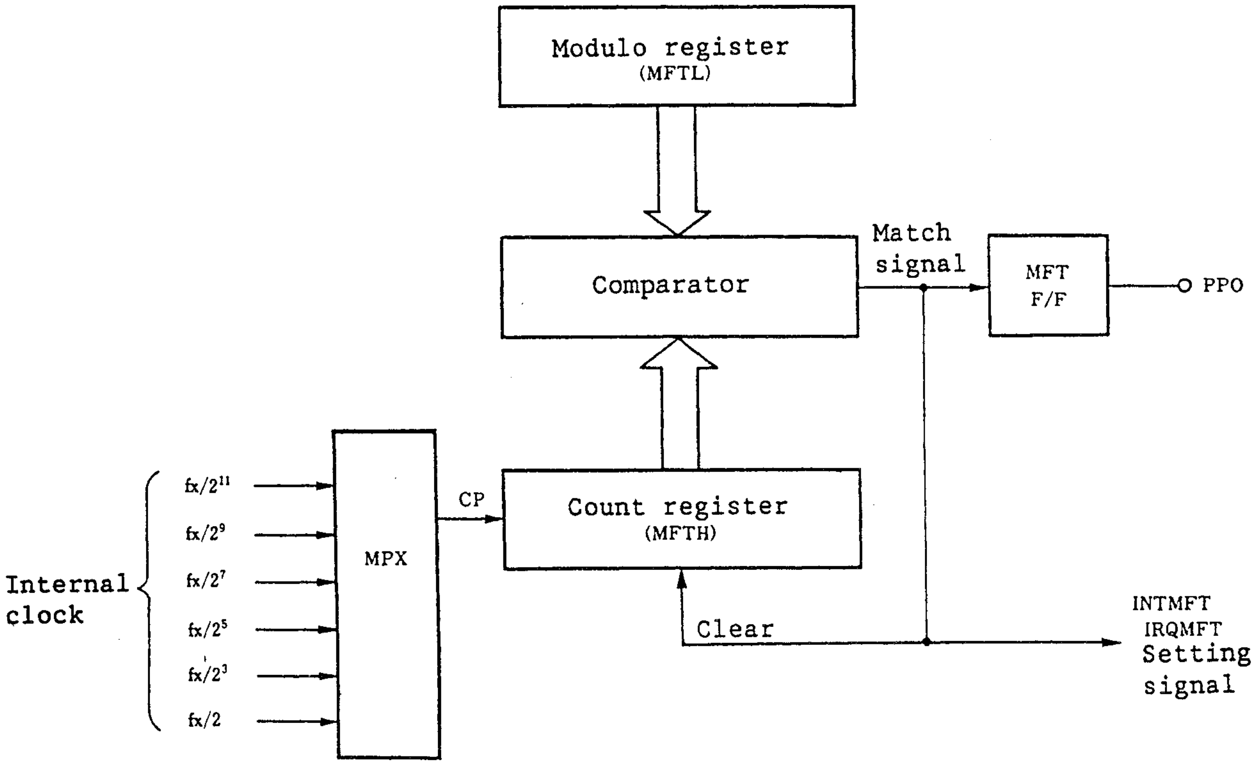
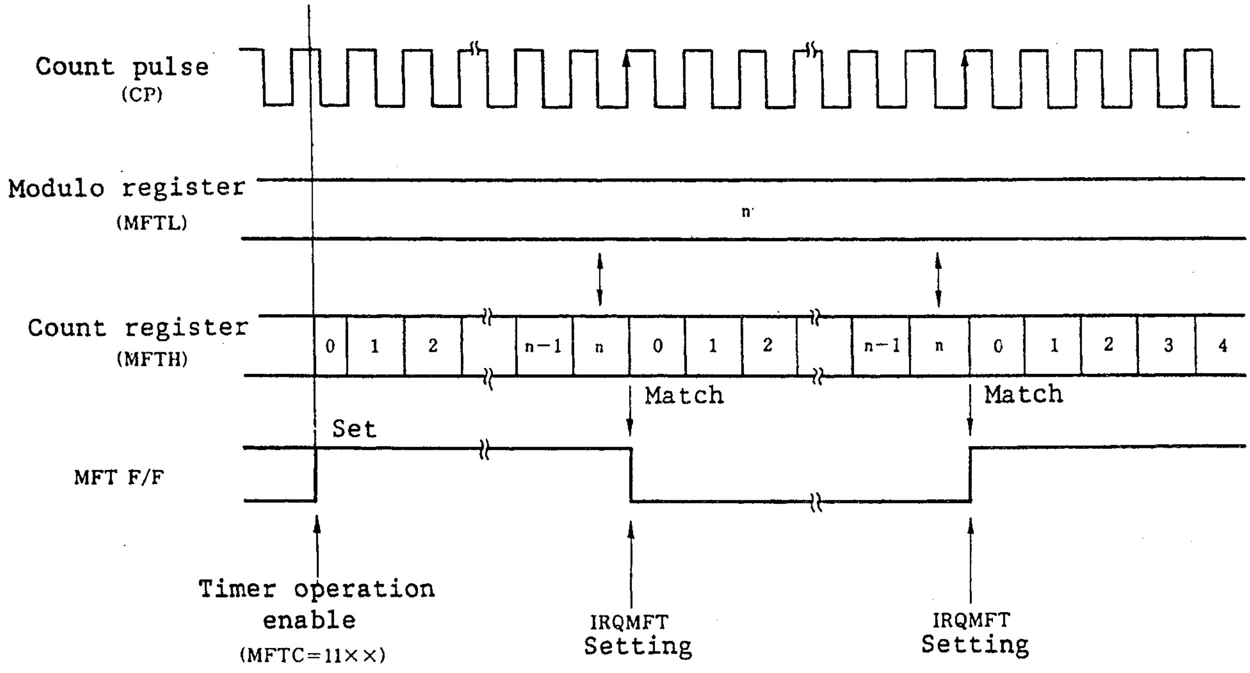


Fig. 4-60 Timer Operation Timing



(b) PWM output mode

(i) Function

. output of 6-, 7-, or 8-bit precision PWM signal to PPO pin

(ii) Operation

Fig. 4-61 shows the MFT block diagram in the PWM output mode. The count register (MFTH) operates as a 6-, 7-, or 8-bit counter by selecting the tap position.

Timer operation is performed as described below:

① Set the operation mode, tap position, and count pulse (CP) in the mode register (MFTM).

② Set count value in the modulo register (MFTL).

③ Set the control register (MFTC) to 11xxB to initialize the count register (MFTH) and enable timer operation.

When the counter is initialized (MFTC.3=1) or the count register (MFTH) overflows, the MFT F/F contents are set to 1.

When the count register (MFTH) overflows, the interrupt request flag (IRQMFT) is set and the modulo register (MFTL) contents are reloaded into the modulo latch.

Fig. 4-62 shows this PWM output operation timing. Assuming that the tap position is bit n (n = 6, 7, or 8), the PWM signal period can be found from the following expression:

PWM signal period = 2^n x resolution

Fig. 4-61 Block Diagram in PWM Output Mode

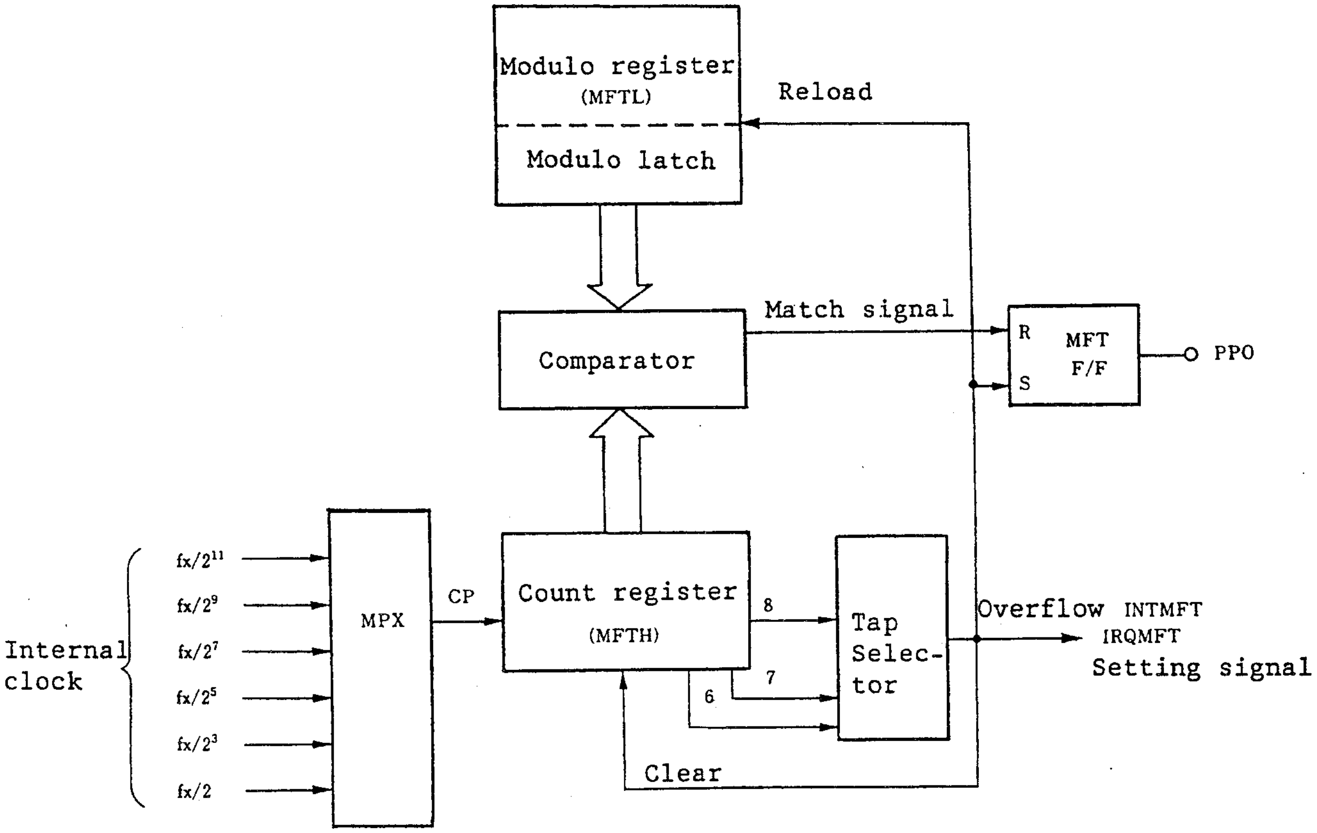
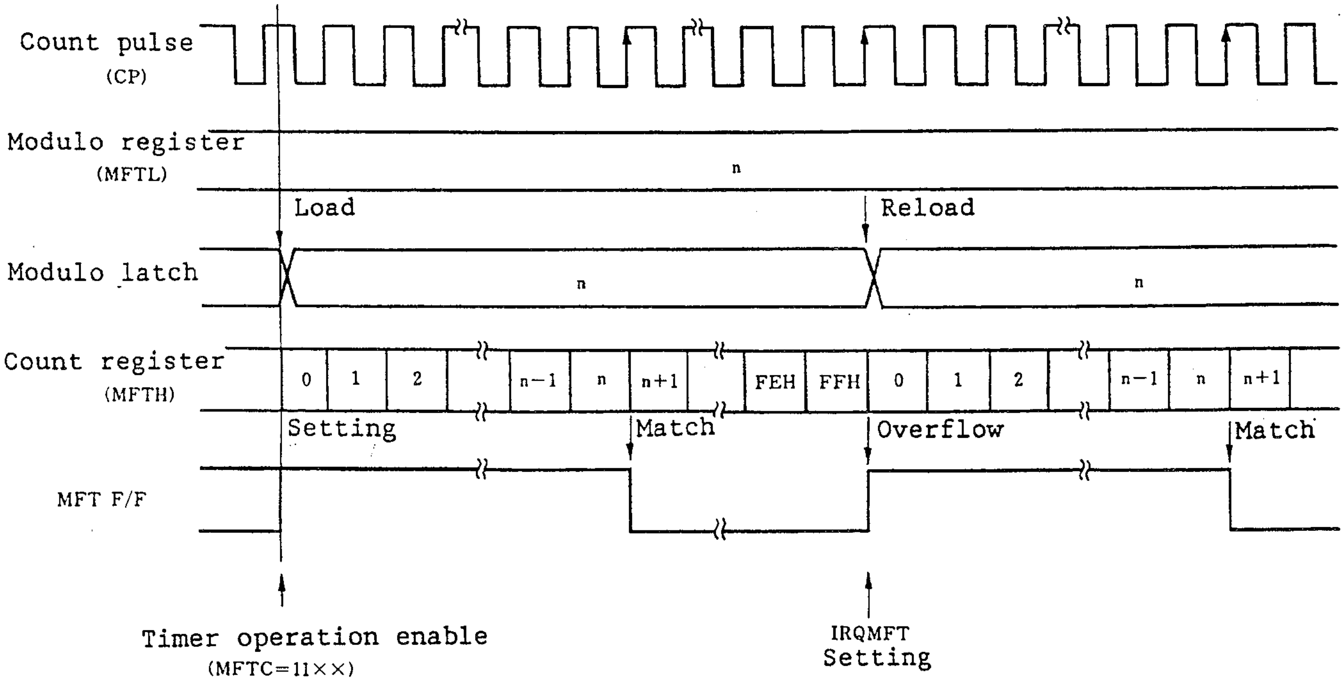


Fig. 4-62 PWM Output Operation Timing (when tap = 8 bit is selected)



(c) 16-bit free-running timer mode

(i) Function

. interval timer operation to cause an interrupt to occur at given time intervals

. applicable as a one-shot timer

(ii) Operation

Fig. 4-63 shows the MFT block diagram in the 16-bit free-running timer mode. The count register consists of MFTL for the low-order eight bits and MFTH for the high-order eight bits. It operates as a 13-, 14-, 15-, or 16-bit count register by selecting the tap position. Timer operation is performed as described below:

① Set the operation mode, tap position, and count pulse (CP) in the mode register (MFTM).

② Set the control register (MFTC) to 11xxB to initialize the count register (MFTH, MFTL) and enable timer operation.

When the count register overflows, the MFT F/F contents are inverted and the interrupt request flag (IRQMFT) is set. Fig. 4-64 shows this free-running timer operation timing.

Assuming that the tap position is bit n (n = 13, 14, 15 or 16), the interrupt occurrence interval time can be found from the following expression:

interrupt occurrence interval time = 2^n x resolution

Fig. 4-63 Block Diagram in Free-running Timer Mode

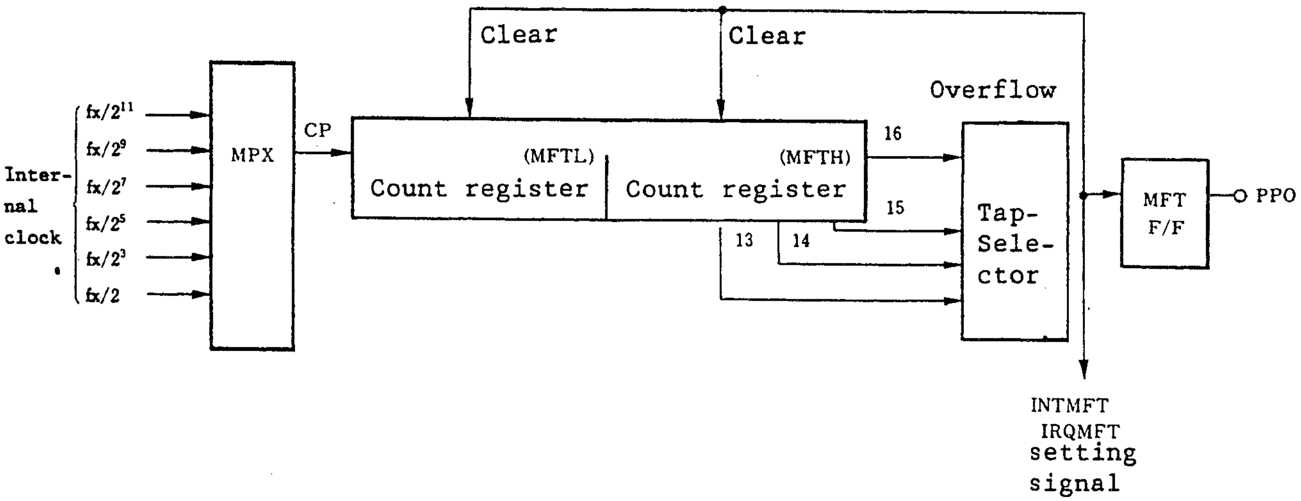
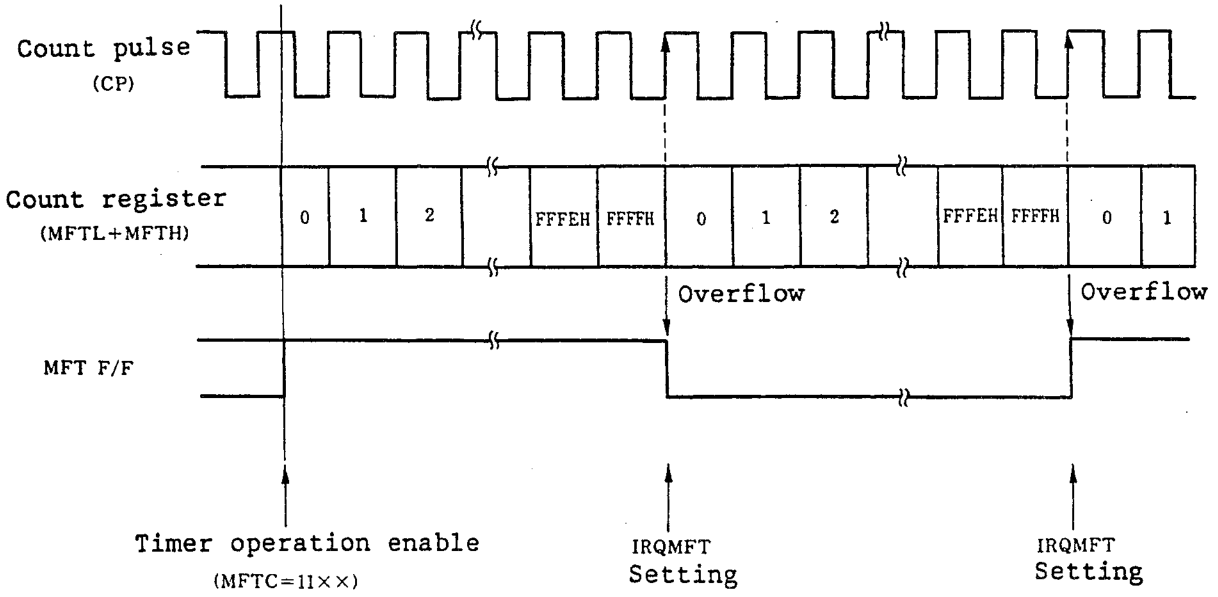


Fig. 4-64 Free-running Timer Operation Timing (when tap = bit 16 is selected)



(d) Integration A/D converter mode

(i) Function

- . output of 16-bit integration A/D converter control signal
- . 13-, 14-, 15-, or 16-bit resolution can be selected

(ii) Operation

Fig. 4-65 shows the MFT block diagram in the integration A/D converter mode. Integration A/D converter can be implemented by providing an external circuit as shown in Fig. 4-66.

A/D conversion operation is started according to the following procedure:

① Write "1111" into the P100-P103 output latch.

② Set port 10 to the output mode.

Caution: P103/$\overline{\text{MAT}}$ functions as an external comparator signal input pin although it is set to the output mode.

③ Set the integration A/D converter mode.

④ Set the operation mode, tap position, falling edge specification, and count pulse (CP) in the mode register (MFTM). The A/D conversion resolution is determined by the tap position.

⑤ Set the auto zero time. Set count value in the count register (MFTL, MFTH). The set time can be found from the following expression:
(2^{16} - count register setup value) x resolution

⑥ Set control register (MFTC) bit MFTC.2 to 1 or MFTC to 01xxB to enable count operation.

After this, A/D conversion operation is performed automatically.

① Auto zero period ($\overline{\text{MAZ}}$-low output)

. Integrator is set to 0 until the counter overflows from the setup value.

② Integration period ($\overline{\text{MAI}}$-low output)

. Integration operation is performed on analog input voltage V_{IN} until the tap selected for the counter overflows.

③ Reverse integration period ($\overline{\text{MAR}}$-low output)

. Reverse integration operation is performed on reference voltage V_{REF} during which the counter performs timer operation.

④ Conversion termination (falling edge input to $\overline{\text{MAT}}$)

. When comparator input is inverted, count operation stops, the interrupt request flag (IRQMFT) is set, and conversion terminates.

Fig. 4-67 shows this integration A/D converter operation timing. After the conversion terminates, the conversion result can be obtained by reading the count register (MFTH, MFTL) contents.

Fig. 4-65 Block Diagram in Integration A/D Converter Mode

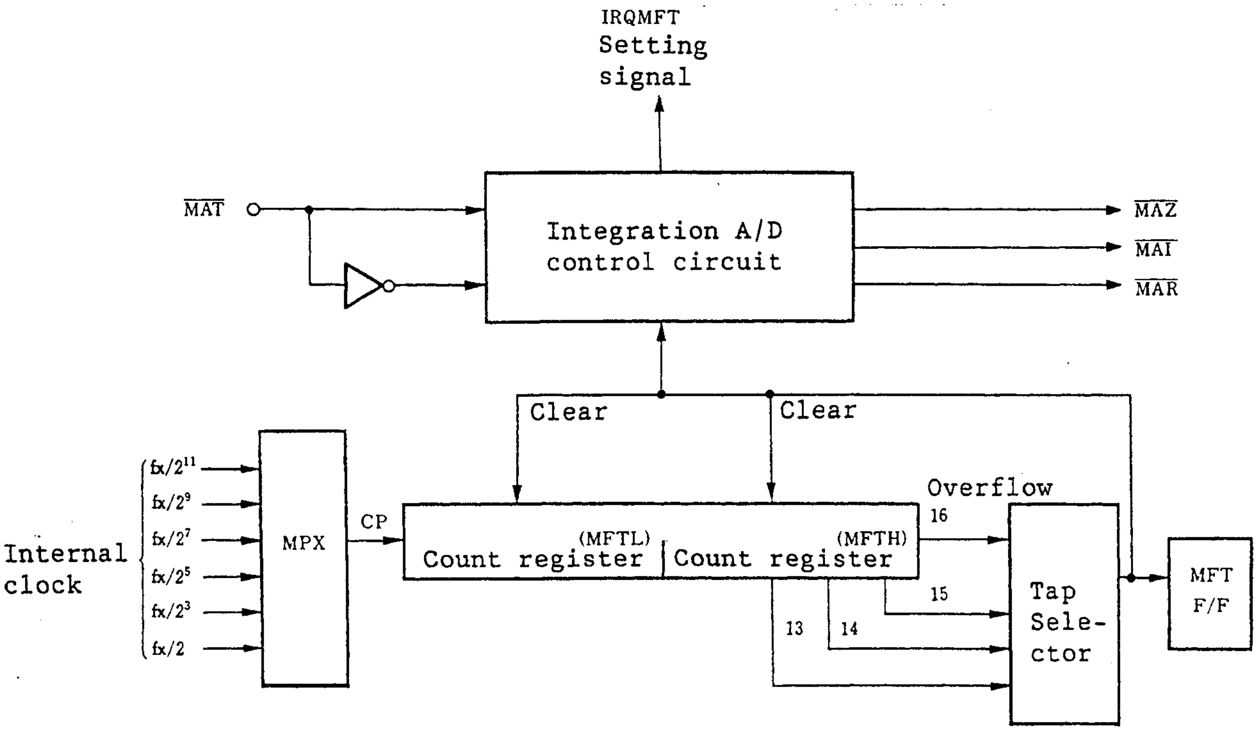
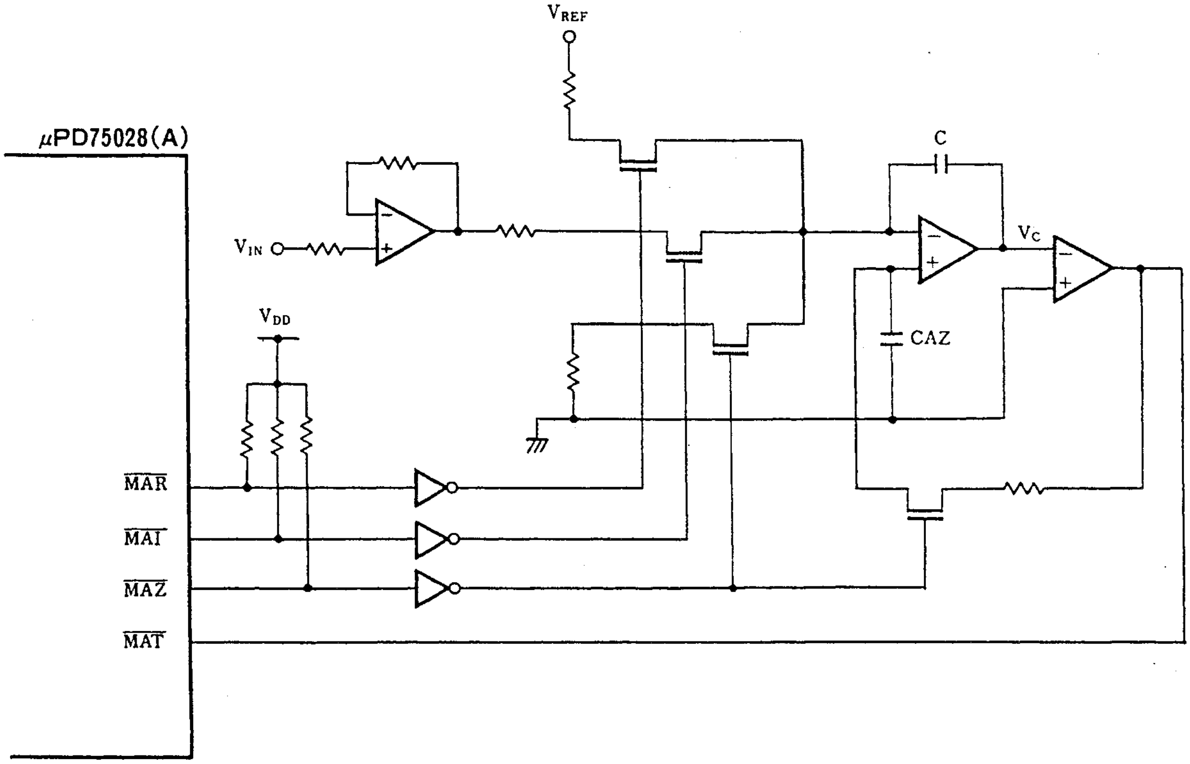
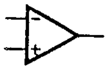
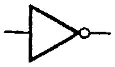


Fig. 4-66 Integration A/D Converter External Circuit Configuration

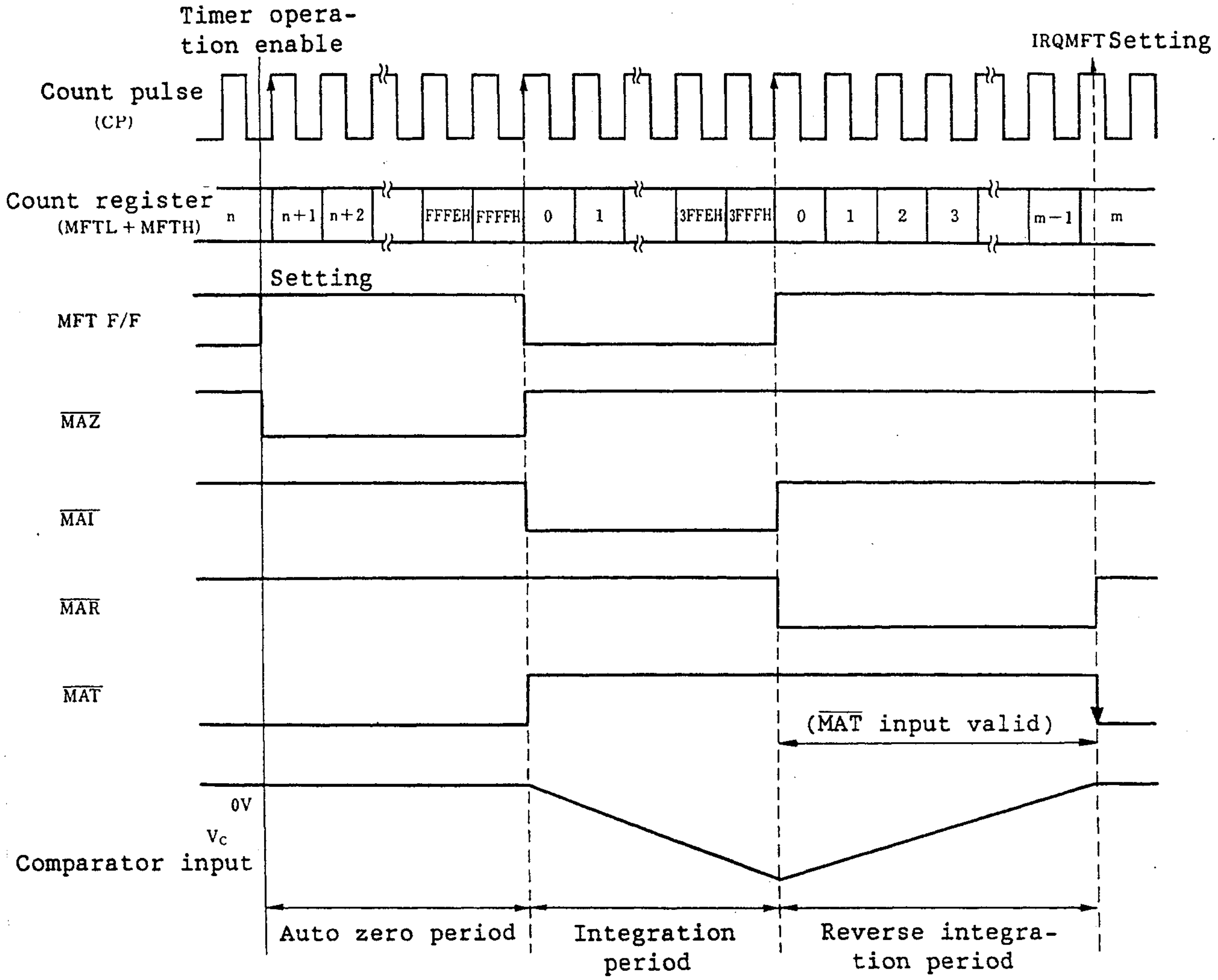


Remarks:

	uPD4066: Bidirectional switch
	uPC4074, 4064: Operational amplifier
	uPD4069: Invertor (Note)

Note: Just after $\overline{\text{RESET}}$ is generated, $\overline{\text{MAR}}$, $\overline{\text{MAI}}$, and $\overline{\text{MAZ}}$ go high (when pull-up resistor are contained). Thus, an inverter must be connected so as not to turn on the bidirectional switch.

Fig. 4-67 Integration A/D Converter Operation Timing (when tap = 14 bit is selected)

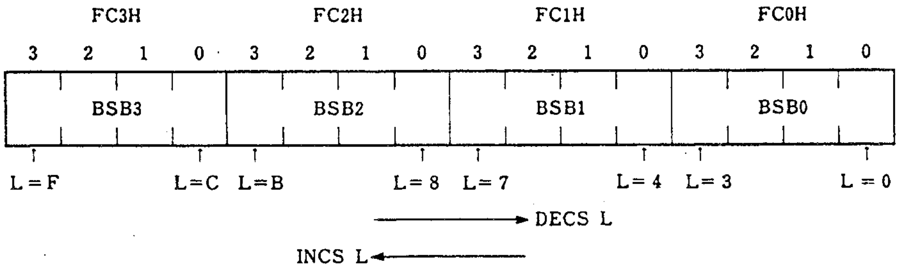


4.10 Bit Sequential Buffer ••• 16 bits

The bit sequential buffer is special data memory for bit manipulation. Particularly, address and bit specification can be changed in sequence for easy bit manipulation, thus the bit sequential buffer is useful when data which is long in bit length is processed bit-wise.

The data memory consists of 16 bits. Bit manipulation instruction pmem. @L addressing can be used and indirect bit specification can be made by using the L register. In this case, processing can be performed while specification bits are being moved in sequence simply by incrementing or decrementing the L register within a program loop.

Fig. 4-68 Bit Sequential Buffer Format



Remarks: In pmem.@L addressing, specification bit moves according to the L register.

Data can also be handled in the direct addressing. 1-bit, 4-bit, 8-bit direct addressing and pmem.@L addressing can be combined for application to consecutive input and consecutive output of 1-bit data, etc. For 8-bit handling, BSB0 and BSB2 are specified and high-order eight bits and low-order eight bits are handled separately.

Example: Output BUFF1, 2 16-bit data from port 3 bit 0 in series.

Program example

```
                CLR1  MBE
                MOV   XA, BUFF1
                MOV   BSB0, XA  ; Set BSB0, 1
                MOV   XA, BUFF2
                MOV   BSB2, XA  ; Set BSB2, 3
                MOV   L, #0
        LOOP0:  SKT   BSB0, @L  ; Test specified bit of BSB
                BR    LOOP1
                NOP             ; Dummy (timing adjustment)
                SET1  PORT3. 0  ; Set Port 3 bit 0
                BR    LOOP2
        LOOP1:  CLR1  PORT3. 0  ; Clear Port 3 bit 0
                NOP             ; Dummy (timing adjustment)
                NOP
          LOOP2: INCS  L        ; L  — L + 1
                BR    LOOP0
                RET
```

5. INTERRUPT FUNCTION

The uPD75028(A), contains seven interrupt sources and enables multiple interrupt under the software control. It also contains two types of test source. Of these, INT2 has two edge detect testable input.

Table 5-1 Interrupt Sources

Interrupt source	Internal/ external	Interrupt priority level (Note 1)	Vectored interrupt request signal (vector table address)
INTBT (reference time interval signal from basic interval timer)	Internal	1	VRQ1 (0002H)
INT4 (both rising edge detection and falling edge detection are valid)	External		
INT0 (rising and falling detect edge selection)	External	2	VRQ2 (0004H)
INT1	External	3	VRQ3 (0006H)
INTCSI (serial data transfer end signal)	Internal	4	VRQ4 (0008H)
INTT0 (match signal between programmable timer/counter count register and modulo register)	Internal	5	VRQ5 (000AH)
INTMFT (when interrupt condition in each mode becomes true) (Note 3)	Internal	6	VRQ6 (000CH)
INT2 (rising edge detection of input to INT2 pin or falling edge detection of input to any of KR0 -KR7) (Note 2) (Note 4)	External	Testable input signal (IRQ2, IRQW setting)	
INTW (signal from watch timer) (Note 4)	Internal		

Note 1: The interrupt priority level is the priority level applied when more than one interrupt request occurs at the same time.

2: See 5.2 (4) for details of INT2

3: See 4.9 MULTI-FUNCTION TIMER (MFT).

4: INT2 and INTW are of test sources. They are affected by interrupt enable flag in the same way as interrupt source, but they do not generate vectored interrupt.

The uPD75028(A) interrupt control circuit has the following functions:

(a) Vectored interrupt function under the hardware control which can control whether or not an interrupt can be acknowledged by setting the interrupt flag (IExxx) and interrupt master enable flag (IME).

(b) Interrupt start address can be set as desired.

(c) Interrupt request flag (IRQxxx) test function (interrupt occurrence can be checked by software).

(d) Standby mode release (an interrupt to release the standby mode can be selected by setting interrupt enable flag).

5.1 Interrupt Control Circuit Configuration

Fig. 5-1 shows the interrupt control circuit block diagram. The hardware is mapped in data memoryspace.

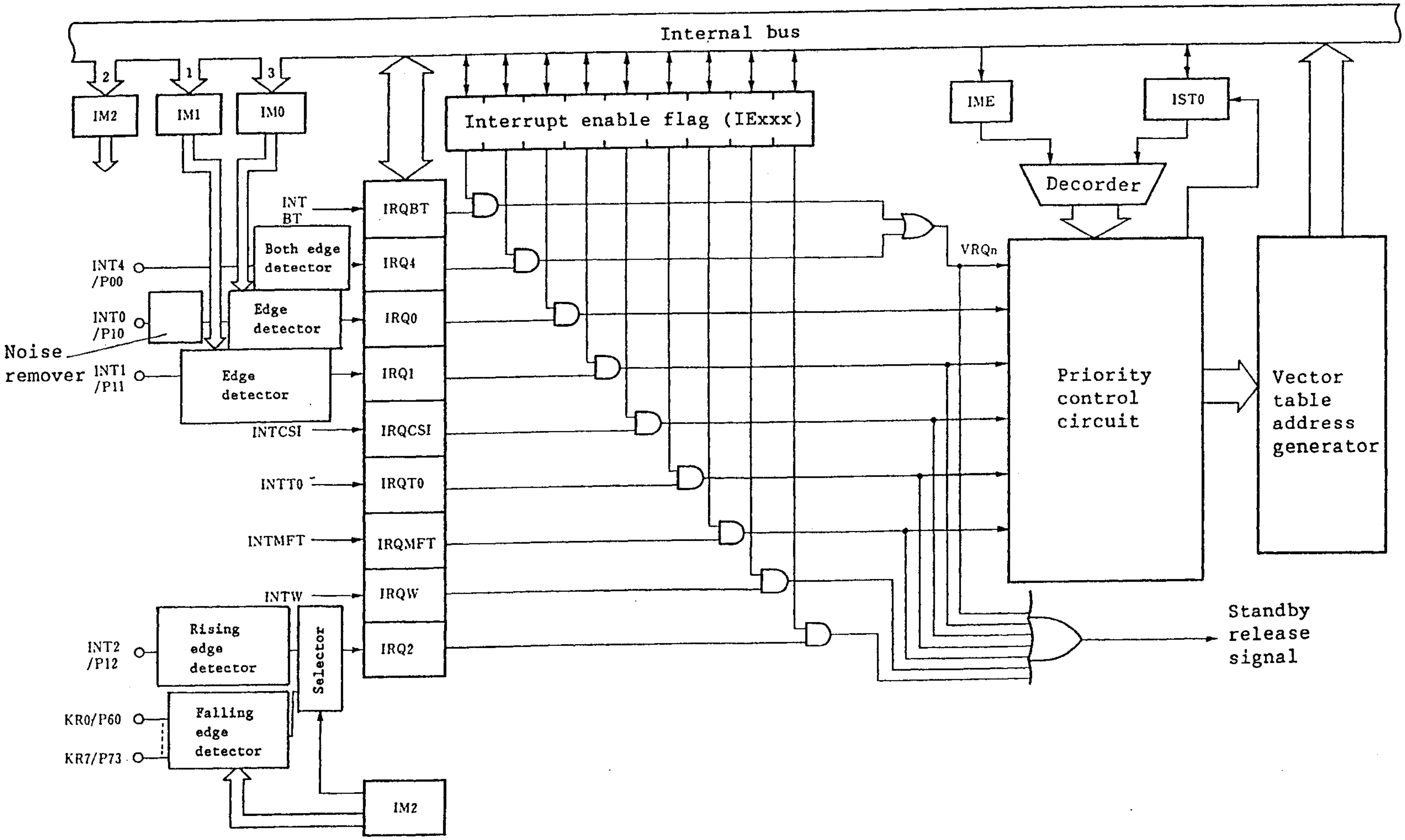


Fig. 5-1 Interrupt Control Circuit Block Diagram

5.2 Interrupt Control Circuit Hardware

(1) Interrupt request flags and interrupt enable flags

The interrupt request flag (IRQxxx) is set to 1 when its corresponding interrupt request occurs. When interrupt processing is performed, the interrupt request flag is automatically cleared. However, IRQBT and IRQ4 share vector address, thus differ in clear operation. (See 5.5.)

The interrupt enable flag (IExxx) is provided for each interrupt request flag. When the interrupt enable flag contains 1, its corresponding interrupt is enabled; when 0, it is disabled.

When an interrupt request flag is set to 1 and its corresponding interrupt enable flag enables interrupt a vectored interrupt request (VRQn) occurs. This signal is also used to release the standby mode.

The interrupt request flags and interrupt enable flags are handled by executing bit manipulation instructions and 4-bit memory handling instructions. Bit manipulation instructions can always be executed for direct flag handling regardless of how MBE is set. The interrupt enable flags are handled by executing EI IExxx instruction and DI IExxx instruction. To test the interrupt request flags, normally the SKTCLR instruction is used.

```
Example:  EI      IE0    ; Enable INT0
          DI      IE1    ; Dissable INT1
          SKTCLR  IRQCSI ; Skip and clear if IRQCSI  is
                           1
```

When the interrupt request flag is set to 1 by executing an instruction, a vectored interrupt is executed as if an interrupt occurred.

When the $\overline{\text{RESET}}$ signal is generated, the interrupt request flags and interrupt enable flags are cleared, disabling all interrupts.

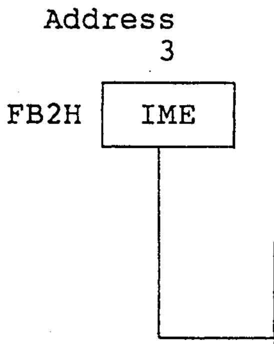
Table 5-2 Interrupt Request Flag Setting Signals

Interrupt request flag	Interrupt request flag setting signal	Interrupt enable flag
IRQBT	Refernce time interval signal by basic interval timer	IEBT
IRQ4	Detection of either rising or falling edge of INT4/P00 pin input signal	IE4
IRQ0	INT0/P10 pin input signal edge detection. The detected edge is selected by setting INT0 mode register (IM0).	IE0
IRQ1	INT1/P11 pin input signal edge detection. The detected edge is selected by setting INT1 mode register (IM1).	IE1
IRQCSI	Serial interface serial data transfer operation end signal.	IECSI
IRQT0	Match signal from timer/event counter #0.	IET0
IRQW	Signal from watch timer	IEW
IRQ2	INT2/P12 pin input signal rising edge detection of falling edge detection of input to any of KR0/P60 to KR7/P73 pins.	IE2
IRQMFT	Timer mode: Counter match signal	IEMFT
	PWM output mode: At reference time intervals	
	Free running timer mode: At reference time intervals	
	Integration A/D converter mode: A/D conversion end signal	

(2) Interrupt master enable flag (IME)

The interrupt master enable flag (IME) is used to specify whether acknowledge of all interrupts is enabled or disabled.
The IME is set by executing the EI or DI instruction.
When the $\overline{\text{RESET}}$ signal is generated, the IME is reset to 0, disabling acknowledge of all interrupts.

Fig. 5-2 IME Format



Interrupt master enable flag (IME)

0	All interrupts are disabled and vectored interrupt is not started.
1	Interrupt enable/disable is controlled by the corresponding interrupt enable flag.

(3) Hardware of INT0, INT1, INT4

(a) Fig. 5-3 (a) shows the INT0 structure. It is an external interrupt input where the rising or falling edge can be selected for the detection edge.
INT0 has the noise elimination function using a sampling clock. The noise eliminator can remove any pulse shorter than than two sampling clock cycles[Note] as noise. A pulse wider than one sampling clock cycle may be captured as an interrupt signal depending on the sampling timing. This eliminator can accept a pulse wider than this as an interrupt signal.

Note $2t_{CY}$ when sampling clock is Φ
$128/f_X$ when sampling clock is $f_X/64$

INT0 has two types of sampling clocks Φ and $f_X/64$, either of which can be selected for use by setting edge detection mode register(IM0) bit 3 (IM03). (See Fig.)
To select the detection edge, set edge detection mode register(IM0) bit 0 (IM00) and bit 1 (IM01).
Fig. 5-4 (a) shows the IM0 format. To set IM0, use a 4-bit manipulation instruction. When the reset signal is generated, all IM0 bits are reset to 0, specifying the rising edge.

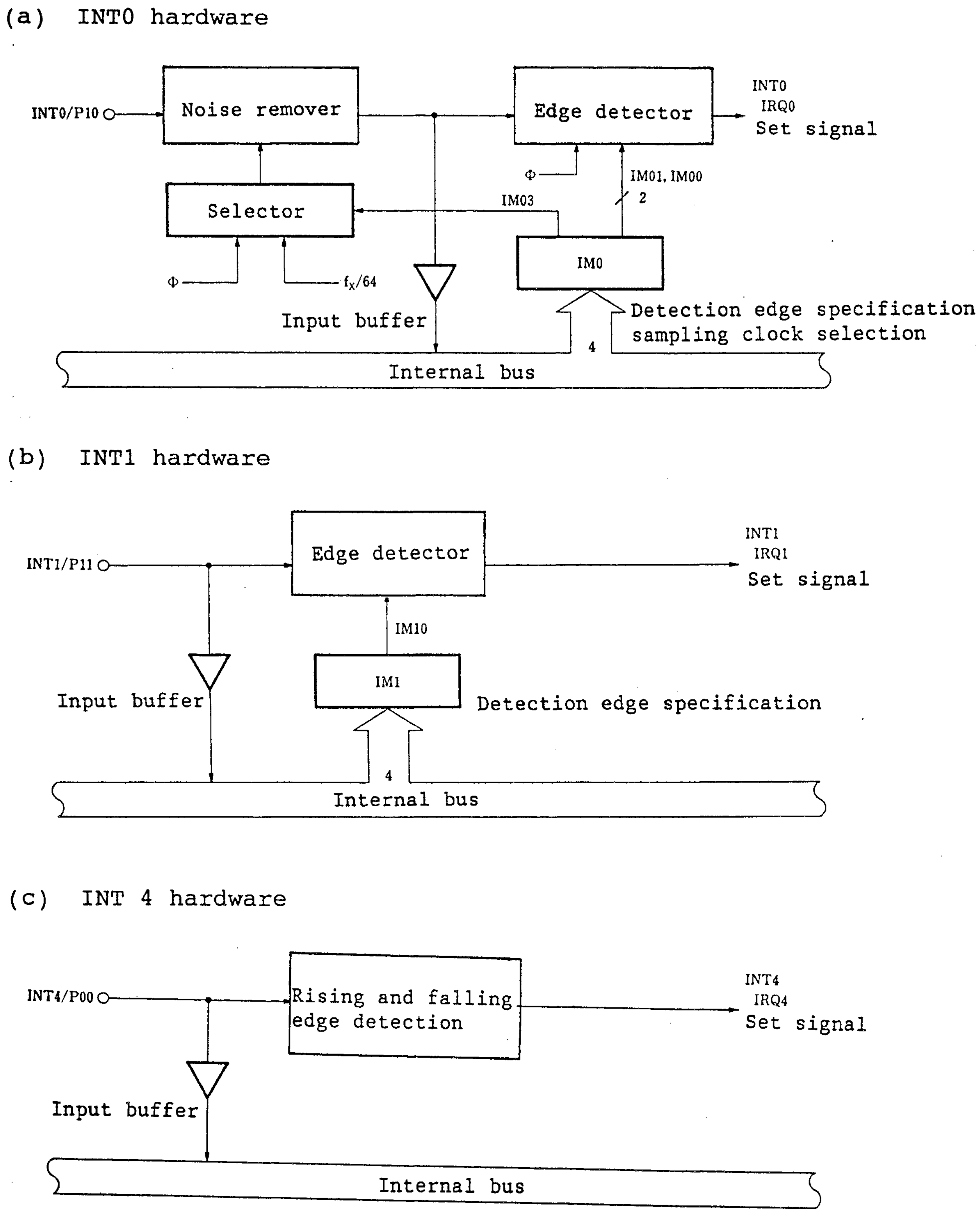
Caution 1: INT0 where sampling is performed with a clock does not operate during the standby mode.

2: Input a pulse wider than two sampling clock cycles to the INT0/P10 pin even when the pin is used as a port. Otherwise, the pulse is suppressed by the noise eliminator.

(b) Fig. 5-3 (b) shows the INT1 format. It is external interrupt input whose detection edge can be selected out of rising and falling edges.
Fig. 5-4 (b) shows the IM1 format. The IM1 is set by executing a 4-bit manipulation instruction. When the RESET signal is generated, all the IM1 bits are reset to 0, specifying the rising edge.

(c) Fig. 5-3 (c) shows the INT4 structure. It is external interrupt input whose rising and falling edges can be detected.

Fig. 5-3 INT0, INT1, and INT4 Structures



(a) INT0 hardware

(b) INT1 hardware

(c) INT 4 hardware

Fig. 5-4 Edge Detection Mode Register Formats

(a) INT0 edge detection mode register (IM0)

Address	3	2	1	0	Symbol
FB4H	IM03	0	IM01	IM00	IM0

Detection edge specification

IM01	IM00	
0	0	Rising edge specification
0	1	Falling edge specification
1	0	Rising and falling edge specification
1	1	Don't care (interrupt request flag is not set)

Sampling period

IM03	
0	1/Φ
1	$64/f_X$ (15.3 us (during 4.19MHz operation))

(b) INT1 edge detection mode register (IM1)

FB5H	0	0	0	IM10	IM1

IM10	
0	Rising edge specification
1	Falling edge specification

(c) INT2 edge detection mode register (IM2)

FB6H	0	0	IM21	IM20	IM0

IM21	IM20	INT2 interrupt source	Interrupt input pins
0	0	INT2 pin input rising edge specification	INT2 (one pin)
0	1	Falling edge specification of any of KR x pin inputs	KR4-KR7 (four pins)
1	0		KR2-KR7 (six pins)
1	1		KR0-KR7 (eight pins)

Caution: If the edge detection mode register is set or changed, interrupt request flag may be set. Previously disable interrupts and change the mode register contents. Clear the interrupt request flag by executing the CLR1 instruction before enabling interrupts. If $f_X/64$ is selected as sampling clock when the IM0 contents are changed, the interrupt request flag must be cleared in 16 machine cycles after the mode register contents are changed.

(4) INT2 and key interrupt (KR0-KR7) hardware

Fig. 5-5 shows the INT2 and KR0-KR7 structures.
IRQ2 is set in either of the following modes:

(a) INT2 pin input rising edge detection
When the rising edge input to the INT2 pin is detected, IRQ2 is set.

(b) Falling edge detection of any of KR0-KR7 pin inputs (key interrupt)
When the falling edge input to any of the KR0-KR7 pins selected in the edge detection mode register (IM2) is detected, IRQ2 is set.

Caution: When a low level is input to even one of the pins selected as falling edge detection, IRQ2 is not set although falling edge is input to any other pin.

The INT2 pin or the KR0-KR7 pins are selected as interrupt signal in the edge detection mode register (IM2). Which pins of KR0-KR7 are selected as falling edge input is also specified in IM2.
Fig. 5-4 (c) shows the IM2 format. The IM2 is set by executing a 4-bit manipulation instruction. When the RESET signal is generated, all the IM2 bits are reset to 0, specifying the INT2 rising edge.

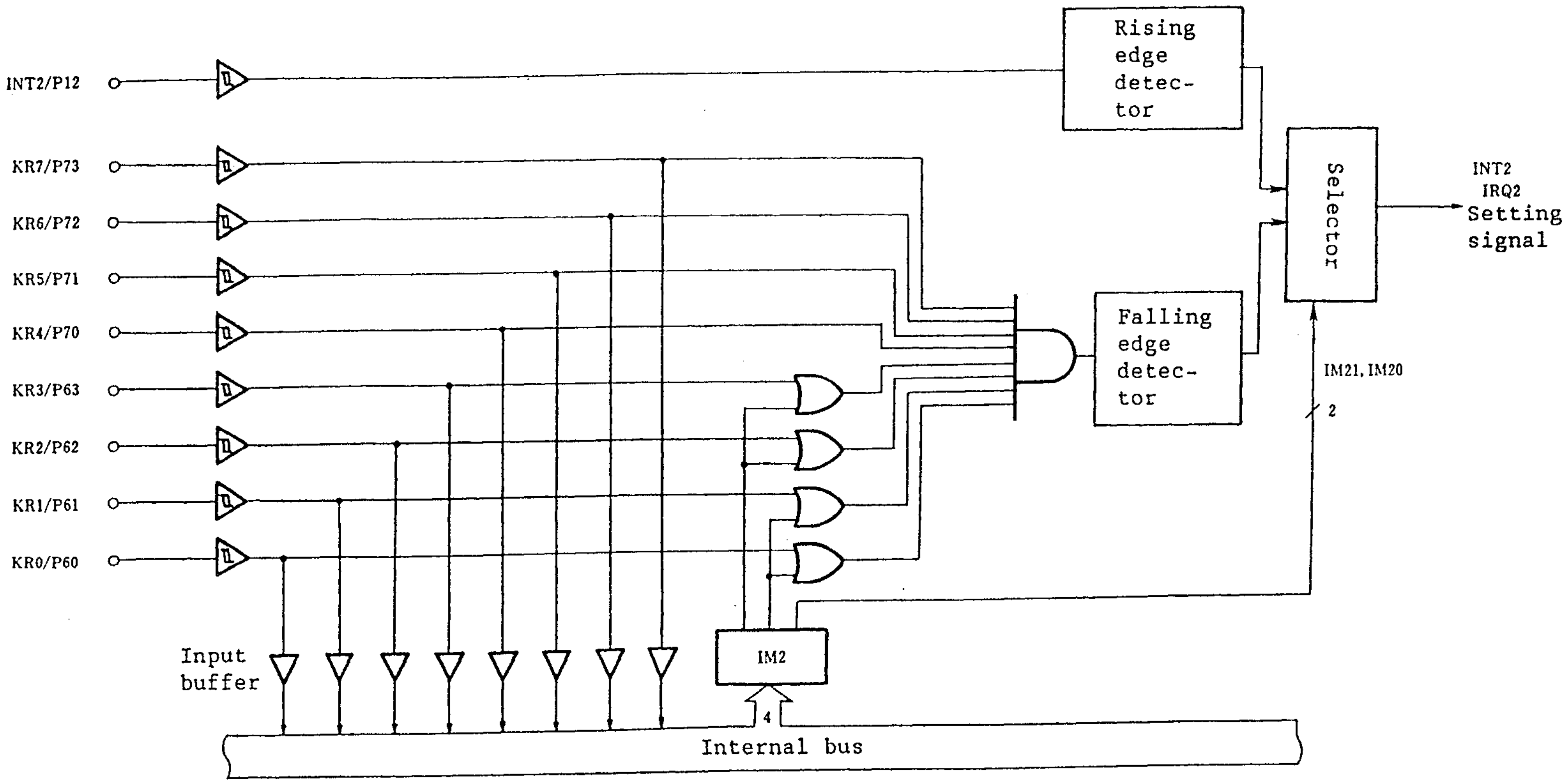
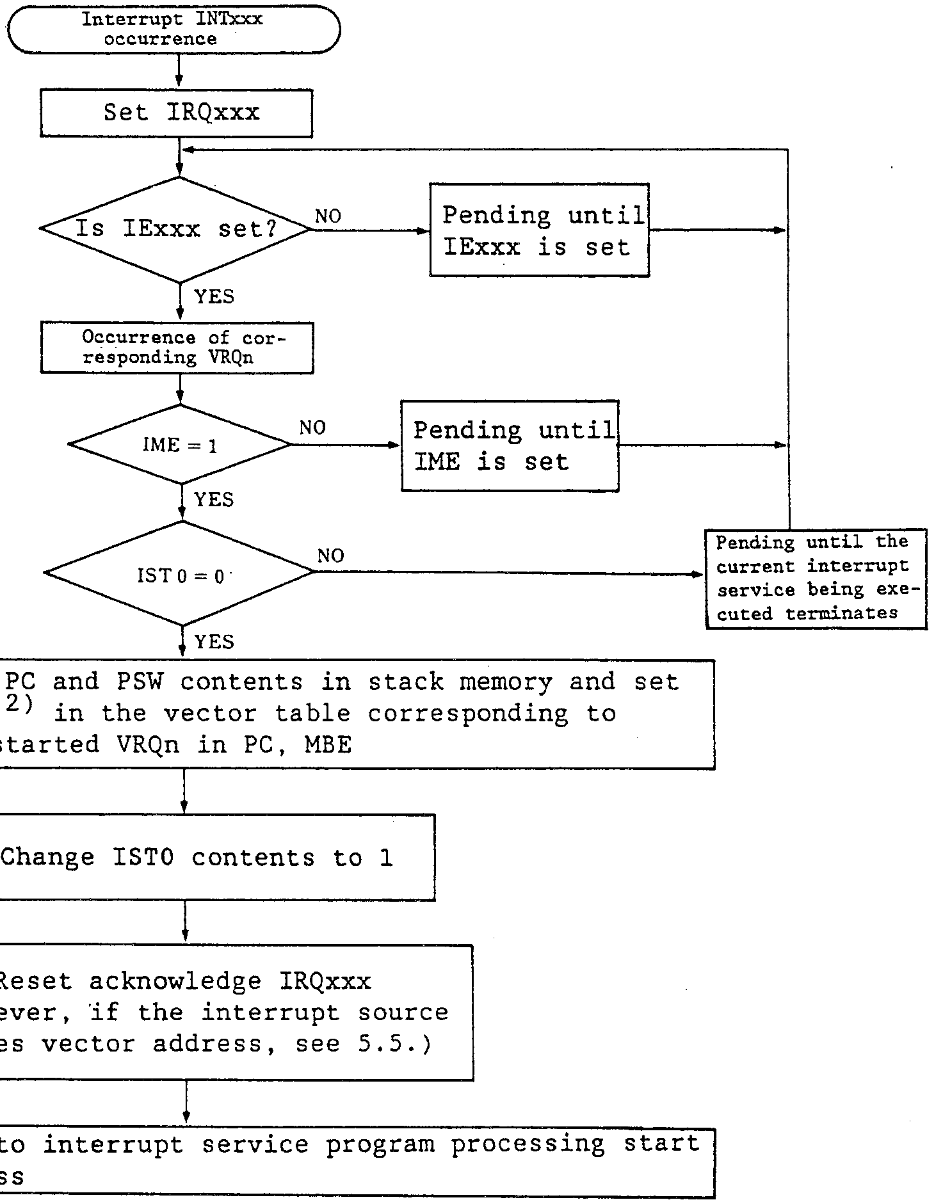


Fig. 5-5 INT2 and KR0-KR7 Configuration

5.3 Interrupt Sequence

When an interrupt occurs, it is processed as shown in Fig. 5-6.

Fig. 5-6 Interrupt Processing Sequence


Interrupt INTxxx occurrence
Set IRQxxx
Is IExxx set?
NO
Pending until IExxx is set
YES
Occurrence of corresponding VRQn
IME = 1
NO
Pending until IME is set
YES
IST 0 = 0
NO
Pending until the current interrupt service being executed terminates
YES
Save PC and PSW contents in stack memory and set data(2) in the vector table corresponding to the started VRQn in PC, MBE
Change IST0 contents to 1
Reset acknowledge IRQxxx (However, if the interrupt source shares vector address, see 5.5.)
Jump to interrupt service program processing start address

Remarks 1: IST0: Interrupt status flag (PWS bit 2). (See Table 5-3.)

2: Prestore the interrupt service program start address and the MBE setup value at the interrupt start in each vector table.

5.4 Multiple Interrupt Processing Control

uPD75028(A) multiple interrupt is enabled as described below:

Table 5-3 IST0 and Interrupt Processing State

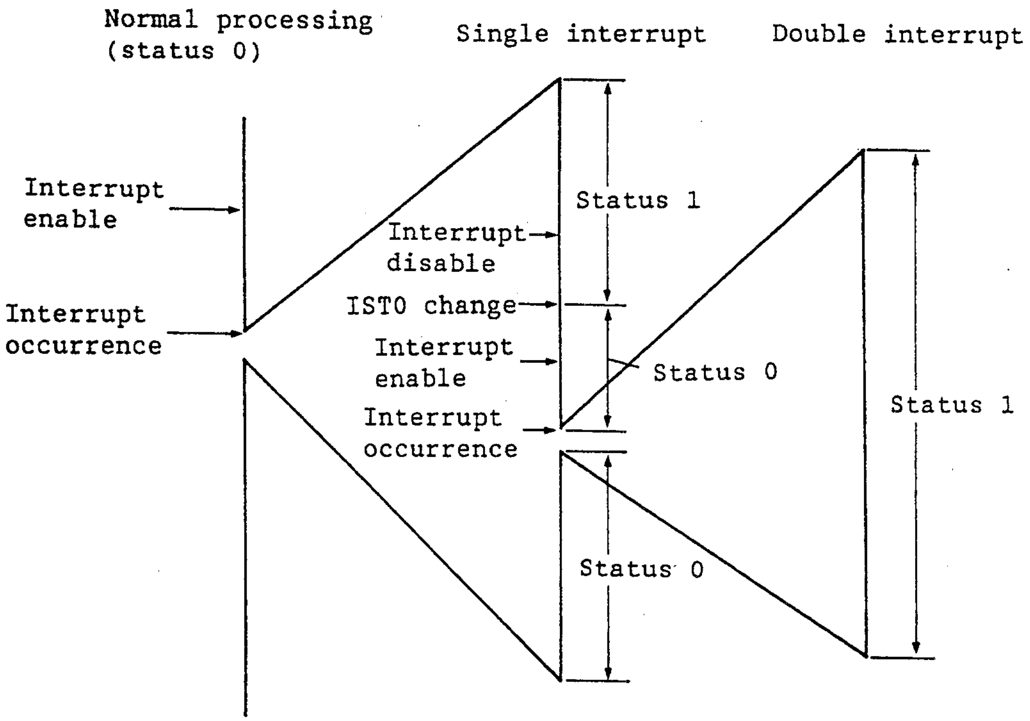
IST0	Status of processing being performed	CPU processing	Interrupt requests that can be acknowledged	After interrupt is acknowledged
				IST0
0	Status 0	Normal program processing	All interrupts can be acknowledged	1
1	Status 1	Interrupt processing	None of interrupts can be acknowledged	——

When an interrupt is acknowledged, IST0 is saved in stack memory together with other PSW bits, then set to 1. When RETI instruction is executed, IST0 is restored to 0.

As understood from Table 5-3, if the interrupt status flag value is changed by a program, multiple interrupt is enabled. That is, if the IST0 value is changed to 0 by the interrupt service program to change the status to status 0, multiple interrupt is enabled.

Change the IST0 value when interrupts are disabled previously by executing the DI instruction.

Fig. 5-7 Multiple Interrupt by Changing the Interrupt Status Flag Value



5.5 Vector Address Sharing Interrupt Processing

Since INTBT and INT4 interrupt sources share vector table, interrupt source is selected as described below:

(1) To use only one interrupt

Set the interrupt enable flag corresponding to the necessary one of the two interrupt sources sharing the vector table to 1 and clear the interrupt enable flag corresponding to the other interrupt source. In this case, an interrupt request is caused by the enabled interrupt source (IExxx=1), and when it is acknowledged, the interrupt request flag is reset to 0.

(2) To use both interrupts

Set both the interrupt enable flags corresponding to the two interrupt sources to 1. In this case, the interrupt request flags of the two interrupt sources are ORed together for an interrupt request.

Even if interrupt request is acknowledged by setting one or both of the interrupt request flags to 1, neigher interrupt request flag is reset to 0.

Thus, in this case, the interrupt service routine must decide which interrupt source causes the interrupt. To do this, DI instruction is executed at the beginning of the interrupt service routine and the SKTCLR instruction is executed to check the interrupt request flags

Remarks: When one interrupt only is enabled, the source of the interrupt that occurred is known, thus the interrupt request flag is reset to 0 by the hardware when interrupt is acknowledged.

When both interrupts are enabled, the source of the interrupt that occurred in unknown, thus the interrupt request flag is not reset to 0 by the hardware when interrupt is acknowledged.

Therefore, to decide wich source causes the interrupt, the interrupt request flags are checked by the software.

6. STANDBY FUNCTION

The standby mode STOP or HALT can be used on the uPD75028(A) to reduce power consumption during program standby.

6.1 Standby Mode Setting and Operation State

Table 6-1 Operation state in standby mode

		STOP mode	HALT mode
Set instruction		STOP instruction	HALT instruction
System clock when the mode is set		Can be set only when main system clock is used	Can be set when either main system clock or subsystem clock is used
Operation state	Clock oscillator	Oscillation of main system clock only stops	CPU clock Φ only stops (oscillation continues)
	Basic interval timer	Operation stop	Operation only during main system clock oscillation (IRQBT setting at reference time intervals)
	Serial interface	Can operate only when extrnal $\overline{SCK}$ input is selected for serial clock	Operation only when external $\overline{SCK}$ input is selected for serial clock or during main system clock oscillation
	Timer/event counter	Can operate only whe TI0 pin input is selected for count clock	Operation only when TI0 pin input is specified for count clock or during main system clock oscillation
	Watch timer	Can operate when f_{XT} is selected for count clock	Can operate
	A/D converter	Operation stop	Can operate (Note)
	Multifunction timer	Operation stop	Can operate (Note)
	External interrupt	INT1, INT2, and INT4 can operate. INT0 only cannot operate.	
	CPU	Operation stop	
Release signal		$\overline{RESET}$ input or interrupt request signal, from hardware that can operate, enabled on interrupt enable flag	$\overline{RESET}$ input or interrupt request signal, from hardware that can operate, enabled on interrupt enable flag

Note: Can operate only when the main system clock oscillates.

Set the STOP mode by executing the STOP instruction to set PCC bit 3 to 1. Set the HALT mode by executing the HALT instruction to set PCC bit 2 to 1.

When CPU operation clock is changed by setting the low-order two bits of PCC, time-lag may occur from PCC rewrite to CPU clock change, as listed in Table 4-5. To change the operation clock before the standby mode is set and the CPU clock after the standby mode is released, set the standby mode in machine cycles required for CPU clock change after PCC is rewritten.

The standby mode holds data in all registers and data memory which stop operation during the standby mode, such as general purpose registers, flags, mode registers, and output latches.

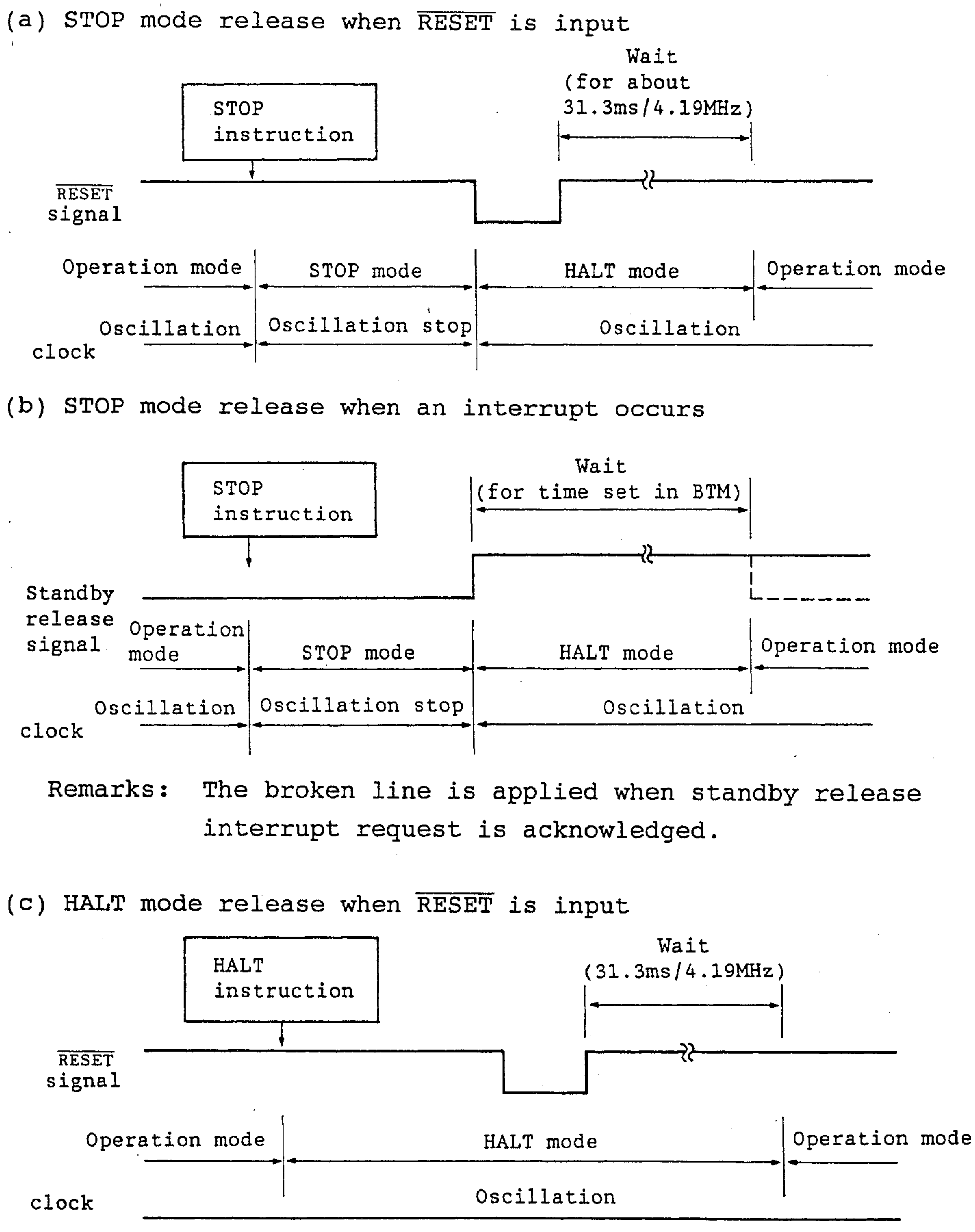
Caution 1: When the STOP mode is set, X1 input is internally connected to V_{SS} (GND potential) to suppress crystal oscillator leakage. In an external clock using system, do not use the STOP mode.

2: Since interrupt request signal is used to release the standby mode, if there is an interrupt source for which both the interrupt request flag and interrupt enable flag are set, the standby mode will be released immediately if the standby mode is entered. Thus, for the STOP mode, the HALT mode is entered just after the STOP instruction is executed, and a wait is made as long as the time set in the BTM register before returning to the operation mode.

6.2 Standby Mode Release

Each of the STOP mode and HALT mode is released when $\overline{\text{RESET}}$ is input or interrupt request signal enabled on interrupt enable flag is generated. Fig. 6-1 shows standby mode release operation.

Fig. 6-1 Standby Mode Release Operation



(a) STOP mode release when $\overline{\text{RESET}}$ is input

(b) STOP mode release when an interrupt occurs

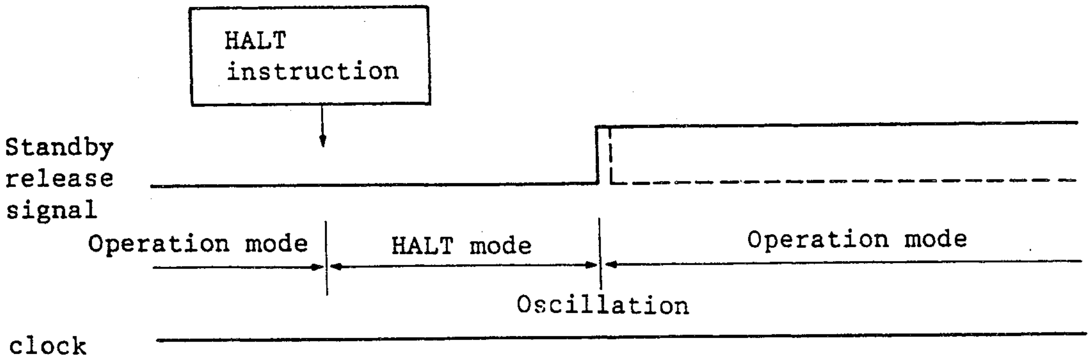
Remarks: The broken line is applied when standby release interrupt request is acknowledged.

(c) HALT mode release when $\overline{\text{RESET}}$ is input

(to be continued)

Fig. 6-1 Standby Mode Release Operation (Cont'd)

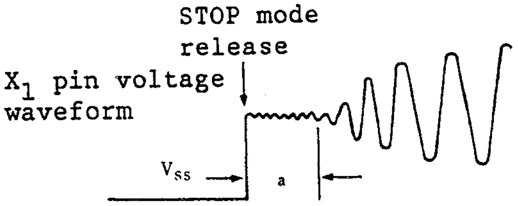
(d) HALT mode release when an interrupt occurs



Remarks: The broken line is applied when standby release interrupt request is acknowledged.

The wait time when the STOP mode is released, regardless of $\overline{\text{RESET}}$ input or interrupt occurrence, does not contain the time until clock oscillation starts after the STOP mode is released. (Fig. 6-2 a)

Fig. 6-2 Oscillation Start State



If the STOP mode is released by causing an interrupt to occur, the wait time is determined by setting BTM. (See Table 6-2.)

Table 6-2 Wait Time Selection by Setting BTM

BTM3	BTM2	BTM1	BTM0	Wait time (Note). The value enclosed in parentheses is applied when f_X=4.19 MHz.
-	0	0	0	About $2^{20}/f_X$ (about 250 ms)
-	0	1	1	About $2^{17}/f_X$ (about 31.3 ms)
-	1	0	1	About $2^{15}/f_X$ (about 7.82 ms)
-	1	1	1	About $2^{13}/f_X$ (about 1.95 ms)
Other than above				Undefined

Note: This time does not contain the time until oscillation starts after the STOP mode is released.

6.3 Operation after Standby Mode is Released

(1) If the stnadby mode is released by inputting $\overline{\text{RESET}}$, normal reset operation is performed.

(2) If the standby mode is released by causing an inter rupt to occur, whether or not a vectored interrupt is executed when the CPU restarts instruction execution is determined by the interrupt master enable flag IME contents.

(a) When IME=0
After the standby mode is released, execution restarts at the instruction following the standby mode setting instruction.
The interrupt request flag is held.

(b) When IME=1
After the standby mode is released, two instructions are executed before a vectored interrupt is executed. However, if the standby mode is released by INTW or INT2 (testable input), no vectored interrupt occurs, and processing as in (a) is performed.

7. RESET FUNCTION

When $\overline{\text{RESET}}$ is input, the uPD75028(A) is reset and the hardware is initialized as listed in Table 7-1.

Fig. 7-1 shows the reset operation timing.

Fig. 7-1 Reset Operation Timing when $\overline{\text{RESET}}$ is Input

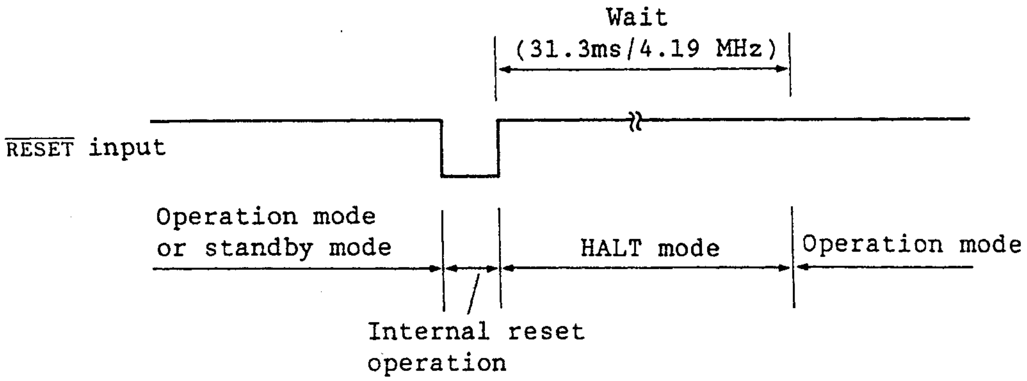


Table 7-1 Hardware State after Reset

Hardware		$\overline{\text{RESET}}$ input during standby mode	$\overline{\text{RESET}}$ input during operation
Program counter (PC)		The low-order five bits of program memory address 0000H are set in PC12-8 and the contents of address 0001H are set in PC7-0.	Same as left
PSW	Carry flag (CY)	Held	Undefined
	Skip flag (SK0-2)	0	0
	Interrupt status flag (IST0)	0	0
	Bank enable flag (MBE)	Bit 7 of program memory address 0000H is set in MBE	Same as left
Stack pointer (SP)		Undefined	Undefined
Data memory (RAM)		Held (Note)	Undefined
General purpose registers (X, A, H, L, D, E, B, and C)		Held	Undefined
Bank selection register (MBS)		0	0
Basic interval timer	Counter (BT)	Undefined	Undefined
	Mode register (BTM)	0	0
Timer event counter	Counter (T0)	0	0
	Modulo register (TMOD0)	FFH	FFH
	Mode register (TM0)	0	0
	TOE0, TOUT F/F	0, 0	0, 0
Watch timer	Mode register (WM)	0	0

Note: When RESET is input, data at data memory addresses 0F8H-0FDH becomes undefined.

Table 7-1 Hardware State after Reset (Cont'd)

Hardware		$\overline{\text{RESET}}$ input during standby mode	$\overline{\text{RESET}}$ input during operation
Serial interface	Shift register (SIO)	Held	Undefined
	Operation mode register (CSIM)	0	0
	SBI control register (SBIC)	0	0
	Slave address register (SVA)	Held	Undefined
Clock generator, clock output circuit	Processor clock control register (PCC)	0	0
	System clock control register (SCC)	0	0
	Clock output mode register (CLOM)	0	0
Interrupt function	Interrput enable flag (IExxx)	0	0
	Interrupt master enable flag (IME)	0	0
	INT0, INT1, and INT2 mode registers (IM0, IM1, and IM2)	0, 0, 0	0, 0, 0
Digital port	Output buffer	Off	Off
	Output latch	Reset to 0	Reset to 0
	Input/output mode register (PMGA, B, C)	0	0
	Pull-up resistor specification register (POGA and POGB)	0	0
	Pull-down resistor specification register (PDGB)	0	0

Table 7-1 Hardware State after Reset (Cont'd)

Hardware		$\overline{\text{RESET}}$ input during standby mode	$\overline{\text{RESET}}$ input during operation
Multifunction timer	Counter (MFTL)	FFH	FFH
	Counter (MFTH)	0	0
	Mode register (MFTM)	0	0
	Control register (MFTC)	0	0
A/D converter	Mode register (ADM)	04H	04H
	SA register (SA)	Held	Undefined
Bit sequential buffer (BSB0-BSB3)		Held	Undefined

8. INSTRUCTION SET

8.1 uPD75028(A) Featuring Instruction

(1) GETI instruction

The GETI instruction is a 1-byte instruction which references 2-byte table in the program memory and executes any operation of the following:

(a) Subroutine call in all space by using the table data as the call instruction call address.

(b) Branch to all space by using the table data as the branch instruction branch address.

(c) Execution of the table data as operation code of a 2-byte instruction or operation codes of two 1-byte instructions.

The GETI instruction is greatly effective for reduction of the number of program steps. Addresses of tables referenced by GETI instruction execution are program memory addresses 0020H-007FH as shown in Fig. 3-2; data can be set in 48 tables.

To describe a table address as an operand, describe an even address.

Remarks 1: 2-byte instructions that can be referenced by GETI instruction execution are limited to 2-machine-cycle instructions.

2: to reference two 1-byte instructions by GETI instruction execution, combinations of the two 1-byte instructions are limited as listed below:

Instruction at the first byte	Instruction at the second byte
MOV A, @HL MOV @HL, A XCH A, @HL	INCS L DECS L INCS H DECS H
MOV A, @DE XCH A, @DE	INCS E DECS E INCS D DECS D
MOV A, @DL XCH A, @DL	INCS L DECS L INCS D DECS D

Since PC is not incremented during GETI instruction execution, processing is continued at the address following the GETI instruction after reference instruction execution.

If the instruction preceding the GETI instruction has the skip function, the GETI instruction is skipped as with any other 1-byte instruction. If the instruction referenced by GETI instruction execution has the skip function, the instruction following the GETI instruction is skipped.

When a string effect instruction is referenced by GETI instruction execution,

- If the instruction proceding the GETI instruction also has the string effect of the same group, when

the GETI instruction is executed, the string effect is lost and the referenced instruction is not skipped.
If the instruction following the GETI instruction also has the string effect of the same group, the string effect caused by the referenced instruction is valid, and the following instruction is skipped.

(2) Bit manipulation instruction

In addition to normal 1-bit manipulation instructions such as set and clear, bit test instruction and bit Boolean instructions (AND, OR, and XOR) are enhanced for the uPD75028. The bit to be manipulated is specified by bit manipulation addressing.

The bit manipulation addressing modes that can be used and the bits manipulated in each addressing mode are listed below:

Addressing	Peripheral hardware that can be addressed	Address range of bits that can be addressed
fmem. bit	MBE/IST0/ IExxx/IRQxxx	FB0H-FBFH
	PORT0-11	FF0H-FFFH
pmem. @L	BSB0 PORT0, 4	FC0H-FFFH
@H+mem. bit	All peripheral hardware that can be handled bit-wise	All bits that can be manipulated bit-wise, of the memory bank indicated by MB

xxx: 0, 1, 2, 4, BT, T0, W, CSI, MFT
MB=MBE•MBS

(3) String effect instruction

If the instructions of the same group of the following three instructions are placed in two or more contiguous addresses in a program, the instruction placed at the start point of the string effect instructions is execued, then one string effect instruction is replaced with one NOP instruction for execution:

```
  Group A:  MOV A, #n4, MOV XA, #n8
  Group B:  MOV HL, #n8
```

(4) Notation adjustment instruction

The uPD75028 contains notation adjustment instruction to adjust the 4-bit data addition or subtraction result to any desired notation.

Assume that the notation to adjust to is m the addition or subtraction result between the accumulator and memory location addressed by register pair HL is adjusted to m notation.
by combining

```
   .  during addition        ADDS A, #16-m
                             ADDC A, @HL
                             ADDS A, #m

   .  during subtraction     SUBC A, @HL
                             ADDS A, #m
```

However, when subtraction is performed, the m complement of the subtraction result is set in the accumulator. An overflow or underflow is left in the carry flag. (When these instructions are combined, the ADDS A, #m instruction skip function is inhibited.)

8.2 Instruction Set and Operation

(1) Operand identifiers and description

In the operand field of each instruction, describe operands comforming to the description for the operand identifiers of the instruction. (For details, see the assembler specifications.) Select one of the entries under the description. Uppercase alphabetic characters and symbols are keywords which should be described exactly as shown.

For immediate data, describe a proper numeric value or label.

The symbols shown in the register and flag formats in Chapters 3-5 can be described as labels in place of mem, fmem, pmem, bit, etc. (However, the labels that can be described in fmem and pmem are limited. See the table on the preceding page.)

Identifier	Description
reg reg1	X, A, B, C, D, E, H, L X, B, C, D, E, H, L
rp rp1 rp2	XA, BC, DE, HL BC, DE, HL BC, DE
rpa rpa1	HL, DE, DL DE, DL
n4 n8	4-bit immediate data or label 8-bit immediate data or label
mem (Note) bit	8-bit immediate data or label 2-bit immediate data or label
fmem pmem	FB0H-FBFH, FF0H-FFFH immediate data or label FC0H-FFFH immediate data or label
addr caddr faddr	0000H-1F7FH immediate data or label 12-bit immediate data or label 11-bit immediate data or label
taddr	20H-7FH immediate data (where bit 0=0) or label
PORTn IExxx MBn	PORT0-PORT11 IEBT, IECSI, IET0, IE0, IE1, IE2, IE4, IEW, IEMFT MB0, MB1, MB15

Note: For 8-bit data processing, only an even address can be specified in mem.

(2) Legend on operation explanation

A	:	A register or 4-bit accumulator
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
X	:	X register or 4-bit accumulator
XA	:	Register pair (XA) or 8-bit accumulator
BC	:	Register pair (BC)
DE	:	Register pair (DE)

HL	:	Register pair (HL)
PC	:	Program counter
SP	:	Stack pointer
CY	:	Carry flag or 1-bit accumulator
PSW	:	Program status word
MBE	:	Memory bank enable flag
PORNn	:	Port n (n=0-11)
IME	:	Interrupt master enable flag
IExxx	:	Interrupt enable flag
MBS	:	Memory bank selection register
PCC	:	Processor clock control register
.	:	Separation of address and bit
(xx)	:	Contents addressed by xx
xxH	:	Hexadecimal data

(3) Symbol explanation under addressing area

*1	MB=MBE·MBS (MBS=0, 1, 15)	Data memory addressing
*2	MB=0	

*3	MBE=0:MB=0 (00H-7FH)
	MB=15 (80H-FFH)
	MBE=1:MB=MBS (MBS=0, 1, 15)	

*4	MB=15, fmem=FB0H-FBFH, FF0H-FFFH	
*5	MB=15, pmem=FC0H-FFFH	
*6	addr=0000H-1F7FH	Program memory addressing
*7	addr= (Current PC) -15 to (Current PC) -1 (Current PC) +2 to (Current PC) +16	
*8	caddr=0000H-0FFFH (PC_{12}=0) or 1000H-1F7FH (PC_{12}=1)	
*9	faddr=0000H-07FFH	
*10	taddr=0020H-007FH	

Remarks 1: MB denotes memory bank that can be access

2: *2: MB=0 regardless of MBE or MBS.

3: *4 and *5: MB=15 regardless of MBE or MBS.

4: *6 to *10: Area that can be addressed is shown.

(4) Explanation of number of machine cycles

S denites the number of machine cycles required when an instruction with skip performs skip operation. The value of S varies as follows:

. when not skipped: S=0
. when the skipped instruction is a 1-byte or 2-byte instruction: S=1
. when the skipped instruction is a 3-byte instruction (BR !addr or CALL !addr): S=2

Caution: The GETI instruction is skipped in one machine cycle.

One machine cycle (t_{CY}) equals to one cycle of CPU clock Φ. One of three types of time can be selected by setting PCC. (See 4.2 (3).)

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Transfer instructions	MOV	A, #n4	1	1	A ← n4		Vertically stored A
		regl, #n4	2	2	regl ← n4		
		XA, #n8	2	2	XA ← n8		Vertically stored A
		HL, #n8	2	2	HL ← n8		Vertically stored B
		rp2, #n8	2	2	rp2 ← n8		
		A, @HL	1	1	A ← (HL)	*1	
		A, @rpal	1	1	A ← (rpal)	*2	
		XA, @HL	2	2	XA ← (HL)	*1	
		@HL, A	1	1	(HL) ← A	*1	
		@HL, XA	2	2	(HL) ← XA	*1	
		A, mem	2	2	A ← (mem)	*3	
		XA, mem	2	2	XA ← (mem)	*3	
		mem, A	2	2	(mem) ← A	*3	
		mem, XA	2	2	(mem) ← XA	*3	
		A, regl	2	2	A ← regl		
		XA, rp	2	2	XA ← rp		
		regl, A	2	2	regl ← A		
		rpl, XA	2	2	rpl ← XA		

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Transfer instructions	XCH	A, @HL	1	1	A ←→ (HL)	*1	
		A, @rpal	1	1	A ←→ (rpal)	*2	
		XA, @HL	2	2	XA ←→(HL)	*1	
		A, mem	2	2	A ←→ (mem)	*3	
		XA, mem	2	2	XA ←→ (mem)	*3	
		A, regl	1	1	A ←→ regl		
		XA, rp	2	2	XA ← rp		
	MOVT	XA, @PCDE	1	3	XA ← $(PC_{12-8}+DE)_{ROM}$		
		XA, @PCXA	1	3	XA ← $(PC_{12-8}+XA)_{ROM}$		
Operating instructions	ADDS	A, #n4	1	1+S	A ← A+n4		carry
		A, @HL	1	1+S	A ← A+(HL)	*1	carry
	ADDC	A, @HL	1	1	A, CY ← A+(HL)+CY	*1	
	SUBS	A, @HL	1	1+S	A ← A-(HL)	*1	borrow
	SUBC	A, @HL	1	1	A, CY ← A-(HL)-CY	*1	
	AND	A, #n4	2	2	A ← A∧n4		
		A, @HL	1	1	A ← A∧(HL)	*1	

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Operating instructions	OR	A, #n4	2	2	$A \leftarrow A \lor n4$		
		A, @HL	1	1	$A \leftarrow A \lor (HL)$	*1	
	XOR	A, #n4	2	2	$A \leftarrow A \veebar n4$		
		A, @HL	1	1	$A \leftarrow A \veebar (HL)$	*1	
Accumulator handling (manipulation) instructions	RORC	A	1	1	$CY \leftarrow A_0,\ A_3 \leftarrow CY,\ A_{n-1} \leftarrow A_n$		
	NOT	A	2	2	$A \leftarrow \overline{A}$		
Increment and Decrement instructions	INCS	reg	1	1+S	reg ← reg+1		reg=0
		@HL	2	2+S	(HL) ← (HL)+1	*1	(HL)=0
		mem	2	2+S	(mem) ← (mem)+1	*3	(mem)=0
	DECS	reg	1	1+S	reg ← reg-1		reg=FH
Comparison instruction	SKE	reg, #n4	2	2+S	Skip if reg=n4		reg=n4
		@HL, #n4	2	2+S	Skip if(HL)=n4	*1	(HL)=n4
		A, @HL	1	1+S	Skip if A=(HL)	*1	A=(HL)
		A, reg	2	2+S	Skip if A=reg		A=reg
Carry flag handling (manipulation) instructions	SET1	CY	1	1	CY ← 1		
	CLR1	CY	1	1	CY ← 0		
	SKT	CY	1	1+S	Skip if CY=1		CY=1
	NOT1	CY	1	1	$CY \leftarrow \overline{CY}$		

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Memory bit manipulation instructions	SET1	mem.bit	2	2	(mem.bit) ← 1	*3	
		fmem.bit	2	2	(fmem.bit) ← 1	*4	
		pmem.@L	2	2	(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$)) ← 1	*5	
		@H+mem.bit	2	2	(H+mem$_{3-0}$.bit) ← 1	*1	
	CLR1	mem.bit	2	2	(mem.bit) ← 0	*3	
		fmem.bit	2	2	(fmem.bit) ← 0	*4	
		pmem.@L	2	2	(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$)) ← 0	*5	
		@H+mem.bit	2	2	(H+mem$_{3-0}$+bit) ← 0	*1	
	SKT	mem.bit	2	2+S	Skip if(mem.bit)=1	*3	(mem.bit)=1
		fmem.bit	2	2+S	Skip if(fmem.bit)=1	*4	(fmem.bit)=1
		pmem.@L	2	2+S	Skip if(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$)=1	*5	(pmem.@L)=1
		@H+mem.bit	2	2+S	Skip if(H+mem$_{3-0}$.bit)=1	*1	(@H+mem.bit)=1
	SKF	mem.bit	2	2+S	Skip if(mem.bit)=0	*3	(mem.bit)=0
		fmem.bit	2	2+S	Skip if(fmem.bit)=0	*4	(fmem.bit)=0
		pmem.@L	2	2+S	Skip if(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$)=0	*5	(pmem.@L)=0
		@H+mem.bit	2	2+S	Skip if(H+mem$_{3-0}$.bit)=0	*1	(@H+mem.bit)=0

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Memory bit manipulation instructions	SKTCLR	fmem.bit	2	2+S	Skip if(fmem.bit)=1 and clear	*4	(fmem.bit)=1
		pmem.@L	2	2+S	Skip if(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$))=1 and clear	*5	(pmem.@L)=1
		@H+mem.bit	2	2+S	Skip if(H+mem$_{3-0}$.bit)=1 and clear	*1	(@H+mem.bit)=1
	AND1	CY, fmem.bit	2	2	CY ← CY $\wedge$(fmem.bit)	*4	
		CY, pmem.@L	2	2	CY ← CY$\wedge$(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$))	*5	
		CY, @H+mem.bit	2	2	CY ← CY $\wedge$(H+mem$_{3-0}$.bit)	*1	
	OR1	CY, fmem.bit	2	2	CY ← CY$\vee$(fmem.bit)	*4	
		CY, pmem.@L	2	2	CY ← CY$\vee$(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$))	*5	
		CY, @H+mem.bit	2	2	CY ← CY$\vee$(H+mem$_{3-0}$.bit)	*1	
	XOR1	CY, fmem.bit	2	2	CY ← CY$\veebar$(fmem.bit)	*4	
		CY, pmem.@L	2	2	CY ← CY$\veebar$(pmem$_{7-2}$+L$_{3-2}$.bit(L$_{1-0}$))	*5	
		CY, @H+mem.bit	2	2	CY ← CY$\veebar$(H+mem$_{3-0}$.bit)	*1	
Branch instructions	BR	addr	-	-	PC$_{12-0}$ ← addr (Select appropriate instruction from among BR !addr, BRCB !caddr, and BR $addr according to the assembler being used.)	*6	
		!addr	3	3	PC$_{12-0}$ ← addr	*6	
		$addr	1	2	PC$_{12-0}$ ← addr	*7	
	BRCB	!caddr	2	2	PC$_{12-0}$ ← PC$_{12-0}$+caddr$_{11-0}$	*8	

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Subroutine stack control instructions	CALL	!addr	3	3	(SP-4) (SP-1) (SP-2) ← PC_{11-0} (SP-3) ← (MBE, 0, 0, PC_{12}) PC_{12-0} ← addr, SP ← SP-4	*6	
	CALLF	!faddr	2	2	(SP-4) (SP-1) (SP-2) ← PC_{11-0} (SP-3) ← (MBE, 0, 0, PC_{12}) PC_{12-0} ← 00, faddr, SP ← SP-4	*9	
	RET		1	3	(MBE, X, X, PC_{12}) ← (SP+1) PC_{11-0} ← (SP) (SP+3) (SP+2) SP ← SP+4		
	RETS		1	3+S	(MBE, X, X, PC_{12}) ← (SP+1) PC_{11-0} ← (SP) (SP+3) (SP+2) SP ← SP+4, then skip unconditionally		Unconditional
	RETI		1	3	(MBE, X, X, PC_{12} ← (SP+1) PC_{11-0} ← (SP) (SP+3) (SP+2) PSW ← (SP+4) (SP+5), SP SP+6		
	PUSH	rp	1	1	(SP-1) (SP-2) ← rp, SP ← SP-2		
		BS	2	2	(SP-1) ← MBS, (SP-2) ← 0, SP ← SP-2		
	POP	rp	1	1	rp ← (SP+1) (SP), SP ← SP+2		
		BS	2	2	MBS ← (SP+1), SP ← SP+2		

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Interrupt control instructions	EI		2	2	IME ← 1		
		IExxx	2	2	IExxx ← 1		
	DI		2	2	IME ← 0		
		IExxx	2	2	IExxx ← 0		
Input/output instructions	(Note) IN	A, PORT$_{n}$	2	2	A ← PORT$_{n}$ (n=0-11)		
		XA, PORT$_{n}$	2	2	XA ← PORT$_{n+1}$, PORTn (n=4,6)		
	(Note) OUT	PORT$_{n}$, A	2	2	PORT$_{n}$ ← A (n=2-10)		
		PORT$_{n}$, XA	2	2	PORT$_{n+1}$, PORT$_{n}$ ← XA (n=4,6)		
CPU control instructions	HALT		2	2	Set HALT Mode (PCC.2 ← 1)		
	STOP		2	2	Set STOP Mode (PCC.3 ← 1)		
	NOP		1	1	No operation		

Instruction group	Mnemonic	Operand	Number of bytes	Number of machine cycles	Operation	Addressing area	Skip condition
Special instructions	SEL	MBn	2	2	MBS — n (n=0,1,15)		
	GETI	taddr	1	3	. When TBR instruction is executed $PC_{12-0} \leftarrow (taddr)_{4-0}+(taddr+1)$	*10	
					. When TCALL instruction executed $(SP-4)\ (SP-1)\ (SP-2) \leftarrow PC_{11-0}$ $(SP-3) \leftarrow (MBE,\ 0,\ 0,\ PC_{12})$ $PC_{12-0} \leftarrow (taddr)_{4-0}+(tddr+1)$ $SP \leftarrow SP-4$		
					. When any instruction other than TBR or TCALL (taddr) (taddr+1) instruction execution		Depending on the reference instruction

Caution: MBE=0, MBE=1 and MBS=15 must have been set during execution of an IN or OUT instruction.

9. ELECTRICAL CHARACTERISTICS

Absolute maximum ratings (Ta = 25°C)

Parameter	Symbol	Test condition		Ratings	Unit
Power supply voltage	V_{DD}			-0.3 to +7.0	V
Input voltage	V_{I1}	Except for port 4, 5, 10		-0.3 to V_{DD}+0.3	V
	V_{I2}	Port 4, 5, 10	Pull-up resistor is contained	-0.3 to V_{DD}+0.3	V
			Open drain	-0.3 to +11	V
Output voltage	V_O			-0.3 to V_{DD}+0.3	V
Output high current	I_{OH}	Per pin		-10	mA
		Total, all outputs		-30	mA
Output low current	I_{OL} (Note)	Port 0, 3, 4, 5 per pin	Peak value	30	mA
			rms value	15	mA
		Except for port 0, 3, 4, 5 per pin	Peak value	20	mA
			rms value	5	mA
		Port 0, 3 to 9, 11 total	Peak value	170	mA
			rms value	120	mA
		Port 0, 2, 10 total	Peak value	30	mA
			rms value	20	mA
Operation temperature	T_{opt}			-40 to +85	°C
Storage temperature	T_{stg}			-65 to +150	°C

Note: To obtain the rms value, calculate
[rms value] = [peak value] x $\sqrt{duty}$

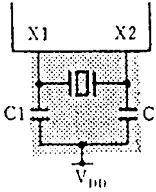
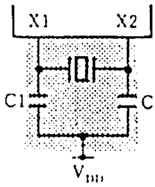
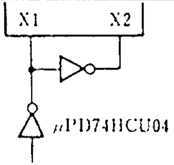
Operation temperature range

Temperature. (Ta) Item		-40 to +70°C	-40 to +85°C
CPU clock	Select the main system clock	Operation enable	
	Select the sub-system clock	Operation enable	Operation disable
Subsystem clock oscillation circuit		Operation enable(Note 1)	
A/D converter		Operation enable(Note 2)	
Other hardware		Operation enable	

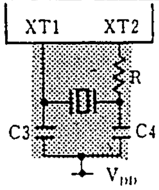
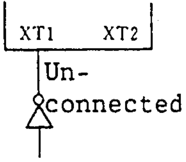
Note 1: Don't exceed +70°C as temperature range when subsystem clock is selected CPU clock.

2: Precision is different by temperature range.

Main systerm clock oscillator characteristics (Ta = -40 to +85°C, V_{DD} = 2.7 to 6.0V)

Resonator	Recommended constant	Parameter	Test condition	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X) (Note 1)	V_{DD} - oscillation frequency	2.0		5.0	MHz
		Oscillation stabilization time (Note 2)	After V_{DD} reaches MIN of oscillation voltage range			4	ms
Crystal resonator		Oscillation frequency (f_X) (Note 1)		2.0	4.19	5.0	MHz
		Oscillation stabilization time (Note 2)	V_{DD} - 4.5 to 6.0V			10	ms
						30	ms
External clock		X1 input frequency (f_X) (Note 1)		2.0		5.0	MHz
		X1 input high, low level width (t_{XH}, t_{XL})		100		250	ns

Subsystem clock oscillator characteristics (Ta = -40 to +85°C, V_{DD} = 2.7 to 6.0V)

Resonator	Recommended constant	Parameter	Test condition	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_X) (Note 1)		32	32.768	35	kHz
		Oscillation stabilization time (Note 2)	V_{DD} = 4.5 to 6.0V		1.0	2	s
						10	s
External clock		XT1 input frequency (f_X) (Note 1)		32		100	kHz
		XT1 input high, low level width (t_{XTH}, t_{XTL})		5		15	us

Note 1: It indicates only the oscillator characteristics. For the instruction execution time, see the AC characteristics.

2: Time required until oscillation becomes stable after V_{DD} application or STOP mode release.

3: When oscillation frequency is 4.19 MHz < f_X 5.0 MHz, do not set PCC = 0011 for the CPU clock frequcy.
When PCC = 0011, 1 machine cycle falls short of 0.95 us, the minimum value of the limit.

4: When selecting subsystem clock as CPU clock, the range of temperature should be within +70 °C.

Caution: Make the shaded portion in the figure as short as possible. (For details, see 4.2 (5).)

Capacitance (Ta = 25°C, V_{DD} =0V)

Parameter	Symbol	Test condition	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f = 1MHz Unmeasured pins returned to 0V			15	pF
Output capacitance	C_O				15	pF
Input/Output capacitance	C_{IO}				15	pF

DC characteristics (Ta = -40 to +85°C, V_{DD} = 2.7 to 6.0V)

Parameter	Symbol	Test condition		MIN.	TYP.	MAX.	Unit
Input high voltage	V_{IH1}	Port 2, 3, 8, 9, 11		$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	Port 0, 1, 6, 7, $\overline{RESET}$		$0.8V_{DD}$		V_{DD}	V
	V_{IH3}	Port 4, 5, 10	Pull-up resistor is contained	$0.7V_{DD}$		V_{DD}	V
			Open drain	$0.7V_{DD}$		10	V
	V_{IH4}	X1, X2, XT1		VDD-0.5		V_{DD}	V
Input low voltage	V_{IL1}	Port 2 to 5, 8 to 11		0		$0.3V_{DD}$	V
	V_{IL2}	Port 0, 1, 6, 7, $\overline{RESET}$		0		$0.2V_{DD}$	V
	V_{IL3}	X1, X2, XT1		0		0.4	V
Output high voltage	V_{OH}	V_{DD} = 4.5 to 6.0 V, I_{OH} = -1mA		V_{DD}-1.0			V
		I_{OH} = -100uA		V_{DD}-0.5			V
Output low voltage	V_{OL}	Port 3, 4, 5	V_{DD} = 4.5 to 6.0V, I_{OL} = 15mA		0.4	2.0	V
		V_{DD} = 4.5 to 6.0V, I_{OL} = 1.6mA				0.4	V
		I_{OL} = 400uA				0.5	V
		SB0, 1	Open drain Pull-up resistor≥1kΩ			$0.2V_{DD}$	V
Input high leakage current	I_{LIH1}	V_I = V_{DD}	Except for below			3	uA
	I_{LIH2}		X1, X2, XT1			20	uA
	I_{LIH3}	V_I = 9V	Port 4, 5, 10 (when open drain is selected)			20	uA
Input low leakage current	I_{LIL1}	V_I = 0V	Except for below			-3	uA
	I_{LIL2}		X1, X2, XT1			-20	uA
Output high leakage current	I_{LOH1}	V_O = V_{DD}	Except for below			3	uA
	I_{LOH2}	V_O = 9V	Port 4, 5, 10 (when open drain is selected)			20	uA
Output low leakage current	I_{LOL}	V_O = 0V				-3	uA
Internal pull-up resistor	R_{U1}	Port 0, 1, 2, 3, 6, 7, 8 (except P00) V_I = 0V	V_{DD} = 5.0V±10%	15	40	80	kΩ
			V_{DD} = 3.0V±10%	30		300	kΩ
	R_{U2}	Port 4, 5, 10 V_O = V_{DD}-2.0V	V_{DD} = 5.0V±10%	15	40	70	kΩ
			V_{DD} = 3.0V±10%	10		60	kΩ
Internal pull-down resistor	R_D	Port 9 V_{IN} = V_{DD}	V_{DD} = 5.0V±10%	10	40	70	kΩ
			V_{DD} = 3.0V±10%	10		60	kΩ

Parameter	Symbol	Test condition			MIN.	TYP.	MAX.	Unit
Power supply (Note 1) current	I_{DD1}	(Note 2) 4.19MHz crystal oscillation C1 = C2 = 22pF	V_{DD} = 5.0V±10% (Note 3)			2.5	8	mA
			V_{DD} = 3V±10% (Note 4)			0.35	1.2	mA
	I_{DD2}		HALT mode	V_{DD} = 5V±10%		500	1500	uA
				V_{DD} = 3V±10%		150	450	uA
	I_{DD3}	(Note 5) 32.768kHz crystal oscillation	V_{DD} = 3V±10%			30	90	uA
	I_{DD4}		HALT mode	V_{DD} = 3V±10%		5	15	uA
	I_{DD5}	XT1 = 0V STOP mode	V_{DD} = 5V±10%			0.5	20	uA
			V_{DD} = 3V±10%			0.1	10	uA
				Ta = 25°C		0.1	5	uA

Note 1: Current flowing into internal pull-up resistor is not contained.

2: Case where subsystem clock is oscillated is also contained.

3: When the processor clock control register (PCC) is set to 0011 and the uPD75028(A) is operated in high speed mode.
When PCC is set to 0000 and the uPD75028(A) is operated in

4: low speed mode.
When the system clock control register (SCC) is set to

5: 1001, main system clock oscillation is stopped and the uPD75028(A) is operated with subsystem clock.

AC characteristics (Ta = -40 to +85°C, V_{DD} = 2.7 to 6.0V)

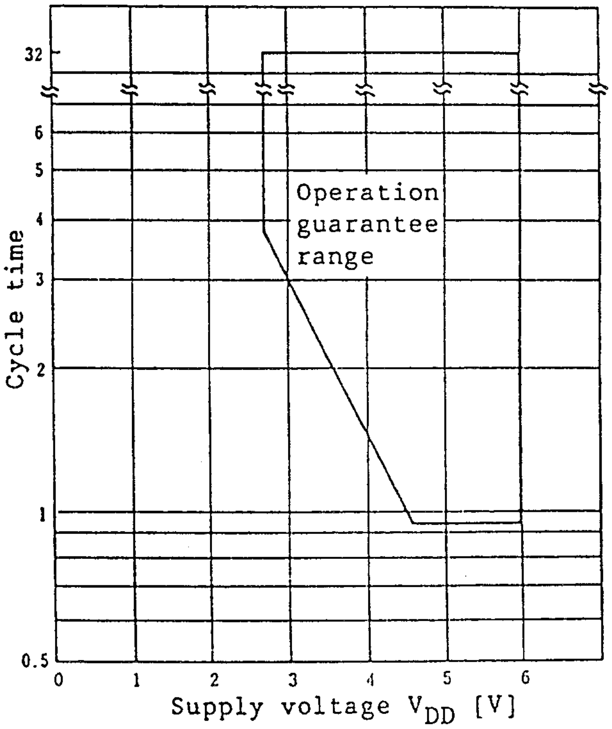
Parameter	Symbol	Test condition		MIN.	TYP.	MAX.	Unit
(Note 1) CPU clock cycle (minimum instruction execution time = 1 machine cycle)	t_{CY}	Operation with main system clock	V_{DD} = 4.5 to 6.0V	0.95		32	us
				3.8		32	us
		Operation with subsystem clock	(Note 2) Ta=-40 to +70°C	114	122	125	us
TIO input frequency	f_{TI}	V_{DD} = 4.5 to 6.0V		0		1	MHz
				0		275	kHz
TIO input high, low level width	t_{TIH}, t_{TIL}	V_{DD} = 4.5 to 6.0V		0.48			us
				1.8			us
Interrupt input high, low level width	t_{INTH}, t_{INTL}	INT0		(Note 2)			us
		INT1, 2, 4		10			us
		KR0 to 7		10			us
$\overline{\text{RESET}}$ low level width	t_{RSL}			10			us

Note 1: The CPU clock (Φ) cycle time is determined by the oscillation frequency of the connected resonator, the system clock control register (SCC), and the processor clock control register (PCC). The right chart shows the cycle time t_{CY} characteristics for power supply voltage V_{DD} during main system clock operation. (See Fig. 4-12 Processor Clock Control Register Format)

2: $2t_{CY}$ or $128/f_X$ depending on how the interrupt mode register (IMO) is set.

3: This value becomes $2t_{CY}$ or $128/f_X$ depensing on the setting of interrupt mode register (IMO).

t_{CY} vs V_{DD} (During main system clock operation)



Serial transfer operation

2-line, 3-line sêrial I/O mode ($\overline{SCK}$ - internal clock output)

Parameter	Symbol	Test condition	MIN.	TYP.	MAX.	Unit
$\overline{SCK}$ cycle time	t_{KCY1}	V_{DD} = 4.5 to 6.0V	1600			ns
			3800			ns
$\overline{SCK}$ high, low level width	t_{KL1}	V_{DD} = 4.5 to 6.0V	$t_{KCY}/2$ - 50			ns
	t_{KH1}		$t_{KCY}/2$ - 150			ns
SI setup time (to $\overline{SCK}$ ↑)	t_{SIK1}		150			ns
SI hold time (from $\overline{SCK}$ ↑)	t_{KSI1}		400			ns
$\overline{SCK}$ ↓ — SO output delay time	t_{KSO1}	(Note) R_L = 1kΩ, C_L = 100pF; V_{DD} = 4.5 to 6.0V			250	ns
					1000	ns

2-line, 3-line serial I/O mode ($\overline{SCK}$ - external clock output)

Parameter	Symbol	Test condition	MIN.	TYP.	MAX.	Unit
$\overline{SCK}$ cycle time	t_{KCY2}	V_{DD} = 4.5 to 6.0V	800			ns
			3200			ns
$\overline{SCK}$ high, low level width	t_{KL2}	V_{DD} = 4.5 to 6.0V	400			ns
	t_{KH2}		1600			ns
SI setup time (to $\overline{SCK}$ ↑)	t_{SIK2}		100			ns
SI hold time (from $\overline{SCK}$ ↑)	t_{KSI2}		400			ns
$\overline{SCK}$ ↓ — SO output delay time	t_{KSO2}	(Note) R_L = 1kΩ, C_L = 100pF; V_{DD} = 4.5 to 6.0V			300	ns
					1000	ns

Note: R_L and C_L are output line load resistance and load capacitance, respectively.

SBI mode ($\overline{SCK}$ - internal clock output (master))

Parameter	Symbol	Test condition		MIN.	TYP.	MAX.	Unit
$\overline{SCK}$ cycle time	t_{KCY3}	V_{DD} = 4.5 to 6.0V		1600			ns
				3800			ns
$\overline{SCK}$ high, low level width	t_{KL3}	V_{DD} = 4.5 to 6.0V		$t_{KCY}/2 - 50$			ns
	t_{KH3}			$t_{KCY}/2 - 150$			ns
SB0, SB1 setup time (to $\overline{SCK}$ ↑)	t_{SIK3}			150			ns
SB0, SB1 hold time (from $\overline{SCK}$ ↑)	t_{KSI3}			$t_{KCY}/2$			ns
$\overline{SCK}$ ↓ → SB0, SB1 output delay time	t_{KSO3}	(Note) R_L = 1kΩ, C_L = 100pF	V_{DD} = 4.5 to 6.0V	0		250	ns
				0		1000	ns
$\overline{SCK}$ ↑ → SB0, 1 ↓	t_{KSB}			t_{KCY}			ns
SB0, 1 ↓ → $\overline{SCK}$	t_{SBK}			t_{KCY}			ns
SB0, SB1 low level width	t_{SBL}			t_{KCY}			ns
SB0, SB1 high level width	t_{SBH}			t_{KCY}			ns

Note: R_L and C_L are SB0, SB1 output line load resistance and load capacitance,respectively.

SBI mode ($\overline{SCK}$ - external clock output (slave))

Parameter	Symbol	Test condition		MIN.	TYP.	MAX.	Unit
$\overline{SCK}$ cycle time	t_{KCY4}	V_{DD} = 4.5 to 6.0V		800			ns
				3200			ns
$\overline{SCK}$ high, low level	t_{KL4}	V_{DD} = 4.5 to 6.0V		400			ns
	t_{KH4}			1600			ns
SB0, SB1 setup time (to $\overline{SCK}$↑)	t_{SIK4}			100			ns
SB0, SB1 hold time (from $\overline{SCK}$↑)	t_{KSI4}			$t_{KCY}/2$			ns
$\overline{SCK}$↓→ SB0, SB1 output delay time	t_{KSO4}	(Note) R_L = 1kΩ, C_L = 100pF	V_{DD} = 4.5 to 6.0V	0		300	ns
				0		1000	ns
$\overline{SCK}$↑ — SB0, 1↓	t_{KSB}			t_{KCY}			ns
SB0, 1↓ — $\overline{SCK}$↓	t_{SBK}			t_{KCY}			ns
SB0, SB1 low level width	t_{SBL}			t_{KCY}			ns
SB0, SB1 high level width	t_{SBH}			t_{KCY}			ns

Note: R_L and C_L are SB0, SB1 output line load resistance and load capacitance,respectively.

A/D converter (Ta = -40 to +85°C, V_{DD} = 2.7 to 6.0V, AV_{SS} = V_{SS} = 0V)

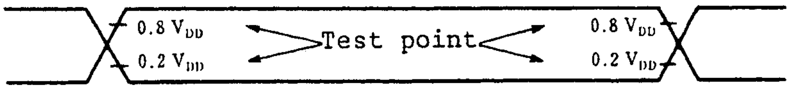
Parameter	Symbol	Test condition		MIN.	TYP.	MAX.	Unit
Resolution						8	bit
Absolute accuracy (Note 1)		$2.5V \leq AV_{REF} \leq AV_{DD}$	$-10 \leq$ Ta $\leq +85°C$			±1.5	LSB
			$-40 \leq$ Ta $< -10°C$			±2.0	
Conversion time (Note 2)	t_{CONV}					$168/f_X$	us
Sampling time (Note 3)	t_{SAMP}					$44/f_X$	us
Analog input voltage	V_{IAN}			AV_{REF-}		AV_{REF+}	V
Analog supply voltage	AV_{DD}			2.5		V_{DD}	V
Reference input voltage	AV_{REF+}	$2.5V \leq (AV_{REF+}) - (AV_{REF-})$.		2.5		AV_{DD}	V
Reference input voltage	AV_{REF-}	$2.5V \leq (AV_{REF+}) - (AV_{REF-})$		0		1.0	V
Analog input high impedance	R_{AN}				1000		MΩ
V_{AREF} current	I_{AREF}				0.25	2.0	mA

Note 1: Absolute accuracy from which quantization error (±1/2LSB) is removed.

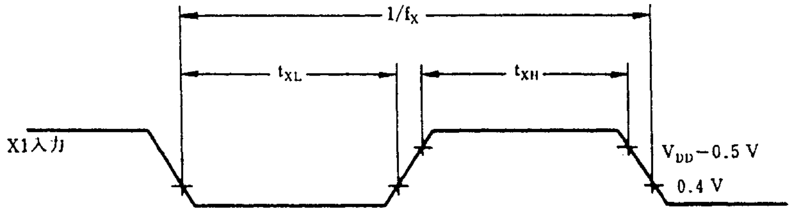
2: Time until conversion end (EOC=1) after conversion start instruction execution (40.1us: During f_X=4.19 MHz operation).

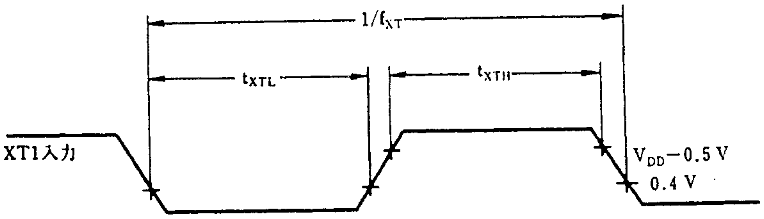
3: Time until sampling end after conversion start instruction execution (10.5 us: During f_X=4.19 MHz operation).

AC timing test points (except for X1 or XT1 input)

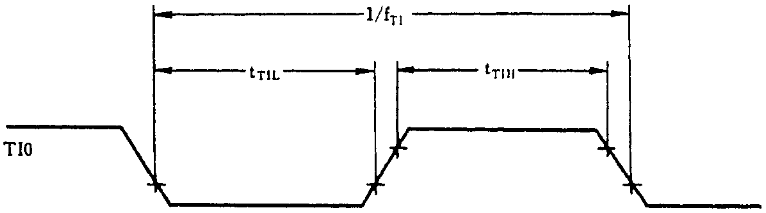


Clock timing



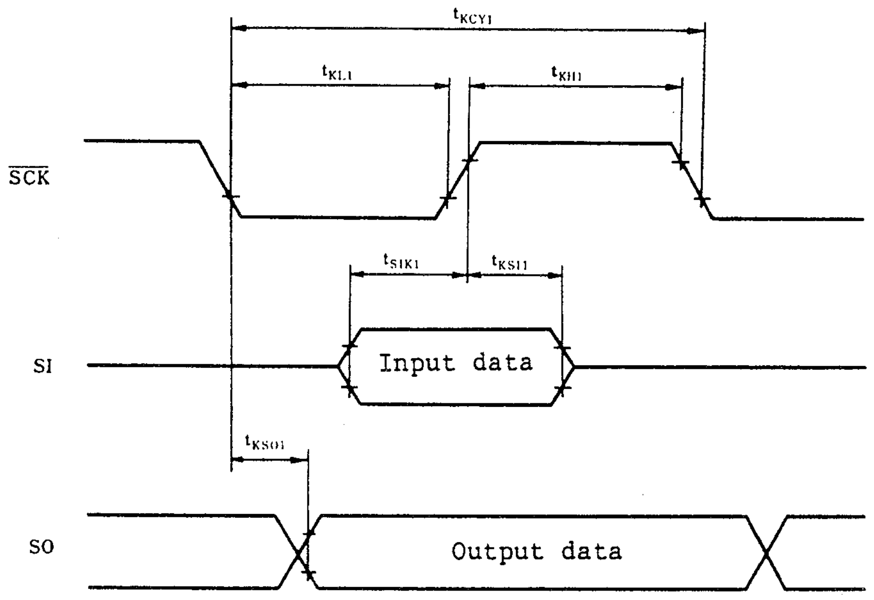


TI0 timing

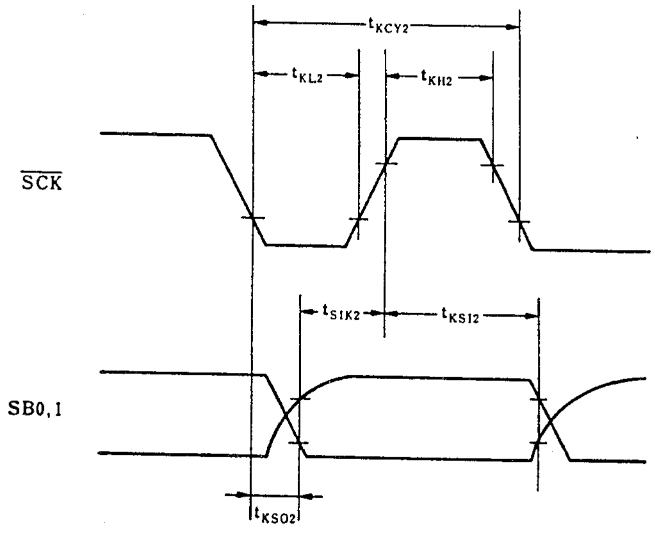


Serial transfer timing

3-line serial I/O mode:

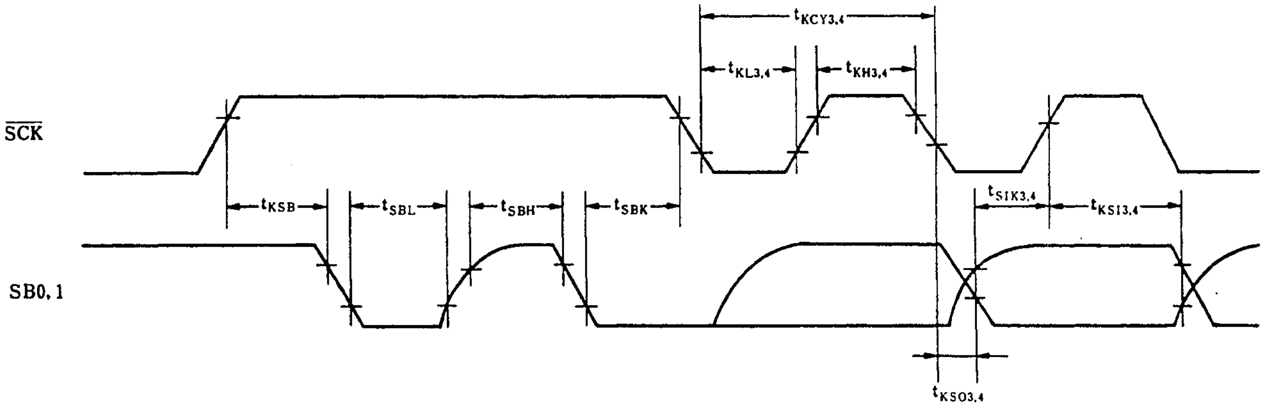


2-line serial I/O mode:

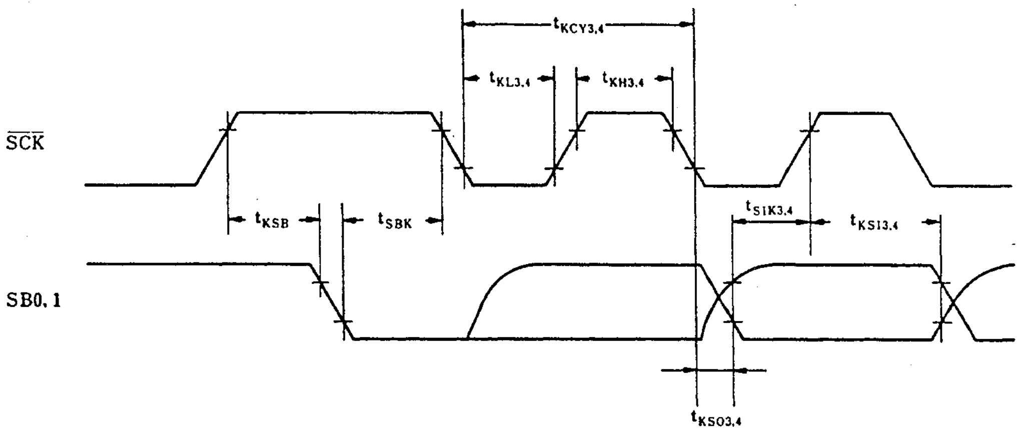


Serial transfer timing

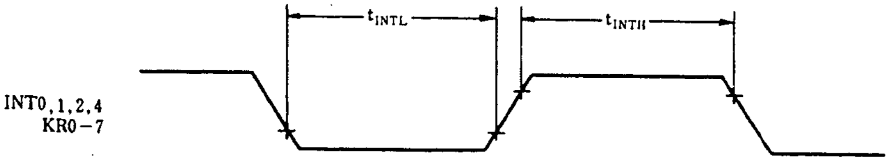
Bus release signal transfer:



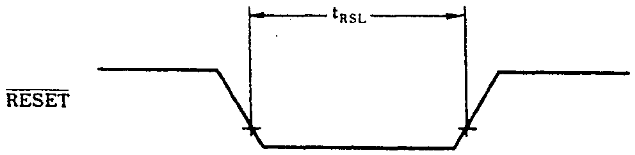
Command signal transfer:



Interrupt input timing:



$\overline{\text{RESET}}$ input timing



Data memory STOP mode low supply voltage data retention characteristics (Ta = -40 to +85°C)

Parameter	Symbol	Test condition	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		2.0		6.0	V
(Note 1) Data retention supply current	I_{DDDR}	$V_{DDDR} = 2.0V$		0.1	10	uA
Release signal set time	t_{SREL}		0			us
(Note 2) Oscillation stable wait time	t_{WAIT}	Release when $\overline{RESET}$ is input		$2^{17}/f_X$		ms
		Release when interrupt request occurs		(Note 3)		ms

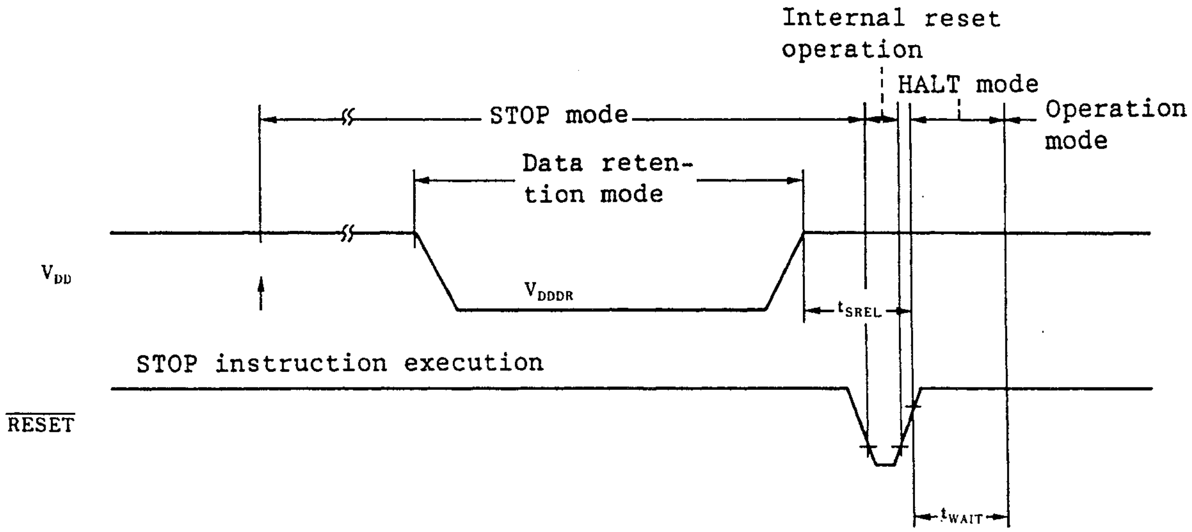
Note 1: Current into internal pull-up resistor is not contained.

2: The oscillation stable wait time is the CPU operation stop time to prevent unstable operation when oscillation starts.

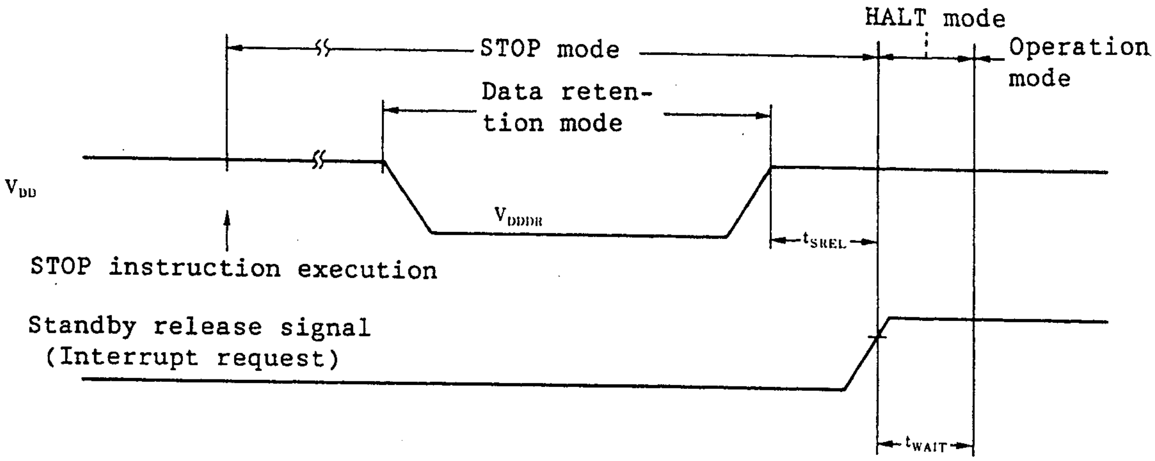
3: Depending on how the basic interval timer mode register (BTM) is set. (See the following table.)

BTM3	BTM2	BTM1	BTM0	Wait time. The values enclosed in parentheses apply when f_X = 4.19 MHz.
—	0	0	0	$2^{20}/f_X$ (about 250 ms)
—	0	1	1	$2^{17}/f_X$ (about 31.3 ms)
—	1	0	1	$2^{15}/f_X$ (about 7.82 ms)
—	1	1	1	$2^{13}/f_X$ (about 1.95 ms)

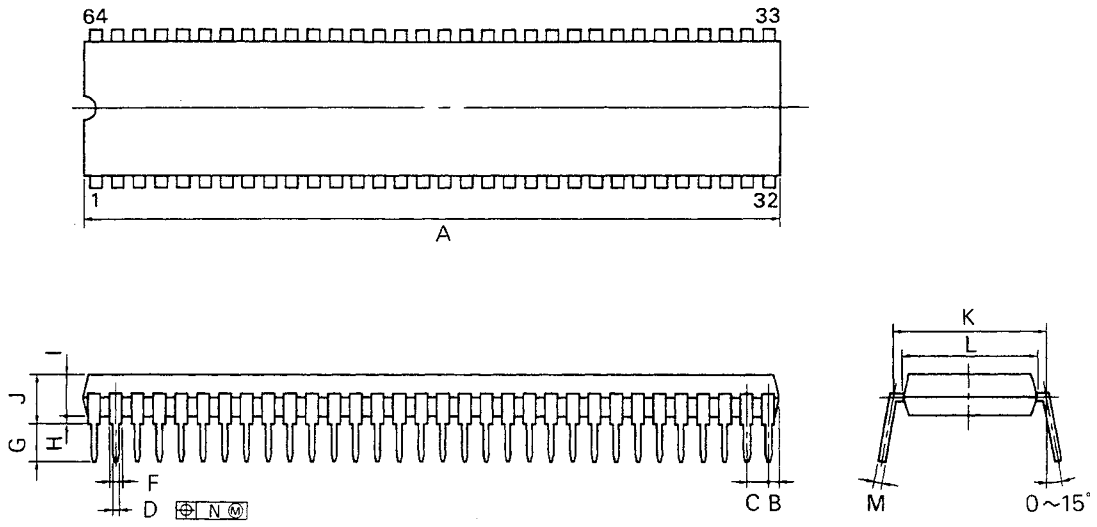
Data retention timing (STOP mode release when $\overline{\text{RESET}}$ is input)



Data retention timing (standby release signal: STOP mode release by interrupt request)



64PIN PLASTIC SHRINK DIP (750 mil)



P64C-70-750A,C

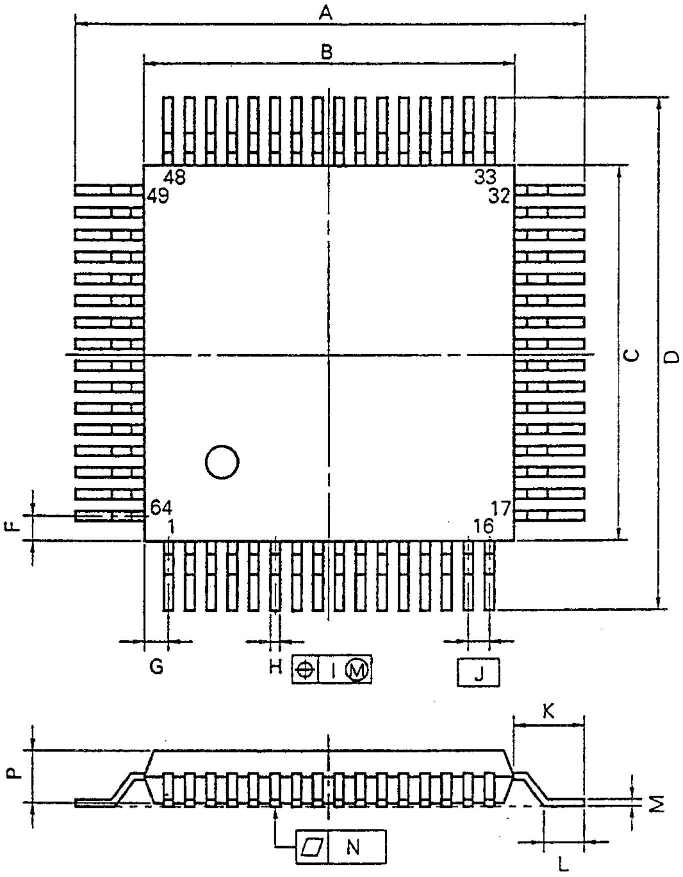
NOTES

1) Each lead centerline is located within 0.17 mm (0.007 inch) of its true position (T.P.) at maximum material condition.

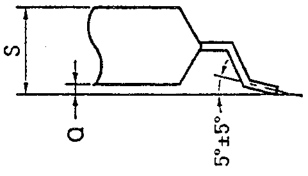
2) Item "K" to center of leads when formed parallel.

ITEM	MILLIMETERS	INCHES
A	58.68 MAX.	2.311 MAX.
B	1.78 MAX.	0.070 MAX.
C	1.778 (T.P.)	0.070 (T.P.)
D	$0.50^{+0.10}$	$0.020^{+0.004}_{-0.005}$
F	0.9 MIN.	0.035 MIN.
G	$3.2^{+0.3}$	$0.126^{\pm 0.012}$
H	0.51 MIN.	0.020 MIN.
I	4.31 MAX.	0.170 MAX.
J	5.08 MAX.	0.200 MAX.
K	19.05 (T.P.)	0.750 (T.P.)
L	17.0	0.669
M	$0.25^{+0.10}_{-0.05}$	$0.010^{+0.004}_{-0.003}$
N	0.17	0.007

64 PIN PLASTIC QFP (□14)



detail of lead end



P64GC-80-AB8-2

NOTE

Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	17.6±0.4	0.693±0.016
B	14.0±0.2	$0.551^{+0.009}_{-0.008}$
C	14.0±0.2	$0.551^{+0.009}_{-0.008}$
D	17.6±0.4	0.693±0.016
F	1.0	0.039
G	1.0	0.039
H	0.35±0.10	$0.014^{+0.004}_{-0.005}$
I	0.15	0.006
J	0.8 (T.P.)	0.031 (T.P.)
K	1.8±0.2	0.071±0.008
L	0.8±0.2	$0.031^{+0.009}_{-0.008}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
N	0.15	0.006
P	2.55	0.100
Q	0.1±0.1	0.004±0.004
S	2.85 MAX.	0.112 MAX.

11. RECOMMENDED CONDITIONS FOR SOLDERING

The following conditions (See table below) must be met when soldering this product.

Please consult with our sales offices in case other soldering process is used, or in case soldering is done under different conditions.

For more details, refer to our document "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (IEI-1207)

Table 11-1 Type of Surface Mount Device

uPD75028GC(A)-XXX-AB8 : 64-pin plastic QFP (14 x 14 mm)

Soldering Process	Soldering Conditions	Symbol
Infrared Ray Reflow	Peak temperature of package surface : 235 °C or lower Reflow time : 30 seconds or less (210°C or higher), Number of reflow processes : 1, Exposure limit* : 7 days (20 hours pre-baking is required at 125°C afterwards)	IR35-207-1
VPS	Peak temperature of package surface : 215°C or lower, Reflow time : 40 seconds or less (200°C or higher), Number of reflow processes : 1, Exposure limit* : 7 days (20 hours pre-baking is required at 125°C afterwards)	VP15-207-1
Wave soldering	Soldering temperature : 260 °C or lower, Flow time : 10 seconds or less, Number of reflow processes : 1, Exposure limit* : 7 days (20 hours pre-backing is required at 125°C afterwards), Temperature of pre-heat : 120°C or lower (Package surface temperature)	WS60-207-1
Partial Heating Method	Pin temperature : 300°C or lower, Time : 3 seconds or less (Per side of the package	—

Note Exposure limit before soldering after dry-pack package is opened. Storage conditions : 25 °C and relative humidity at 65 % or less.

Caution Do not apply more than one soldering method at any one time, except for "Partial heating method."

Table 11-2 Type of through hole device

uPD75028CW(A)-XXX : 64-pin plastic shrink DIP (750 mil)

Soldering Process	Soldering Conditions
Wave soldering (only lead part)	Solder temperature : 260°C or lower, Flow time : 10 seconds or less
Partial Heating Method	Pin temperature : 260°C or lower, Time : 10 seconds or less

Caution Do not jet molten solder on the surface of package.

APPENDIX DEVELOPMENT TOOLS

The following development tools are provided for system development using the uPD75028(A):

Language processor

RA75X relocatable assember	Host machine			Ordering code (product name)
		OS	Distribution media	
	PC-9800 series	MS-DOS^{TM} (Ver. 3.10 Ver. 3.30 Ver. 3.30A)	3.5-inch 2HD	uS5A13RA75X
			5-inch 2HD	uS5A10RA75X
	IBM PC series	PC DOS^{TM} (Ver. 3.1)	5-inch 2D	uS7B11RA75X

PROM write tools

<table>
<tr><td rowspan="3">Hard-ware</td><td>PG-1500</td><td colspan="4">PG-1500 is a PROM programmer which enables you to program single chip microcomputers containing PROM by stand-alone or host machine operation by connecting attached board and option programmer adapter.
If also enables you to program typical PROM devices of 256K bits to 1M bits.</td></tr>
<tr><td>PA-75P036CW</td><td colspan="4">PROM programmer adapter for uPD75P036CW.
Connect the programmer adapter to PG-1500 for use.</td></tr>
<tr><td>PA75P036GC</td><td colspan="4">PROM programmer adapter for uPD75P036GC.
Connect the programmer adapter to PG-1500 for use.</td></tr>
<tr><td rowspan="5">Soft-ware</td><td rowspan="5">PG-1500 controller</td><td colspan="4">PG-1500 and a host machine are connected by serial and parallel interfaces and PG-1500 is controlled on the host machine.</td></tr>
<tr><td>Host machine</td><td>OS</td><td>Distribution media</td><td>Ordering code (product neme)</td></tr>
<tr><td rowspan="2">PC-9800 series</td><td rowspan="2">MS-DOS
(Ver.3.10
Ver. 3.30
Ver. 3.30A)</td><td>3.5-inch 2HD</td><td>uS5A13PG1500</td></tr>
<tr><td>5-inch 2HD</td><td>uS5A10PG1500</td></tr>
<tr><td>IBM PC series</td><td>PC DOS
(Ver. 3.1)</td><td>5-inch 2D</td><td>uS7B11PG1500</td></tr>
</table>

Remarks: Relocatable assembler operation is guaranteed only on the host machine under the operating system listed above.

Debugging tools

Hard-ware	IE-75000-R (Note 1)		IE-75000-R is an in-circuit emulator for the 75X series. To develop the uPD75028(A), use IE-75000-R and an emulation probe in combination. IE-75000-R can be connected to a host machine and PROM programmer for efficient debugging.
		IE-75000-R-EM	Emulation board for IE-75000-R and IE-75001-R. IE-75000-R contains the board. Use the emulation board and IE-75000-R or IE-75001-R in combination for evaluation of uPD75028(A).
	IE-75001-R		IE-75001-R is an in-circuit emulator for the 75X series. To developm the uPD75028(A), use IE-75001-R, emulation board IE-75000-R-EM (Note 2), and an emulation probe in combination. IE-75001-R can be connected to a host machine and PROM programmer for efficient debugging.
	EP-75028CW-R		Emulation probe for uPD75208CW(A) (64-pin plastic shrunk DIP). Connect the emulation probe to IE-75000-R or IE-75001-R and IE-75000-R-EM for use.
	EP-75028GC-R		Emulation probe for uPD75028GC(A) (64-pin plastic QFP). Connect the emulation probe to IE-75000-R or IE-75001-R and IE-75000-R-EM for use. 64-pin conversion socket EV-9200GC-64 is attached to facilitate connection to user system.
		EV-9200GC-64	
Soft-ware	IE control program		IE-75000-R or IE-75001-R and a host machine are connected by RS-232-C and IE-75000-R or IE-75001-R is controlled on the host machine.

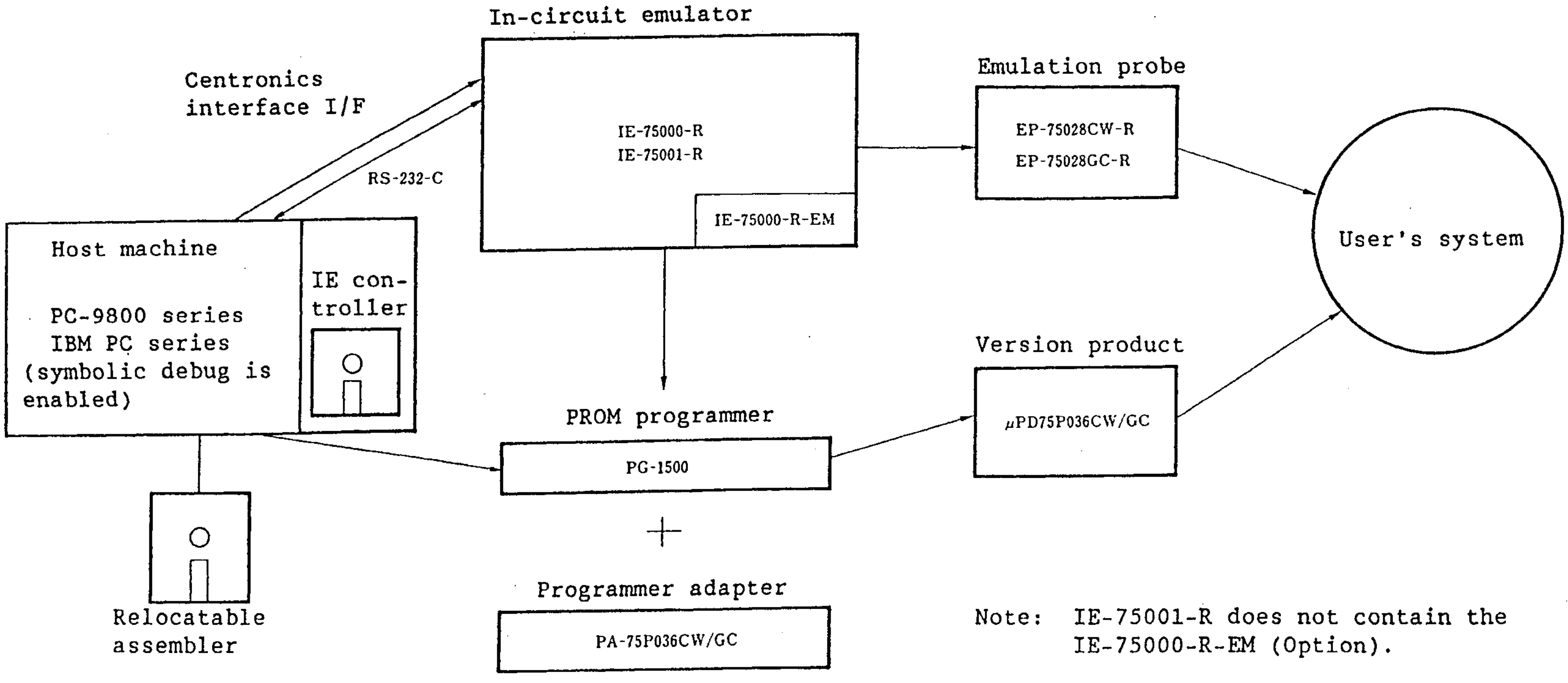
Host machine	OS	Distribution media	Ordering code (product name)
PC-9800 series	MS-DOS (Ver.3.10 to Ver.3.30C)	3.5-inch 2HD	uS5A13IE75X
		5-inch 2HD	uS5A101E75X
IBM PC series	PC DOS (Ver.3.1)	5-inch 2HC	uS7B101E75X

Note 1: Maintenence product

2: IE-75000-R-EM is a option.

Remarks: IE control program operation is gauranteed only on the host machine under the operating system listed above.

Development tool configuration



Note: IE-75001-R does not contain the IE-75000-R-EM (Option).

〔MEMO〕

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.
The devices listed in this document are not suitable for use in aerospace equipment, submarine cables, nuclear reactor control systems and life support systems. If customers intend to use NEC devices for above applications or they intend to use "Standard" quality grade NEC devices for applications not intended by NEC, please contact our sales people in advance.
Application examples recommended by NEC Corporation

Standard:	Computer, Office equipment, Communication equipment, Test and Measurement equipment, Machine tools, Industrial robots, Audio and Visual equipment, Other consumer products, etc.
Special:	Automotive and Transportation equipment, Traffic control systems, Antidisaster systems, Anticrime systems, etc.

M4 92.6