DATA SHEET

NEC

MOS INTEGRATED CIRCUIT

μPD78310A(A), 78312A(A)

16/8 BIT SINGLE-CHIP MICROCOMPUTER

The μPD78312A(A) is a CMOS 16/8-bit microcomputer. It contains a high-performance 16-bit CPU, enabling highly advanced internal arithmetic/logical operations. Up to 56K bytes of external memory can be added.

The μPD78310A(A) is a ROM-less version of the μPD78312A(A), and can directly access external memory of up to 64K bytes. The μPD78310A(A) and μPD78312A(A) are members of the 78K/III series.

With the μPD78312A(A), the μPD78P312A, which contains a one-time PROM or EPROM, is also available for system evaluation.

The μPD78310A(A) and μPD78312A(A) are enhanced versions of the μPD78310(A) and μPD78312(A). The following user's manual describes the details of functions. Be sure to read it before design.

μPD78312A User's Manual: IEM-5086

FEATURES

- More reliable than the μPD78310A and μPD78312A
- Upward-compatible to the μPD78312(A) with enhanced functions including:
  - 4-time count mode
  - 16-bit data transfer instructions
- Various built-in peripheral hardware suited for machine system control:
  - Multi-function pulse I/O unit
  - 8-bit general-purpose serial interface
  - High-precision 8-bit A/D converter
  - Pseudo-static RAM refresh function
- Fast instruction execution by the instruction prefetch function:
  - Instruction cycle: 500 ns/12 MHz
- Ninety-six types of basic instructions suitable for control applications:
  - Multiply/divide instructions
    (Instructions such as 16 bits × 16 bits, 32 bits ÷ 16 bits)
- Advanced, built-in interrupt controller:
  - Three types of interrupt processing
    (Vectored interrupt, macro service, context switching)

Some references made to the μPD78312A(A) in this material pertain to functions common to the μPD78310A(A) and μPD78312A(A).

Document No. IC-8837O

D
P

ORDERING INFORMATION

Part number	Package	Internal ROM
μPD78310ACW(A)	64-pin plastic shrink DIP (750 mil)	None
μPD78310AGQ(A)-36	64-pin plastic QUIP	None
μPD78310AGF(A)-3BE	64-pin plastic QFP (14 × 20 mm)	None
μPD78310AL(A)	68-pin plastic QFJ (□950 mil)	None
μPD78312ACW(A)-xxx	64-pin plastic shrink DIP (750 mil)	Included
μPD78312AGQ(A)-xxx-36	64-pin plastic QUIP	Included
μPD78312AGF(A)-xxx-3BE	64-pin plastic QFP (14 × 20 mm)	Included
μPD78312AL(A)-xxx	68-pin plastic QFJ (□950 mil)	Included

Remark xxx: ROM code number

QUALITY GRADE

Special

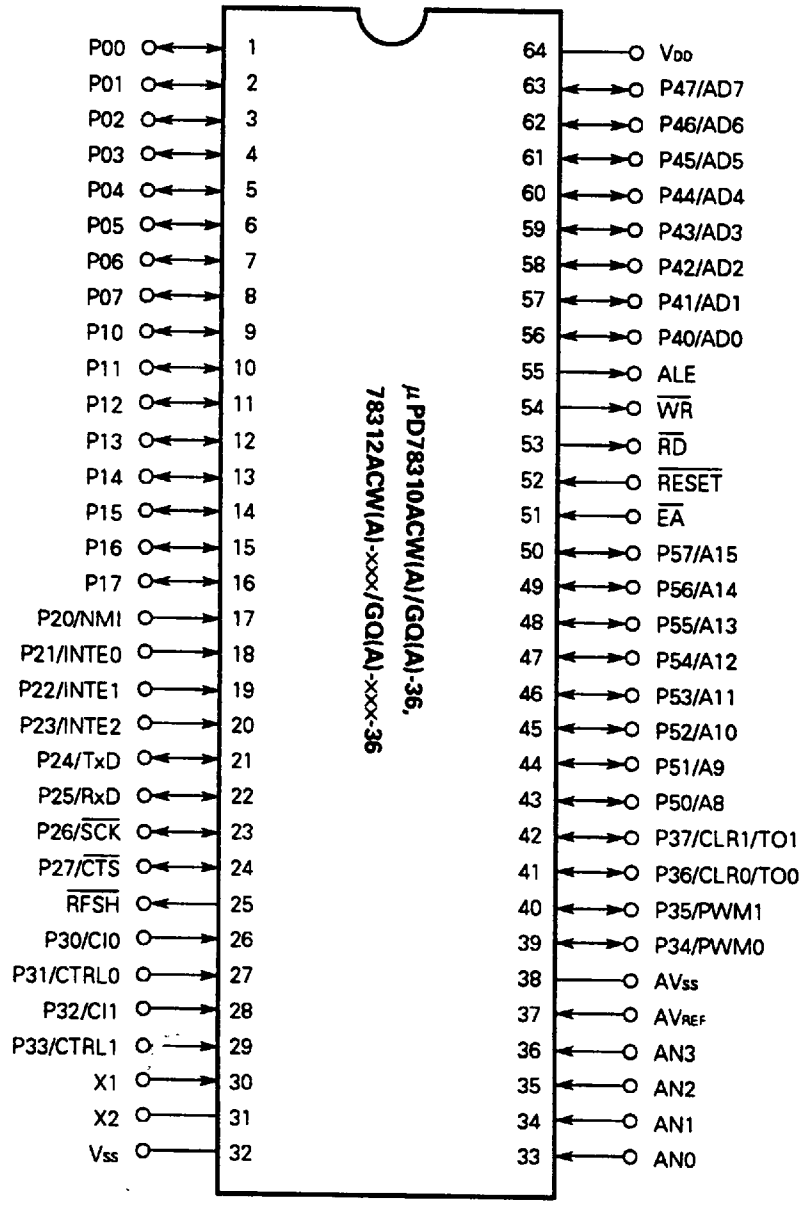
Please refer to *Quality Grades on NEC Semiconductor Devices* (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

DIFFERENCES BETWEEN THE μPD78310A(A) OR μPD78312A(A) AND THE μPD78310A OR μPD78312A

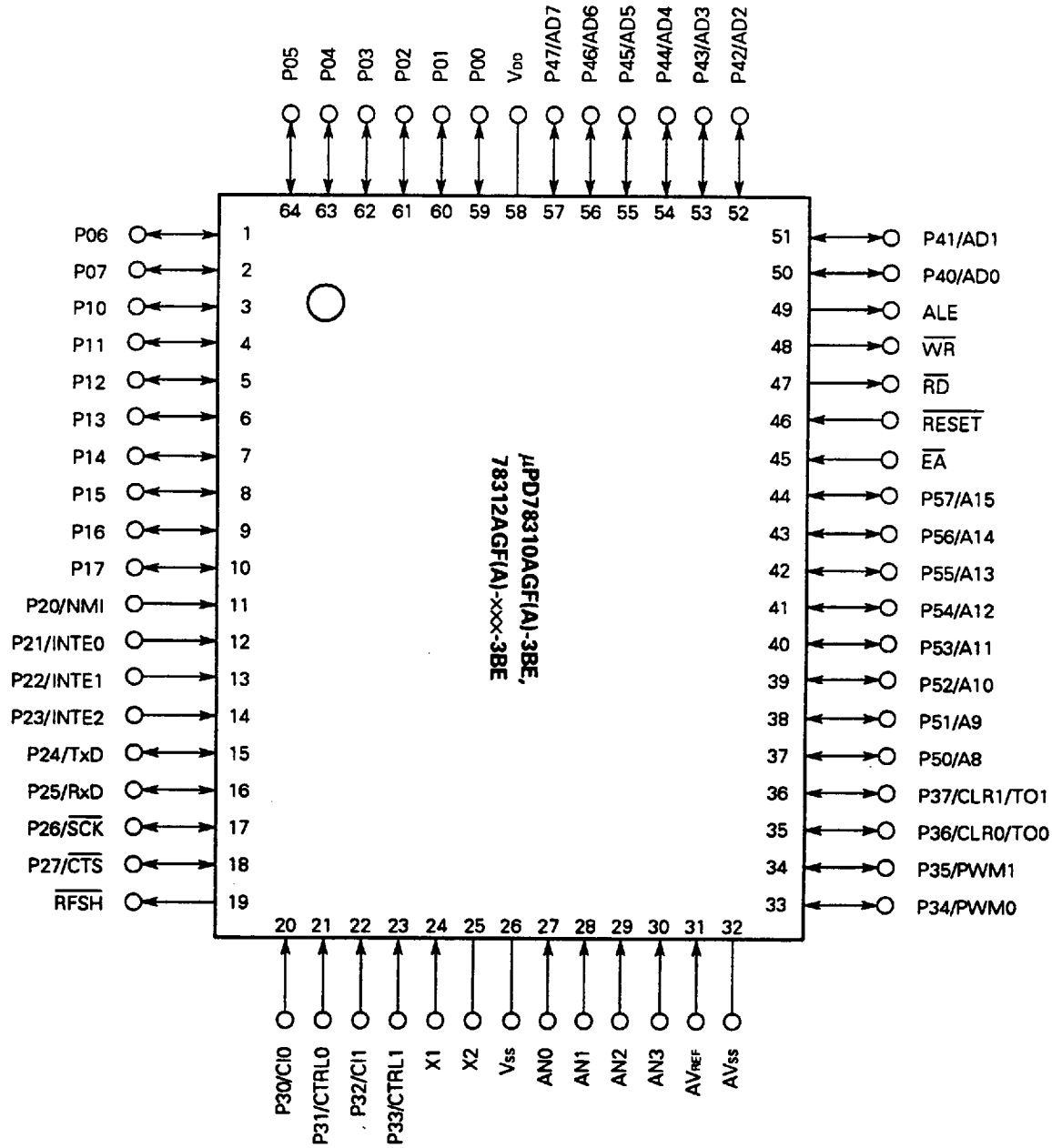
Item \ Product		μPD78310A(A) and μPD78312A(A)	μPD78310A and μPD78312A
Quality grade		Special	Standard
Electrical characteristics	Absolute maximum ratings	Differs from each other in the operating temperature	
	Recommended operating condition	Differs from each other in T_a (ambient temperature)	
	DC characteristics	Differs from each other in the data retention current	

PIN CONFIGURATION (TOP VIEW)

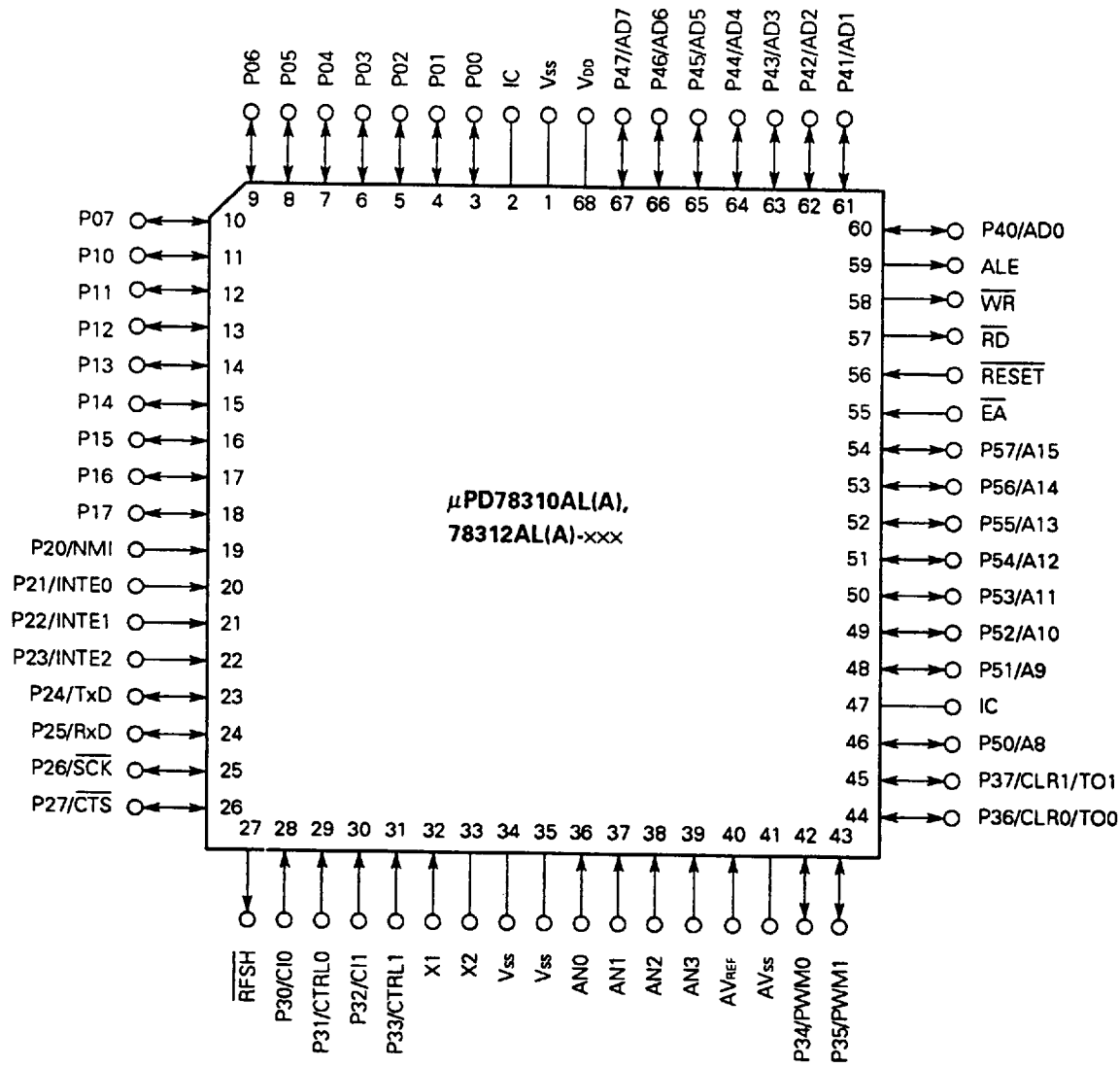
64-pin plastic shrink DIP (750 mil) and 64-pin plastic QUIP



64-pin plastic QFP (14 × 20 mm)



68-pin plastic QFJ (□950 mil)



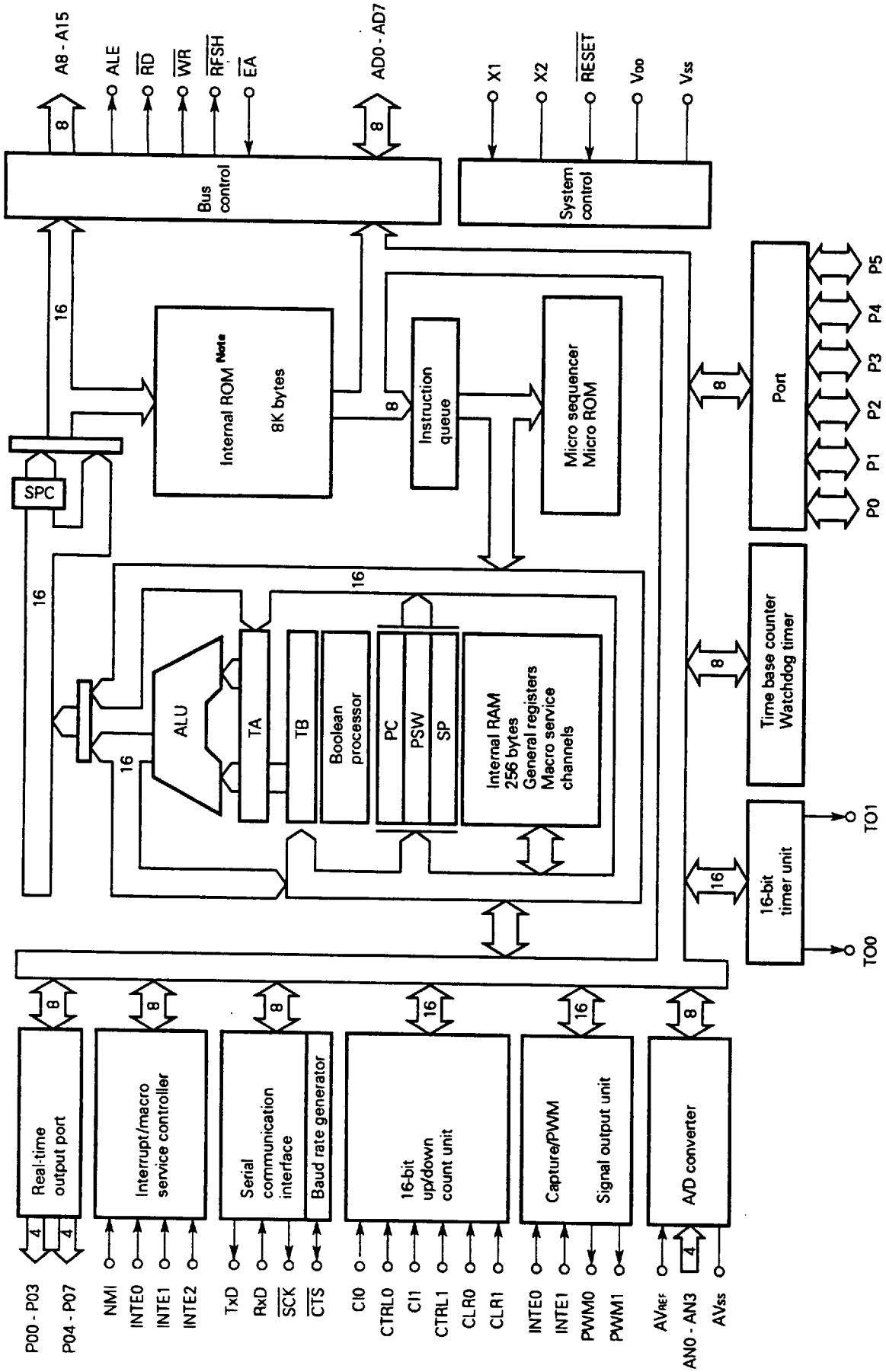
P00 - P07 : Port 0
P10 - P17 : Port 1
P20 - P27 : Port 2
P30 - P37 : Port 3
P40 - P47 : Port 4
P50 - P57 : Port 5
AD0 - AD7 : Address/data
A8 - A15 : Address
$\overline{RD}$: Read strobe
$\overline{WR}$: Write strobe
ALE : Address Latch enable
$\overline{EA}$: External access
$\overline{RFSH}$: Refresh
X1, X2 : Crystal
$\overline{RESET}$: Reset

CI0, CI1 : Count pulse input
CTRL0, CTRL1 : Control pulse input
CLR0, CLR1 : Timer clear input
PWM0, PWM1 : Pulse width modulation output
TO0, TO1 : Timer output
NMI : Nonmaskable interrupt
INTE0 - INTE2 : Interrupt from externals
AN0 - AN3 : Analog input
AV_{REF} : Reference voltage
AVss : Analog Vss
RxD : Receive serial data
TxD : Transfer serial data
$\overline{SCK}$: Serial clock
$\overline{CTS}$: Clear to send
IC : Internally connected

FUNCTION OVERVIEW

Number of basic instructions	96
Minimum instruction execution time	500 ns (when operating at 12 MHz)
Built-in memory	• ROM: 8192 × 8 (μPD78312A(A) only) • RAM: 256 × 8
Memory space	64K bytes
General register	8 bits × 16 × 8 banks (memory mapping)
I/O line	• Input ports : 8 • I/O ports : 24 (μPD78310A(A)), 40 (μPD78312A(A)) • Output port : 1 • Analog inputs: 4
Multi-function pulse I/O unit	• 16-bit presettable up/down counter × 2 • 16-bit interval timer × 2 • 16-bit free running counter capture function × 2 • High-precision PWM output × 2 • Real-time output port: 4 bits × 2
Serial communication interface	• 8 bits (Send and receive in full duplex mode) • Special baud rate generator included • Asynchronous mode or I/O interface mode
A/D converter	• 8-bit precision (4 analog inputs)
Interrupt function	• 17 sources (external: 4, internal: 13) • 8 priority levels can be programmed. • Three types of interrupt processing modes can be selected. (Vectored interrupt function, macro service function, and context switching function)
Stand-by	STOP mode/HALT mode
Instruction set	16-bit arithmetic/logical instructions, multiply/divide instructions, bit manipulation instructions, BCD correction instructions, user stack manipulation instructions, and string instructions
Others	• Watchdog timer included • 20-bit time base counter included • Pseudo-static RAM refresh function
Package	• 64-pin plastic shrink DIP (750 mil) • 64-pin plastic QUIP • 64-pin plastic QFP (14 × 20 mm) • 68-pin plastic QFJ (□ 950 mil)

BLOCK DIAGRAM OF THE μPD78310A(A) AND μPD78312A(A)



Note No internal ROM is provided in the μPD78310A(A).

CONTENTS

1. PIN FUNCTIONS

1.1 PORT PINS

Pin name	I/O	Dual-function	Function
P00 - P07	I/O or real-time output	-	(Port 0) • 8-bit I/O port. Inputs and outputs can be specified bit by bit. • This port also functions as a 2-channel, 4-bit real-time output port.
P10 - P17	I/O	-	(Port 1) 8-bit I/O port. Inputs and outputs can be specified bit by bit.
P20	Input	NMI	(Port 2) P20 - P23 function as an input port. P24 - P27 function as an I/O port. Inputs and outputs can be specified bit by bit.
P21		INTE0	
P22		INTE1	
P23		INTE2	
P24	I/O	TxD	
P25		RxD	
P26		$\overline{SCK}$	
P27		$\overline{CTS}$	
P30	Input	CI0	(Port 3) P30 - P33 function as an input port. P34 - P37 function as an I/O port. Inputs and outputs can be specified bit by bit.
P31		CTRL0	
P32		CI1	
P33		CTRL1	
P34	I/O	PWM0	
P35		PWM1	
P36		TO0/CLR0	
P37		TO1/CLR1	
P40 - P47	I/O	AD0 - AD7	(Port 4) 8-bit I/O port. Inputs and outputs can be specified in units of 8 bits.
P50 - P57	I/O	A8 - A15	(Port 5) 8-bit I/O port. Inputs and outputs can be specified bit by bit.

1.2 NON-PORT PINS

Pin name	I/O	Dual-function	Function
NMI	Input	P20	Nonmaskable interrupt request input pin. The rising or falling edge can be specified as the detective edge through the mode register.
INTE0	Input	P21	External interrupt request input pin. The detective edge can be selected through the mode register.
INTE1		P22	
INTE2		P23	
TxD	Output	P24	Serial data output pin
RxD	Input	P25	Serial data input pin
$\overline{SCK}$	I/O	P26	Serial clock output pin
$\overline{CTS}$	I/O	P27	• In asynchronous mode, the pin receives an input of a send enable control signal. • In I/O interface mode, the pin functions as a serial clock I/O pin.
CI0	Input	P30	Inputs of external count clocks for the count unit
CI1		P31	
CTRL0	Input	P32	Inputs of a count operation switching control signal for the count unit
CTRL1		P33	
CLR0	Input	P36/TO0	Inputs of a clear signal for the count unit
CLR1		P37/TO1	
PWM0	Output	P34	PWM output pins
PWM1		P35	
TO0	Output	P36/CLR0	Pulse output pins for the timer unit
TO1		P37/CLR1	
AD0 - AD7	I/O	P40 - P47	Multiplexed address/data bus when an external memory is connected
A8 - A15	Output	P50 - P57	Address bus when an external memory is connected
$\overline{WR}$	Output	–	External memory write signal output
$\overline{RD}$	Output	–	External memory read signal output
ALE	Output	–	Output pin of a timing signal for externally latching an address output when external memory is accessed
AN0 - AN3	Input	–	Analog inputs to the A/D converter
AV_{REF}	–	–	A/D converter reference voltage input
AV_{SS}	–	–	Ground of the A/D converter
X1	Input	–	Crystal or ceramic input for system clock generation A clock signal provided externally is applied to the X1 pin, and its inverted signal is applied to the X2 pin.
X2	–	–	
$\overline{RFSH}$	Output	–	Refresh pulse output to an external pseudo-static memory
$\overline{RESET}$	Input	–	System reset input
V_{DD}	–	–	Positive power supply pin
V_{SS}	–	–	Ground pin
$\overline{EA}$	Input	–	Normally, connect the $\overline{EA}$ pin to V_{DD}. When the $\overline{EA}$ pin is connected to V_{SS}, the microcomputer enters the ROM-less mode and accesses the external memory. The level at the $\overline{EA}$ pin cannot be changed during operation.
IC	–	–	Internally connected. Leave this pin open.

1.3 INPUT/OUTPUT CIRCUITS OF EACH PIN

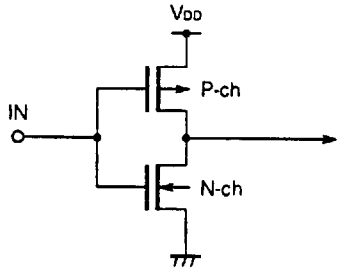
Table 1-1 and Fig. 1-1 show the input and output circuit of each pin in a simplified format.

Table 1-1

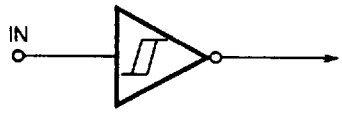
Pin	I/O circuit type	Pin	I/O circuit type
P00 - P07	5	P34/PWM0	5
P10 - P17	5	P35/PWM1	
P20/NMI	2	P36/TO0/CLR0	6
P21/INTE0	1	P37/TO1/CLR1	
P22/INTE1		P40 - P47/AD0 -AD7	5
P23/INTE2		P50 - P57/A8 - A15	5
P24/TxD	5	$\overline{WR}$	3
P25/RxD		$\overline{RD}$	
P26/$\overline{SCK}$		ALE	
P27/$\overline{CTS}$		$\overline{EA}$	1
P30/CI0	1	AN0 - AN3	7
P31/CTRL0		$\overline{RFSH}$	3
P32/CI1		$\overline{RESET}$	2
P33/CTRL1			

Fig. 1-1 Input/Output Circuits of Pins

Type 1

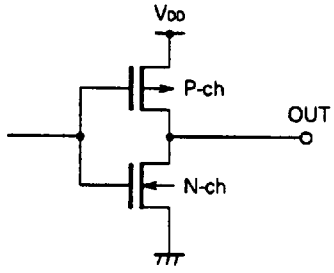


Type 2

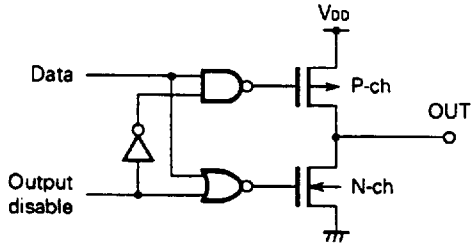


Schmitt trigger input with hysteresis characteristics

Type 3

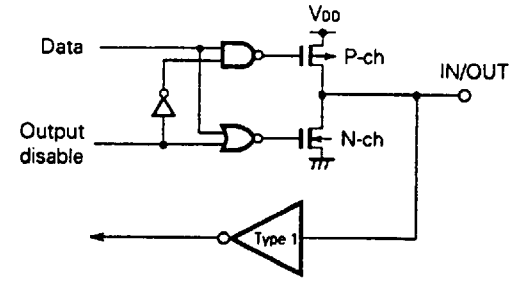


Type 4



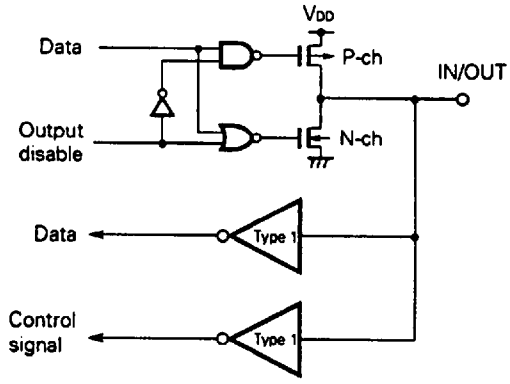
Push-pull output which can output high impedance (Both the positive and negative channels are off.)

Type 5



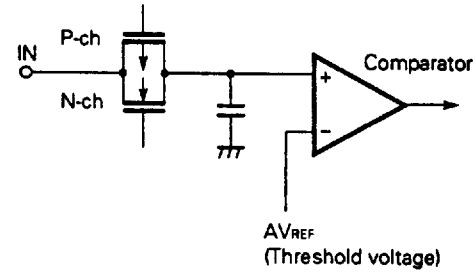
I/O circuit consisting of type 4 push-pull output and type 1 input buffer

Type 6



I/O circuit consisting of type 4 push-pull output and type 1 input buffer

Type 7



(Threshold voltage)

1.4 CONNECTION OF UNUSED PINS

Pin	Recommended connection
P00 - P07 P10 - P17	Input : Connected to V_{DD} via a pull-up resistor Output: Open
P20 - P23	Connected to V_{SS}
P30 - P33	Connected to V_{SS} or V_{DD}
P24 - P27 P34 - P37 P40 - P47 P50 - P57	Input : Connected to V_{DD} via a pull-up resistor Output: Open
$\overline{WR}$ $\overline{RD}$ ALE $\overline{RFSH}$	Open
AN0 - AN3	Connected to V_{SS} or V_{DD}
AV_{REF} AV_{SS}	Connected to V_{SS}

2. DIFFERENCES BETWEEN μPD78310A(A), μPD78312A(A), AND μPD78P312A

The μPD78310A(A) is similar to the μPD78312A(A) except that the μPD78310A(A) does not contain a mask ROM.

The μPD78P312A uses a one-time PROM or EPROM instead of the mask ROM in the μPD78312A(A).

Table 2-1 lists the differences between these products. Features other than noted here are common to all three units.

Table 2-1 Differences between μPD78310A(A), μPD78312A(A), and μPD78P312A

Item		μPD78310A(A)	μPD78312A(A)	μPD78P312A
Program memory		Not included	• Mask ROM • 8192 × 8 bits	• One-time PROM or EPROM • 8192 × 8 bits
Pin function	PROM mode	Not available	Not available	Available
	Ports 4 and 5	Not available (Always function as an address bus and a data bus)	Available	Available
	$\overline{EA}$	Available (Be sure to set the low level.)	Available	
External memory access		64K bytes of external memory can always be accessed regardless of the memory expansion mode register (MM) specification.	External memory can be expanded to 256 bytes, to 4K bytes, to 16K bytes, and to 56K bytes in steps according to the memory expansion mode register (MM) specification.	Same as the μPD78312A(A)
Electrical characteristics	Absolute maximum ratings	Differs from each other in the operating temperature		
	Recommended operating condition	Differs from each other in T_a (ambient temperature)		
	DC characteristics	Differs from each other in the data retention current		
Quality grade		Special		Standard
Package	Without window	• 64-pin plastic shrink DIP (750 mil) • 64-pin plastic QUIP • 64-pin plastic QFP (14 × 20 mm) • 68-pin plastic QFJ (□950 mil)		
	With window	None		• 64-pin ceramic DIP (750 mil) • 64-pin ceramic QUIP

3. DIFFERENCES BETWEEN μPD78312A(A) AND μPD78312(A)

The μPD78312A(A) is an extended version of the μPD78312(A). Table 3-1 lists the differences between the μPD78312A(A) and the μPD78312(A). Similar differences exist between the μPD78310A(A) and the μPD78310(A).

Table 3-1 Differences between the μPD78312A(A) and μPD78312(A)

Item	μPD78310A(A) μPD78312A(A)	μPD78310(A) Note μPD78312(A) Note
Counter unit mode 4 (4-time mode)	Available	Not available
Start of interval timer count operation by external trigger	Available	Not available
16-bit data transfer instructions between memory and a register pair • MOVW rp1, !addr16 instruction • MOVW !addr16, rp1 instruction	Available	Not available

Note Discontinued products

4. CPU ARCHITECTURE

4.1 MEMORY SPACE

The μPD78312A(A) can address a memory of up to 64K bytes. Fig. 4-1 shows the memory space. The memory map varies depending on the $\overline{EA}$ pin status.

Caution The $\overline{EA}$ pin of the μPD78310A(A) must be held at the low level.

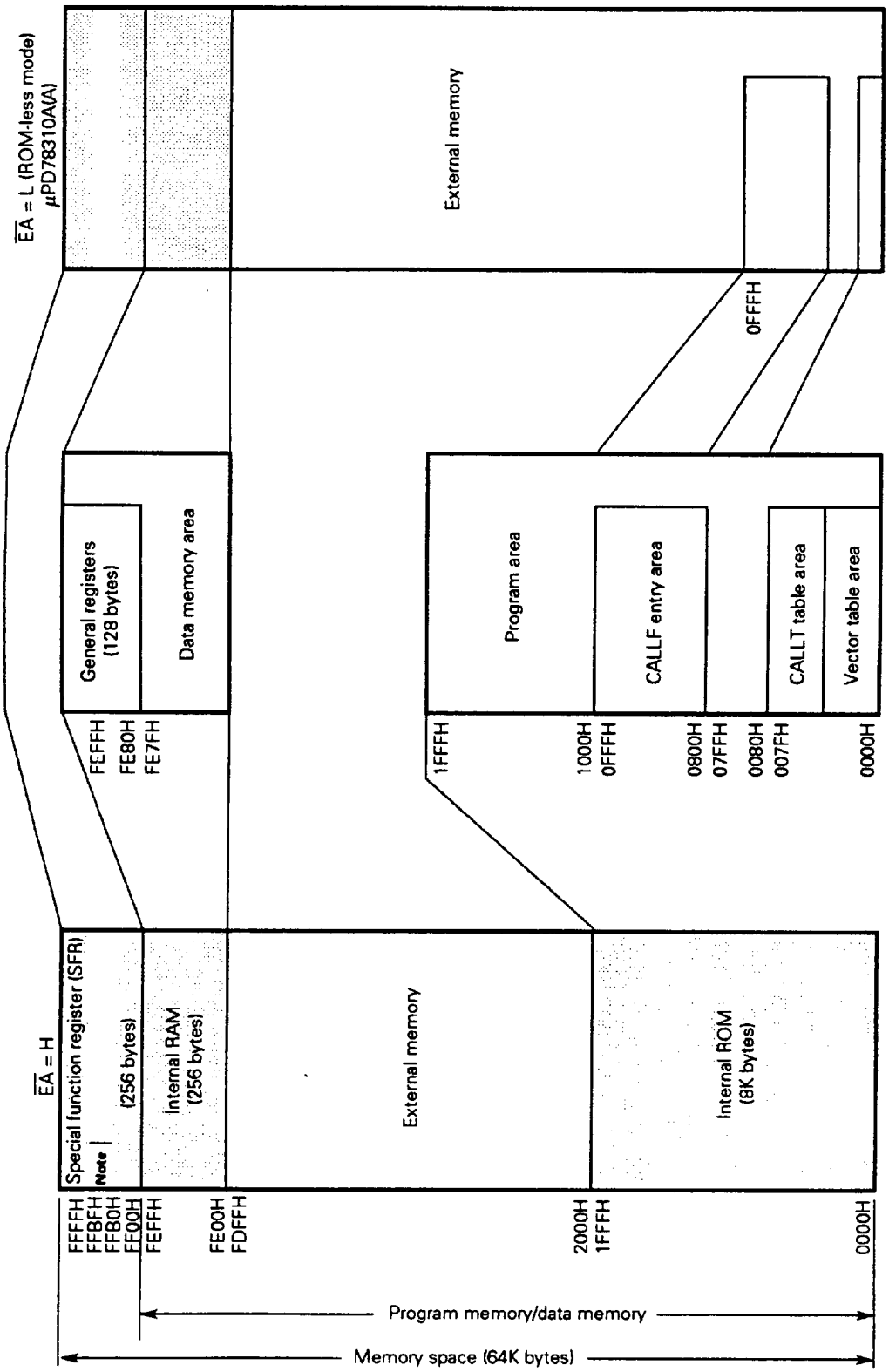


Fig. 4-1 Memory Map

Remarks 1. The shaded areas represent internal memory.
2. The μPD78310A(A) is applicable only when $\overline{EA}$ is low.

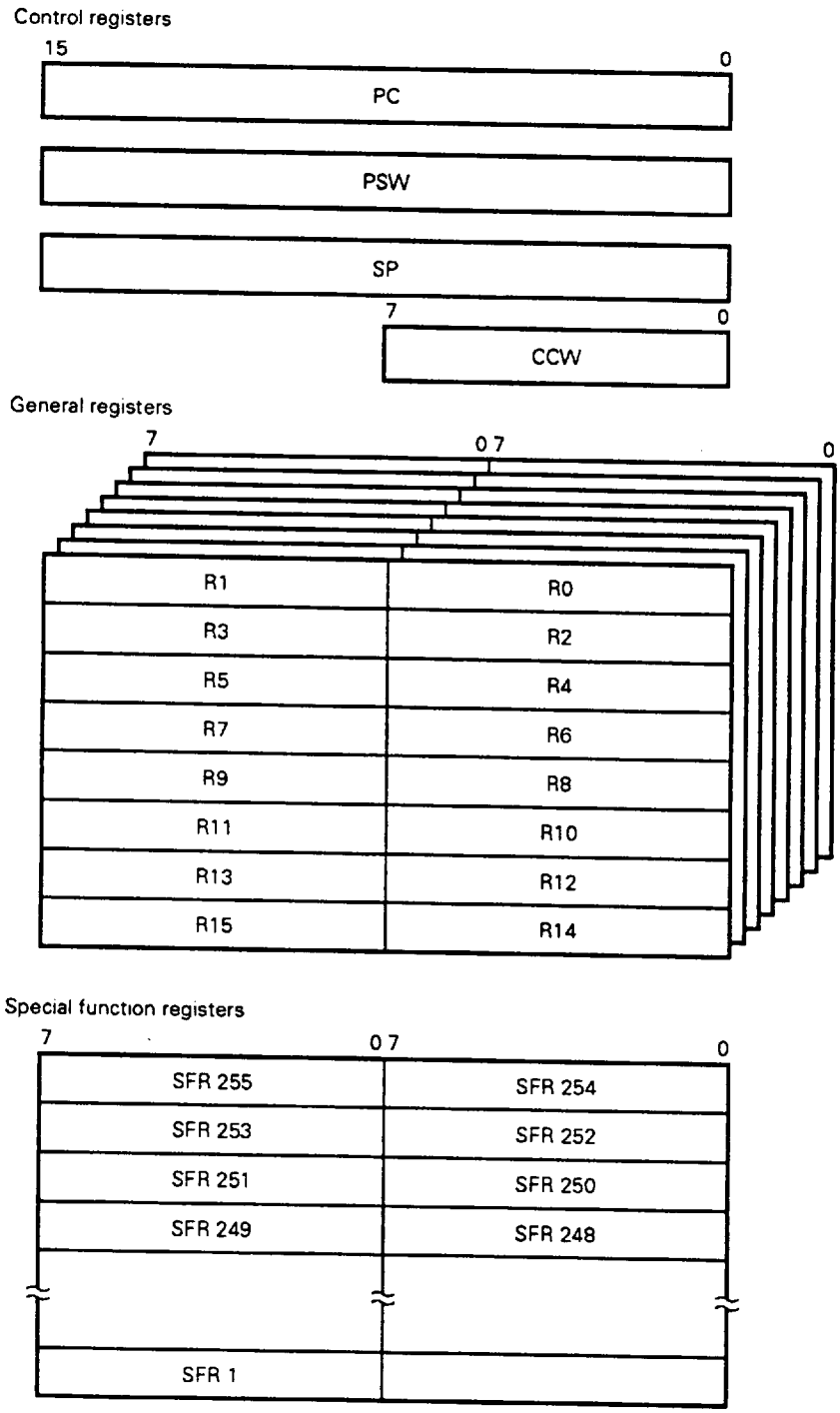
Note Enables access to external memory.

4.2 PROCESSOR REGISTERS

The processor registers include the following:

- Eight banks of general registers, each bank consisting of 16 8-bit registers
- Control registers including one 8-bit register and three 16-bit registers
- Special function registers assigned special functions, such as I/O mode registers for the peripheral hardware

Fig. 4-2 Register Configuration



Remark CCW, a control register, is mapped to the special function register (SFR) area.

4.2.1 Control Registers

The control registers include three 16-bit registers and one 8-bit register which are provided with special functions such as the control of program execution.

(1) Program counter (PC)

The program counter is a 16-bit register for holding address information of the program to be executed next. The program counter is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data or register contents are set in the PC.

(2) Program status word (PSW)

The program status word is a 16-bit register consisting of flags set or reset according to the instruction execution results.

(3) Stack pointer (SP)

The stack-pointer is a 16-bit register which holds the start address of the stack area in memory (in LIFO form).

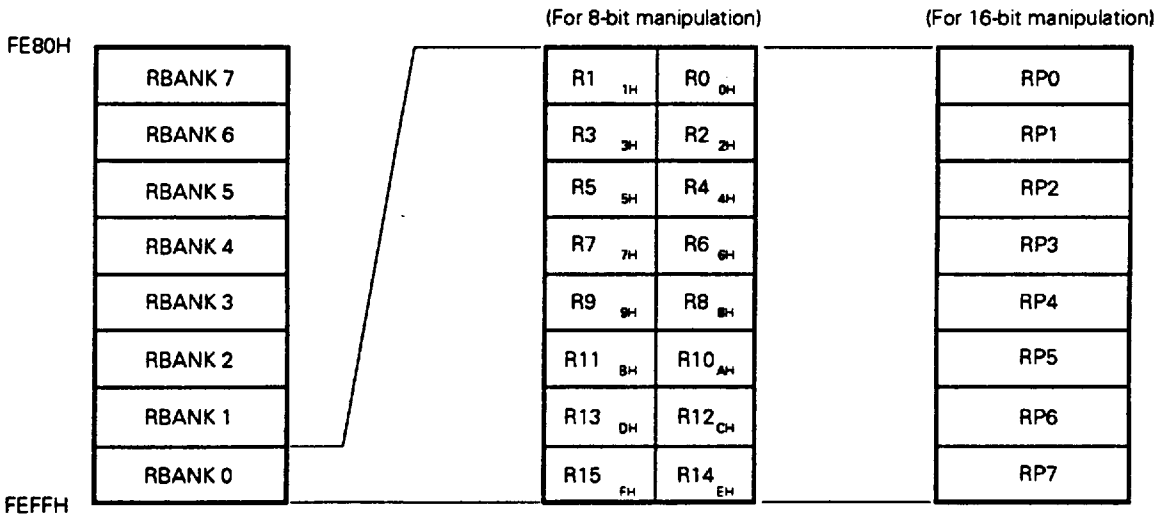
(4) CPU control word (CCW)

The CPU control word is an 8-bit register consisting of flags pertaining to CPU control. By setting the table position flag (TPF) in the CCW by software, the vector table area can be moved from the normal locations at 0000H - 007FH to 8000H - 807FH.

4.2.2 General Registers

The general registers, consisting of 128 bytes in total, are mapped to a certain area (FE80H to FEFFH) of the internal RAM space. They are grouped into eight register banks, each bank consisting of 16 8-bit general registers.

Fig. 4-3 Memory Location of General Registers



A pair of 8-bit registers can function as a 16-bit register pair (RP0 - RP7).

A function name listed in Table 4-1 is assigned to each of the 16 8-bit registers. The X register functions as the low-order bits of a 16-bit accumulator. The A register functions as an 8-bit accumulator or the high-order bits of a 16-bit accumulator. The B and C registers function as counters. The DE, HL, VP, and UP registers, in a pair, function as an address register. The VP register functions as a base register, and the UP register functions as a user stack pointer.

The value of the register set selection flag (RSS) in the PSW changes registers having specific functions, as shown in Table 4-1.

The μPD78312A(A) allows two ways of addressing process data: implied addressing and register addressing. Implied addressing uses a function name which places much importance on the specific function of a register. Register addressing uses an absolute name to create a program which is easy to describe and which performs less data transfer operations, enabling high-speed data processing.

Table 4-1 General Register Configuration

Absolute name	Function name	
	RSS = 0	RSS = 1
R0	X	
R1	A	
R2	C	
R3	B	
R4		X
R5		A
R6		C
R7		B
R8	VP_L	VP_L
R9	VP_H	VP_H
R10	UP_L	UP_L
R11	UP_H	UP_H
R12	E	E
R13	D	D
R14	L	L
R15	H	H

Absolute name	Function name	
	RSS = 0	RSS = 1
RP0	AX	
RP1	BC	
RP2		AX
RP3		BC
RP4	VP	VP
RP5	UP	UP
RP6	DE	DE
RP7	HL	HL

4.2.3 Special Function Registers (SFR)

In contrast to general registers, special function registers (SFRs) are given special functions. The SFRs are assigned to the 256-byte memory address space from FF00H to FFFFH.

Short direct memory addressing is applicable to the 32-byte area from FF00H to FF1FH. This means that SFRs assigned to this area can be processed with shorter word length and less clocks than SFRs assigned to other areas. Assigned to this area are those SFRs frequently accessed, such as a timer compare register, capture register, and ports.

SFRs can be manipulated in the same way as general registers by using arithmetic/logical instructions, transfer instructions, and bit manipulation instructions. They can be manipulated in units of 1 and 8 bits, and some SFRs can also be manipulated in units of 16 bits (see Table 4-2). The following explains how to specify an SFR when it is manipulated in units of 1, 8, and 16 bits:

- 1-bit manipulation
  An abbreviation and bit are coded in an operand (sfr. bit) of a bit manipulation instruction. An address can also be specified.
- 8-bit manipulation
  An abbreviation is coded in an operand (sfr) of an 8-bit manipulation instruction. An address can also be specified.
- 16-bit manipulation
  An abbreviation is coded in an operand (sfrp) of a 16-bit manipulation instruction. SFRs that can be manipulated in units of 16 bits are assigned in consecutive 2-byte locations at even-odd addresses. When the address of such an SFR is specified, the even address must be coded.

Table 4-2 lists SFRs. The items in Table 4-2 mean:

- Abbreviation
  A symbol indicating the address of a built-in SFR. This can be specified in the operand field of an instruction.
- R/W
  Indicates whether data can be read from the SFR and/or written into the SFR.
  R/W: Can be read from and written to.
  R : Can be read from.
  W : Can be written to.
- 16-bit manipulation
  Indicates whether an SFR can be manipulated in units of 16 bits. SFRs applicable to 16-bit manipulation are marked with a circle (O).
- At resetting
  Indicates the status of each register existing when $\overline{\text{RESET}}$ input occurs.

Caution In the area from FF00H to FFFFH, locations which are not assigned any SFR cannot be accessed. Access to such a location may cause malfunction.

Table 4-2 Special Function Registers (SFRs) (1/3)

Address	Special function register (SFR) name	Abbreviation		R/W	16-bit manipulation	At resetting
FF00H	Port 0	P0		R/W	–	Undefined
FF01H	Port 1	P1				
FF02H	Port 2	P2		R/W Note 1	–	
FF03H	Port 3	P3				
FF04H	Port 4	P4		R/W	–	
FF05H	Port 5	P5				
FF08H	Capture/compare register 00	CR00L	CR00		○	
FF09H		CR00H				
FF0AH	Capture/compare register 01	CR01L	CR01		○	
FF0BH		CR01H				
FF0CH	Capture/compare register 10	CR10L	CR10		○	
FF0DH		CR10H				
FF0EH	Capture/compare register 11	CR11L	CR11		○	
FF0FH		CR11H				
FF10H	Capture register 0	CPT0L	CPT0	R/W	○	
FF11H		CPT0H				
FF12H	Capture register 1	CPT1L	CPT1		○	
FF13H		CPT1H				
FF14H	PWM register 0	PWM0L	PWM0		○	
FF15H		PWM0H				
FF16H	PWM register 1	PWM1L	PWM1		○	
FF17H		PWM1H				
FF1CH	Presettable up/down count register 0	UDC0L	UDC0		○	
FF1DH		UDC0H				
FF1EH	Presettable up/down count register 1	UDC1L	UDC1		○	
FF1FH		UDC1H				
FF20H	Port 0 mode register	PM0		R/W	–	FFH
FF21H	Port 1 mode register	PM1				FFH
FF22H	Port 2 mode register	PM2			–	FFH
FF23H	Port 3 mode register	PM3				FFH
FF25H	Port 5 mode register	PM5			–	FFH
FF32H	Port 2 mode control register	PMC2		R/W	–	0FH
FF33H	Port 3 mode control register	PMC3				0FH
FF38H	Real-time output port control register	RTPC			–	08H
FF3AH	Port 0 buffer register Note 2	P0L			–	Undefined
FF3BH		P0H				

Notes 1. Bits 0 to 3 of P2 and P3 can be read only.

2. P0H and P0L are 4-bit buffers. P0H is assigned to the upper 4 bits, and P0L is assigned to the lower 4 bits. P0H and P0L, in a pair, can be manipulated on an 8-bit basis.

Table 4-2 Special Function Registers (SFRs) (2/3)

Address	Special function register (SFR) name	Abbreviation		R/W	16-bit manipulation	At resetting
FF40H	Memory expansion mode register	MM		R/W	–	30H
FF41H	Refresh mode register	RFM				10H
FF42H	Watchdog timer mode register	WDM			–	00H
FF44H	Standby control register	STBC			–	2×H Note
FF46H	Time base mode register	TBM			–	00H
FF48H	External interrupt mode register	INTM			–	00H
FF4AH	In-service priority register	ISPR		R	–	00H
FF4EH	CPU control word	CCW		R/W	–	00H
FF50H	Serial communication mode register	SCM			–	00H
FF52H	Serial communication control register	SCC			–	00H
FF53H	Baud rate generator	BRG				00H
FF56H	Serial communication receive buffer	RxB		R	–	Undefined
FF57H	Serial communication send buffer	TxB		W		
FF60H	Free-running counter control register	FRCC		R/W	–	00H
FF64H	Capture mode register	CPTM			–	00H
FF66H	PWM mode register	PWMM			–	00H
FF68H	A/D converter mode register	ADM			–	00H
FF6AH	A/D conversion result register	ADCR		R	–	Undefined
FF70H	Count unit input mode register	CUIM		R/W	–	00H
FF72H	Up/down counter control register 0	UDCC0			–	00H
FF74H	Capture/compare register control register	CRC			–	00H
FF7AH	Up/down counter control register 1	UDCC1			–	00H
FF80H	Timer control register 0	TMC0		R/W	–	00H
FF82H	Timer control register 1	TMC1			–	00H
FF88H	Timer register 0	TM0L	TM0		○	Undefined
FF89H		TM0H				
FF8AH	Modulo/timer register 0	MD0L	MD0		○	
FF8BH		MD0H				
FF8CH	Timer register 1	TM1L	TM1		○	
FF8DH		TM1H				
FF8EH	Modulo/timer register 1	MD1L	MD1		○	
FF8FH		MD1H				

Note Bit 3 is not affected by $\overline{\text{RESET}}$ input, thus, the lower 4 bits show either 0 or 8.

Table 4-2 Special Function Registers (SFRs) (3/3)

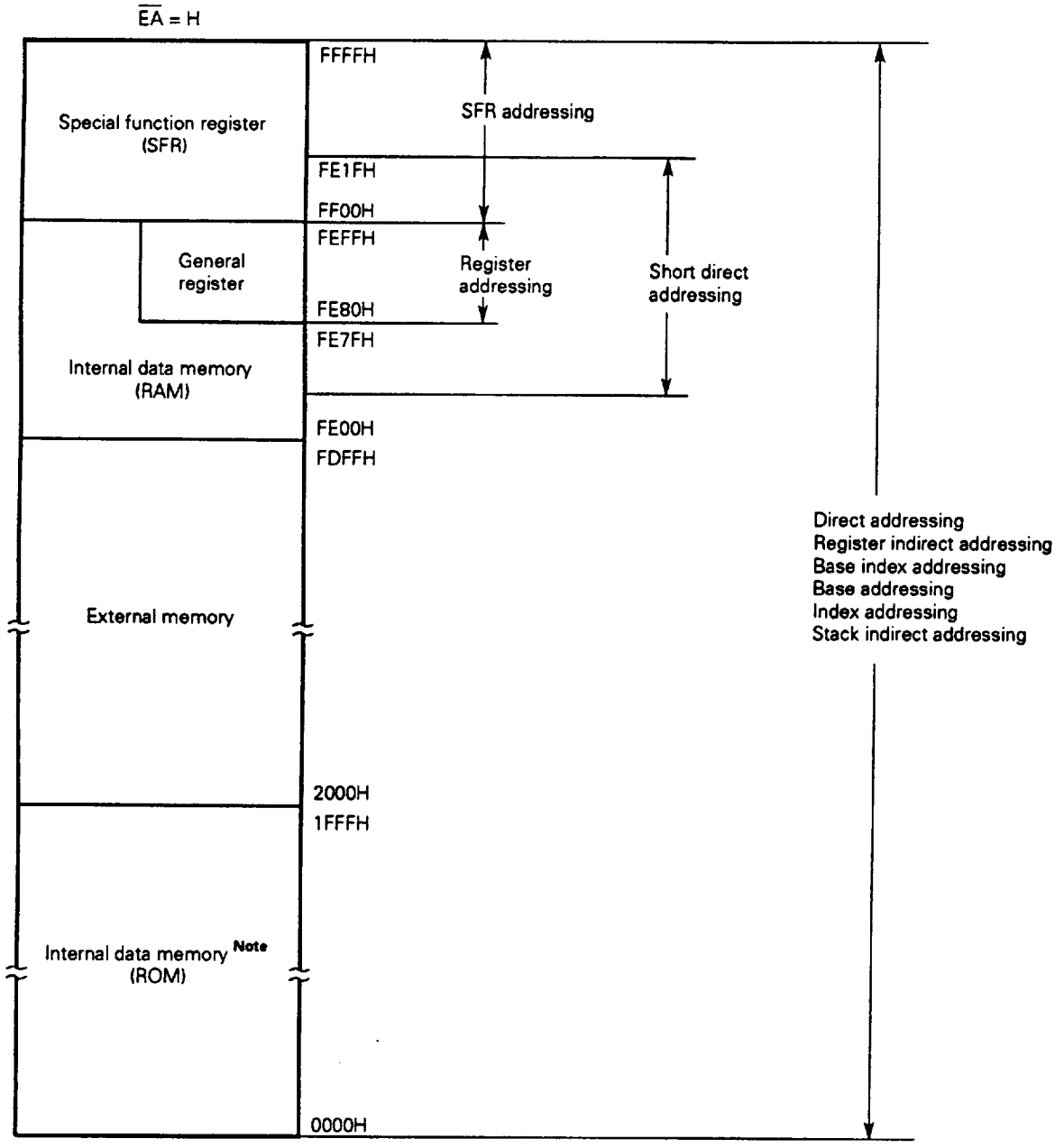
Address	Special function register (SFR) name	Abbreviation	R/W	16-bit manipulation	At resetting
FFB0H to FFBFH	External access area [Note]			–	
FFC0H	Count unit 0 interrupt request control register 0	CRIC00	R/W	–	47H
FFC1H	Count unit 0 macro service control register 0	CRMS00			Undefined
FFC2H	Count unit 0 interrupt request control register 1	CRIC01		–	47H
FFC4H	Count unit 1 interrupt request control register 0	CRIC10		–	47H
FFC5H	Count unit 1 macro service control register 0	CRMS10			Undefined
FFC6H	Count unit 1 interrupt request control register 1	CRIC11		–	47H
FFC8H	External interrupt pin interrupt request control register 0	EXIC0		–	47H
FFC9H	External interrupt pin macro service control register 0	EXMS0			Undefined
FFCAH	External interrupt pin interrupt request control register 1	EXIC1		–	47H
FFCBH	External interrupt pin macro service control register 1	EXMS1			Undefined
FFCCH	External interrupt pin interrupt request control register 2	EXIC2		–	47H
FFCDH	External interrupt pin macro service control register 2	EXMS2			Undefined
FFCEH	Timer unit interrupt request control register 0	TMIC0		–	47H
FFCFH	Timer unit macro service control register 0	TMMS0			Undefined
FFD0H	Timer unit interrupt request control register 1	TMIC1	R/W	–	47H
FFD1H	Timer unit macro service control register 1	TMMS1			Undefined
FFD2H	Timer unit interrupt request control register 2	TMIC2		–	47H
FFD3H	Timer unit macro service control register 2	TMMS2			Undefined
FFDAH	Serial communication receive error interrupt request control register	SEIC		–	47H
FFDCH	Serial communication receive completion interrupt request control register	SRIC		–	47H
FFDDH	Serial communication receive completion macro service control register	SRMS			Undefined
FFDEH	Serial communication send completion interrupt request control register	STIC		–	47H
FFDFH	Serial communication send completion macro service control register	STMS			Undefined
FFE0H	A/D converter interrupt request control register	ADIC	R/W	–	47H
FFE1H	A/D converter macro service control register	ADMS			Undefined
FFE2H	Timer base counter interrupt request control register	TBIC		–	47H

Note This space enables access to external memory by SFR addressing.

4.3 DATA MEMORY ADDRESSING

Fig. 4-4 shows how the μPD78312A(A) memory space is addressed when it is accessed as data memory.

Fig. 4-4 Memory Map of Data Memory and its Addressing (μPD78312A(A))



Note If $\overline{EA}$ is low, or when the μPD78310A(A) is used, external memory is assigned.

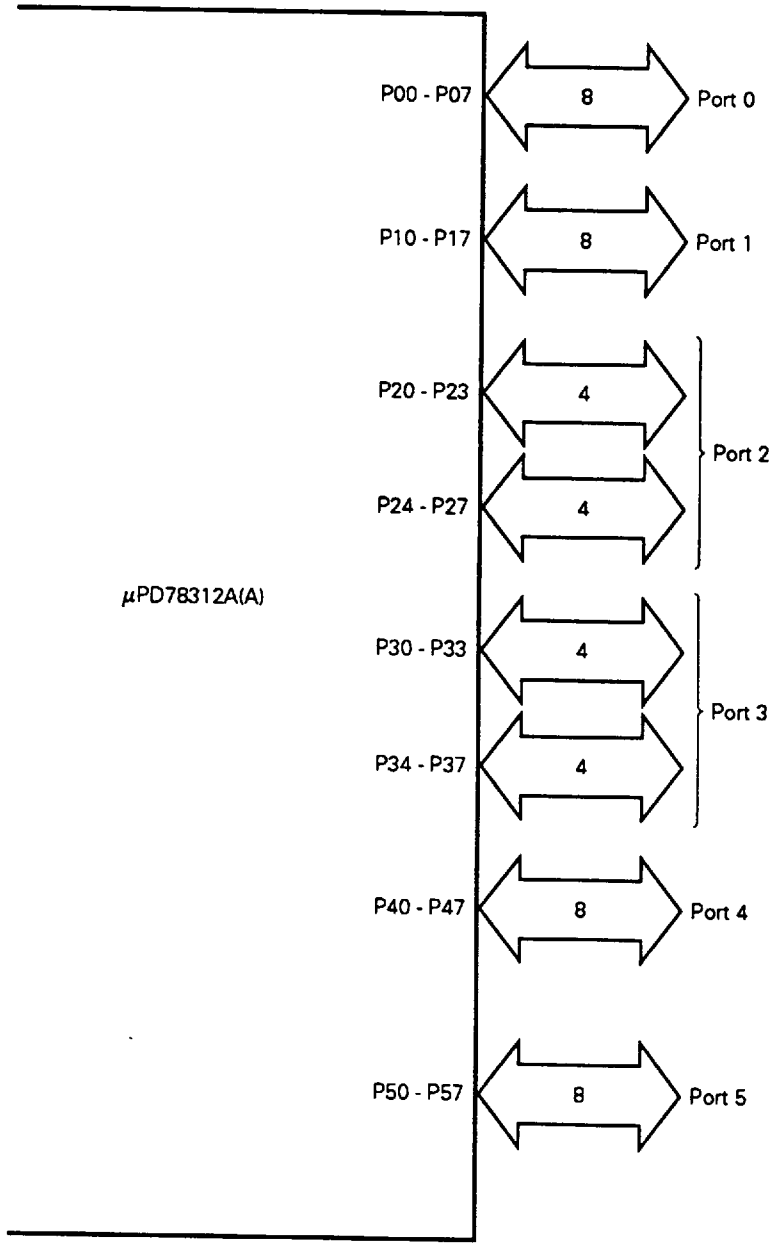
Caution Do not set the stack pointer in the SFR area or ROM area.

5. PERIPHERAL HARDWARE

5.1 PORT

The μPD78312A(A) has ports as shown in Fig. 5-1.
Table 5-1 lists the functions and features of the ports.

Fig. 5-1 Port Configuration



Caution In the μPD78310A(A), port 4 functions as a multiplexed bus for the lower address part and data. Port 5 functions as an address bus for the higher address part.

Table 5-1 Functions and Features of Ports

Port	Function	Operation features	Remarks
Port 0	8-bit I/O	Can be specified as input or output bit by bit. Can be specified as a real-time output port in units of 4 bits.	See Section 5.2.5 for the real-time output port function.
Port 1	8-bit I/O	Can be specified as input or output bit by bit.	–
Port 2	8-bit I/O (Input only on P20 - P23)	Can be specified as a port pin or control pin bit by bit. P24 - P27 can be specified as input or output bit by bit.	Shares pins with NMI, INTE0 - 2, TxD, RxD, $\overline{SCK}$, and $\overline{CTS}$.
Port 3	8-bit I/O (Input only on P30 - P33)	Can be specified as a port pin or control pin bit by bit. P34 - P37 can be specified as input or output bit by bit.	Shares pins with CI0 - 1, CTRL0 - 1, PWM0 - 1, TO0 - 1, and CLR0 - 1
Port 4	8-bit I/O	Can be specified as input or output in units of 8 bits. In the external memory expansion mode, this port functions as a multiplexed address/data bus (AD0 - AD7).	In the μPD78310A(A), this port always functions as a multiplexed address/data bus.
Port 5	8-bit I/O	Can be specified as input or output bit by bit. In the external memory expansion mode, this port functions as an address bus (A8 - A15). Pins not used for the address bus can be used as ports.	In the μPD78310A(A), this port always functions as an address bus.

5.2 PULSE I/O UNIT

The pulse I/O unit consists mainly of the following five blocks:

- Count unit
- Capture unit
- PWM unit
- Timer unit
- Real-time output port

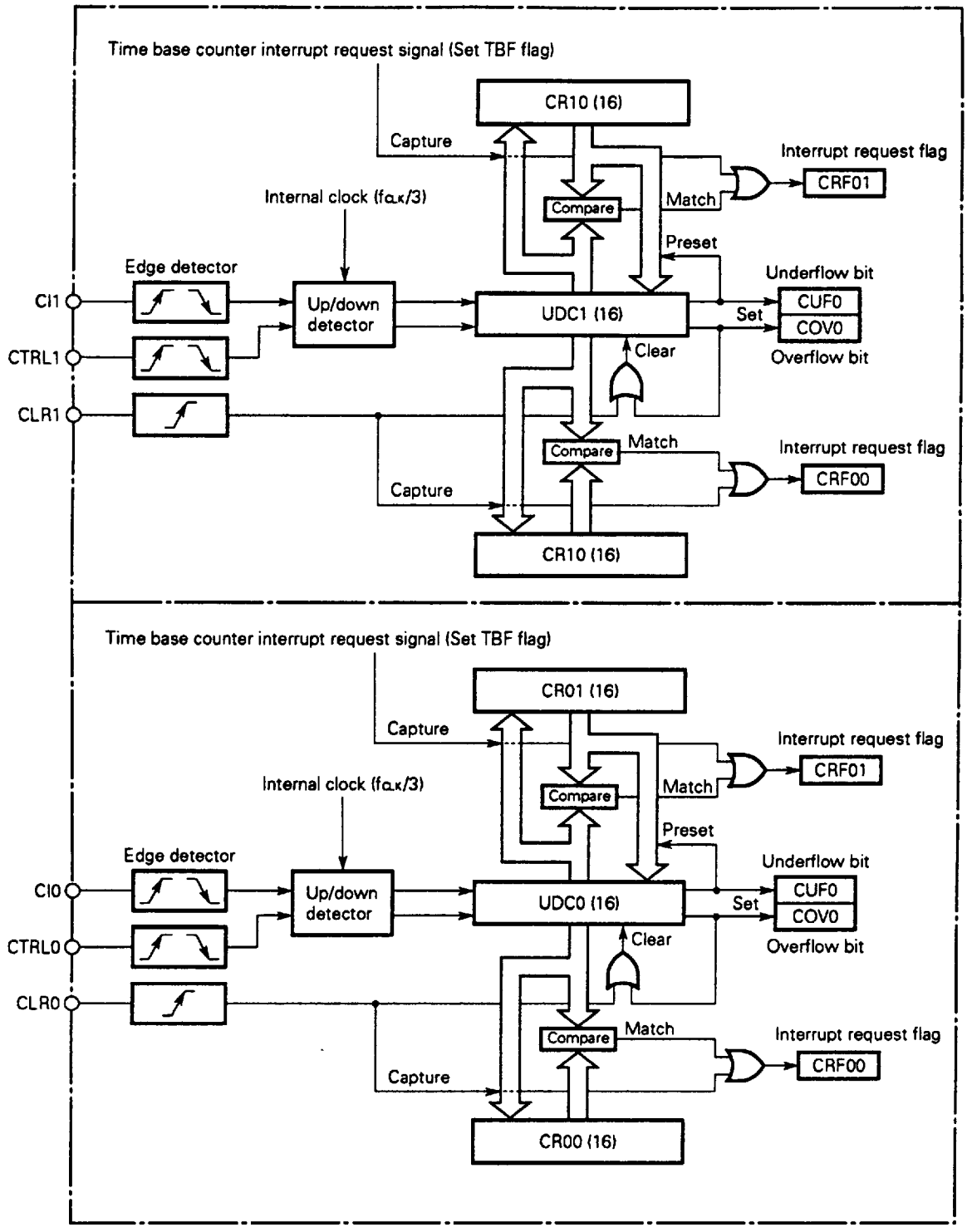
5.2.1 Count Unit

The count unit includes 16-bit presettable up/down counters (UDC0 and UDC1) as its main function. Fig. 5-2 shows the configuration of the count unit. As peripheral registers for the UDC0 and UDC1, capture/compare registers (CR00, CR01, CR10, and CR11) are provided, enabling a wide range of pulse input control.

As UDC0 and UDC1 can automatically determine up/down count operation, that they are useful for phase determination in servo motor control.

The μPD78312A(A) includes the count unit shown in Fig. 5-2.

Fig. 5-2 Block Diagram of the Count Unit



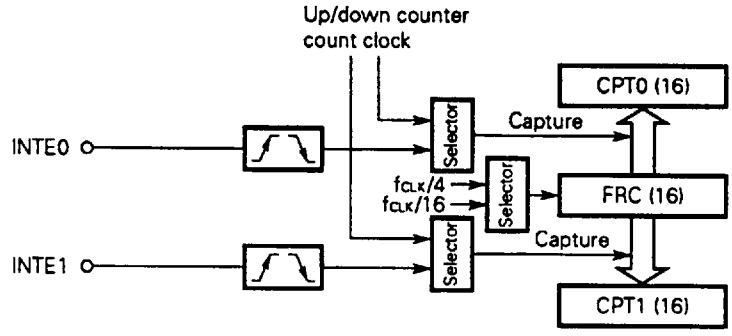
5.2.2 Capture Unit

The capture unit receives and holds a count value of the free-running counter (FRC). The FRC shares hardware with the time base counter. Fig. 5-3 shows the block diagram of the capture unit.

A capture register (CPT0 or CPT1) is triggered by a count clock input from the count unit or an effective edge input on the INTE0 (INTE1) pin, and receives and holds a count value of the FRC.

An input pulse cycle can be detected by performing the difference operation for the captured value.

Fig. 5-3 Block Diagram of the Capture Unit



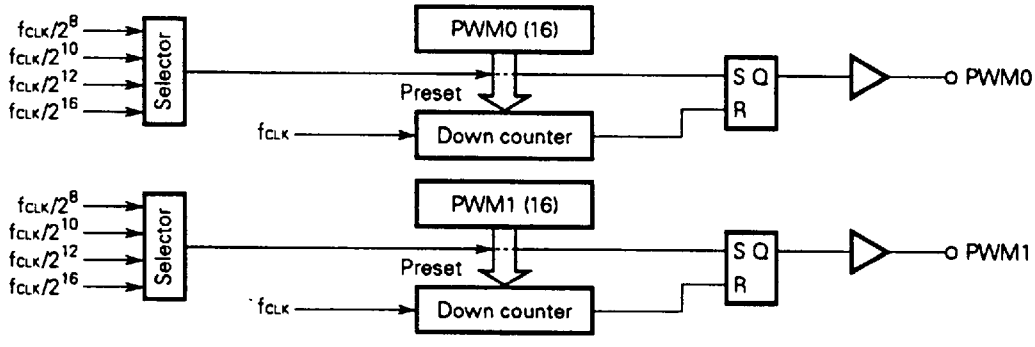
Remark f_{CLK}: Internal system clock frequency

5.2.3 PWM Unit

(1) Configuration of the PWM unit

The PWM unit has a PWM output function. Fig. 5-4 shows the configuration.

Fig. 5-4 Configuration of the PWM Unit



Remark f_{CLK}: Internal system clock frequency

One of four levels of PWM output precision can be selected: 8 bits, 10 bits, 12 bits, or 16 bits of precision. Table 5-2 lists PWM output frequencies.

Table 5-2 Setting of PWM Period

PWM precision	PWM output frequency	PWM period
8-bit PWM output	23.4 kHz	42 μs
10-bit PWM output	5.9 kHz	171 μs
12-bit PWM output	1.5 kHz	683 μs
16-bit PWM output	91.6 Hz	10.9 ms

Remark f_{CLK} = 6 MHz

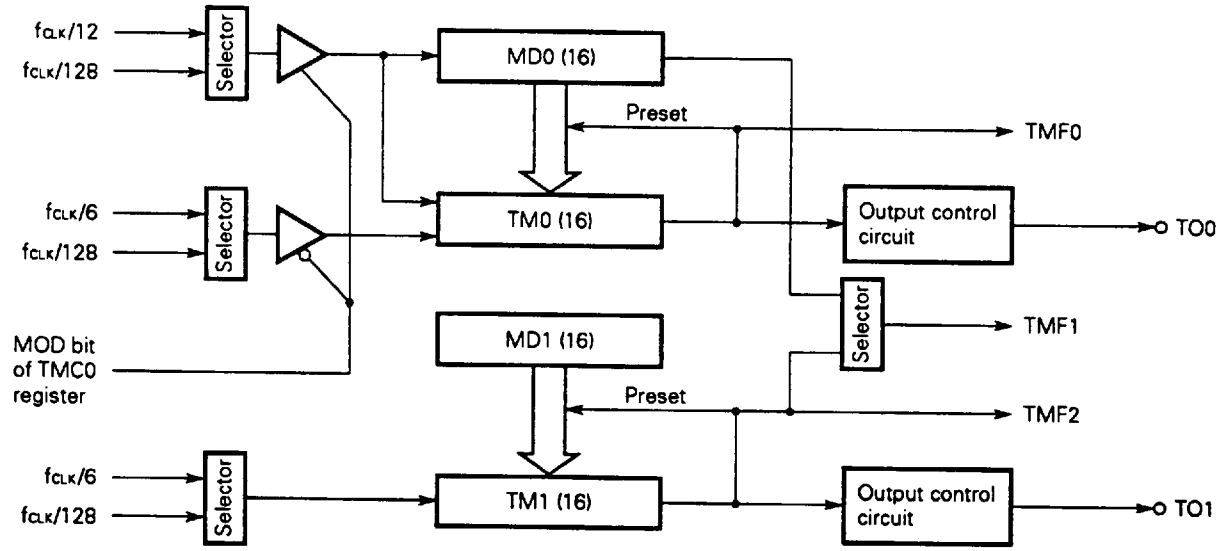
5.2.4 Timer Unit

(1) Configuration of the timer unit

The timer unit can be used as an interval timer, a one-shot timer, or a timer for controlling square wave output or the output of a real-time output port.

As shown in Fig. 5-5, the timer unit consists of 16-bit timer registers (TM0 and TM1), 16-bit modulo/timer registers (MD0 and MD1), 8-bit timer control registers (TMC0 and TMC1), and an output control circuit.

Fig. 5-5 Block Diagram of the Timer Unit



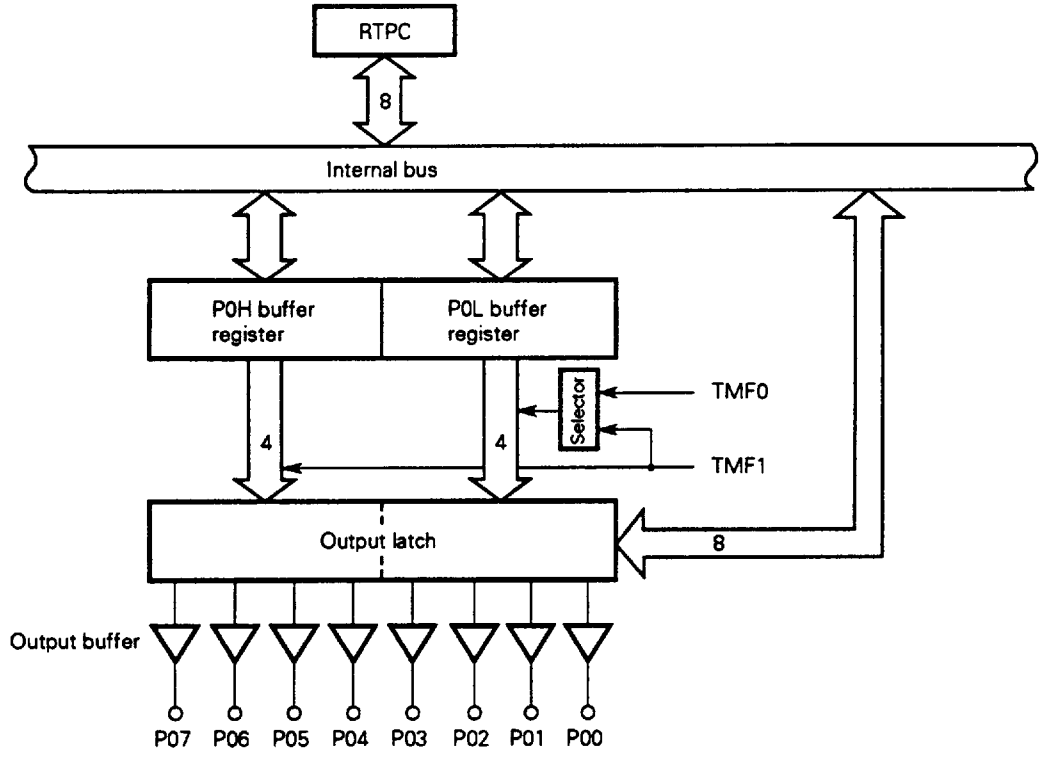
Remark f_{CLK}: Internal system clock frequency

5.2.5 Real-Time Output Port

The real-time output port can output the contents of the port 0 buffer register. The real-time output port is included in port 0, and it functions in units of 4 or 8 bits at an interval set by software.

The real-time output port has a master-slave configuration as shown in Fig. 5-6. It automatically transfers the buffer register contents to the output latch on the timing signal issued from the timer unit. Thus, output without jitters can be obtained on given interval timing generated by the timer unit. This function is suitable for controlling a stepping motor.

Fig. 5-6 Block Diagram of the Real-Time Output Port



5.3 A/D CONVERTER

The μPD78312A(A) contains an analog/digital (A/D) converter with four multiplexed analog inputs (AN3 - AN0). The converter uses the successive approximation method for conversion and stores the conversion result in the A/D conversion result register (ADCR). (Conversion takes 30 μs when the 6-MHz internal clock is used for operation.)

A $\overline{\text{RESET}}$ input does not affect the contents of the ADCR.

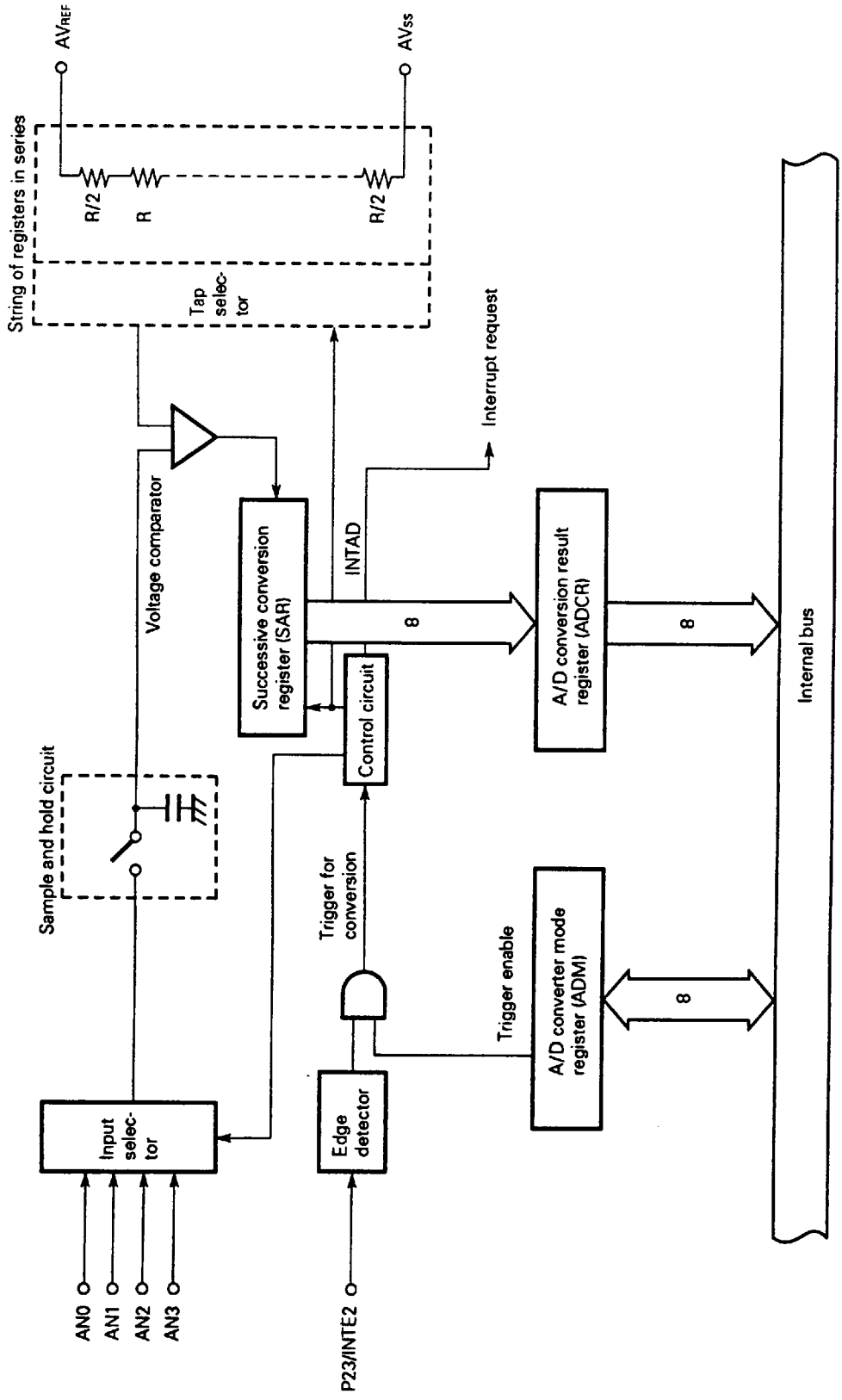


Fig. 5-7 Block Diagram of the A/D Converter

5.4 SERIAL COMMUNICATION INTERFACE

5.4.1 Configuration of the Serial Communication Interface

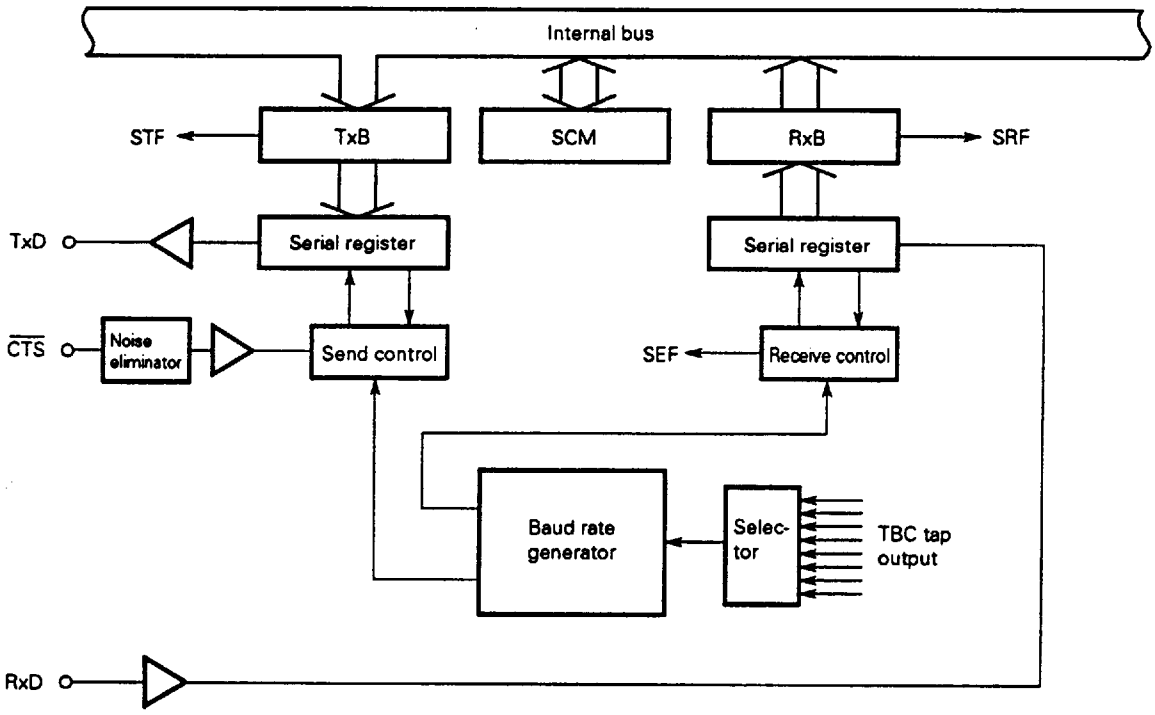
The μPD78312A(A) has a serial communication interface with a special baud rate generator.

The serial communication interface provides two types of operation modes: the asynchronous mode and the I/O interface mode. In the asynchronous mode, start and stop bits are used to indicate the beginning and end of bits or characters in transmission. In the I/O interface mode, data transfer is synchronized with a controlled serial clock in the same way as the serial data transfer method used in the 87AD series.

The serial communication interface consists of four pins including serial data input (RxD), serial data output (TxD), serial clock output ($\overline{SCK}$), and send enable control input ($\overline{CTS}$), a transfer control section, 8-bit serial registers for send and receive operations, a send buffer, a receive buffer, and a baud rate generator. Separate serial registers and buffers are provided for send and receive operations so that the send and receive operations can be performed independently. In the I/O interface mode, the $\overline{CTS}$ pin functions as receive clock I/O, enabling serial communication in the full-duplex mode.

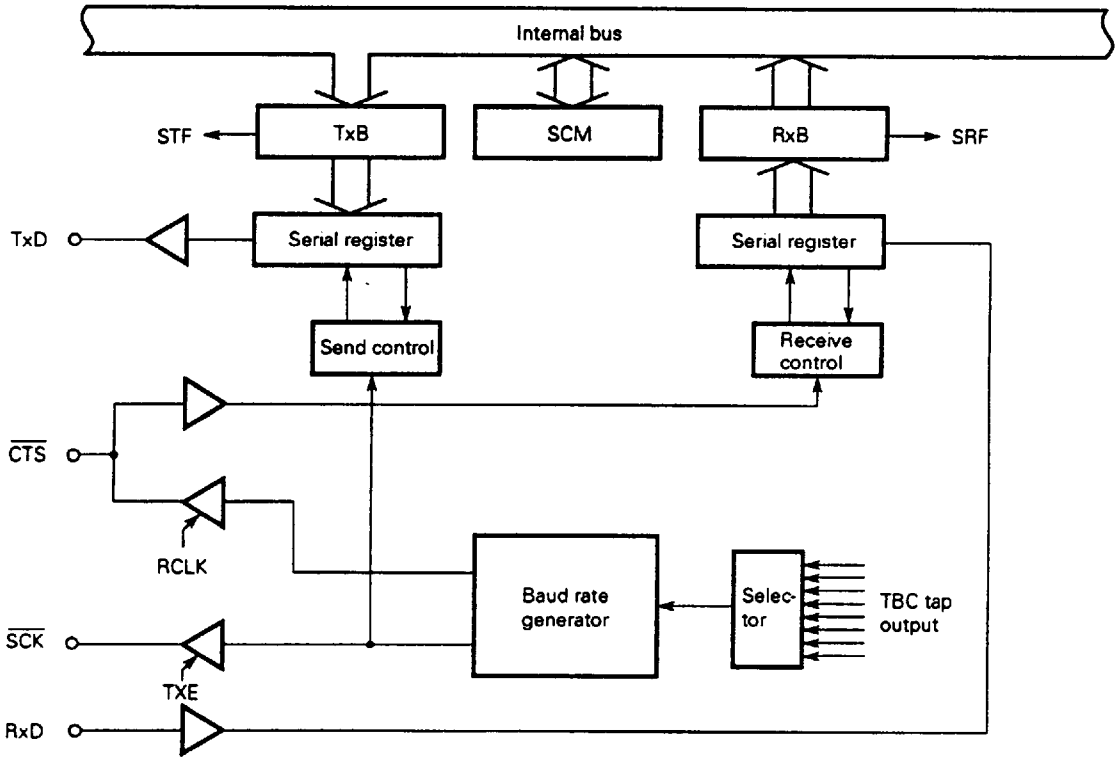
Fig. 5-8 Block Diagram of the Serial Communication Interface

(a) When the asynchronous mode is set:



Remark The $\overline{SCK}$ pin is held high.

(b) When the I/O interface mode is set:



5.4.2 Configuration of the Baud Rate Generator (BRG)

The BRG is an 8-bit timer dedicated to the serial communication interface. It generates a shift clock signal for send and receive operations. There is a send BRG and a receive BRG (see Fig. 5-9).

Table 5-3 lists samples of transfer baud rate setting with the BRG.

Fig. 5-9 BRG Configuration

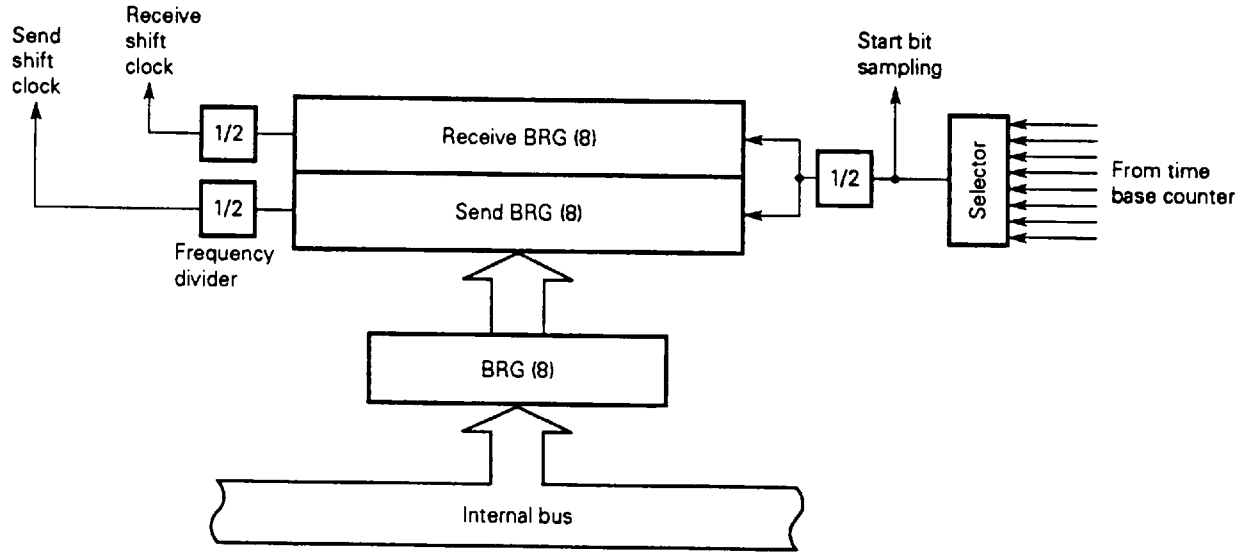


Table 5-3 BRG Setting (Reference)

Transfer baud rate	n	BRG set value G	Error (%)
110	7	107	0.43
150	7	78	0.16
300	6	78	0.16
600	5	78	0.16
1200	4	78	0.16
2400	3	78	0.16
4800	2	78	0.16
9600	1	78	0.16
19200	0	78	0.16
38400	0	39	0.16
1.5M	0	1	0

Remark n: Specification number of the input clock to BRG
f_{CLK} = 6 MHz

Caution In the I/O interface mode, set a value of 2 or greater for the BRG. (When G is 2 with the 6-MHz internal system clock, the transfer baud rate is 750 kbps.)

5.5 TIME BASE COUNTER AND WATCHDOG TIMER

The μPD78312A(A) has a 20-bit time base counter and an 8-bit watchdog timer that receives the tap output of the time base counter as a clock source.

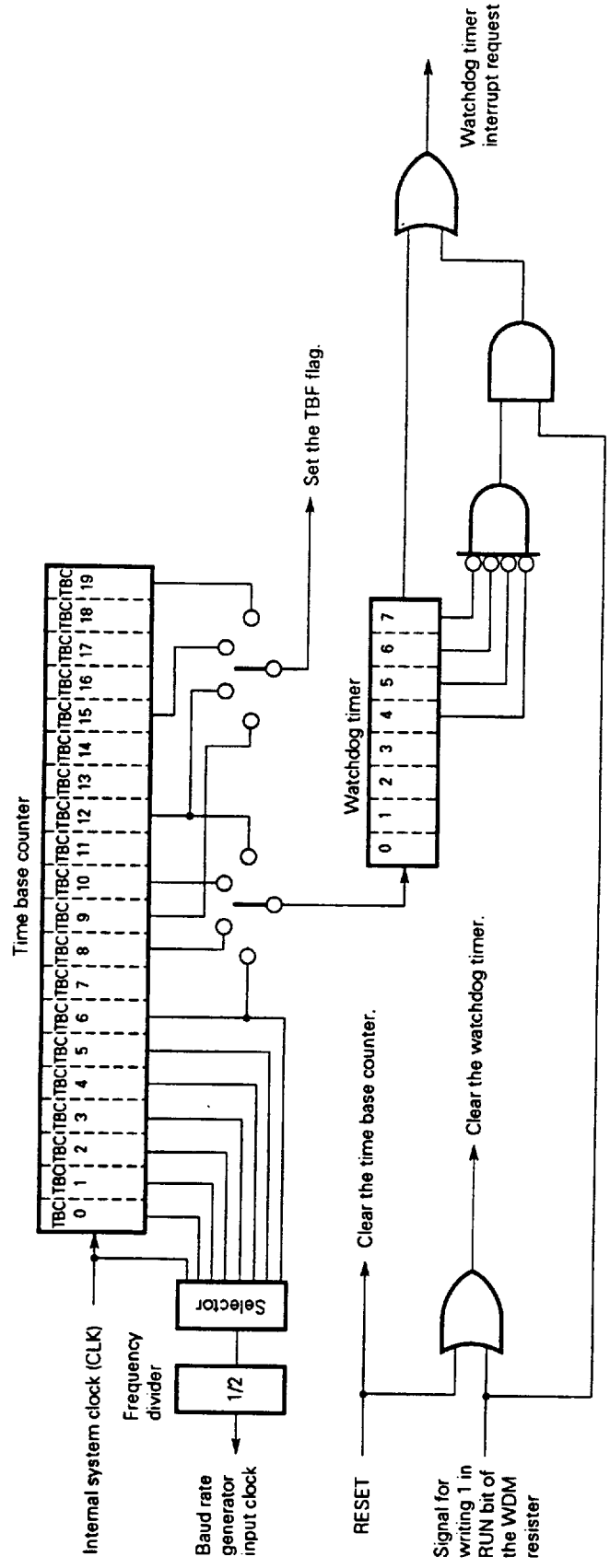


Fig. 5-10 Configuration of the Time Base Counter and Watchdog Timer

5.5.1 Time Base Counter

For system controller applications, various types of processing may be executed on a certain time interval basis.

To obtain such a reference time base output, the μPD78312A(A) contains a 20-bit time base counter that frequency-divides the internal system clock (CLK).

Table 5-4 lists available interrupt request time intervals which may be generated by the time base counter.

Table 5-4 Intervals Set by the Time Base Counter for Interrupt Generation

Time interval
$2^{10}/f_{CLK}$ 170 μsec
$2^{13}/f_{CLK}$ 1.36 msec
$2^{16}/f_{CLK}$ 10.9 msec
$2^{20}/f_{CLK}$ 175 msec

Remark f_{CLK} = 6 MHz

5.5.2 Watchdog Timer

The μPD78312A(A) has a watchdog timer for detecting program crashes.

When 1 is written in bit 7 (RUN) of the watchdog timer mode register (WDM), the watchdog timer is cleared to 00H and starts counting the tap output of the time base counter specified by the watchdog timer mode register (WDM).

If the watchdog timer is cleared to 00H before bit 4 of the watchdog timer is set to 1 for the first time, or if the watchdog timer overflows before it is cleared, a nonmaskable interrupt request is generated.

The nonmaskable interrupt request from the watchdog timer can specify a priority relative to the nonmaskable interrupt request generated by the input on the NMI pin.

Table 5-5 lists the times it takes the watchdog timer to overflow, and the times it takes to set bit 4 to 1 after count operation starts (6.25 % of the overflow time), where the internal system clock operates at 6 MHz.

Table 5-5 Watchdog Timer Count Clock and Overflow Time

Watchdog timer count clock	Overflow time	6.25 % of overflow time
$f_{CLK}/2^7$ (TBC6 tap output)	5.5 ms	343 μs
$f_{CLK}/2^9$ (TBC8 tap output)	21.8 ms	1.36 ms
$f_{CLK}/2^{11}$ (TBC10 tap output)	87.4 ms	5.46 ms
$f_{CLK}/2^{13}$ (TBC12 tap output)	349.5 ms	22 ms

Remark f_{CLK} = 6 MHz

6. INTERRUPT FUNCTION

6.1 INTERRUPT

The μPD78312A(A) has interrupt sources listed in Table 6-1. They are classified into the following three types:

- Software interrupts
- Nonmaskable interrupts
- Maskable interrupts

The μPD78312A(A) provides eight priority levels in accepting maskable interrupt requests. These priority levels can be set by software. In addition, one of the processing modes listed in Table 6-2 can be selected.

Table 6-1 Interrupt Sources

Interrupt request type	Default priority	Interrupt request source		Macro service	Vector table address
Software	–	BRK	Execution of BRK or BRKCS instruction	–	003EH
Nonmaskable	–	NMI	NMI pin input	–	0002H
	–	WDT	Watchdog timer	–	000AH
Maskable	0	CRF00	Count unit	Provided	001AH
	1	CRF01	Count unit	–	001CH
	2	CRF10	Count unit	Provided	001EH
	3	CRF11	Count unit	–	0020H
Maskable	4	EXIF0	INTE0 pin input	Provided	0004H
	5	EXIF1	INTE1 pin input	Provided	0006H
	6	EXIF2	INTE2 pin input	Provided	0008H
Maskable	7	TMF0	Timer unit	Provided	000EH
	8	TMF1	Timer unit	Provided	0010H
	9	TMF2	Timer unit	Provided	0012H
Maskable	10	SEF	Serial interface error	–	0022H
	11	SRF	Serial reception	Provided	0024H
	12	STF	Serial transmission	Provided	0026H
Maskable	13	ADF	A/D converter	Provided	0028H
	14	TBF	Time base counter	–	000CH
Reset	–	$\overline{\text{RESET}}$	Reset input	–	0000H

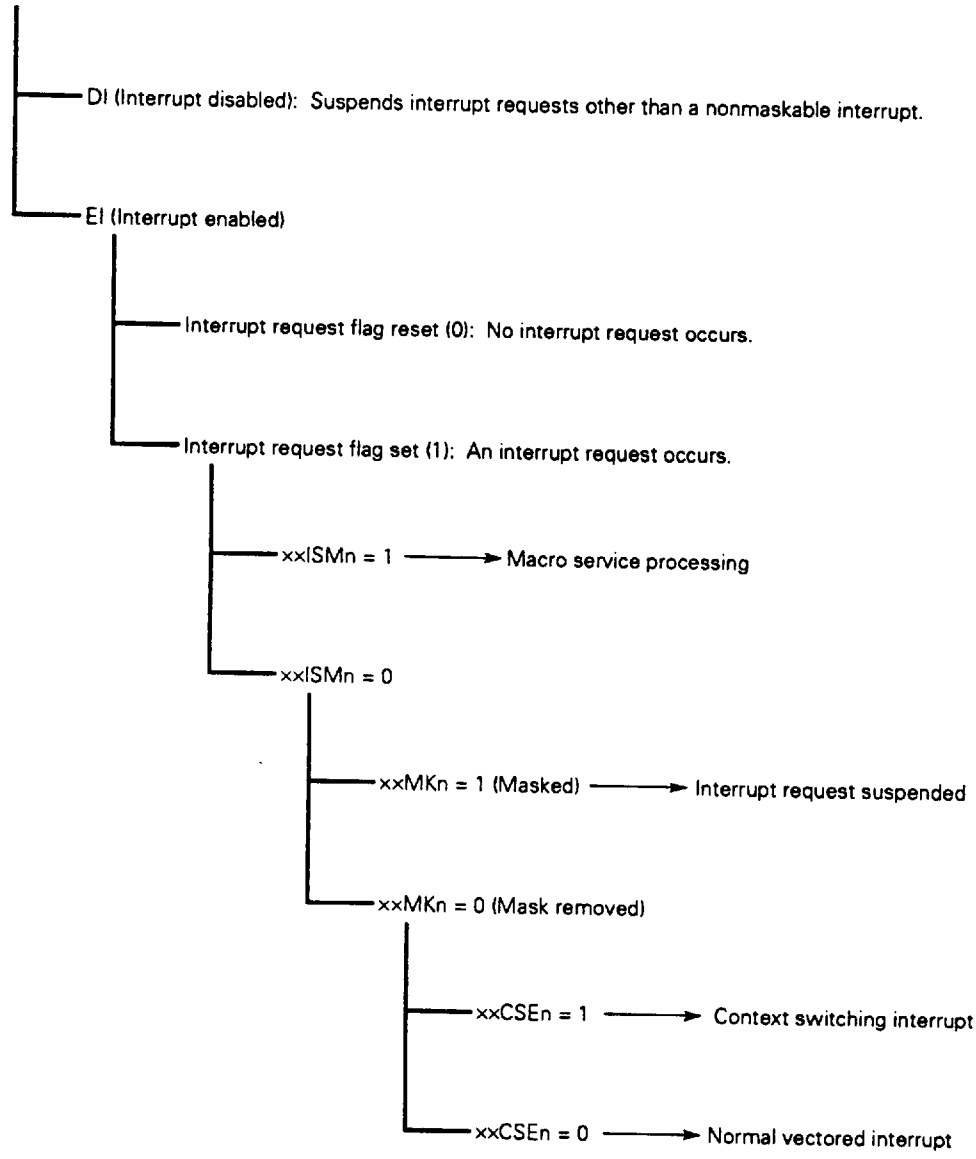
Remark The default priority is the priority set by hardware.

Table 6-2 Interrupt Processing Mode

Processing mode	Handled by	Processing	PC and PSW contents
Vectored interrupt	Software	Loads a vector address into the program counter (PC), then branches to an interrupt service routine. General register banks are left unchanged.	Saved, then restored (stack)
Context switching		Changes the general register bank, and branches to an interrupt service routine.	Saved, then restored (within register banks)
Macro service	Firmware	Transfers data between memory and SFR.	Held

Fig. 6-1 shows the processing form for the maskable interrupt request.

Fig. 6-1 Processing for Interrupt Request



6.2 MASKABLE INTERRUPT PRIORITY CONTROL

Maskable interrupt requests are classified into five groups as shown in Table 6-3. A priority can be specified for each group by software.

Up to eight priority levels, 0 to 7, can be set. The same priority level can be specified for more than one group.

If more than one interrupt is generated at the same time within the same group or within groups having the same priority level, the interrupts are accepted in order of default priority shown in Table 6-3.

Caution If an interrupt request having the same or lower priority than an interrupt request being processed is generated, it is not regarded as a multiple interrupt and is rejected.

Table 6-3 Grouping of Maskable Interrupts

Group	Default priority	Interrupt source
Count unit	0	CRF00
	1	CRF01
	2	CRF10
	3	CRF11
External interrupt	4	EXIF0
	5	EXIF1
	6	EXIF2
Timer unit	7	TMF0
	8	TMF1
	9	TMF2
Serial interface	10	SEF
	11	SRF
	12	STF
A/D converter time base counter	13	ADF
	14	TBF

6.3 MACRO SERVICE

When an interrupt request is generated, the macro service function temporarily stops the CPU executing a program, and automatically transfers 1 or 2 bytes of data between a special function register (SFR) and memory. This function neither saves nor restores the CPU state, nor includes data, so that high-speed transfer operation can be performed.

(1) Example of using the macro service function

In the following example, analog-to-digital conversion is performed with an A/D converter six times, then the conversion results are stored in the area from FE40H to FE45H in the internal RAM. Macro service channel 4 is used here.

After data has been transferred six times, the transferred data is processed by an interrupt service program.

Fig. 6-2 Example of Macro Service Operation

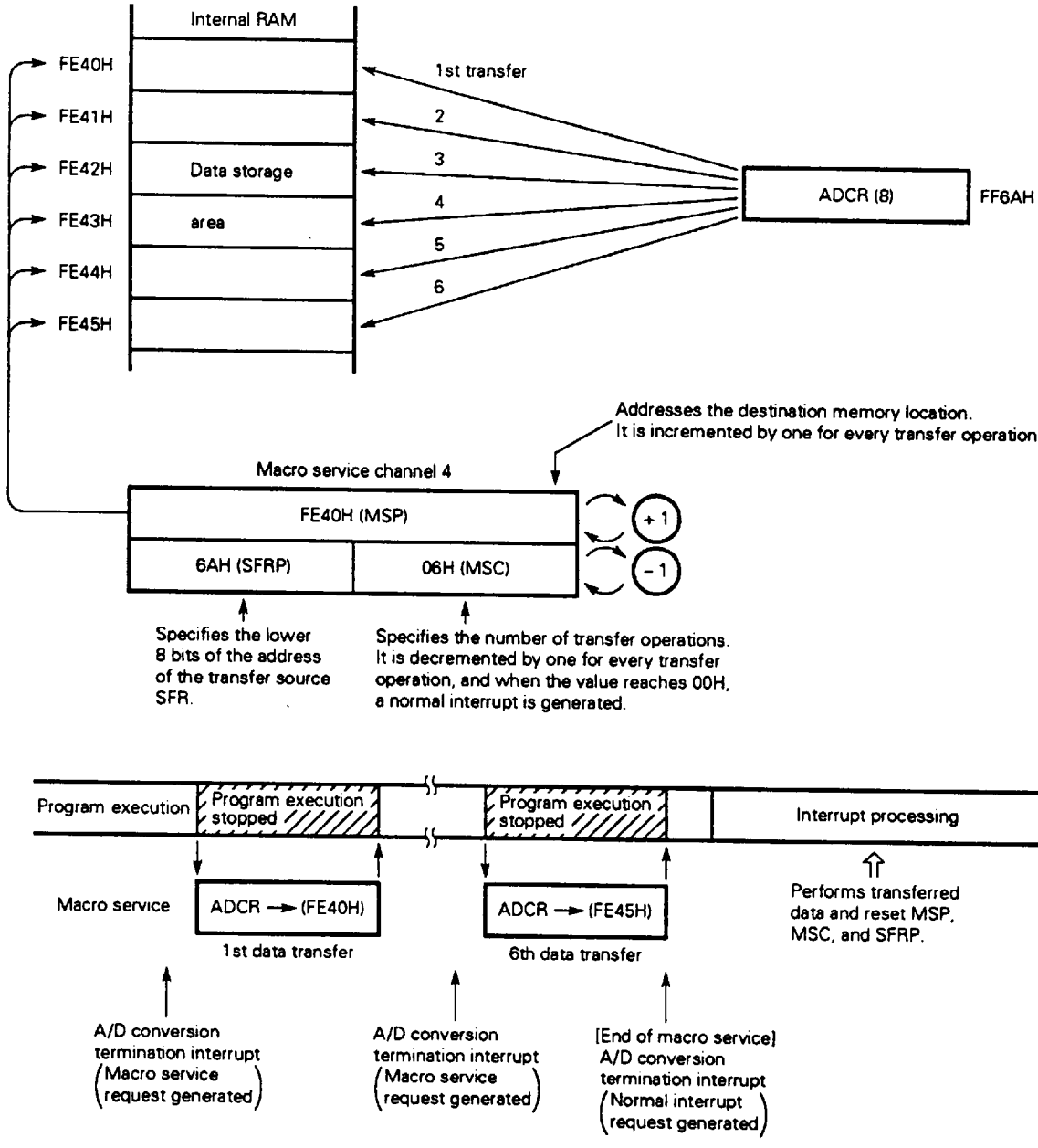
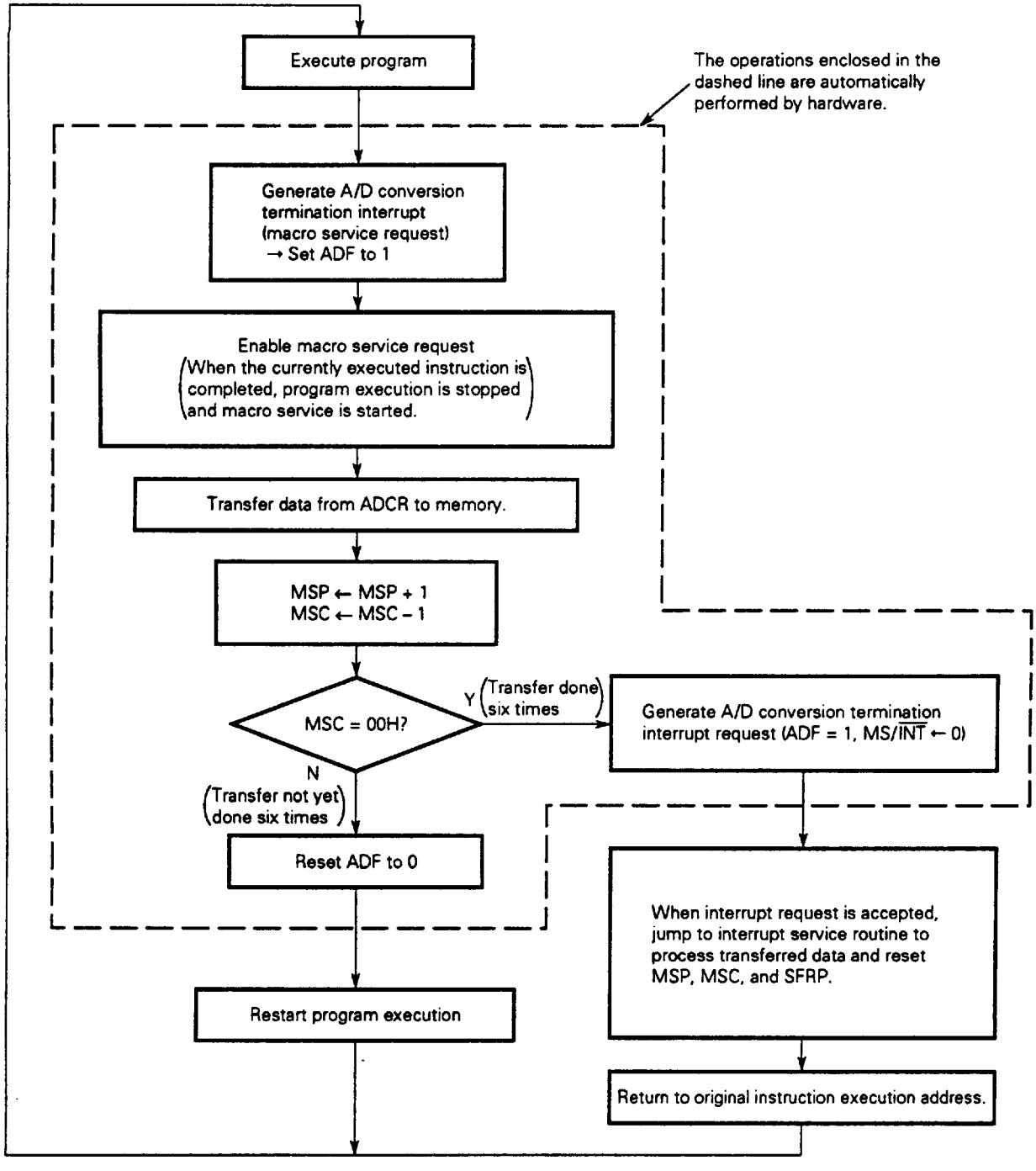


Fig. 6-3 Flowchart of Macro Service Operation Example



ADF: A/D conversion interrupt request flag

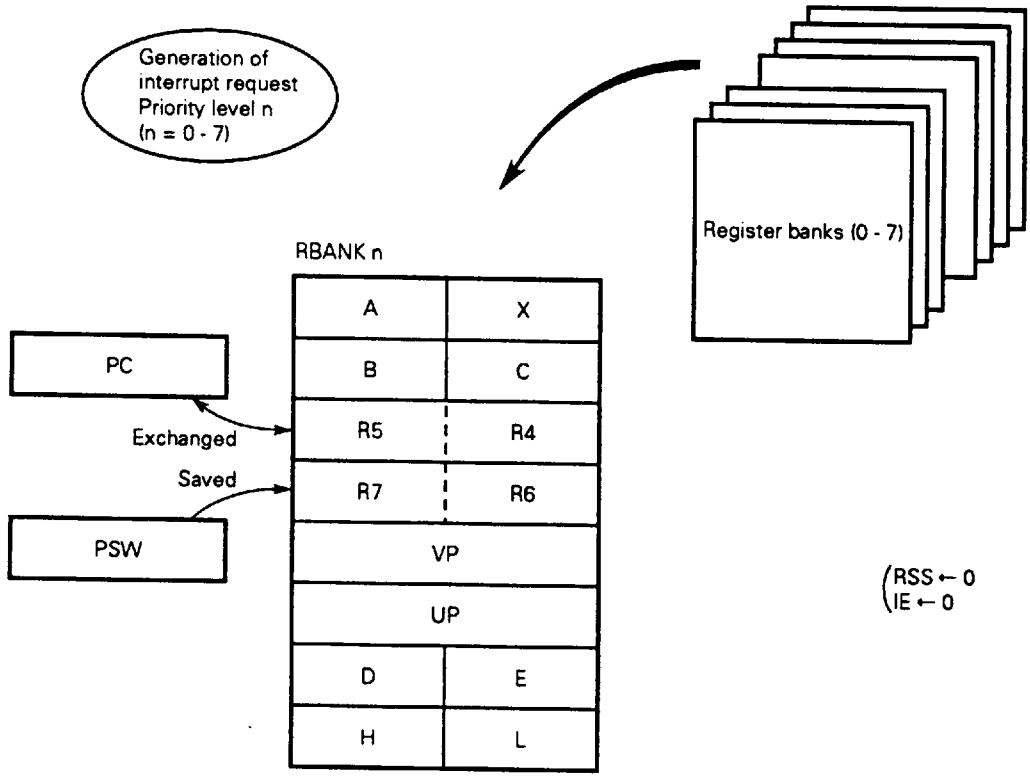
6.4 CONTEXT SWITCHING

When an interrupt request is generated or a BRKCS instruction is executed, the context switching function selects an appropriate register bank by hardware, and causes a branch to the vector address stored beforehand in the register bank. It also stacks the current PC and PSW contents in the register bank at the same time. The context switching function can be specified only for maskable interrupts.

6.4.1 Context Switching Due to Interrupt Request Generation

The activation of the context switching function is enabled by setting the ENCS bit of the interrupt request control register to 1 in the source of an interrupt request. When an interrupt request occurs in the EI status, which is not masked, and for which the context switching function is enabled, a register bank associated with the priority level of the group to which the interrupt request belongs is selected. The context switching function loads into the PC the vector address stored in the selected register bank beforehand, and saves the contents stored so far in the PC and PSW and branches to the interrupt service routine at the same time (Fig. 6-4).

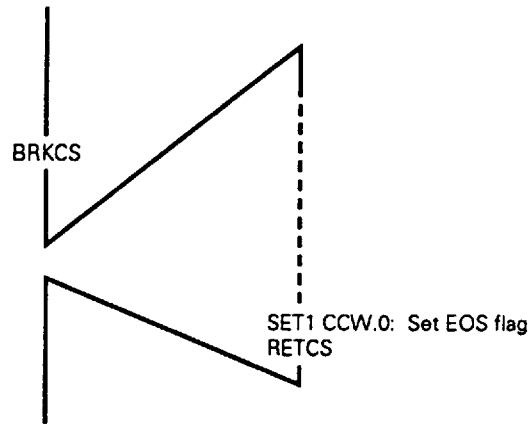
Fig. 6-4 Context Switching Due to Interrupt Request Generation



6.4.2 Return from the Service Routine to Which a Branch Is Caused by Context Switching

Return from branch processing caused by the context switching function is performed by executing the RETCS instruction.

When the RETCS instruction is executed, the contents of the R4 and R5 registers in the register bank selected for execution are loaded into the PC, and the contents of the R6 and R7 are loaded into the PSW. At the same time, 16-bit immediate data coded in an operand of the RETCS instruction is stored in the R4 and R5 registers in the register bank. When the same register bank is again selected by the context switching function, the 16-bit immediate data for an operand of the RETCS instruction is used as the branch address.



7. STANDBY FUNCTION

As a standby function, the μPD78312A(A) provides the following two modes:

- HALT mode: The CPU operation clock is stopped.
  Intermittent operation can be performed by combining the HALT mode and the normal operation mode to reduce the total system power consumption.
- STOP mode: The oscillator is stopped.
  In this mode, data can be held with low power consumption.

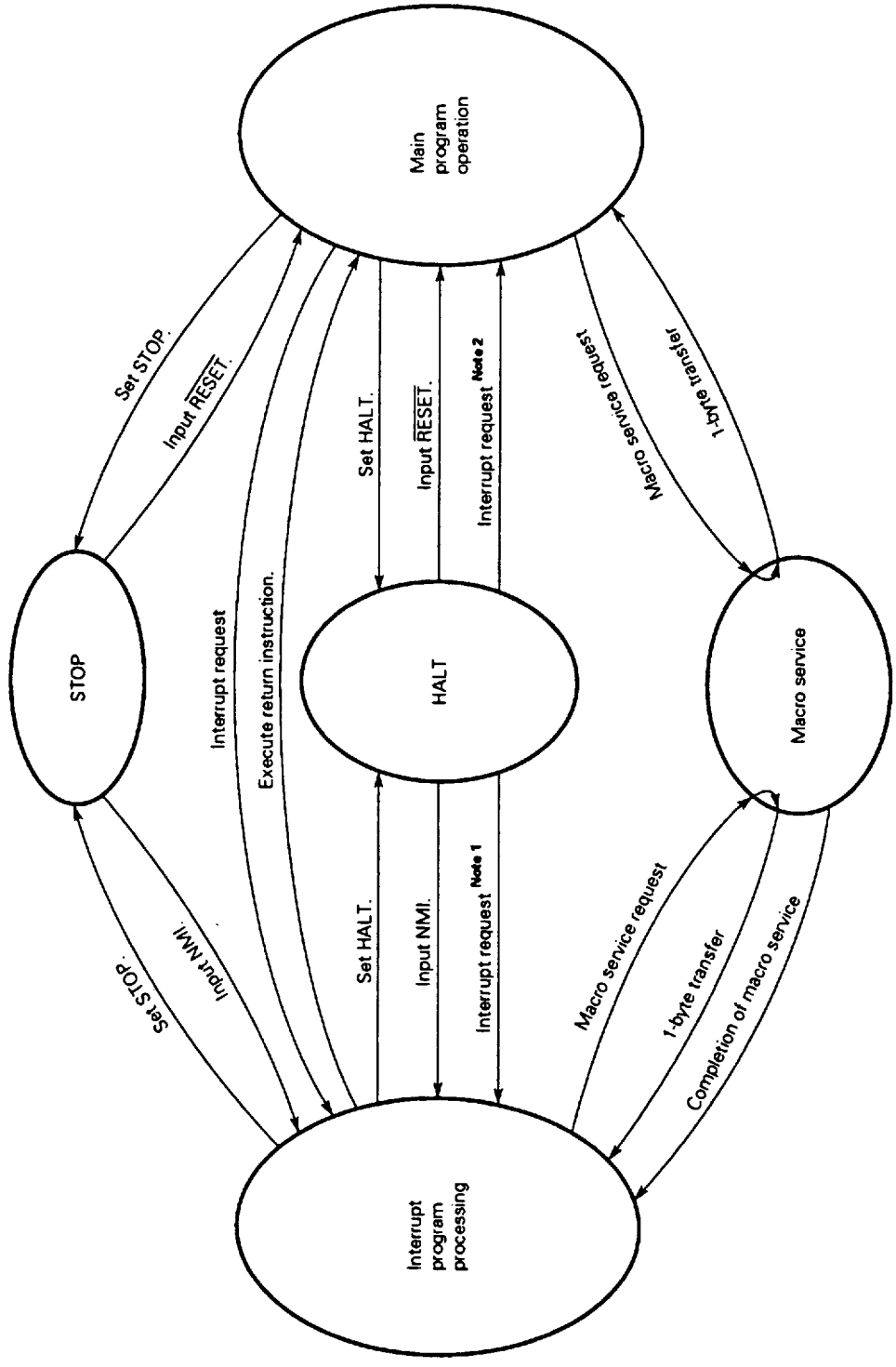


Fig. 7-1 Standby Status Transition

Notes 1. Maskable interrupt enabled (EI status)
2. Maskable interrupt enabled (DI status)

8. LOCAL BUS INTERFACE

The μPD78312A(A) allows external memories (ROM and RAM), other than internal memory, and input/output devices to be connected.

8.1 EXTERNAL DEVICE ACCESS FUNCTION

External devices are accessed through a multiplexed address/data bus on P47 to P40 and an address bus on P57 to P50 by using the $\overline{RD}$, $\overline{WR}$, and ALE signals.

The μPD78312A(A) allows external memory and input/output devices to be expanded according to the memory expansion mode register (MM) in steps from 256 bytes to 4K bytes to 16K bytes and to 56K bytes.

The μPD78312A(A) can connect memory and input/output devices to external 64K-byte space regardless of the MM register.

8.2 PROGRAMMABLE WAIT FUNCTION

The μPD78312A(A) has a programmable wait function that can automatically insert up to three waits into external access cycles when the μPD78312A(A) is connected to a low-speed memory or input/output devices.

8.3 PSEUDO-STATIC RAM REFRESH FUNCTION

The μPD78312A(A) has a pseudo-static RAM refresh function so that a pseudo-static RAM can directly be connected to the μPD78312A(A).

Refresh operations include:

- Pulse refresh : Outputs refresh pulses on the $\overline{RFSH}$ pin in synchronization with the bus cycle.
- Power-down self-refresh: In the standby mode, outputs a low on the $\overline{RFSH}$ pin to hold the contents of the pseudo-static RAM.

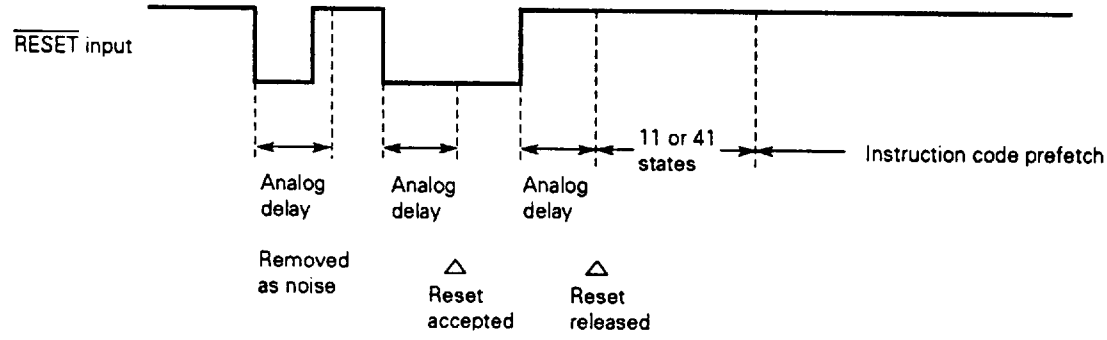
9. RESET

A low level input on the $\overline{\text{RESET}}$ input pin resets the system, placing each hardware component in a certain state as listed in Table 9-1. When the $\overline{\text{RESET}}$ input becomes high, the reset status is released and program execution starts. Registers must be initialized in a program as required.

The $\overline{\text{RESET}}$ input pin has a noise eliminator using analog delay to prevent malfunction due to noise.

It takes 11 states, when internal ROM is accessed, to start prefetching an instruction code after the reset status is released, or it takes 41 states when external memory is accessed.

Fig. 9-1 Acceptance of a Reset Signal



To settle oscillation at a power-on reset, a time period of about 40 ms is required from when the microcomputer is turned on to when it receives a reset signal.

Fig. 9-2 Power-On Reset

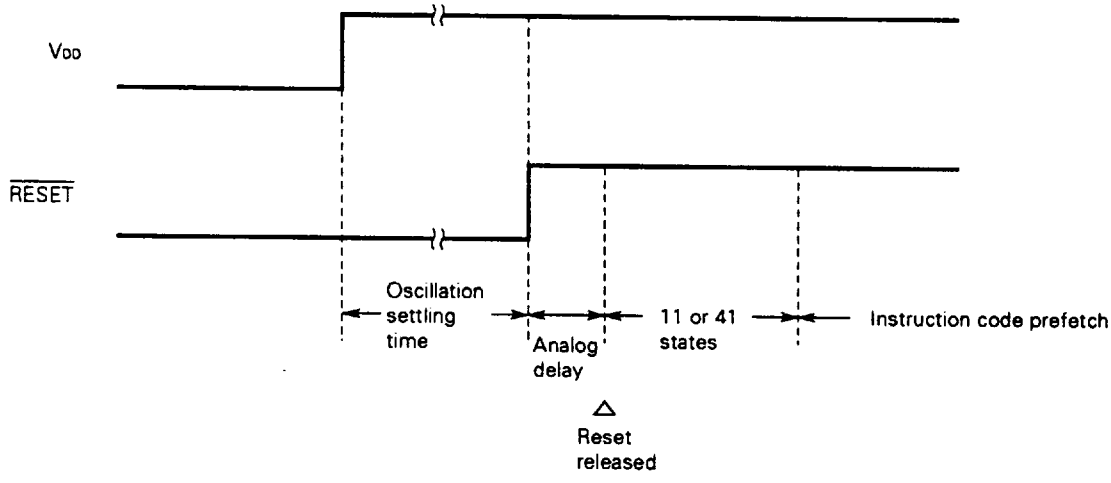


Table 9-1 Hardware States after Reset (1/2)

Hardware		State after reset
Program counter (PC)		00H
Stack pointer (SP)		Undefined
Program status word (PSW)		00H
CPU control word (CCW)		00H
Internal RAM	Data memory	Undefined Note
	General registers (R0 - R15)	
Port	Output latch (P0 - P5)	Undefined
	Mode registers (PM0 - PM3, PM5)	FFH (input mode)
	Mode control registers (PMC2 and PMC3)	0FH
Count unit	Capture compare registers (CR00, CR01, CR10, CR11)	Undefined
	Up/down count registers (UDC0 and UDC1)	Undefined
	Input mode register (CUIM)	00H
	UDC control registers (UDCC0 and UDCC1)	00H
	Capture compare register control register (CRC)	00H
Capture PWM unit	Capture register (CPT0 and CPT1)	Undefined
	PWM registers (PWM0 and PWM1)	Undefined
	FRC control register (FRCC)	00H
	Capture mode register (CPTM)	00H
	PWM mode register (PWMM)	00H
Real-time output port	Control register (RTPC)	08H
	Port 0L buffer register (P0L)	Undefined
	Port 0H buffer register (P0H)	Undefined
Timer unit	Timer registers (TM0 and TM1)	Undefined
	Modulo/timer registers (MD0 and MD1)	Undefined
	Timer control registers (TMC0 and TMC1)	00H
A/D converter	Mode register (ADM)	00H
	Conversion result register (ADCR)	Undefined

Note When the standby mode is released by $\overline{\text{RESET}}$ input, the state before the standby mode was set is restored.

Table 9-1 Hardware States after Reset (2/2)

Hardware		State after reset
Serial communication interface	Serial mode register (SCM)	00H
	Serial control register (SCC)	00H
	Baud rate generator set value (BRG)	00H
	Receive buffer register (RxB)	Undefined
	Send buffer register (TxB)	Undefined
Time base counter		00H
Time base mode register (TBM)		00H
Standby control register (STBC)		2xH Note
Watchdog timer mode register (WDM)		00H
Memory expansion mode register (MM)		30H
Refresh mode register (RFM)		10H
Interrupt request	External interrupt mode register (INTM)	00H
	In-service priority register (ISPR)	00H
	Interrupt request control register	47H
	Macro service control register	Undefined

Note Bit 3 of STBC is not affected by $\overline{\text{RESET}}$ input, so the lower 4 bits of STBC are either 0 or 8.

10. INSTRUCTION SET

10.1 INSTRUCTION SET AND OPERATIONS OF INDIVIDUAL INSTRUCTIONS

(1) Operand notation and coding format

Operands are coded in the operand field of each instruction as listed in the coding column of Table 10-1. For details of the operand format, refer to the relevant assembler specifications. When several coding forms are presented, any one of them is selected. Uppercase letters and the symbols, +, –, #, $, !, and [ ] are keywords and must be written as they are.

For immediate data, an appropriate numeric or label must be written.

Table 10-1 Operand Notation and Coding Format

Notation	Coding
r r1 r2	R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15 R0, R1, R2, R3, R4, R5, R6, R7 C, B
rp rp1 rp2	RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 DE, HL, VP, UP
sfr sfrp	Special function register abbreviation (See Table 4-2.) Special function register abbreviation (16-bit manipulation register: See Table 4-2.)
post	RP0, RP1, RP2, RP3, RP4, RP5/PSW, RP6, RP7 (Can be coded more than once. However, RP5 can only be used in a PUSH or POP instruction and PSW can only be used in a PUSHU or POPU instruction.)
mem	[DE], [HL], [DE+], [HL+], [DE–], [HL–], [VP], [UP] : Register indirect mode [DE+A], [HL+A], [DE+B], [HL+B], [VP+DE], [VP+HL] : Base index mode [DE+byte], [HL+byte], [VP+byte], [UP+byte], [SP+byte]: Base mode word [A], word [B], word [DE], word [HL] : Index mode
saddr saddrp	FE20H - FF1FH Immediate data or label FE20H - FF1EH Immediate data (bit 0 = 0, however) or label (for 16-bit manipulation)
$addr16 !addr16 addr11 addr5	0000H - FEFFH Immediate data or label: Relative addressing 0000H - FEFFH Immediate data or label: Immediate addressing (Data up to FFFFH can be coded in an MOV instruction.) 800H - FFFH Immediate data or label 40H - 7EH Immediate data (bit 0 = 0) or label
word byte bit n	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label 3-bit immediate data (0 to 7)

Remarks 1. The same register name can be specified in rp and rp1, but different codes are generated.

2. Functional names (X, A, C, B, E, D, L, H, AX, BC, DE, HL, VP, and UP) can be specified in r, r1, rp, rp1, and post, as well as absolute names (R0 to R15 and RP0 to RP7). See **Table 4-1** for the correspondence between the absolute names and functional names.
Refer to *μPD78312A User's Manual* (IEM-5086) for details.

3. Immediate addressing is effective for entire address spaces. Relative addressing is effective for the locations within a displacement range of –128 to +127 from the starting address of the next instruction.

(2) **Legend**

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
R0 - R15	:	Register 0 to register 15 (absolute name)
AX	:	Register pair (AX); 16 bit accumulator
BC	:	Register pair (BC)
DE	:	Register pair (DE)
HL	:	Register pair (HL)
RP0 - RP7	:	Register pair 0 to register pair 7 (absolute name)
PC	:	Program counter
SP	:	Stack pointer
UP	:	User stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
P/V	:	Parity/overflow flag
S	:	Sign flag
SUB	:	Subtraction flag
TPF	:	Table position flag
RBS	:	Register bank select flag
RSS	:	Register set select flag
IE	:	Interrupt enable flag
EOS	:	End-of-software interrupt flag
STBC	:	Standby control register
WDM	:	Watchdog timer mode register
( )	:	Contents at an address enclosed in parentheses or at an address indicated in a register indicated in parentheses. (+) and (−) indicate that an address or the contents of a register indicated in parentheses are incremented and decremented by one after execution of the instruction, respectively.
(( ))	:	Contents at an address indicated by the contents at an address indicated in parentheses (( )).
xxH	:	Hexadecimal number
xH, xL	:	Eight high-order bits and eight low-order bits of 16-bit register
!xx	:	Address indicated through immediate addressing
$xx	:	Address indicated through relative addressing

10.2 CALCULATING THE NUMBER OF INSTRUCTION EXECUTION STATES

A number in the state field indicates the number of instruction execution states excluding fetch cycles. Thus, it sometimes differs from the number of states required for actual execution of an instruction. To approximate the number of instruction execution states, obtain the basic value from the following expression first:

If the calculated result is smaller than the number of states, this number of states is used as the basic value for the number of states required for executing an instruction. (For details, refer to the user's manual.)

- When a program is loaded in internal ROM
  (Number of states) + 1 × (number of bytes of execution instruction) − (number of free states)
- When a program is loaded in external ROM
  (Number of states) + (4 + m) × (number of bytes of execution instruction) − (number of free states)
  m: Number of waits to be inserted according to the MM register specification
- When a program is loaded in internal RAM
  (Number of states) + 3 × (number of bytes of execution instruction)

Next, obtain the number of states to be added by using the basic value. Except for the following cases, there need be no additional states, and the basic value obtained from the above expression is used as the number of instruction execution states. The numbers of states to be added are given below.

(a) When the vector table or CALLT table is set up in external memory, only the CALLT and BRK instructions require (12 + 4m) additional states (where m is the number of waits specified by the MM register).

(b) When a special function register (SFR) or external memory is accessed, additional states are required as shown in Table 10-2.

Table 10-2 Number of Additional States per Access

Access target	Number of additional states per access
Internal memory (ROM, RAM)	0
Special function register (SFR)	Number of accesses to SFR × k
External memory	Number of accesses to external memory × (2 + m)

k: Number of waits generated when the timer unit or count unit is accessed, which varies within the following range depending on the counter access condition:

Accessed SFR	Value of k
Timer unit (TMn, MDn)	0 to 5
Counter unit (UDCn, CRnn)	0 to 2
Other SFRs	0

(n = 0, 1)

m: Number of waits specified by the MM register when external memory is accessed

Each state takes 167 ns when 6 MHz is specified as the internal system clock.

Table 10-3 Numbers of Accesses to SFR by Instructions

Mnemonic	Operand	No. of accesses to SFR
MOV	sfr, #byte	1
	A, sfr	
	sfr, A	
XCH	A, sfr	2
MOVW	sfrp, #word	1
	AX, sfrp	
	sfrp, AX	
XCHW	AX, sfrp	2
ADD, ADDC, SUB, SUBC, AND, OR, XOR	sfr, #byte	2
	A, sfr	1
CMP	sfr, #byte	1
	A, sfr	
ADDW, SUBW	sfrp, #word	2
	AX, sfrp	1
CMPW	sfrp, #word	1
	AX, sfrp	
MOV1	CY, sfr.bit	1
	sfr.bit, CY	2
AND1, OR1, XOR1	CY, sfr.bit	1
SET1, CLR1, NOT1	sfr.bit	2
BT, BF	sfr.bit, $addr16	1
BTCLR, BFSET	sfr.bit, $addr16	1/2 Note

Note If a condition is met and a branch is made, SFR is accessed twice.

Table 10-4 Numbers of Accesses to Memory by Instructions

Mnemonic	Operand	No. of accesses to memory
MOV	A, mem	1
	mem, A	
	A, [saddrp]	
	[saddrp], A	
	A, !addr16	
	!addr16, A	
XCH	A, mem	2
	mem, A	
MOVW	rp1, !addr16	2
	!addr16, rp1	2
ADD, ADDC, SUB, SUBC, AND, OR, XOR	A, mem	1
	mem, A	2
CMP	A, mem	1
	mem, A	
ROR4, ROL4	[rp1]	2
CALL	!addr16	2
	rp1	
	[rp1]	4
CALLF	!addr11	2
CALLT	[addr5]	2
BRK		4
RET		2
RETI		4
PUSH, POP	post	2 × n
	PSW	2
PUSHU, POPU	post	2 × n
BR	[rp1]	2
MOVM, CMPME, CMPMNE, CMPMC, CMPMNC	[DE+], A	1 × s
	[DE−], A	
MOVBK, CMPBKE, CMPBKNE, CMPBKC, CMPBKNC	[DE+], [HL+]	2 × s
	[DE−], [HL−]	
XCHM	[DE+], A	2 × s
	[DE−], A	
XCHBK	[DE+], [HL+]	4 × s
	[DE−], [HL−]	

Remark n: Number of registers coded for post in the PUSH or POP instruction

s: Iteration count in a string instruction

10.3 SYMBOLS IN STATE FIELD

(i) If n is indicated in the state field, the value of n is determined as follows:

- Stack manipulation instruction : Number of registers to be saved or restored
- Shift rotate instruction : Number of bits by which a shift operation is performed
- String instruction : Number of times the instruction is repeated until a certain condition is met to exit from a loop

(ii) A number in parentheses in the state field of a conditional branch instruction indicates the minimum number of execution states required when no branch takes place.

(iii) In an instruction with an operand of saddr or saddrp, when an SFR is accessed with a value from FF00H to FF1FH being coded in saddr or saddrp, the number after the slash (/) is used as the number of states.

(iv) A number in parentheses in the state field of a string instruction indicates the number of states required when an interrupt request is accepted during execution of the string instruction.

10.4 NUMBERS OF BYTES AND STATES FOR INSTRUCTIONS WITH THE mem OPERAND

The number of bytes and the number of states required vary according to the mode coded in mem, as shown in Table 10-5.

Table 10-5 Numbers of Bytes and States for Instructions with the mem Operand

mem mode			Register indirect mode	Base index mode	Base mode	Index mode
Number of bytes			1 Note/2	2	3	4
Number of states (varies depending on the instruction)	MOV	A, mem	5	6	6	6
		mem, A				
	XCH	A, mem	7	8	8	8
		mem, A				
	ADD, ADDC, SUB, SUBC, AND, OR, XOR	A, mem	6	7	7	7
		mem, A	7	8	8	8
	CMP	A, mem	6	7	7	7
		mem, A				

Note Only when [DE], [HL], [DE+], [HL+], [DE−], or [HL−] is specified for mem in the MOV instruction, it is treated as a special 1-byte instruction.

10.5 THE MEANING OF SYMBOLS IN THE FLAG OPERATION FIELD

Table 10-6 Symbols in the Flag Operation Field

Symbol	Explanation
(Blank)	Left unchanged
0	Cleared to 0
1	Set to 1
×	Set or cleared according to the result
P	P/V flag operates as a parity flag.
V	P/V flag operates as an overflow flag.
U	Undefined
R	Previously saved value is restored.

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
8-bit data transfer instructions	MOV	r1, #byte	2	3	3	r1 ← byte						
		saddr, #byte	3	3/4	0	(saddr) ← byte						
		sfr, #byte Note	3	4	0	sfr ← byte						
		r, r1	2	3	3	r ← r1						
		A, r1	1	3	3	A ← r1						
		A, saddr	2	3/4	1	A ← (saddr)						
		saddr, A	2	3/4	0	(saddr) ← A						
		saddr, saddr	3	4/6	0	(saddr) ← (saddr)						
		A, sfr	2	4	1	A ← sfr						
		sfr, A	2	4	0	sfr ← A						
		A, mem	1 - 4	5 - 6	3 - 4	A ← (mem)						
		mem, A	1 - 4	5 - 6	2	(mem) ← A						
		A, [saddrp]	2	5/6	1	A ← ((saddrp))						
		[saddrp, A]	2	4/5	0	((saddrp)) ← A						
		A, !addr16	4	5	3	A ← (!addr16)						
		!addr16, A	4	4	2	(!addr16) ← A						
		PSWL, #byte	3	4	0	PSW_L ← byte	×	×	×	×	×	×
		PSWH, #byte	3	4	0	PSW_H ← byte						
		PSWL, A	2	4	0	PSW_L ← A	×	×	×	×	×	×
		PSWH, A	2	4	0	PSW_H ← A						
		A, PSWL	2	4	1	A ← PSW_L						
		A, PSWH	2	4	1	A ← PSW_H						
	XCH	A, r1	1	4	4	A ↔ r1						
		r, r1	2	4	4	r ↔ r1						
		A, mem	2 - 4	7 - 8	3 - 4	A ↔ (mem)						
		A, saddr	2	4/6	0	A ↔ (saddr)						
		A, sfr	3	8	3	A ↔ sfr						
		A, [saddrp]	2	6/7	0	A ↔ ((saddrp))						
		saddr, saddr	3	8/12	0	(saddr) ↔ (saddr)						

Note If STBC or WDM is coded in sfr, a different instruction having the different byte count and state count are generated.

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
16-bit data transfer instructions	MOVW	rp1, #word	3	3	3	rp1 ← word						
		saddrp, #word	4	3/4	0	(saddrp) ← word						
		sfrp, #word	4	4	0	sfrp ← word						
		rp, rp1	2	3	3	rp ← rp1						
		AX, saddrp	2	3/4	1	AX ← (saddrp)						
		saddrp, AX	2	3/4	0	(saddrp) ← AX						
		saddrp, saddrp	3	4/6	0	(saddrp) ← (saddrp)						
		AX, sfrp	2	4	1	AX ← sfrp						
		sfrp, AX	2	4	0	sfrp ← AX						
		rp1, !addr16	4	10	6	rp1 ← (addr16)						
		!addr16, rp1	4	8	4	(addr16) ← rp1						
	XCHW	AX, saddrp	2	4/6	0	AX ↔ (saddrp)						
		AX, sfrp	3	9	3	AX ↔ sfrp						
		saddrp, saddrp	3	8/12	0	(saddrp) ↔ (saddrp)						
		rp, rp1	2	5	5	rp ↔ rp1						
8-bit arithmetic/ logical instructions	ADD	A, #byte	2	3	3	A, CY ← A + byte	×	×	×	V	0	×
		saddr, #byte	3	5/7	0	(saddr), CY ← (saddr) + byte	×	×	×	V	0	×
		sfr, #byte	4	10	3	sfr, CY ← sfr + byte	×	×	×	V	0	×
		r, r1	2	3	3	r, CY ← r + r1	×	×	×	V	0	×
		A, saddr	2	3/4	1	A, CY ← A + (saddr)	×	×	×	V	0	×
		A, sfr	3	7	4	A, CY ← A + sfr	×	×	×	V	0	×
		saddr, saddr	3	6/9	0	(saddr), CY ← (saddr) + (saddr)	×	×	×	V	0	×
		A, mem	2 - 4	6 - 7	4 - 5	A, CY ← A + (mem)	×	×	×	V	0	×
		mem, A	2 - 4	7 - 8	2 - 3	(mem), CY ← (mem) + A	×	×	×	V	0	×
	ADDC	A, #byte	2	3	3	A, CY ← A + byte + CY	×	×	×	V	0	×
		saddr, #byte	3	5/7	0	(saddr), CY ← (saddr) + byte + CY	×	×	×	V	0	×
		sfr, #byte	4	10	3	sfr, CY ← sfr + byte + CY	×	×	×	V	0	×
		r, r1	2	3	3	r, CY ← r + r1 + CY	×	×	×	V	0	×
		A, saddr	2	3/4	1	A, CY ← A + (saddr) + CY	×	×	×	V	0	×
		A, sfr	3	7	4	A, CY ← A + sfr + CY	×	×	×	V	0	×
		saddr, saddr	3	6/9	0	(saddr), CY ← (saddr) + (saddr) + CY	×	×	×	V	0	×
		A, mem	2 - 4	6 - 7	4 - 5	A, CY ← A + (mem) + CY	×	×	×	V	0	×
		mem, A	2 - 4	7 - 8	2 - 3	(mem), CY ← (mem) + A + CY	×	×	×	V	0	×

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
8-bit arithmetic/ logical instructions	SUB	A, #byte	2	3	3	A, CY ← A – byte	×	×	×	V	1	×
		saddr, #byte	3	5/7	0	(saddr), CY ← (saddr) – byte	×	×	×	V	1	×
		sfr, #byte	4	10	3	sfr, CY ← sfr – byte	×	×	×	V	1	×
		r, r1	2	3	3	r, CY ← r – r1	×	×	×	V	1	×
		A, saddr	2	3/4	1	A, CY ← A – (saddr)	×	×	×	V	1	×
		A, sfr	3	7	4	A, CY ← A – sfr	×	×	×	V	1	×
		saddr, saddr	3	6/9	0	(saddr), CY ← (saddr) – (saddr)	×	×	×	V	1	×
		A, mem	2 - 4	6 - 7	4 - 5	A, CY ← A – (mem)	×	×	×	V	1	×
		mem, A	2 - 4	7 - 8	2 - 3	(mem), CY ← (mem) – A	×	×	×	V	1	×
	SUBC	A, #byte	2	3	3	A, CY ← A – byte – CY	×	×	×	V	1	×
		saddr, #byte	3	5/7	0	(saddr), CY ← (saddr) – byte – CY	×	×	×	V	1	×
		sfr, #byte	4	10	3	sfr, CY ← sfr – byte – CY	×	×	×	V	1	×
		r, r1	2	3	3	r, CY ← r – r1 – CY	×	×	×	V	1	×
		A, saddr	2	3/4	1	A, CY ← A – (saddr) – CY	×	×	×	V	1	×
		A, sfr	3	7	4	A, CY ← A – sfr – CY	×	×	×	V	1	×
		saddr, saddr	3	6/9	0	(saddr), CY ← (saddr) – (saddr) – CY	×	×	×	V	1	×
		A, mem	2 - 4	6 - 7	4 - 5	A, CY ← A – (mem) – CY	×	×	×	V	1	×
		mem, A	2 - 4	7 - 8	2 - 3	(mem), CY ← (mem) – A – CY	×	×	×	V	1	×
	AND	A, #byte	2	3	3	A ← A ∧ byte	×	×		P	0	
		saddr, #byte	3	5/7	0	(saddr) ← (saddr) ∧ byte	×	×		P	0	
		sfr, #byte	4	10	3	sfr ← sfr ∧ byte	×	×		P	0	
		r, r1	2	3	3	r ← r ∧ r1	×	×		P	0	
		A, saddr	2	3/4	1	A ← A ∧ (saddr)	×	×		P	0	
		A, sfr	3	7	4	A ← A ∧ sfr	×	×		P	0	
		saddr, saddr	3	6/9	0	(saddr) ← (saddr) ∧ (saddr)	×	×		P	0	
		A, mem	2 - 4	6 - 7	4 - 5	A ← A ∧ (mem)	×	×		P	0	
		mem, A	2 - 4	7 - 8	2 - 3	(mem) ← (mem) ∧ A	×	×		P	0	

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag					
							S	Z	AC	P/V	SUB	CY
8-bit arithmetic/ logical instructions	OR	A, #byte	2	3	3	A ← A ∨ byte	×	×		P	0	
		saddr, #byte	3	5/7	0	(saddr) ← (saddr) ∨ byte	×	×		P	0	
		sfr, #byte	4	10	3	sfr ← sfr ∨ byte	×	×		P	0	
		r, r1	2	3	3	r ← r ∨ r1	×	×		P	0	
		A, saddr	2	3/4	1	A ← A ∨ (saddr)	×	×		P	0	
		A, sfr	3	7	4	A ← A ∨ sfr	×	×		P	0	
		saddr, saddr	3	6/9	0	(saddr) ← (saddr) ∨ (saddr)	×	×		P	0	
		A, mem	2 - 4	6 - 7	4 - 5	A ← A ∨ (mem)	×	×		P	0	
		mem, A	2 - 4	7 - 8	2 - 3	(mem) ← (mem) ∨ A	×	×		P	0	
	XOR	A, #byte	2	3	3	A ← A ⊻ byte	×	×		P	0	
		saddr, #byte	3	5/7	0	(saddr) ← (saddr) ⊻ byte	×	×		P	0	
		sfr, #byte	4	10	3	sfr ← sfr ⊻ byte	×	×		P	0	
		r, r1	2	3	3	r ← r ⊻ r1	×	×		P	0	
		A, saddr	2	3/4	1	A ← A ⊻ (saddr)	×	×		P	0	
		A, sfr	3	7	4	A ← A ⊻ sfr	×	×		P	0	
		saddr, saddr	3	6/9	0	(saddr) ← (saddr) ⊻ (saddr)	×	×		P	0	
		A, mem	2 - 4	6 - 7	4 - 5	A ← A ⊻ (mem)	×	×		P	0	
		mem, A	2 - 4	7 - 8	2 - 3	(mem) ← (mem) ⊻ A	×	×		P	0	
	CMP	A, #byte	2	3	3	A – byte	×	×	×	V	1	×
		saddr, #byte	3	5/7	1	(saddr) – byte	×	×	×	V	1	×
		sfr, #byte	4	10	4	sfr – byte	×	×	×	V	1	×
		r, r1	2	3	3	r – r1	×	×	×	V	1	×
		A, saddr	2	3/4	1	A – (saddr)	×	×	×	V	1	×
		A, sfr	3	7	4	A – sfr	×	×	×	V	1	×
		saddr, saddr	3	6/8	1	(saddr) – (saddr)	×	×	×	V	1	×
		A, mem	2 - 4	6 - 7	4 - 5	A – (mem)	×	×	×	V	1	×
		mem, A	2 - 4	6 - 7	3 - 4	(mem) – A	×	×	×	V	1	×

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag					
							S	Z	AC	P/V	SUB	CY
16-bit arithmetic/ logical instructions	ADDW	AX, #word	3	4	4	AX, CY ← AX + word	×	×	×	V	0	×
		saddrp, #word	4	5/7	0	(saddrp), CY ← (saddrp) + word	×	×	×	V	0	×
		sfrp, #word	5	10	3	sfrp, CY ← sfrp + word	×	×	×	V	0	×
		rp, rp1	2	4	4	rp, CY ← rp + rp1	×	×	×	V	0	×
		AX, saddrp	2	4/5	2	AX, CY ← AX + (saddrp)	×	×	×	V	0	×
		AX, sfrp	3	8	5	AX, CY ← AX + sfrp	×	×	×	V	0	×
		saddrp, saddrp	3	6/9	0	(saddrp), CY ← (saddrp) + (saddrp)	×	×	×	V	0	×
	SUBW	AX, #word	3	4	3	AX, CY ← AX – word	×	×	×	V	1	×
		saddrp, #word	4	5/7	0	(saddrp), CY ← (saddrp) – word	×	×	×	V	1	×
		sfrp, #word	5	10	3	sfrp, CY ← sfrp – word	×	×	×	V	1	×
		rp, rp1	2	4	4	rp, CY ← rp – rp1	×	×	×	V	1	×
		AX, saddrp	2	4/5	2	AX, CY ← AX – (saddrp)	×	×	×	V	1	×
		AX, sfrp	3	8	5	AX, CY ← AX – sfrp	×	×	×	V	1	×
		saddrp, saddrp	3	6/9	0	(saddrp), CY ← (saddrp) – (saddrp)	×	×	×	V	1	×
	CMPW	AX, #word	3	4	3	AX – word	×	×	×	V	1	×
		saddrp, #word	4	4/5	1	(saddrp) – word	×	×	×	V	1	×
		sfrp, #word	5	8	4	sfrp – word	×	×	×	V	1	×
		rp, rp1	2	4	4	rp – rp1	×	×	×	V	1	×
		AX, saddrp	2	4/5	1	AX – (saddrp)	×	×	×	V	1	×
		AX, sfrp	3	8	4	AX – sfrp	×	×	×	V	1	×
		saddrp, saddrp	3	5/7	1	(saddrp) – (saddrp)	×	×	×	V	1	×
Multiply/ divide instructions	MULU	r1	2	18	18	AX ← A × r1						
	DIVUW	r1	2	26	26	AX (quotient), r1 (remainder), ← AX ÷ r1						
	MULUW	rp1	2	27	27	AX (16 high-order bits), rp1 (16 low-order bits) ← AX × rp1						
	DIVUX	rp1	2	50	50	AXDE (quotient), rp1 (remainder) ← AXDE ÷ rp1						

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
Increment/ decrement instructions	INC	r1	1	3	3	r1 ← r1 + 1	×	×	×	V	0	
		saddr	2	4/6	0	(saddr) ← (saddr) + 1	×	×	×	V	0	
	DEC	r1	1	3	3	r1 ← r1 − 1	×	×	×	V	1	
		saddr	2	4/6	0	(saddr) ← (saddr) − 1	×	×	×	V	1	
	INCW	rp2	1	3	3	rp2 ← rp2 + 1						
		saddrp	3	6/8	2	(saddrp) ← (saddrp) + 1						
	DECW	rp2	1	3	3	rp2 ← rp2 − 1						
		saddrp	3	6/8	2	(saddrp) ← (saddrp) − 1						
Shift/ rotate instructions	ROR	r1, n	2	4 + 3n	4 + 3n	(CY, $r1_7$ ← $r1_0$, $r1_{m-1}$ ← $r1_m$) × n times				P	0	×
	ROL	r1, n	2	4 + 3n	4 + 3n	(CY, $r1_0$ ← $r1_7$, $r1_{m+1}$ ← $r1_m$) × n times				P	0	×
	RORC	r1, n	2	4 + 3n	4 + 3n	(CY ← $r1_0$, $r1_7$ ← CY, $r1_{m-1}$ ← $r1_m$) × n times				P	0	×
	ROLC	r1, n	2	4 + 3n	4 + 3n	(CY ← $r1_7$, $r1_0$ ← CY, $r1_{m+1}$ ← $r1_m$) × n times				P	0	×
	SHR	r1, n	2	4 + 3n	4 + 3n	(CY ← $r1_0$, $r1_7$ ← 0, $r1_{m-1}$ ← $r1_m$) × n times	×	×	0	P	0	×
	SHL	r1, n	2	4 + 3n	4 + 3n	(CY ← $r1_7$, $r1_0$ ← 0, $r1_{m+1}$ ← $r1_m$) × n times	×	×	0	P	0	×
	SHRW	rp1, n	2	4 + 3n	4 + 3n	(CY ← $rp1_0$, $rp1_{15}$ ← 0, $rp1_{m-1}$ ← $rp1_m$) × n times	×	×	0	P	0	×
	SHLW	rp1, n	2	4 + 3n	4 + 3n	(CY ← $rp1_{15}$, $rp1_0$ ← 0, $rp1_{m+1}$ ← $rp1_m$) × n times	×	×	0	P	0	×
	ROR4	[rp1]	2	7	3	A_{3-0} ← $(rp1)_{3-0}$, $(rp1)_{7-4}$ ← A_{3-0}, $(rp1)_{3-0}$ ← $(rp1)_{7-4}$						
	ROL4	[rp1]	2	7	3	A_{3-0} ← $(rp1)_{7-4}$, $(rp1)_{3-0}$ ← A_{3-0}, $(rp1)_{7-4}$ ← $(rp1)_{3-0}$,						
BCD correction instruction	ADJ4		1	3	3	Decimal adjust accumulator	×	×	×	P		×

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
Bit manipulation instructions	MOV1	CY, saddr.bit	3	6/7	4	CY ← (saddr.bit)						×
		CY, sfr.bit	3	7	4	CY ← sfr.bit						×
		CY, A.bit	2	6	6	CY ← A.bit						×
		CY, X.bit	2	6	6	CY ← X.bit						×
		CY, PSWH.bit	2	6	6	CY ← PSW_H.bit						×
		CY, PSWL.bit	2	6	6	CY ← PSW_L.bit						×
		saddr.bit, CY	3	7/8	3	(saddr.bit) ← CY						
		sfr.bit, CY	3	8	3	sfr.bit ← CY						
		A.bit, CY	2	8	8	A.bit ← CY						
		X.bit, CY	2	8	8	X.bit ← CY						
		PSWH.bit, CY	2	9	9	PSW_H.bit ← CY						
		PSWL.bit, CY	2	9	9	PSW_L.bit ← CY	×	×	×	×	×	
	AND1	CY, saddr.bit	3	6/7	4	CY ← CY ∧ (saddr.bit)						×
		CY, /saddr.bit	3	6/7	4	CY ← CY ∧ $\overline{(saddr.bit)}$						×
		CY, sfr.bit	3	7	4	CY ← CY ∧ sfr.bit						×
		CY, /sfr.bit	3	7	4	CY ← CY ∧ $\overline{sfr.bit}$						×
		CY, A.bit	2	6	6	CY ← CY ∧ A.bit						×
		CY, /A.bit	2	6	6	CY ← CY ∧ $\overline{A.bit}$						×
		CY, X.bit	2	6	6	CY ← CY ∧ X.bit						×
		CY, /X.bit	2	6	6	CY ← CY ∧ $\overline{X.bit}$						×
		CY, PSWH.bit	2	6	6	CY ← CY ∧ PSW_H.bit						×
		CY, /PSWH.bit	2	6	6	CY ← CY ∧ $\overline{PSW_H.bit}$						×
		CY, PSWL.bit	2	6	6	CY ← CY ∧ PSW_L.bit						×
		CY, /PSWL.bit	2	6	6	CY ← CY ∧ $\overline{PSW_L.bit}$						×
	OR1	CY, saddr.bit	3	6/7	4	CY ← CY ∨ (saddr.bit)						×
		CY, /saddr.bit	3	6/7	4	CY ← CY ∨ $\overline{(saddr.bit)}$						×
		CY, sfr.bit	3	7	4	CY ← CY ∨ sfr.bit						×
		CY, /sfr.bit	3	7	4	CY ← CY ∨ $\overline{sfr.bit}$						×
		CY, A.bit	2	6	6	CY ← CY ∨ A.bit						×
		CY, /A.bit	2	6	6	CY ← CY ∨ $\overline{A.bit}$						×
		CY, X.bit	2	6	6	CY ← CY ∨ X.bit						×
		CY, /X.bit	2	6	6	CY ← CY ∨ $\overline{X.bit}$						×
		CY, PSWH.bit	2	6	6	CY ← CY ∨ PSW_H.bit						×
		CY, /PSWH.bit	2	6	6	CY ← CY ∨ $\overline{PSW_H.bit}$						×
		CY, PSWL.bit	2	6	6	CY ← CY ∨ PSW_L.bit						×
		CY, /PSWL.bit	2	6	6	CY ← CY ∨ $\overline{PSW_L.bit}$						×

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
Bit manipulation instructions	XOR1	CY, saddr.bit	3	6/7	4	CY ← CY ∀ (saddr.bit)						×
		CY, sfr.bit	3	7	4	CY ← CY ∀ sfr.bit						×
		CY, A.bit	2	6	6	CY ← CY ∀ A.bit						×
		CY, X.bit	2	6	6	CY ← CY ∀ X.bit						×
		CY, PSWH.bit	2	6	6	CY ← CY ∀ PSW_H.bit						×
		CY, PSWL.bit	2	6	6	CY ← CY ∀ PSW_L.bit						×
	SET1	saddr.bit	2	5/7	1	(saddr.bit) ← 1						
		sfr.bit	3	8	2	sfr.bit ← 1						
		A.bit	2	7	7	A.bit ← 1						
		X.bit	2	7	7	X.bit ← 1						
		PSWH.bit	2	8	8	PSW_H.bit ← 1						
		PSWL.bit	2	8	8	PSW_L.bit ← 1	×	×	×	×	×	×
	CLR1	saddr.bit	2	5/7	1	(saddr.bit) ← 0						
		sfr.bit	3	8	2	sfr.bit ← 0						
		A.bit	2	7	7	A.bit ← 0						
		X.bit	2	7	7	X.bit ← 0						
		PSWH.bit	2	8	8	PSW_H.bit ← 0						
		PSWL.bit	2	8	8	PSW_L.bit ← 0	×	×	×	×	×	×
	NOT1	saddr.bit	3	6/8	2	(saddr.bit) ← $\overline{\text{(saddr.bit)}}$						
		sfr.bit	3	8	2	sfr.bit ← $\overline{\text{sfr.bit}}$						
		A.bit	2	7	7	A bit ← $\overline{\text{A.bit}}$						
		X.bit	2	7	7	X.bit ← $\overline{\text{X.bit}}$						
		PSWH.bit	2	8	8	PSW_H.bit ← $\overline{PSW_H.bit}$						
		PSWL.bit	2	8	8	PSW_L.bit ← $\overline{PSW_L.bit}$	×	×	×	×	×	×
	SET1	CY	1	3	3	CY ← 1						1
	CLR1	CY	1	3	3	CY ← 0						0
	NOT1	CY	1	3	3	CY ← $\overline{CY}$						×

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
Call/ return instructions	CALL	!addr16	3	8	0	(SP – 1) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← !addr16, SP ← SP – 2						
	CALLF	!addr11	2	8	0	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15-11 ← 00001, PC10-0 ← !addr11, SP ← SP – 2						
	CALLT	[addr5]	1	13	0	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (TPF, 00000000, addr5 + 1). PCL ← (TPF, 00000000, addr5), SP ← SP – 2						
	CALL	rp1	2	9	0	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PCH ← rp1H, PCL ← rp1L, SP ← SP – 2						
		[rp1]	2	11	0	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PCH ← (rp1 + 1), PCL ← (rp1), SP ← SP – 2						
	BRK		1	20	0	(SP – 1) ← PSWH, (SP – 2) ← PSWL, (SP – 3) ← (PC + 1)H, (SP – 4) ← (PC + 1)L, PCL ← (003EH), PCH ← (003FH), SP ← SP – 4, IE ← 0						
	RET		1	8	0	PCL ← (SP), PCH ← (SP + 1), SP + 2						
	RETI		1	14	0	PCL ← (SP), PCH ← (SP + 1), PSWL ← (SP + 2), PSWH ← (SP + 3), SP ← SP + 4, EOS ← 0	R	R	R	R	R	R
Stack manipulation instructions	PUSH	post	2	41 + 4n	41	{(SP – 1) ← postH, (SP – 2) ← postL, SP ← SP – 2} × n times Note						
		PSW	1	5	1	(SP – 1) ← PSWH, (SP – 2) ← PSWL, SP ← SP – 2						
	PUSHU	post	2	42 + 4n	42	{(UP – 1) ← postH, (UP – 2) ← postL, UP ← UP – 2} × n times Note						
	POP	post	2	41 + 5n	41 + n	{postL ← (SP), postH ← (SP + 1), SP ← SP + 2} × n times Note						
		PSW	1	6	2	PSWL ← (SP), PSWH ← (SP + 1), SP ← SP + 2	R	R	R	R	R	R
	POPU	post	2	42 + 5n	42 + n	{postL ← (UP), postH ← (UP + 1), UP ← UP + 2} × n times Note						
	MOVW	SP, #word	4	4	0	SP ← word						
		SP, AX	2	4	0	SP ← AX						
		AX, SP	2	4	1	AX ← SP						
	INCW	SP	2	5	5	SP ← SP + 1						
	DECW	SP	2	5	5	SP ← SP – 1						

Note n indicates the number of registers specified in post.

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S Z AC P/V SUB CY
Unconditional branch instructions	BR	!addr16	3	4	0	PC ← !addr16	
		rp1	2	5	0	$PC_H \leftarrow rp1_N$, $PC_L \leftarrow rp1_L$	
		[rp1]	2	8	0	$PC_H \leftarrow (rp1 + 1)$, $PC_L \leftarrow (rp1)$	
		$addr16	2	7	0	PC ← $addr16	
Conditional branch instructions	BC	$addr16	2	7(3)	0(3)	PC ← $addr16 if CY = 1	
	BL						
	BNC	$addr16	2	7(3)	0(3)	PC ← $addr16 if CY = 0	
	BNL						
	BZ	$addr16	2	7(3)	0(3)	PC ← $addr16 if Z = 1	
	BE						
	BNZ	$addr16	2	7(3)	0(3)	PC ← $addr16 if Z = 0	
	BNE						
	BV	$addr16	2	7(3)	0(3)	PC ← $addr16 if P/V = 1	
	BPE						
	BNV	$addr16	2	7(3)	0(3)	PC ← $addr16 if P/V = 0	
	BPO						
	BN	$addr16	2	7(3)	0(3)	PC ← $addr16 if S = 1	
	BP	$addr16	2	7(3)	0(3)	PC ← $addr16 if S = 0	
	BGT	$addr16	3	9(5)	0(5)	PC ← $addr16 if $(P/V \veebar S) \vee Z = 0$	
	BGE	$addr16	3	9(5)	0(5)	PC ← $addr16 if $P/V \veebar S = 0$	
	BLT	$addr16	3	9(5)	0(5)	PC ← $addr16 if $P/V \veebar S = 1$	
	BLE	$addr16	3	9(5)	0(5)	PC ← $addr16 if $(P/V \veebar S) \vee Z = 1$	
	BH	$addr16	3	9(5)	0(5)	PC ← $addr16 if $Z \vee CY = 0$	
	BNH	$addr16	3	9(5)	0(5)	PC ← $addr16 if $Z \vee CY = 1$	
	BT	saddr.bit, $addr16	3	9(6)/ 10(7)	0(4)	PC ← $addr16 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	11(8)	0(5)	PC ← $addr16 if sfr.bit = 1	
		A.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if A.bit = 1	
		X.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if X.bit = 1	
		PSWH.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if PSW_H.bit = 1	
		PSWL.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if PSW_L.bit = 1	
	BF	saddr.bit, $addr16	4	10(7)/ 11(8)	0(5)	PC ← $addr16 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	11(8)	0(5)	PC ← $addr16 if sfr.bit = 0	
		A.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if A.bit = 0	
		X.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if X.bit = 0	
		PSWH.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if PSW_H.bit = 0	
		PSWL.bit, $addr16	3	10(7)	0(7)	PC ← $addr16 if PSW_L.bit = 0	

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
Conditional branch instructions	BTCLR	saddr.bit, $addr16	4	12(7)/ 14(8)	0(5)	PC ← $addr16 if (saddr.bit) = 1 then reset (saddr.bit)						
		sfr.bit, $addr16	4	14(8)	0(5)	PC ← $addr16 if sfr.bit = 1 then reset sfr.bit						
		A.bit, $addr16	3	11(7)	0(7)	PC ← $addr16 if A.bit = 1 then reset A.bit						
		X.bit, $addr16	3	11(7)	0(7)	PC ← $addr16 if X.bit = 1 then reset X.bit						
		PSWH.bit, $addr16	3	12(7)	0(7)	PC ← $addr16 if PSW_H.bit = 1 then reset PSW_H.bit						
		PSWL.bit, $addr16	3	12(7)	0(7)	PC ← $addr16 if PSW_L.bit = 1 then reset PSW_L.bit	×	×	×	×	×	×
	BFSET	saddr.bit, $addr16	4	12(7)/ 14(8)	0(5)	PC ← $addr16 if (saddr.bit) = 0 then set (saddr.bit)						
		sfr.bit, $addr16	4	14(8)	0(5)	PC ← $addr16 if sfr.bit = 0 then set sfr.bit						
		A.bit, $addr16	3	11(7)	0(7)	PC ← $addr16 if A.bit = 0 then set A.bit						
		X.bit, $addr16	3	11(7)	0(7)	PC ← $addr16 if X.bit = 0 then set X.bit						
		PSWH.bit, $addr16	3	12(7)	0(7)	PC ← $addr16 if PSW_H.bit = 0 then set PSW_H.bit						
		PSWL.bit, $addr16	3	12(7)	0(7)	PC ← $addr16 if PSW_L.bit = 0 then set PSW_L.bit	×	×	×	×	×	×
	DBNZ	r2, $addr16	2	8(5)	0(5)	r2 ← r2 − 1, then PC ← $addr16 if r2 ≠ 0						
		saddr, $addr16	3	9(6)/ 11(8)	0(2)	(saddr) ← (saddr) − 1, then PC ← $addr16 if (saddr) ≠ 0						
Context switching instructions	BRKCS	RBn	2	12	0	PC_H ← R5, PC_L ↔ R4, R7 ← PSW_H, R6 ← PSW_L, RBS2 − 0 ← n, RSS ← 0, IE ← 0						
	RETCS	!addr16	3	6	0	PC_H ← R5, PC_L ← R4, R5, R4 ← !addr16, PSW_H ← R7, PSW_L ← R6, EOS ← 0	R	R	R	R	R	R

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag S	Z	AC	P/V	SUB	CY
String instructions	MOVM	[DE+], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE+) ← A, C ← C – 1 End if C = 0						
		[DE–], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE–) ← A, C ← C – 1 End if C = 0						
	MOVBK	[DE+], [HL+]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE+) ← (HL+), C ← C – 1 End if C = 0						
		[DE–], [HL–]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE–) ← (HL–), C ← C – 1 End if C = 0						
	XCHM	[DE+], A	2	2 + 12n (4 + 12n)	2 + 6n (3 + 6n)	(DE+) ↔ A, C ← C – 1 End if C = 0						
		[DE–], A	2	2 + 12n (4 + 12n)	2 + 6n (3 + 6n)	(DE–) ↔ A, C ← C – 1 End if C = 0						
	XCHBK	[DE+], [HL+]	2	2 + 15n (4 + 15n)	2 + 7n (3 + 7n)	(DE+) ↔ (HL+), C ← C – 1 End if C = 0						
		[DE–], [HL–]	2	2 + 15n (4 + 15n)	2 + 7n (3 + 7n)	(DE–) ↔ (HL–), C ← C – 1 End if C = 0						
	CMPME	[DE+], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE+) – A, C ← C – 1 End if C = 0 or Z = 0	×	×	×	V	1	×
		[DE–], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE–) – A, C ← C – 1 End if C = 0 or Z = 0	×	×	×	V	1	×
	CMPBKE	[DE+], [HL+]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE+) – (HL+), C ← C – 1 End if C = 0 or Z = 0	×	×	×	V	1	×
		[DE–], [HL–]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE–) – (HL–), C ← C – 1 End if C = 0 or Z = 0	×	×	×	V	1	×
	CMPMNE	[DE+], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE+) – A, C ← C – 1 End if C = 0 or Z = 1	×	×	×	V	1	×
		[DE–], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE–) – A, C ← C – 1 End if C = 0 or Z = 1	×	×	×	V	1	×
	CMPBKNE	[DE+], [HL+]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE+) – (HL+), C ← C – 1 End if C = 0 or Z = 1	×	×	×	V	1	×
		[DE–], [HL–]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE–) – (HL–), C ← C – 1 End if C = 0 or Z = 1	×	×	×	V	1	×
	CMPMC	[DE+], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE+) – A, C ← C – 1 End if C = 0 or CY = 0	×	×	×	V	1	×
		[DE–], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE–) – A, C ← C – 1 End if C = 0 or CY = 0	×	×	×	V	1	×
	CMPBKC	[DE+], [HL+]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE+) – (HL+), C ← C – 1 End if C = 0 or CY = 0	×	×	×	V	1	×
		[DE–], [HL–]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE–) – (HL–), C ← C – 1 End if C = 0 or CY = 0	×	×	×	V	1	×

Instruction set	Mnemonic	Operand	Byte	State	Free state	Operation	Flag					
							S	Z	AC	P/V	SUB	CY
String instructions	CMPMNC	[DE+], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE+) – A, C ← C – 1 End if C = 0 or CY = 1	×	×	×	V	1	×
		[DE–], A	2	2 + 7n (4 + 7n)	2 + 5n (3 + 5n)	(DE–) – A, C ← C – 1 End if C = 0 or CY = 1	×	×	×	V	1	×
	CMPBKNC	[DE+], [HL+]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE+) – (HL+), C ← C – 1 End if C = 0 or CY = 1	×	×	×	V	1	×
		[DE–], [HL–]	2	2 + 10n (4 + 10n)	2 + 6n (3 + 6n)	(DE–) – (HL–), C ← C – 1 End if C = 0 or CY = 1	×	×	×	V	1	×
CPU control instructions	MOV	STBC, #byte	4	6	1	STBC ← byte						
		WDM, #byte	4	6	1	WDM ← byte						
	SWRS		1	3	3	RSS ← $\overline{\text{RSS}}$						
	SEL	RBn	2	4	4	RBS2 – 0 ← n, RSS ← 0						
		RBn, ALT	2	4	4	RBS2 – 0 ← n, RSS ← 1						
	NOP		1	3	3	No Operation						
	EI		1	3	3	IE ← 1 (Enable interrupt)						
	DI		1	3	3	IE ← 0 (Disable interrupt)						

11. ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS (T_a = 25 °C)

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{DD}		−0.5 to +7.0	V
	AV_{REF}		−0.5 to V_{DD} + 0.3	V
	AV_{SS}		−0.5 to +0.5	V
Input voltage	V_I		−0.5 to +V_{DD} + 0.5	V
Output voltage	V_O		−0.5 to +V_{DD} + 0.5	V
Low-level output current	I_{OL}	1 pin	4.0	mA
		Total of all output pins	100	mA
High-level output current	I_{OH}	1 pin	−2	mA
		Total of all output pins	−25	mA
Operating temperature	T_{opt}		−40 to +85	°C
Storage temperature	T_{stg}		−65 to +150	°C

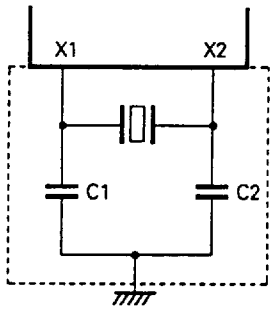
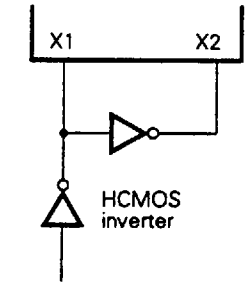
RECOMMENDED OPERATING CONDITIONS

Parameter / Oscillator frequency	T_a	V_{DD}
4 MHz ≤ fxx ≤ 12 MHz	−40 to +85 °C	+5.0 V ±10 %

CAPACITANCE (T_a = 25 °C, V_{DD} = Vss = 0 V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input capacitance	C_I	f = 1 MHz 0 V on pins other than measured pins			10	pF
Output capacitance	C_O				20	pF
I/O capacitance	C_{IO}				20	pF

OSCILLATOR CHARACTERISTICS (T_a = −40 to +85 °C, V_{DD} = +5.0 V ±10 %, V_{SS} = AV_{SS} = 0 V, 4.0 V ≤ AV_{REF} ≤ V_{DD})

Oscillator	Recommended circuit	Parameter	Min.	Typ.	Max.	Unit
Ceramic or crystal resonator		Oscillator frequency (f_{XX})	4		12	MHz
External clock		X1 input frequency (f_X)	4		12	MHz
		X1 input rising and falling times (t_{XR}, t_{XF})	0		30	ns
		X1 input high and low level widths (t_{WXH}, t_{WXL})	30		130	ns

Cautions 1. The oscillating circuit should be placed as close to the X1 and X2 pins as possible.

2. On a printed circuit board, do not draw other signal lines in the area corresponding to the circuit enclosed in broken lines.

★ **Recommended Capacitors in an Oscillation Circuit**

Ceramic resonator

Manufacturer	Part number	Frequency [MHz]	Capacitance of externally-connected capacitors [pF]	
			C1	C2
Murata Mfg.	CSA8.00MT CSA10.0MT CSA12.0MT	8.0 10.0 12.0	30	30
	CST8.00MT CST10.0MT CST12.0MT	8.0 10.0 12.0	Contained	Contained
Kyocera	KBR-8.0M KBR-10.0M KBR-12.0M	8.0 10.0 12.0	33	33
TDK	FCR10.0MC FCR12.0MC	10.0 12.0	Contained	Contained

Crystal resonator

Manufacturer	Part number	Frequency [MHz]	Capacitance of externally-connected capacitors [pF]	
			C1	C2
Kinseki	HC-49U	8.0 10.0 12.0	22	22

DC CHARACTERISTICS (T_a = −40 to +85 °C, V_{DD} = +5.0 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Low-level input voltage	V_{IL1}	Other than $\overline{EA}$		0		0.8	V
	V_{IL2}	$\overline{EA}$		0		0.5	V
High-level input voltage	V_{IH1}	Other than P20/NM1, X1, X2, and $\overline{RESET}$		2.2		V_{DD}	V
	V_{IH2}	P20/NM1, X1, X2, and $\overline{RESET}$		3.8		V_{DD}	V
Low-level output voltage	V_{OL}	I_{OL} = 2.0 mA				0.45	V
High-level output voltage	V_{OH}	I_{OH} = −1.0 mA		V_{DD} − 1			V
Input current	I_{LI}	P20/NMI, $\overline{RESET}$ 0.45 V < V_I < V_{DD}				±10	μA
Input leakage current	I_{LI}					±10	μA
I/O leakage current	I_{LO}					±10	μA
AV_{REF} current	AI_{REF}	f_{CLK} = 6 MHz			1.5	5	mA
V_{DD} supply current	I_{DD1}	Operating mode, f_{CLK} = 6 MHz			30	60	mA
	I_{DD2}	HALT mode, f_{CLK} = 6 MHz			5	15	mA
Data retention voltage	V_{DDDR}	STOP mode		2.5			V
Data retention current	I_{DDDR}	STOP mode	V_{DDDR} = 2.5 V		3	30	μA
			V_{DDDR} = 5.0 V ±10 %		10	100	μA

AC CHARACTERISTICS

READ/WRITE OPERATION (T_a = −40 to +85 °C, V_{DD} = +5.0 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Max.	Unit
Internal system clock cycle type Note 1	t_{CYK}		166	1000	ns
Address setup time (to ALE↓)	t_{SAL}		150		ns
Address hold time (to ALE↓)	t_{HLA}	C_L = 100 pF, R_L = 2 kΩ Note 4	30		ns
Delay from address to $\overline{RD}$↓	t_{DAR}		233		ns
Delay from $\overline{RD}$↓ to address float	t_{FRA}			0	ns
Delay from address to data input	t_{DAID}			413	ns
Delay from ALE↓ to data input	t_{DLID}			233	ns
Delay from $\overline{RD}$↓ to data input	t_{DRID}			180	ns
Delay from ALE↓ to $\overline{RD}$↓	t_{DLR}		63		ns
Data hold time (to $\overline{RD}$↑)	t_{HRID}		0		ns
Delay from $\overline{RD}$↑ to address active	t_{DRA}		53		ns
Delay from $\overline{RD}$↑ to ALE↑	t_{DRL}		116		ns
$\overline{RD}$ low-level width	t_{WRL}		200		ns
ALE high-level width	t_{WLH}		126		ns
Delay from address to $\overline{WR}$↓	t_{DAW}		233		ns
Delay from ALE↓ to data output	t_{DLOD}			193	ns
Delay from $\overline{WR}$↓ to data output	t_{DWOD}			100	ns
Delay from ALE↓ to $\overline{WR}$↓ Note 2	t_{DLW}		63		ns
		Refresh mode	116		ns
Data setup time (to $\overline{WR}$↑)	t_{SODWR}		150		ns
Data setup time (to $\overline{WR}$↓) Note 3	t_{SODWF}	Refresh mode	33		ns
Data hold time (to $\overline{WR}$↑)	t_{HWOD}		20		ns
Delay from $\overline{WR}$↑ to ALE↑	t_{DWL}		116		ns
$\overline{WR}$ low-level width	t_{WWL}		200		ns
		Refresh mode	116		ns

Notes 1. The internal system clock (f_{CLK}) is obtained by dividing the oscillator clock (f_{XX}) by 2 or 8 according to the STBC register specification. In the above table, f_{XX} = 12 MHz and f_{CLK} = f_{XX}/2 are assumed.

2. When pulses are refreshed, the $\overline{WR}$ signal goes low a half clock later than the ALE signal, so t_{DLW} is the value in the lower row in the table.
3. When access is made to a pseudo-static RAM that takes data on the falling edge of the $\overline{WR}$ signal, the data setup time is t_{SODWF}, instead of t_{SODWR}.
4. The hold time includes the times to maintain V_{OH} and V_{OL} under load conditions C_L = 100 pF and R_L = 2 kΩ.

Remark The AC characteristics in the above table apply when the number of wait cycles is zero.

SERIAL OPERATION (T_a = −40 to +85 °C, V_{DD} = +5.0 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions			Min.	Max.	Unit
Serial clock cycle time	t_{CYSK}	Output	$\overline{SCK}$	Note 1	1.33		μs
			$\overline{CTS}$	Note 2	1.33		μs
		Input	$\overline{CTS}$	Note 3	1		μs
Serial clock low-level width	t_{WSKL}	Output	$\overline{SCK}$	Note 1	580		ns
			$\overline{CTS}$	Note 2	580		ns
		Input	$\overline{CTS}$	Note 3	420		ns
Serial clock high-level width	t_{WSKH}	Output	$\overline{SCK}$	Note 1	580		ns
			$\overline{CTS}$	Note 2	580		ns
		Input	$\overline{CTS}$	Note 3	420		ns
$\overline{CTS}$ high/low level width	t_{WCSH}, t_{WCSL}	Note 4			3		t_{CYK}
RxD setup time (to $\overline{CTS}$↑)	t_{SRXSK}				80		ns
RxD hold time (to $\overline{CTS}$↑)	t_{HSKRX}				80		ns
Delay from $\overline{SCK}$↓ to TxD	t_{DSKTX}					210	ns

Notes 1. When a send operation is performed in the I/O interface mode at a data transfer rate of 750 kbps

2. When a receive operation is performed in the I/O interface mode at a data transfer rate of 750 kbps

3. When a receive operation is performed in the I/O interface mode at a data transfer rate of 1 Mbps

4. In the asynchronous mode

A/D CONVERTER CHARACTERISTICS (T_a = −40 to +85 °C, V_{DD} = +5 V ±10 %, 4.0 V ≤ AV_{REF} ≤ V_{DD}, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution			8			bit
Total error Note					0.4	%
Quantization error					±1/2	LSB
Conversion time	t_{CONV}	166 ns ≤ t_{CYK} ≤ 250 ns	180			t_{CYK}
		250 ns ≤ t_{CYK} ≤ 500 ns	120			t_{CYK}
Sampling time	t_{SAMP}	166 ns ≤ t_{CYK} ≤ 250 ns	36			t_{CYK}
		250 ns ≤ t_{CYK} ≤ 500 ns	24			t_{CYK}
Analog input voltage	V_{IAN}		0		AV_{REF}	V
Analog input impedance	R_{AN}			1000		MΩ
Reference voltage	AV_{REF}		4.0		V_{DD}	V
AV_{REF} current	AI_{REF}	f_{CLK} = 6 MHz		1.5	5.0	mA

Note Quantization error is excluded. It is represented in percent with respect to a full-scale value.

COUNT UNIT OPERATION (T_a = −40 to +85 °C, V_{DD} = +5.0 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Max.	Unit
CI0/CI1 high/low level width	t_{WCIH}, t_{WCIL}		3		t_{CYK}
CTRL0/CTRL1 high/low level width	t_{WCTH}, t_{WCTL}		3		t_{CYK}
CTRL0/CTRL1 setup time (to CI↑)	t_{SCTCI}	The count unit operation mode is set to mode 3, and the rising edge on the CI pin input is validated.	2		t_{CYK}
CTRL0/CTRL1 hold time (to CI↑)	t_{HCICT}	The count unit operation mode is set to mode 3, and the rising edge on the CI pin input is validated.	5		t_{CYK}
CLR0/CLR1 high/low level width	t_{WCRH}, t_{WCRL}		3		t_{CYK}
CI0/CI1 setup time (to CI↑)	t_{S4CTCI}	The count unit operation mode is set to mode 4.	6		t_{CYK}
CI0/CI1 hold time (to CI↑)	t_{H4CICT}	The count unit operation mode is set to mode 4.	6		t_{CYK}
CI0/CI1, CTRL0/CTRL1 cycle time	t_{CYC4}	The count unit operation mode is set to mode 4.	4		μs

OTHER OPERATIONS (T_a = −40 to +85 °C, V_{DD} = +5.0 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Max.	Unit
NMI high/low level width	t_{WNIH}, t_{WNIL}		10		μs
INTE0 high/low level width	t_{WI0H}, t_{WI0L}		3		t_{CYK}
INTE1 high/low level width	t_{WI1H}, t_{WI1L}		3		t_{CYK}
INTE2 high/low level width	t_{WI2H}, t_{WI2L}		3		t_{CYK}
$\overline{\text{RESET}}$ high/low level width	t_{WRSH}, t_{WRSL}		10		μs
V_{DD} rising time (when the SBF bit is used)	t_{R0VD}		4		ms
V_{DD} rising/falling time	t_{RVD}, t_{FVD}		200		μs

EXTERNAL CLOCK TIMING (T_a = −40 to +85 °C, V_{DD} = +5.0 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Max.	Unit
X1 input high-level width	t_{WXH}		30	130	ns
X1 input low-level width	t_{WXL}		30	130	ns
X1 input rising time	t_{XR}		0	30	ns
X1 input falling time	t_{XF}		0	30	ns
X1 input cycle time	t_{CYX}		83	250	ns

t_{CYK}-DEPENDENT BUS TIMING DEFINITION

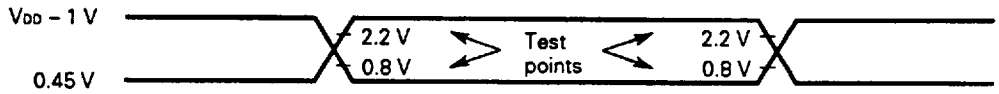
Parameter	Formula		Min./Max.	Unit
t_{SAL}	1.5T − 100		Min.	ns
t_{DAR}	2T − 100		Min.	ns
t_{DAID}	(3.5 + n)T − 170		Max.	ns
t_{DLID}	(2 + n)T − 100		Max.	ns
t_{DRID}	(1.5 + n)T − 70		Max.	ns
t_{DLR}	0.5T − 20		Min.	ns
t_{DRL}	T − 50		Min.	ns
t_{DRA}	0.5T − 30		Min.	ns
t_{WRL}	(1.5 + n)T − 50		Min.	ns
t_{WLH}	T − 40		Min.	ns
t_{DAW}	2T − 100		Min.	ns
t_{DLOD}	0.5T + 110		Max.	ns
t_{DLW}	0.5T − 20	(normal operation)	Min.	ns
	T − 50	(refresh mode)	Min.	ns
t_{SODWR}	(1.5 + n)T − 100		Min.	ns
t_{SODWF}	0.5T − 50		Min.	ns
t_{DWL}	T − 50		Min.	ns
t_{WWL}	(1.5 + n)T − 50	(normal operation)	Min.	ns
	(1 + n)T − 50	(refresh mode)	Min.	ns

Remarks 1. n represents the number of wait cycles to be inserted according to the MM register specification.

2. T = t_{CYK} = 1/f_{CLK} (f_{CLK} is the internal system clock frequency.)

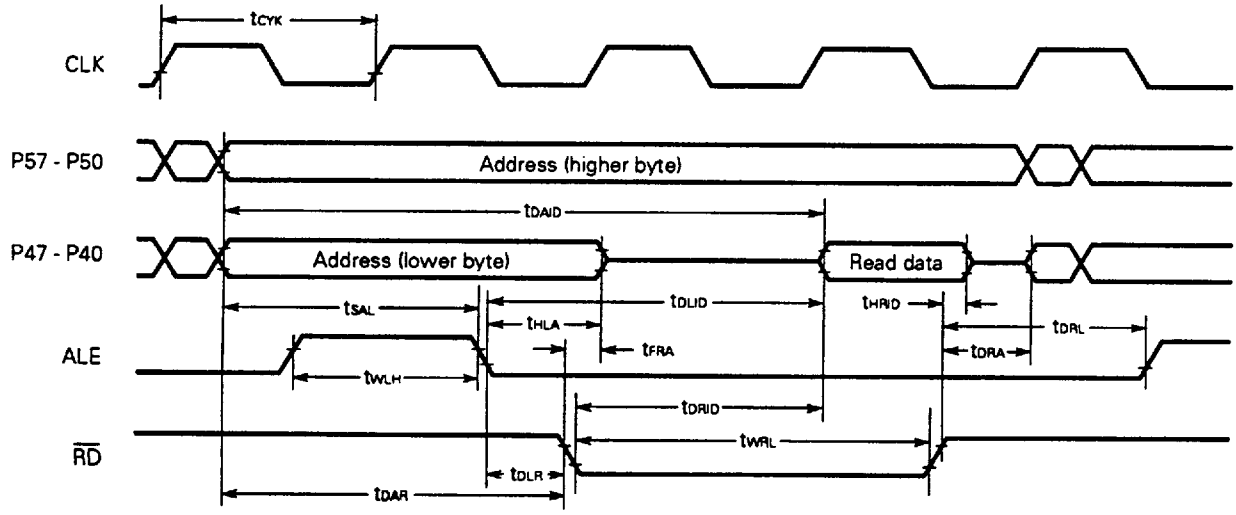
3. Items other than listed above are not dependent on the internal system clock frequency (f_{CLK}).

AC Timing Test Points

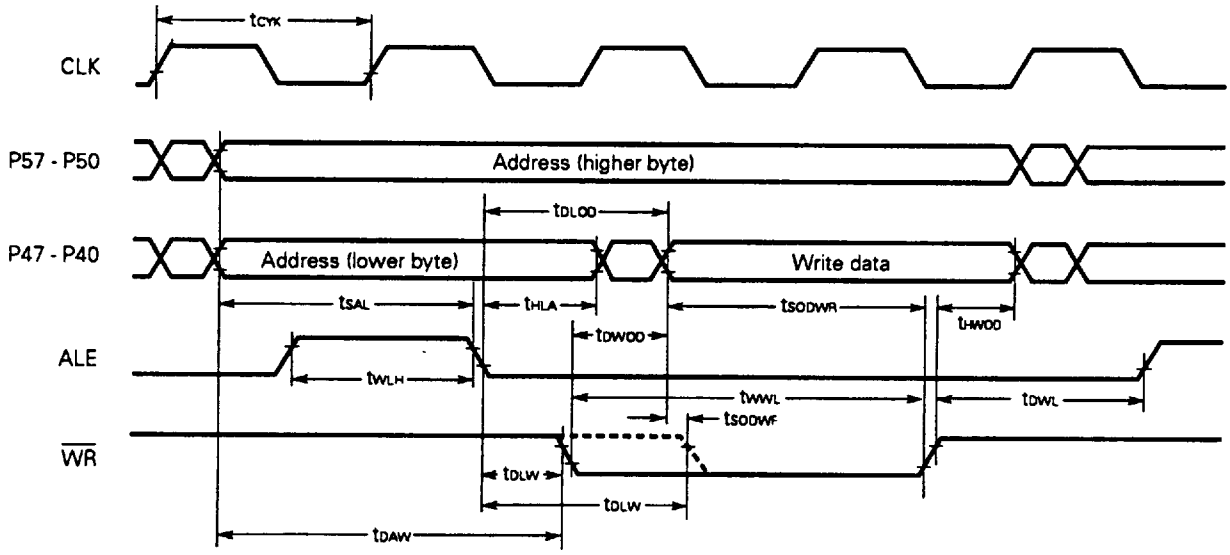


Timing Waveform

Read operation:

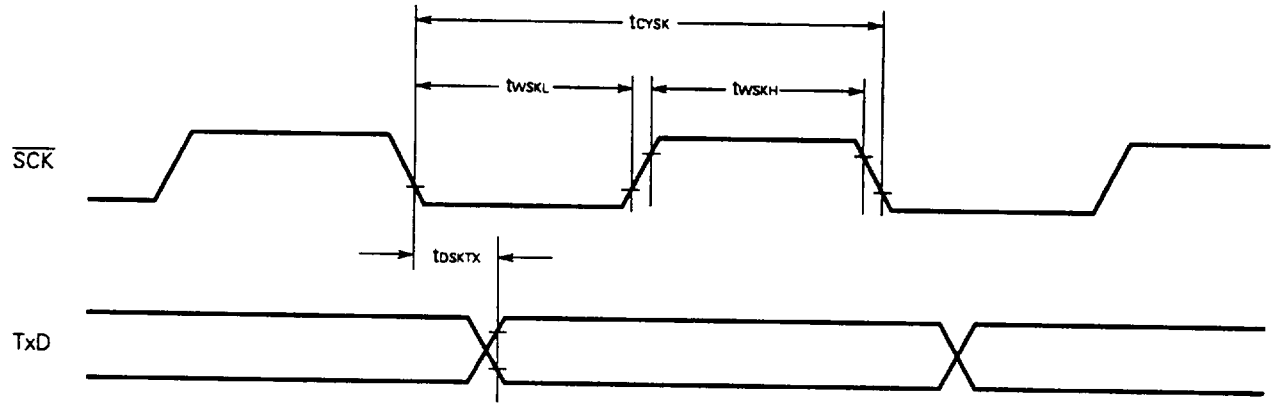


Write operation:

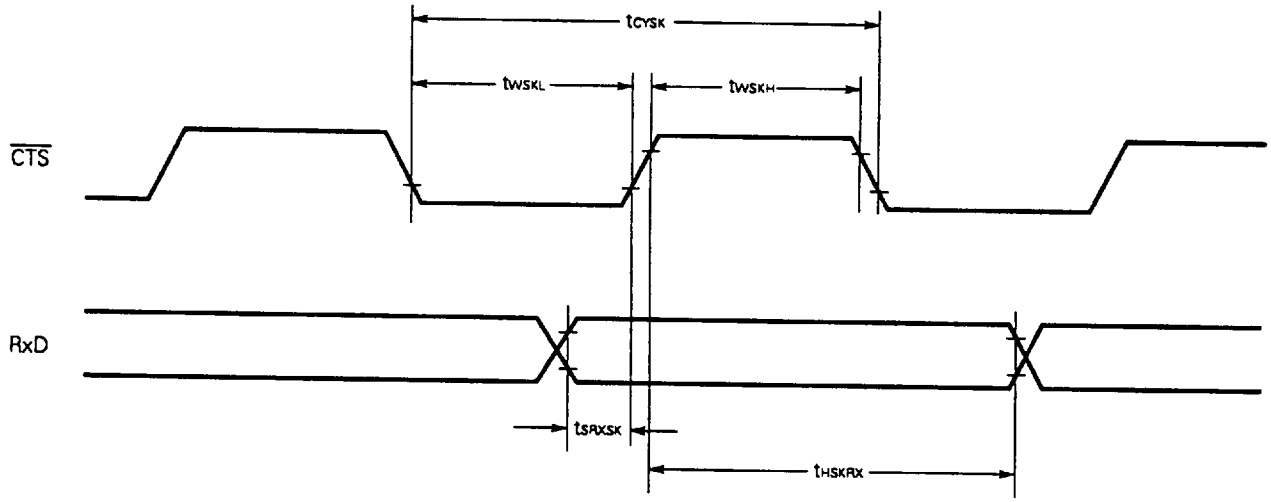


Serial Operation

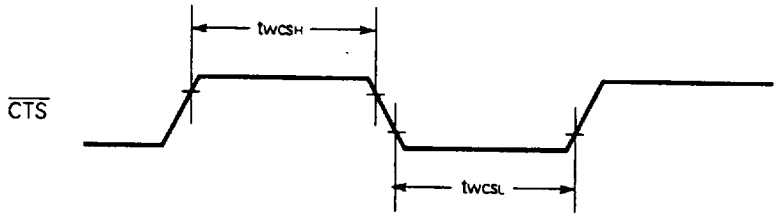
Send operation in the I/O interface mode:



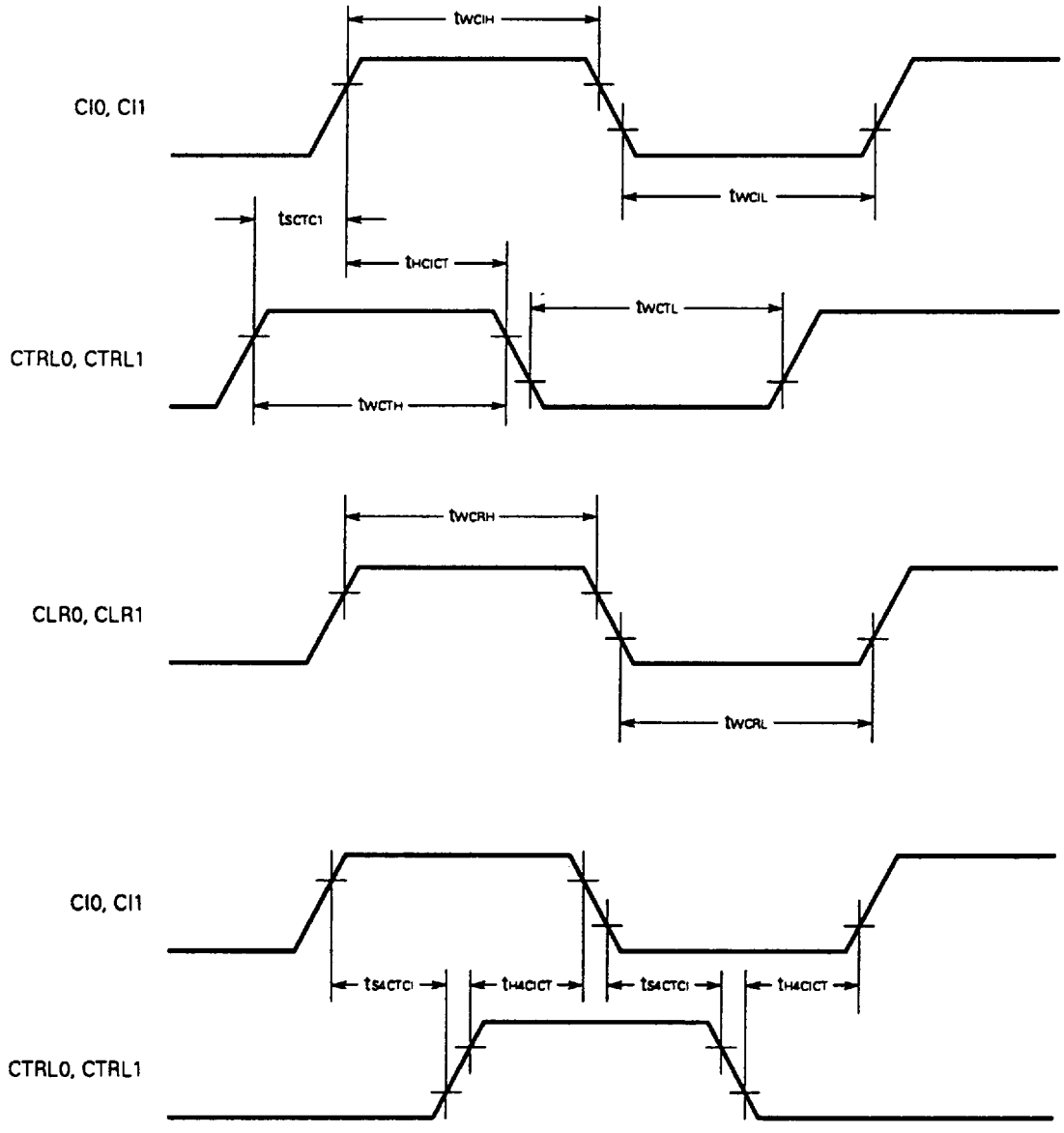
Receive operation in the I/O interface mode:



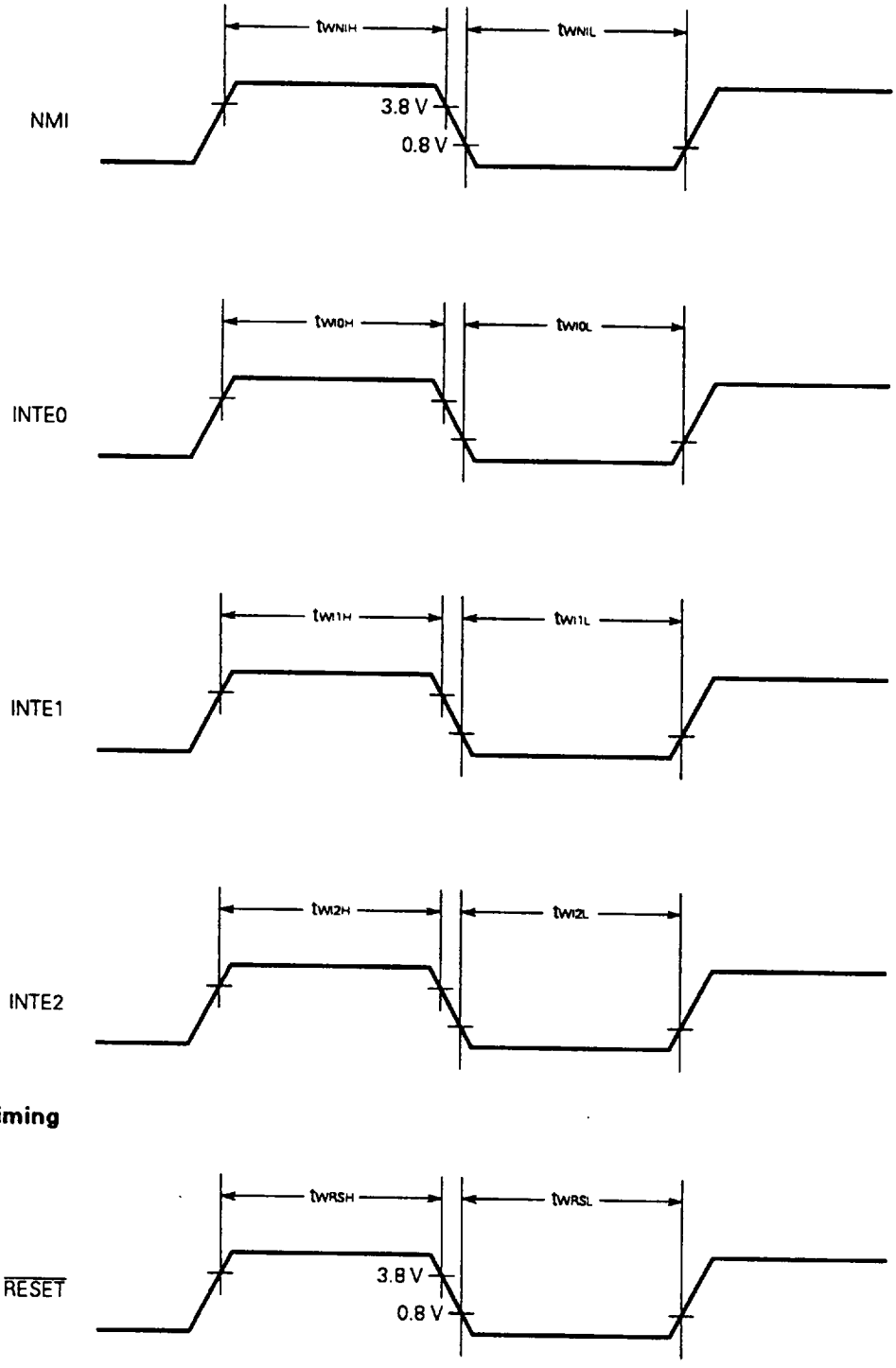
Send enable input timing (asynchronous mode):



Count Unit Input Timing

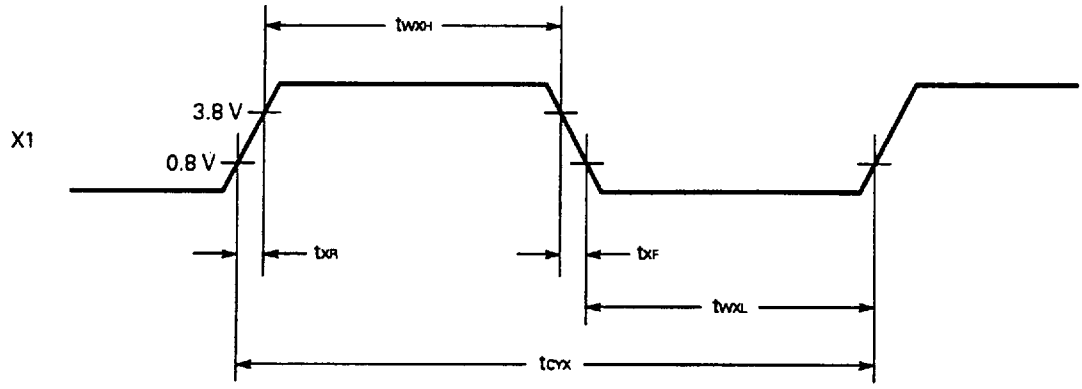


Interrupt Input Timing

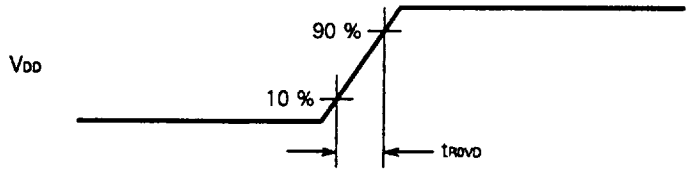


Reset Input Timing

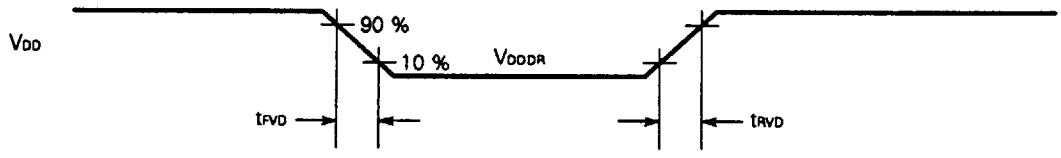
External Clock Timing



Power-On Timing

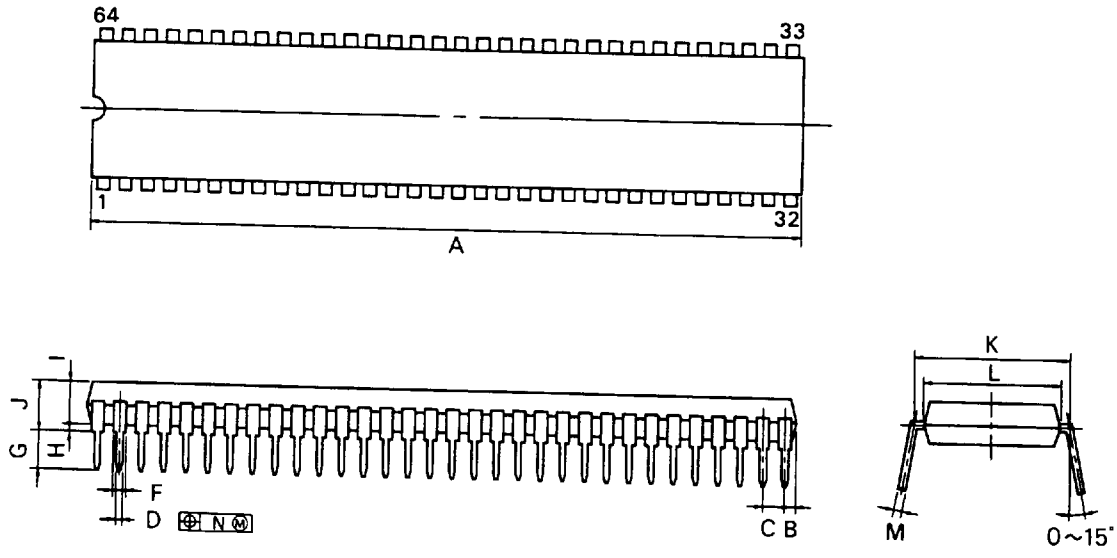


Data Retention Timing



12. PACKAGE DIMENSIONS

64PIN PLASTIC SHRINK DIP (750 mil)



P64C-70-750A,C

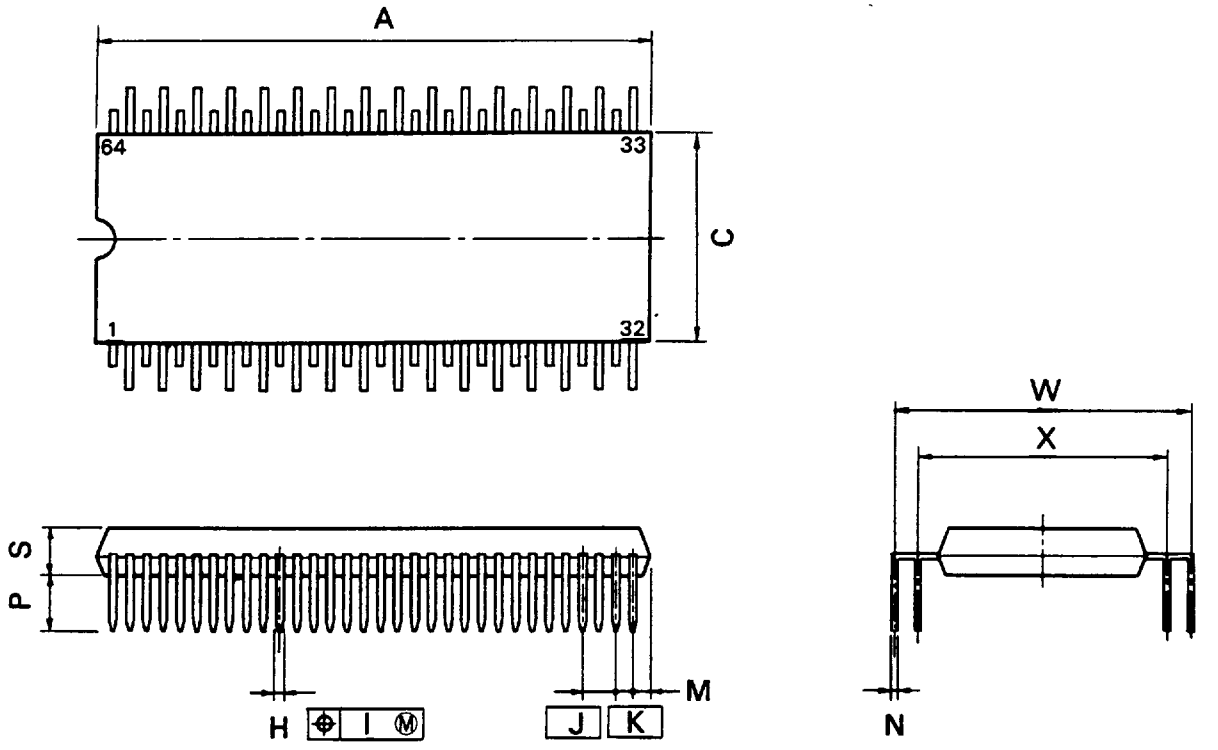
NOTES

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	$0.50^{\pm 0.10}$	$0.020^{+0.004}_{-0.005}$
F	0.9 MIN.	0.035 MIN.
G	$3.2^{\pm 0.3}$	$0.126^{\pm 0.012}$
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 QUIP

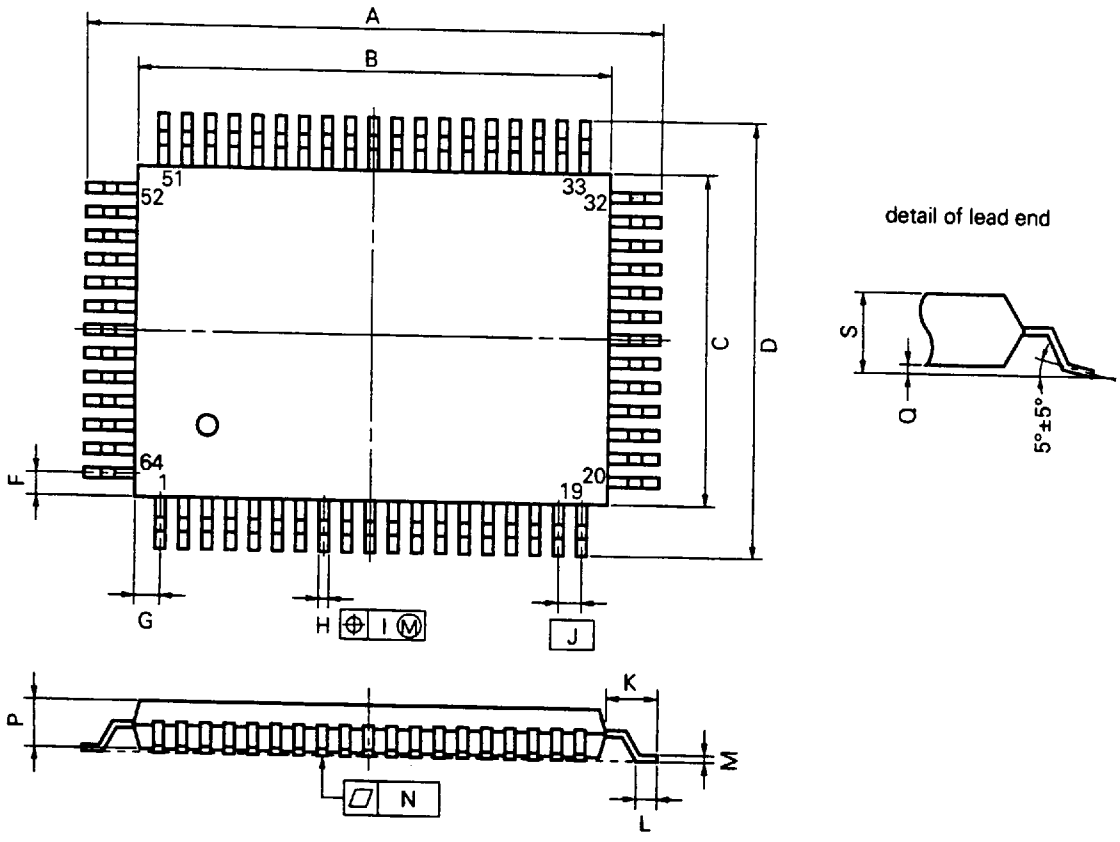


P64GQ-100-36

NOTE
Each lead centerline is located within 0.25 mm (0.010 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	$41.5^{\pm 0.3}$	$1.634^{\pm 0.012}$
C	16.5	0.650
H	$0.50^{\pm 0.10}$	$0.020^{\pm 0.004}$
I	0.25	0.010
J	2.54 (T.P.)	0.100 (T.P.)
K	1.27 (T.P.)	0.050 (T.P.)
M	$1.1^{\pm 0.25}$	$0.043^{\pm 0.011}$
N	$0.25^{\pm 0.10}$	$0.010^{\pm 0.004}$
P	$4.0^{\pm 0.3}$	$0.157^{\pm 0.012}$
S	$3.6^{\pm 0.1}$	$0.142^{\pm 0.004}$
W	$24.13^{\pm 1.05}$	$0.950^{\pm 0.042}$
X	$19.05^{\pm 1.05}$	$0.750^{\pm 0.042}$

64 PIN PLASTIC QFP (14×20)



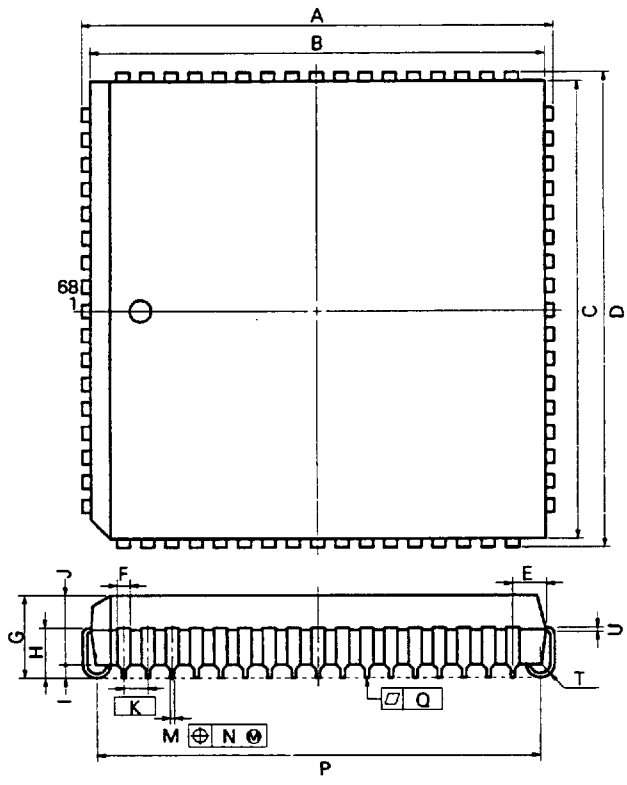
NOTE

Each lead centerline is located within 0.20 mm (0.008 inch) of its true position (T.P.) at maximum material condition.

P64GF-100-3B8,3BE,3BR-1

ITEM	MILLIMETERS	INCHES
A	23.6±0.4	0.929±0.016
B	20.0±0.2	$0.795^{+0.009}_{-0.008}$
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	$0.40^{+0.10}_{-0.05}$	$0.016^{+0.004}_{-0.005}$
I	0.20	0.008
J	1.0 (T.P.)	0.039 (T.P.)
K	1.8±0.2	$0.071^{+0.008}_{-0.009}$
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	0.12	0.005
P	2.7	0.106
Q	0.1±0.1	0.004±0.004
S	3.0 MAX.	0.119 MAX.

68 PIN PLASTIC QFJ (□950 mil)



P68L-50A1-2

NOTE

Each lead centerline is located within 0.12 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	25.2±0.2	0.992±0.008
B	24.20	0.953
C	24.20	0.953
D	25.2±0.2	0.992±0.008
E	1.94±0.15	$0.076^{+0.007}_{-0.006}$
F	0.6	0.024
G	4.4±0.2	$0.173^{+0.009}_{-0.008}$
H	2.8±0.2	$0.110^{+0.009}_{-0.008}$
I	0.9 MIN.	0.035 MIN.
J	3.4	0.134
K	1.27 (T.P.)	0.050 (T.P.)
M	0.40±1.0	$0.016^{+0.004}_{-0.005}$
N	0.12	0.005
P	23.12±0.20	$0.910^{+0.009}_{-0.008}$
Q	0.15	0.006
T	R 0.8	R 0.031
U	$0.20^{+0.10}_{-0.05}$	$0.008^{+0.004}_{-0.002}$

13. RECOMMENDED SOLDERING CONDITIONS

The following conditions (see table below) must be met when soldering this product.

For the details of the recommended soldering conditions refer to our document *SMD Surface Mount Technology Manual* (IEI-1207).

Please consult with our sales offices in case other soldering process is used, or in case soldering is done under different conditions.

Table 13-1 Soldering Conditions for Surface-Mount Devices (1)

μPD78310AGF(A)-3BE : 64-pin plastic QFP (14 × 20 mm)
μPD78312AGF(A)-xxx-3BE: 64-pin plastic QFP (14 × 20 mm)

Soldering process	Soldering conditions	Symbol
Infrared ray reflow	Peak package's surface temperature: 230 °C Reflow time: 30 seconds or less (at 210 °C or more) Number of reflow processes: 1 Exposure limit[Note]: 2 days (16 hours of pre-baking is required at 125 °C afterward.)	IR30-162-1
VPS	Peak package's surface temperature: 215 °C Reflow time: 40 seconds or less (at 200 °C or more) Number of reflow processes: 1 Exposure limit[Note]: 2 days (16 hours of pre-baking is required at 125 °C afterward.)	VP15-162-1
Partial heating method	Terminal temperature: 300 °C or less Flow time: 3 seconds or less (one side per device)	–

Note Exposure limit before soldering after dry-pack package is opened.
Storage conditions: Temperature of 25 °C and maximum relative humidity at 65 % or less

Caution Do not apply more than a single process at once, except for "Partial heating method."

Table 13-2 Soldering Conditions for Surface-Mount Devices (2)

μPD78310AL(A) : 68-pin plastic QFJ (□950 mil)
μPD78312AL(A)-xxx: 68-pin plastic QFJ (□950 mil)

Soldering process	Soldering conditions	Symbol
Infrared ray reflow	Peak package's surface temperature: 230 °C Reflow time: 30 seconds or less (at 210 °C or more) Number of reflow processes: 1	IR30-00-1
VPS	Peak package's surface temperature: 215 °C Reflow time: 40 seconds or less (at 200 °C or more) Number of reflow processes: 1	VP15-00-1
Partial heating method	Terminal temperature: 300 °C or less Flow time: 3 seconds or less (one side per device)	–

Caution Do not apply more than a single process at once, except for "Partial heating method."

Table 13-3 Soldering Conditions for Inserted Devices

μPD78310ACW(A) : 64-pin plastic shrink DIP (750 mil)
μPD78310AGQ(A)-36 : 64-pin plastic QUIP
μPD78312ACW(A)-xxx : 64-pin plastic shrink DIP (750 mil)
μPD78312AGQ(A)-xxx-36 : 64-pin plastic QUIP

Soldering process	Soldering conditions
Wave soldering (only for leads)	Temperature in the soldering vessel: 260 °C or less Soldering time: 10 seconds or less
Partial heating method	Terminal temperature: 260 °C or less Flow time: 10 seconds or less

Caution In wave soldering, apply solder only to the lead section. Care must be taken that jet solder does not contact the main body of the package.

★ **Notice**

Other versions of the products are available. For these versions, the recommended reflow soldering conditions have been mitigated as follows:
Higher peak temperature (235 °C), two-stage, and longer exposure limit.
Contact an NEC representative for details.

14. NOTES ON USE

14.1 NOTES ON THE SERIAL COMMUNICATION INTERFACE FUNCTION

When the asynchronous serial interface mode is selected, do not change the input signal level of the $\overline{CTS}$ pin during data transmission. Also do not change the status of the TXRDY bit (bit 7) in the serial communication mode register (SCM).

When the $\overline{CTS}$ pin input level is changed from low to high or the TXRDY bit is set to 1, an extra transmission completion interrupt may be generated. This erases send data by overwriting in the transmission buffer register (TxB).

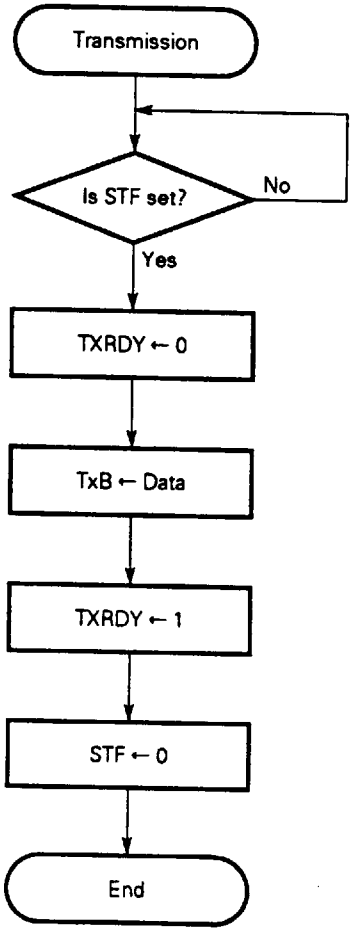
To prevent this, take the following action:

[Action to be taken]

(1) Fix the $\overline{CTS}$ pin input level to low, and keep the TXRDY bit set to 1 during serial communication.

(2) Poll the transmission completion interrupt by software.

Fig. 14-1 Flowchart for Polling



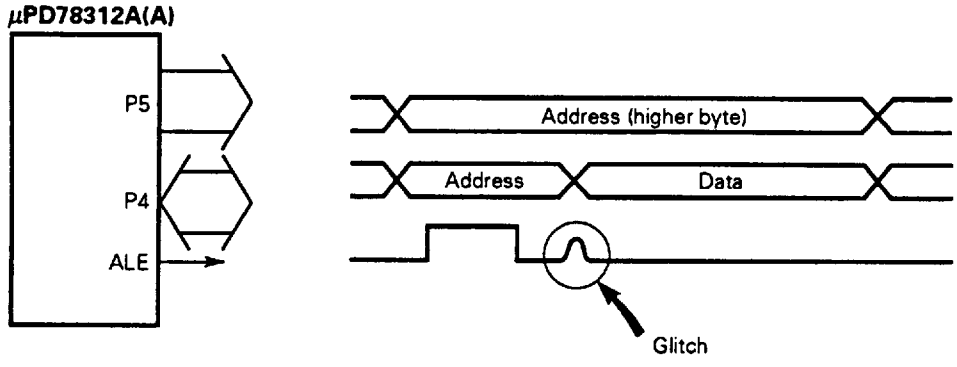
14.2 NOTES ON USE WHEN AN EXTERNAL DEVICE IS EXPANDED

In some application systems, when external memory is accessed with the μPD78310A(A) or μPD78312A(A), a glitch of approximately up to 2.0 V may appear on the ALE pin.

(1) Generation of glitches

A glitch is likely to appear as the status of the address/data multiplexed bus (port 4) changes from output of address FFH to output of data 00H. The glitch may cause the address latch circuit to malfunction and latch the data being regarded improperly as the lower address part. This may prevent normal operation of the application product.

Fig. 14-2 Generation of a Glitch on the ALE Pin



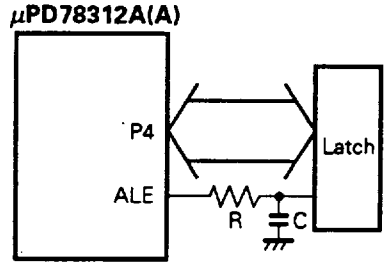
(2) Protection

To suppress glitch outputs to a level that allows normal operation of an application system, bear the following in mind when mounting the device:

① Strengthen patterns for the power supply and ground. (For example, use a multi-layer board.)
② Directly mount the device, instead of inserting it into a socket.
③ Reduce the load capacity of the bus.

The circuit shown in Fig. 14-3 can be immediately implemented to prevent malfunction.

Fig. 14-3 Sample Protection Circuit



Reference value
R: 100 Ω
C: 100 pF

(3) Causes

Possible factors contributing to glitch generation include:

① Device factor

As operation becomes faster, switching noise is easier to generate.

② System factors

- As the load capacity on the bus increases, an instantaneous quantity of traveling charge increases, increasing a chance of generating glitches.
- As the impedance on the power line increases, the more likely it is that glitches may be generated.

★ **14.3 NOTES ON THE COMBINATION OF INSTRUCTIONS**

An instruction for accessing the saddr area accesses the SFR area (FF20H to FFFFH) instead of the saddr area (FE20H to FEFFH) if the instruction is executed immediately after an arithmetic or logical instruction which manipulates a specific special function register (SFR).

This occurs with certain combinations of instructions regardless of the power voltage or operating frequency.

When instructions are specified as shown in the example below, the BF instruction accesses 0FF21H.0H instead of 0FE21H.0H. To prevent this, a NOP instruction must be inserted immediately after an arithmetic or logical instruction.

We recommend using an assembler of Ver. 4.10 or later. These versions of assembler detect relevant combinations of instructions and automatically insert a NOP instruction after the arithmetic or logical instruction (see the example below).

Example Source file

```
                    ADDW  CR11, #01H
                    BF    0FE21H.0H, $JMP
                    .
                    .
                    .
JMP:
                    .
                    .
                    .
                    Assembler
                    (Ver. 4.10)
Object module file
                    .
                    .
                    .
                    ADDW  CR11, #01H
                    NOP                ; NOP instruction automatically inserted
                    BF    0FE21H.0H, $JMP
                    .
                    .
                    .
JMP:
                    .
                    .
                    .
```

When the assembler inserts a NOP instruction, a message appears in the assemble list file (*.PRN) to indicate the insertion. The total number of inserted NOP instructions is also indicated.

Tables 14-1 to 14-3 list the relevant SFRs, arithmetic/logical instructions, and saddr instructions.

Table 14-1 Relevant sfr and sfrp

sfr	CR11L	(0FF0EH)
	UDC1L	(0FF1EH)
	CCW	(0FF4EH)
	MD1L	(0FF8EH)
	EXTSFR14[Note]	(0FFBEH)
	TMIC0	(0FFCEH)
	STIC	(0FFDEH)
sfrp	CR11	(0FF0EH)
	UDC1	(0FF1EH)
	MD1	(0FF8EH)

Note External SFR

Remark Addresses are indicated in parentheses.

Table 14-2 Relevant Instructions

Mnemonic	Operand
XCH	A, sfr
ADD ADDC SUB SUBC AND OR XOR	sfr, #byte
XCHW	AX, sfrp
ADDW SUBW	sfrp, #word

Table 14-3 saddr Instructions

Mnemonic	Operand
MOV ADD ADDC SUB SUBC AND OR XOR CMP	saddr, #byte
MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	A, saddr
MOV	saddr, A
MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	saddr, saddr
MOV XCH	A, [saddrp]
MOV	[saddrp], A
MOVW ADDW SUBW CMPW	saddrp, #word
MOVW XCHW ADDW SUBW CMPW	AX, saddrp
MOVW	saddrp, AX
MOVW XCHW ADDW SUBW CMPW	saddrp, saddrp
INC DEC	saddr
INCW DECW	saddrp
MOV1 AND1 OR1 XOR1	CY, saddr.bit
AND1 OR1	CY, /saddr.bit
MOV1	saddr.bit, CY
SET1 CLR1 NOT1	saddr.bit
BT BF BTCLR BFSET	saddr.bit, $addr16
DBNZ	saddr, $addr16

★ **14.4 NOTES ON THE POP PSW INSTRUCTION**

The CPU may crash when an interrupt occurs (that is, when an interrupt is accepted and the corresponding interrupt request is issued to the CPU) immediately after a POP PSW instruction with a two-byte or longer instruction following it is executed. To prevent this, insert a one-byte instruction, such as a NOP instruction, immediately after the POP PSW instruction.

Whether this occurs depends on the state of the instruction prefetch queue before the POP PSW instruction is executed.

Example

```
        ⋮
MOVW    RP2, #1234H
POP     PSW         ← Does not handle interrupt even when an interrupt request is issued.
CLR1    CRIC11.6    ← Does not access CRIC11 (FFC6H) but generates an invalid sfr
                      address.
        ⋮
```

<Preventive measure>

```
        ⋮
MOVW    RP2, #1234H
POP     PSW         ← Handles interrupt when an interrupt request is issued.
NOP                 ← Insert one-byte instruction such as NOP.
CLR1    CRIC11.6    ← Accesses sfr normally.
        ⋮
```

APPENDIX DEVELOPMENT TOOLS ★

The following tools are provided for developing a system that uses the μPD78310A(A) and μPD78312A(A):

Hardware

IE-78310A-R	In-circuit emulator for developing and debugging application systems. For debugging, connect the emulator to the host machine. Since object files can be transferred to/from the host machine, efficient debugging is enabled.
EP-78310CW EP-78310GF EP-78310GQ EP-78310L	Emulation probe for connecting the IE-78310A-R to a user system.
PG-1500	The PG-1500 PROM programmer is used together with an accessory board and optional program adapter. It allows the user to program a single chip microcomputer containing PROM independently or from a host machine. The PG-1500 can be used to program typical 256K-bit to 4M-bit PROMs.
PA-78P312CW PA-78P312GF PA-78P312GQ PA-78P312L	PROM-programmer adapter for writing a program into the μPD78P312A. It is used with the general PROM programmer such as the PG-1500. PA-78P312CW : For μPD78P312ACW and μPD78P312ADW PA-78P312GF : For μPD78P312AGF-3BE PA-78P312GQ : For μPD78P312AGQ-36 and μPD78P312AR PA-78P312L : For μPD78P312AL

Other PROM programmers

The following PROM programmers can be used for writing a program into the μPD78P312A:

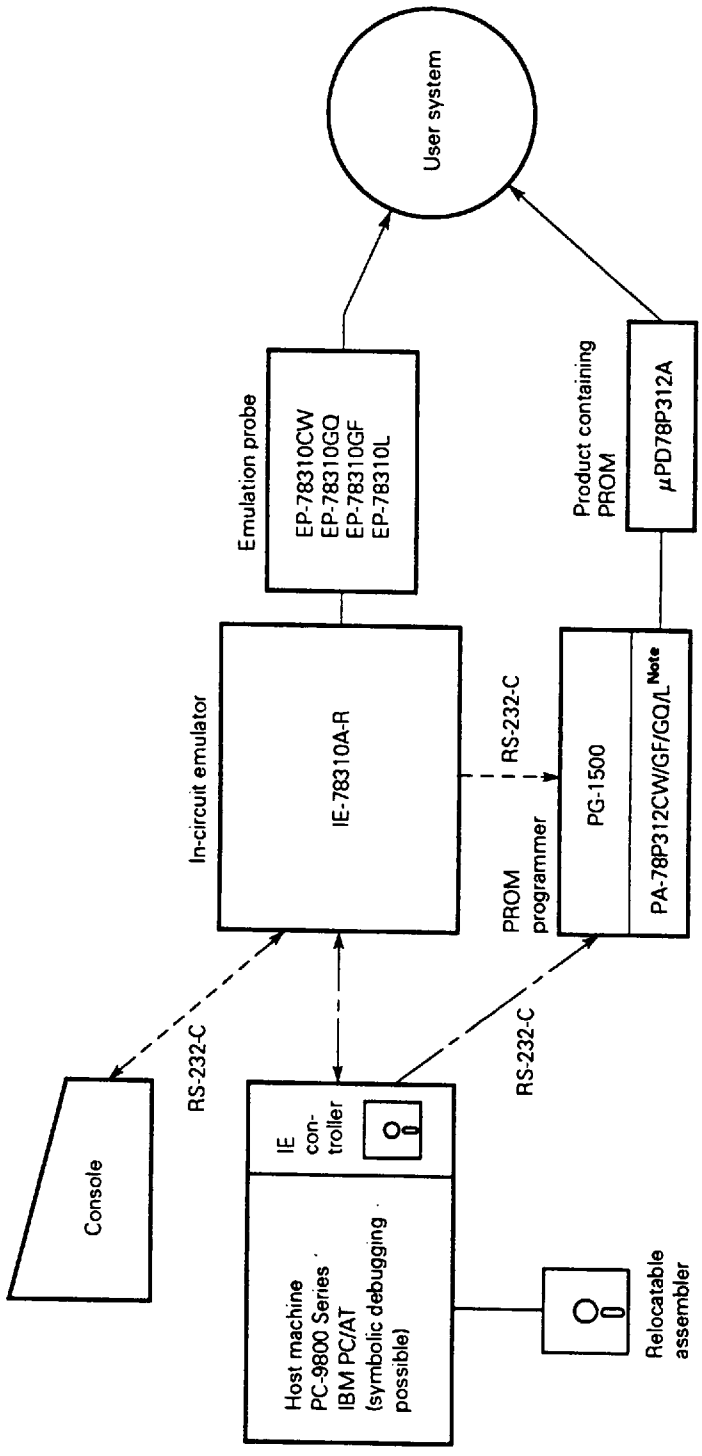
Manufacturer	Part number
Data I/O Japan	UNISITE 2900
Ando Electric	AF-9704 AF-9705

Software

IE-78310A-R control program (IE controller)	Host machine	OS	Distribution media	Part number
	PC-9800 series	MS-DOS™	3.5-inch 2HD	μS5A13IE78310
			5.25-inch 2HD	μS5A10IE78310-P01
	IBM PC/AT™	PC DOS™	5.25-inch 2HC	μS7B10IE78310
78K/III series relocatable assembler	Host machine	OS	Distribution media	Part number
	PC-9800 series	MS-DOS	3.5-inch 2HD	μS5A13RA78K3
			5.25-inch 2HD	μS5A10RA78K3
	IBM PC/AT	PC DOS	5.25-inch 2HC	μS7B10RA78K3
PG-1500 controller	Host machine	OS	Distribution media	Part number
	PC-9800 series	MS-DOS	3.5-inch 2HD	μS5A13PG1500
			5.25-inch 2HD	μS5A10PG1500
	IBM PC/AT	PC DOS	5.25-inch 2HC	μS7B10PG1500

Remark It is guaranteed that the software described above only runs on the host machine under the OS mentioned above.

Development tool configuration



— — — : When the in-circuit emulator is connected to a host machine

– – – – : When the in-circuit emulator is connected to a console so that it is used as a stand-alone emulator

Note PA-78P312CW: For μPD78P312ACW and μPD78P312ADW

PA-78P312GF : For μPD78P312AGF-3BE

PA-78P312GQ: For μPD78P312AGQ-36 and μPD78P312AR

PA-78P312L : For μPD78P312AL

Cautions on CMOS Devices

① **Countermeasures against static electricity for all MOSs**

Caution **When handling MOS devices, take care so that they are not electrostatically charged.**
Strong static electricity may cause dielectric breakdown in gates. When transporting or storing MOS devices, use conductive trays, magazine cases, shock absorbers, or metal cases that NEC uses for packaging and shipping. Be sure to ground MOS devices during assembling. Do not allow MOS devices to stand on plastic plates or do not touch pins. Also handle boards on which MOS devices are mounted in the same way.

② **CMOS-specific handling of unused input pins**

Caution **Hold CMOS devices at a fixed input level.**
Unlike bipolar or NMOS devices, if a CMOS device is operated with no input, an intermediate-level input may be caused by noise. This allows current to flow in the CMOS device, resulting in a malfunction. Use a pull-up or pull-down resistor to hold a fixed input level. Since unused pins may function as output pins at unexpected times, each unused pin should be separately connected to the V_{DD} or GND pin through a resistor.
If handling of unused pins is documented, follow the instructions in the document.

③ **Statuses of all MOS devices at initialization**

Caution **The initial status of a MOS device is unpredictable when power is turned on.**
Since characteristics of a MOS device are determined by the amount of ions implanted in molecules, the initial status cannot be determined in the manufacture process. NEC has no responsibility for the output statuses of pins, input and output settings, and the contents of registers at power on. However, NEC assures operation after reset and items for mode setting if they are defined.
When you turn on a device having a reset function, be sure to reset the device first.

[MEMO]

[MEMO]

NEC
NEC Electronics Inc.
CORPORATE HEADQUARTERS*
2880 Scott Boulevard
P.O. Box 58062
Santa Clara, CA 95052-8062
TEL 408-588-6000
* As of September 3, 1996

For literature, call toll-free 7 a.m. to 6 p.m. Pacific time: **1-800-366-9782**
or FAX your request to: **1-800-729-9288**

IC-2837C