NEC

MOS INTEGRATED CIRCUIT

μPD78327, 78328

16/8 BIT SINGLE-CHIP MICROCOMPUTER

The μPD78328 is a member of the 78K/III series. It contains a 16-bit high-performance CPU. A PWM signal output function for the real-time pulse unit is implemented in the μPD78328. It is similar to one for the real-time pulse unit of the μPD78322, but has a more advanced inverter control function.

The μPD78327 is a ROM-less version of the μPD78328.

The following user's manual describes the details of functions. Be sure to read it before design.

μPD78328 User's Manual: IEU-1268

FEATURES

- High-speed data processing using the pipeline control system and instruction prefetching
  Minimum instruction execution time: 250 ns (when the internal clock frequency is 8 MHz, or when the external clock frequency is 16 MHz)
- An instruction set suited for control applications
  - 16-bit arithmetic/logical instructions
  - Multiply/divide instruction (16 bits × 16 bits, 32 bits ÷ 16 bits), etc.
- Eight built-in 10-bit A/D (analog-to-digital) converters
- High-speed 8-bit PWM signal output function
- Built-in advanced function interrupt controller
- Real-time pulse unit for controlling the inverter
  - One of the two timer output modes can be selected (six set/reset output channels or eight buffer output channels).
  - Six-phase PWM signals can be easily output.
- Powerful serial interface (A dedicated baud rate generator is built in.)
  - UART
  - SBI (NEC standard serial bus interface) or 3-wire serial I/O
- Watchdog timer function for detecting a program crash

APPLICATIONS

- PWM inverter control field: inverter air-conditioner, etc.

ORDERING INFORMATION

Part number	Package	Internal ROM
μPD78327CW	64-pin plastic shrink DIP (750 mil)	Not provided
μPD78327GF-3BE	64-pin plastic QFP (14 × 20 mm)	Not provided
μPD78328CW-×××	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78328GF-×××-3BE	64-pin plastic QFP (14 × 20 mm)	Mask ROM

Remark ×××: ROM code number

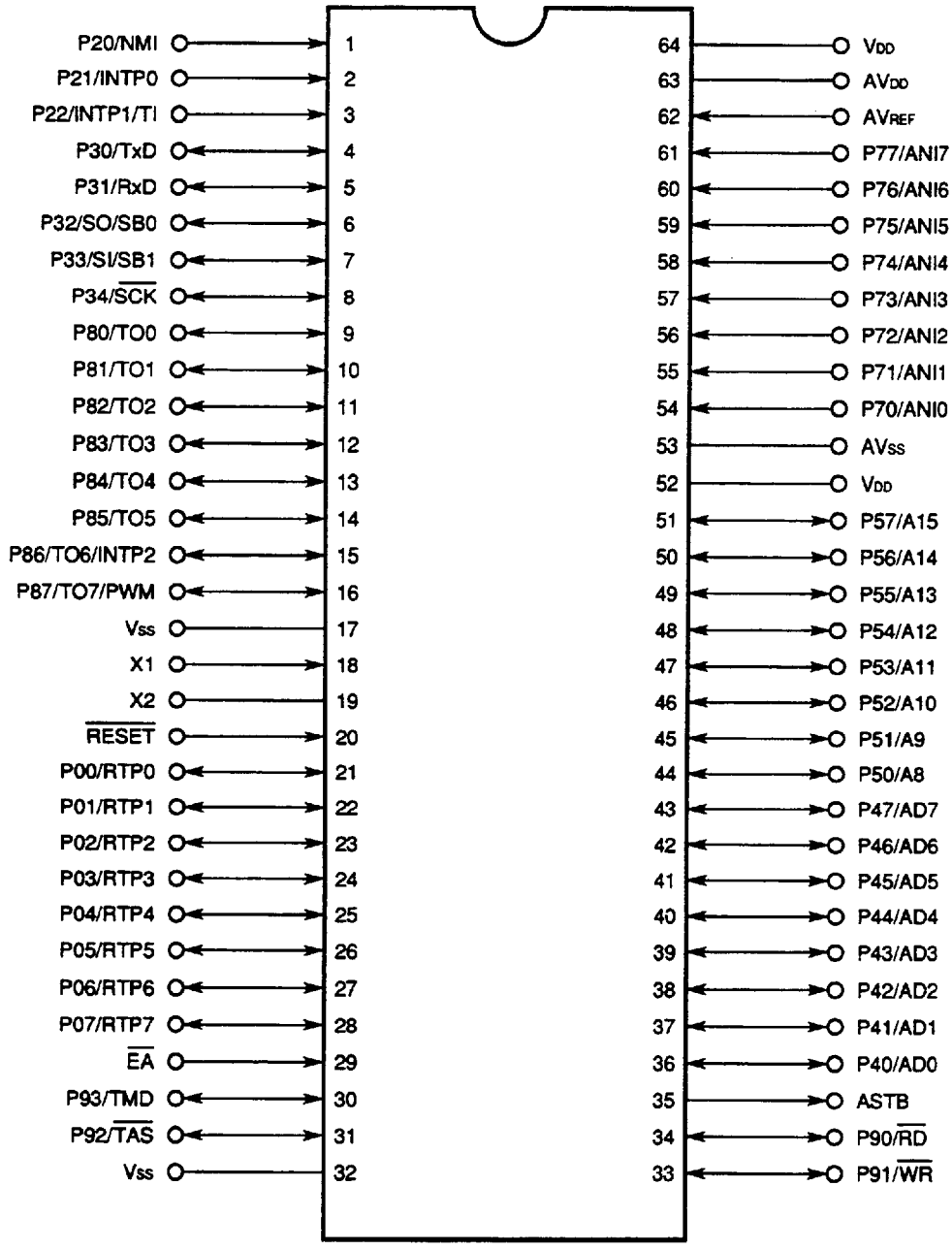
Document No. U10208EJ5V0DS00 (5th edition)
(Previous No. IC-2521)
Date Published September 1995 P
Printed in Japan

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

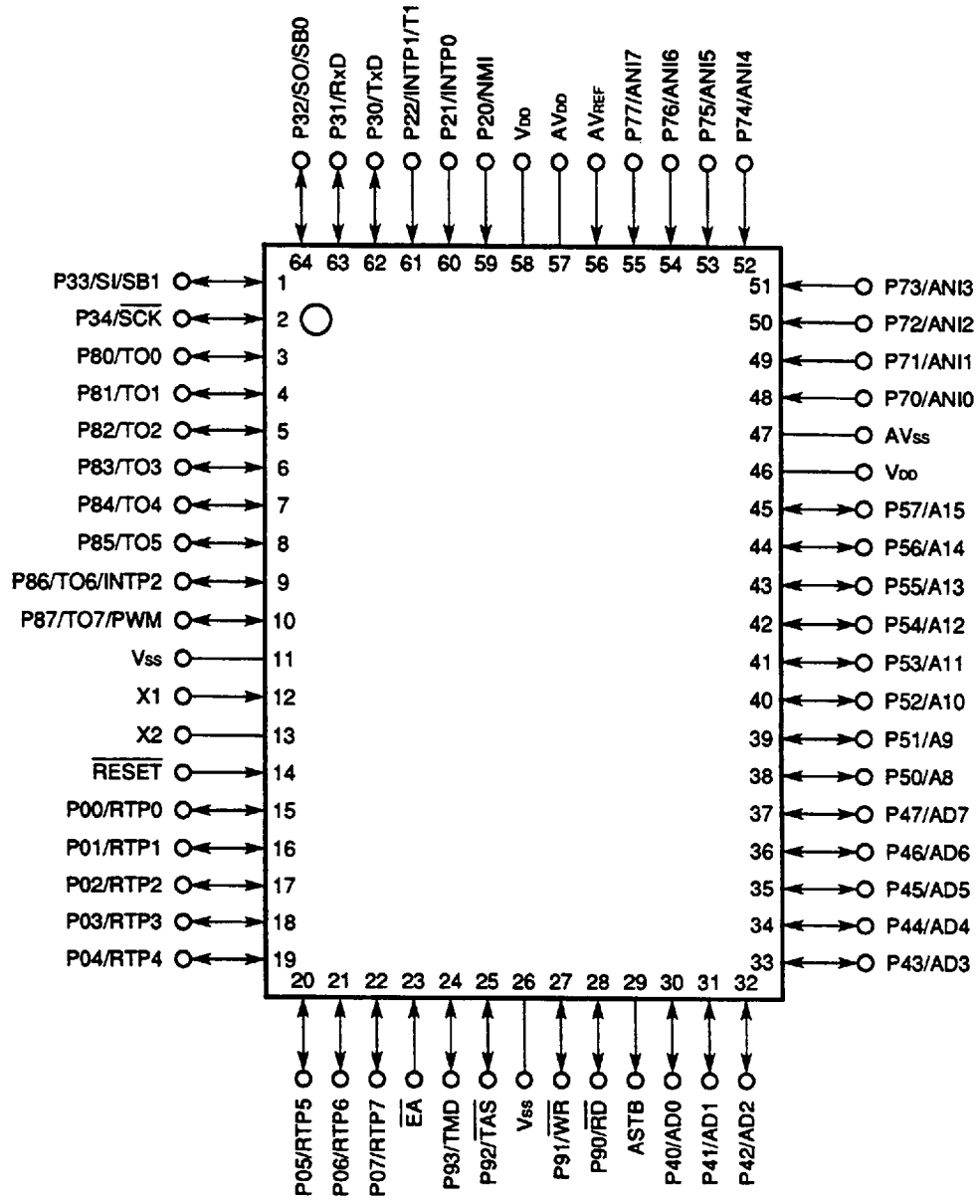
6427525 0086574 550

PIN CONFIGURATION

- **64-pin plastic shrink DIP (750 mil)**
  μPD78327CW
  μPD78328CW-xxx



- **64-pin plastic QFP (14 × 20 mm)**
  μPD78327GF-3BE
  μPD78328GF-xxx-3BE



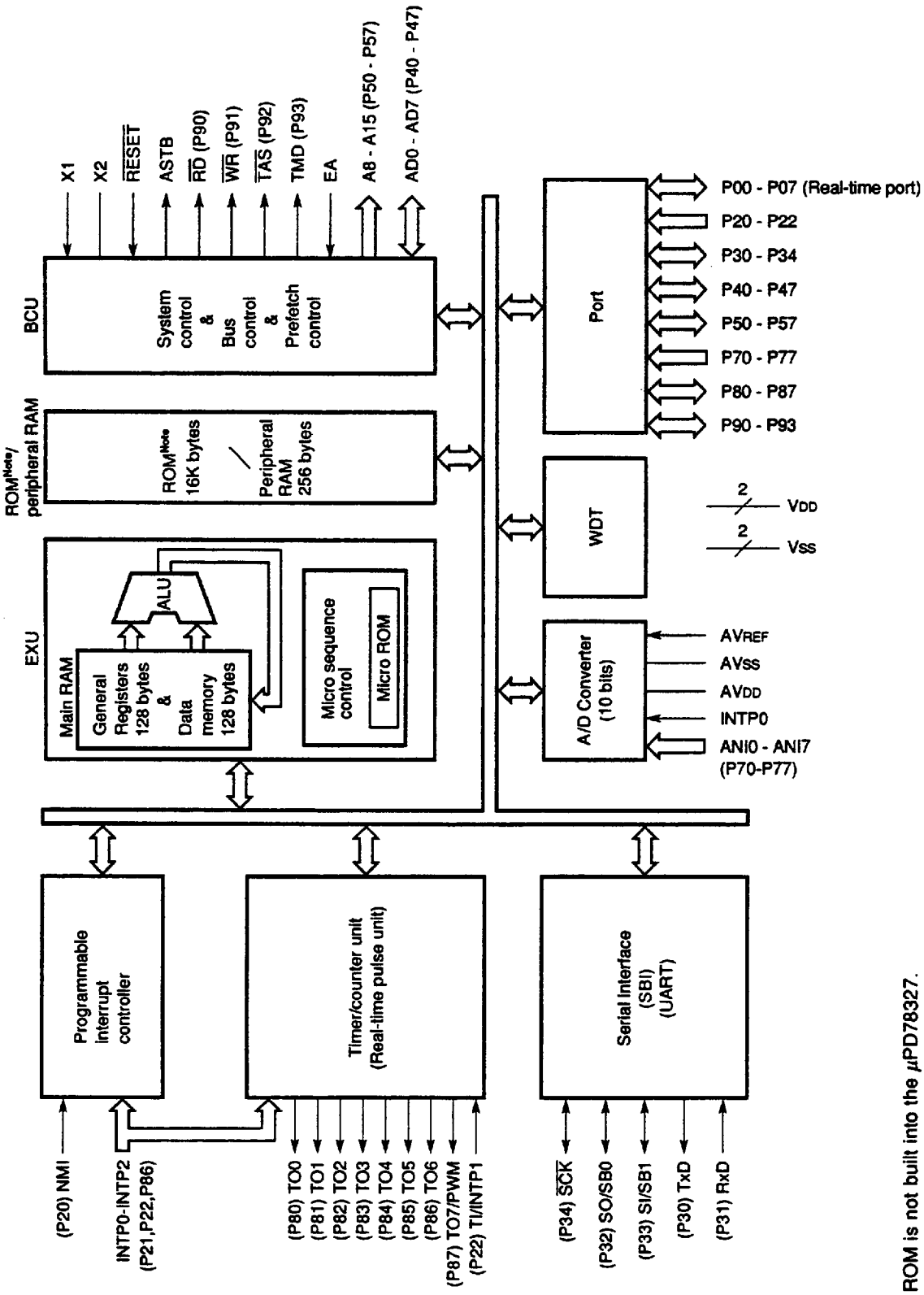
P00-P07 : Port 0
P20-P22 : Port 2
P30-P34 : Port 3
P40-P47 : Port 4
P50-P57 : Port 5
P70-P77 : Port 7
P80-P87 : Port 8
P90-P93 : Port 9
NMI : Nonmaskable interrupt
INTP0-INTP2 : Interrupt from peripherals
RTP0-RTP7 : Real-time ports
TI : Timer input
TxD : Transmit data
RxD : Receive data
SB0/SO : Serial bus or serial output
SB1/SI : Serial bus or serial input
$\overline{SCK}$: Serial clock
TO0-TO7 : Timer outputs

$\overline{RESET}$: Reset
X1, X2 : Crystal
PWM : Pulse width modulation output
$\overline{EA}$: External access
TMD : Turbo mode
$\overline{TAS}$: Turbo access strobe
$\overline{WR}$: Write strobe
$\overline{RD}$: Read strobe
ASTB : Address strobe
AD0-AD7 : Address or data buses
A8-A15 : Address buses
ANI0-ANI7 : Analog inputs
AV_{REF} : Analog reference voltage
AVss : Analog Vss
AV_{DD} : Analog V_{DD}
V_{DD} : Power supply
Vss : Ground

FUNCTION OVERVIEW

	μPD78327	μPD78328
Number of basic instructions	111	
Minimum instruction execution time	250 ns (when the internal clock frequency is 8 MHz, or when the external clock frequency is 16 MHz)	
Internal memory	• ROM: Not provided • RAM: 512 bytes	• ROM: 16K bytes • RAM: 512 bytes
Addressing space	64K bytes	
General register	8 bits × 16 × 8 banks (memory mapping)	
Real-time pulse unit	• Two 16-bit timers • 16-bit timer/event counter • Fourteen 16-bit compare registers • 16-bit capture/compare register • Two output modes can be selected. Mode 0 Six set/reset output channels One toggle output channel Mode 1 Eight buffer output channels	
PWM signal output function	8-bit resolution for one channel	
A/D converter function	10-bit resolution for eight channels	
Interrupt function	• Four external sources and 16 internal sources • A 3-level priority can be specified by the software. • Three types of interrupt processing modes can be selected, vector interrupt, context switching, and macro service functions.	
Test source	One internal source	
I/O line	• 11 Input ports • 23 I/O ports	• 11 Input ports • 41 I/O ports
Real-time output port	Two 4-bit channels or 8-bit channel	
Serial interface	• Dedicated baud rate generator • UART • SBI (NEC serial bus interface)	
Package	• 64-pin plastic shrink DIP (750 mil) • 64-pin plastic QFP (14 × 20 mm)	
Others	• Standby function (STOP/HALT) • Watchdog timer function	

BLOCK DIAGRAM



Note ROM is not built into the μPD78327.

CONTENTS

1. PIN FUNCTIONS

1.1 PORT PINS

Pin name	I/O	Function	Dual-function pin
P00-P07	I/O	Port 0 4- or 8-bit I/O port Inputs and outputs can be specified bit by bit. These pins also function as a real-time output port.	RTP0-RTP7
P20	I	Port 2 3-bit input dedicated port	NMI
P21			INTP0
P22			INTP1/TI
P30	I/O	Port 3 5-bit I/O port Inputs and outputs can be specified bit by bit.	TxD
P31			RxD
P32			SO/SB0
P33			SI/SB1
P34			$\overline{SCK}$
P40-P47	I/O	Port 4 8-bit I/O port Inputs and outputs can be specified in 8-bit units.	AD0-AD7
P50-P57	I/O	Port 5 8-bit I/O port Inputs and outputs can be specified bit by bit.	A8-A15
P70-P77	I	Port 7 8-bit input dedicated port	ANI0-ANI7
P80	I/O	Port 8 8-bit I/O port Inputs and outputs can be specified bit by bit.	TO0
P81			TO1
P82			TO2
P83			TO3
P84			TO4
P85			TO5
P86			TO6/INTP2
P87			TO7/PWM
P90	I/O	Port 9 4-bit I/O port Inputs and outputs can be specified bit by bit.	$\overline{RD}$
P91			$\overline{WR}$
P92			$\overline{TAS}$
P93			TMD

1.2 NON-PORT PINS (1/2)

Pin name	I/O	Function	Dual-function pin
RTP0-RTP7	O	Synchronized with a trigger signal sent from the real-time pulse unit (RPU) and outputs a pulse in real-time.	P00-P07
NMI	I	Receives a nonmaskable interrupt request signal. The mode register can specify the rising or falling edge as the effective edge for this request.	P20
INTP0	I	Receive an external interrupt request signal for which the mode register can specify the effective edge.	P21
INTP1			P22/TI
INTP2			P86/TO6
TI	I	Receives an external count clock signal to the timer 1 (TM1).	P22/INTP1
RxD	I	Receives serial data using the asynchronous serial interface (UART).	P30
TxD	O	Outputs serial data using the asynchronous serial interface (UART).	P31
SO	O	Outputs serial data in the 3-wire mode of the clock synchronous serial interface.	P32/SB0
SI	I	Receives serial data in the 3-wire mode of the clock synchronous serial interface.	P33/SB1
SB0	I/O	Receive or output serial data in the SBI mode for the clock synchronous serial interface.	P32/SO
SB1			P33/SI
$\overline{SCK}$	I/O	Receives or outputs the serial clock signal for the clock synchronous serial interface.	P34
AD0-AD7	I/O	Multiplexed address/data bus when the memory is ex-panded externally	P40-P47
A8-A15	O	Address bus when the memory is expanded externally	P50-P57
TO0	O	Output a pulse output from the real-time pulse unit.	P80
TO1			P81
TO2			P82
TO3			P83
TO4			P84
TO5			P85
TO6			P86/INTP2
TO7			P87/PWM
PWM	O	Outputs a PWM signal output from the real-time pulse unit.	P87/TO7
$\overline{RD}$	O	Outputs a strobe signal for reading data from external memory.	P90
$\overline{WR}$		Outputs a strobe signal for writing data into external memory.	P91
★ $\overline{TAS}$		Outputs a control signal to access the turbo access manager (μPD71P301)Note.	P92
TMD			P93
ASTB	O	Outputs a timing signal to externally latch lower address information output from port 4 for accessing an external memory.	—
$\overline{EA}$	I	For the μPD78328, normally the $\overline{EA}$ pin is connected to the V_{DD} pin. Connecting the $\overline{EA}$ pin to the Vss pin enters the ROM- less mode, enabling access to external memory. For the μPD78327, this pin must be fixed to 0 (low level). The level of the $\overline{EA}$ pin cannot be changed during operations.	—

Note The turbo access manager (μPD71P301) is maintenance service only.

1.2 NON-PORT PINS (2/2)

Pin name	I/O	Function	Dual-function pin
ANI0-ANI7	I	Analog input to the A/D converter	P70-P77
AV_{REF}	I	A/D converter reference voltage input	—
AV_{DD}	—	Analog power supply to the A/D converter	—
AV_{SS}	—	Ground of the A/D converter	—
$\overline{RESET}$	I	System reset input	—
X1	I	Crystal input pin for system clock oscillation An external clock signal input to the X1 pin is inverted and input to the X2 pin (the X2 pin can be left open).	—
X2	—		—
V_{DD}	—	Positive power supply pin	—
V_{SS}	—	Ground pin	—

1.3 PIN INPUT/OUTPUT CIRCUITS AND CONNECTIONS FOR UNUSED PINS

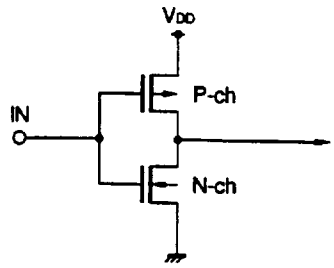
Table 1-1 lists the input/output circuit types of the pins and connections for unused pins, and Fig. 1-1 shows schematics for pin input/output circuits.

Table 1-1 Pin Input/Output Circuit Types and Recommended Connections for Unused Pins

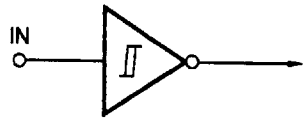
Pin	I/O circuit type	Recommended connection
P00-P07/RTP0-RTP7	5	Input mode : Individually connected to V_{DD} or Vss with a resistor. Output mode : Open
P20/NMI P21/INTP0 P22/INTP1/TI	2	Connected to Vss.
P30/TxD P31/RxD	5	Input mode : Individually connected to V_{DD} or Vss with a resistor. Output mode : Open
P32/SO/SB0 P33/SI/SB1 P34/$\overline{SCK}$	8	
P40-P47/AD0-AD7 P50-P57/A8-A15	5	
P70-P77/ANI0-ANI7	9	Connected to Vss.
P80-P85/TO0-TO5	5	Input mode : Individually connected to V_{DD} or Vss with a resistor. Output mode : Open
P86/TO6/INTP2	6	
P87/TO7/PWM	5	
P90/$\overline{RD}$ P91/$\overline{WR}$ P92/$\overline{TAS}$ P93/TMD	5	
ASTB	4	Open
$\overline{EA}$	1	—
$\overline{RESET}$	2	—
AV_{REF}, AVss	—	Connected to Vss.
AV_{DD}	—	Connected to V_{DD}.

Fig. 1-1 Input/Output Circuits of Each Pin

Type 1

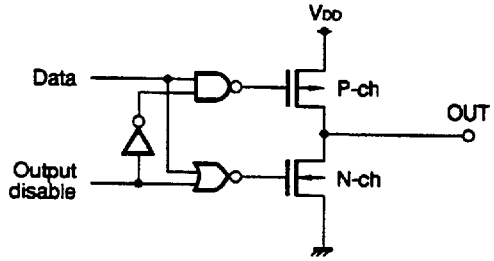


Type 2



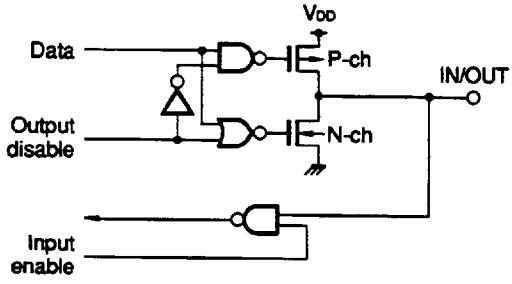
Schmitt trigger input with hysteresis characteristics

Type 4

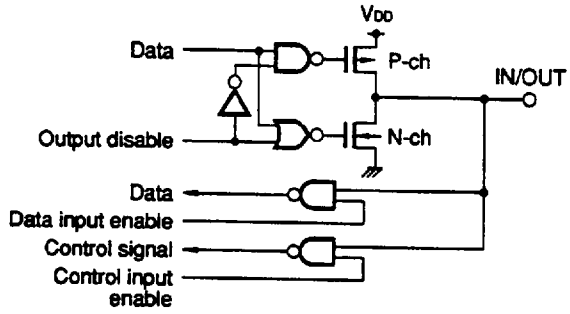


Push-pull output which can output high impedance
(Both the positive and negative channels are off.)

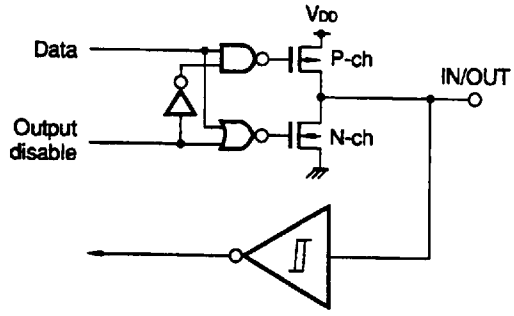
Type 5



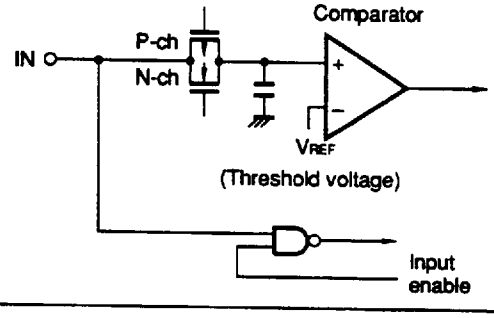
Type 6



Type 8



Type 9



2. CPU ARCHITECTURE

2.1 MEMORY SPACE

The μPD78327 and μPD78328 can address memory of up to 64K bytes (see **Fig. 2-1**).

A program can be fetched from the space 0000H to FDFFH. When the external memory is expanded to the space FE00H to FFFFH (for the main RAM and special function registers), a program can be also fetched from the external memory in this space, not from the main RAM and special function registers.

(1) Vector table area

Interrupt requests from peripheral hardware, reset inputs, external interrupt requests, and branch addresses for interrupts caused by break instructions are stored in the 64-byte area from 0000H to 003FH. When an interrupt request is issued, the contents of the even addresses in each table are set to the eight low-order bits of the program counter (PC) and the contents of the odd addresses are set to the eight high-order bits for branching.

Interrupt source		Vector table address
$\overline{\text{RESET}}$	(Input to the $\overline{\text{RESET}}$ pin)	0000H
NMI	(Input to the NMI pin)	0002H
WDT	(Watchdog timer)	0004H
INTOV0	(Timer 0 overflow)	0006H
INTP0	(Input to the INTP0 pin)	0008H
INTP1/TI	(Input to the INTP1/TI pin)	000AH
INTP2	(Input to the INTP2 pin)	000CH
INTOV1	(Timer 1 overflow)	000EH
INTCM00	(CM00 match signal)	0010H
INTCM01	(CM01 match signal)	0012H
INTCM02	(CM02 match signal)	0014H
INTCM03	(CM03 match signal)	0016H
INTCM04	(CM04 match signal)	0018H
INTCM05	(CM05 match signal)	001AH
INTCM06	(CM06 match signal)	001CH
INTCC10	(CC10 match signal)	001EH
INTCM20	(CM20 match signal)	0020H
INTSR	(Serial reception completion interrupt)	0024H
INTST	(Serial transmission completion interrupt)	0026H
INTCSI	(Serial transmission/reception interrupt)	0028H
INTAD	(A/D conversion completion interrupt)	002AH
Operation code trap		003CH
BRK	(Break instruction)	003EH

When bit 1 (TPF) of the CPU control word (CCW) is set to 1, the interrupt vector table is stored in the external memory area 8002H to 803FH, instead of the area 0002H to 003FH.

(2) CALLT table area

Thirty-two tables of the addresses called by single-byte call instructions (CALLT) can be stored in the 64-byte area from 0040H to 007FH. This area is the CALLT table area.

When bit 1 (TPF) of the CPU control word (CCW) is set to 1, the CALLT instruction table is stored in the external memory area 8040H to 807FH, instead of the area 0040H to 007FH.

(3) CALLF entry area

A subroutine can be directly called from the area 0800H to 0FFFH by using a double-byte call instruction (CALLF).

(4) Internal RAM area

Addresses FD00H to FEFFH are assigned to 512-byte internal RAM. This area is divided into two sections called:

- Peripheral RAM : FD00H to FDFFH (256 bytes)
- Main RAM : FE00H to FEFFH (256 bytes)

High-speed access to the main RAM is possible.

Macro service control words are mapped to the 36-byte area from FE06H to FE2BH in the main RAM area.

General registers are mapped to the 128-byte area from FE80H to FEFFH in the main RAM area consisting of eight banks.

(5) Special function register (SFR) area

Registers having special functions, such as mode and control registers for the on-chip peripheral hardware, are mapped to the area from FF00H to FFFFH. Addresses to which the registers are not mapped cannot be accessed.

(6) External memory area

The μPD78328 can expand an external memory (ROM or RAM) step by step in the 48K-byte area (4000H to FFFFH).

The μPD78327 can connect an external memory (ROM or RAM) in the 64K-byte area (0000H to FFFFH).

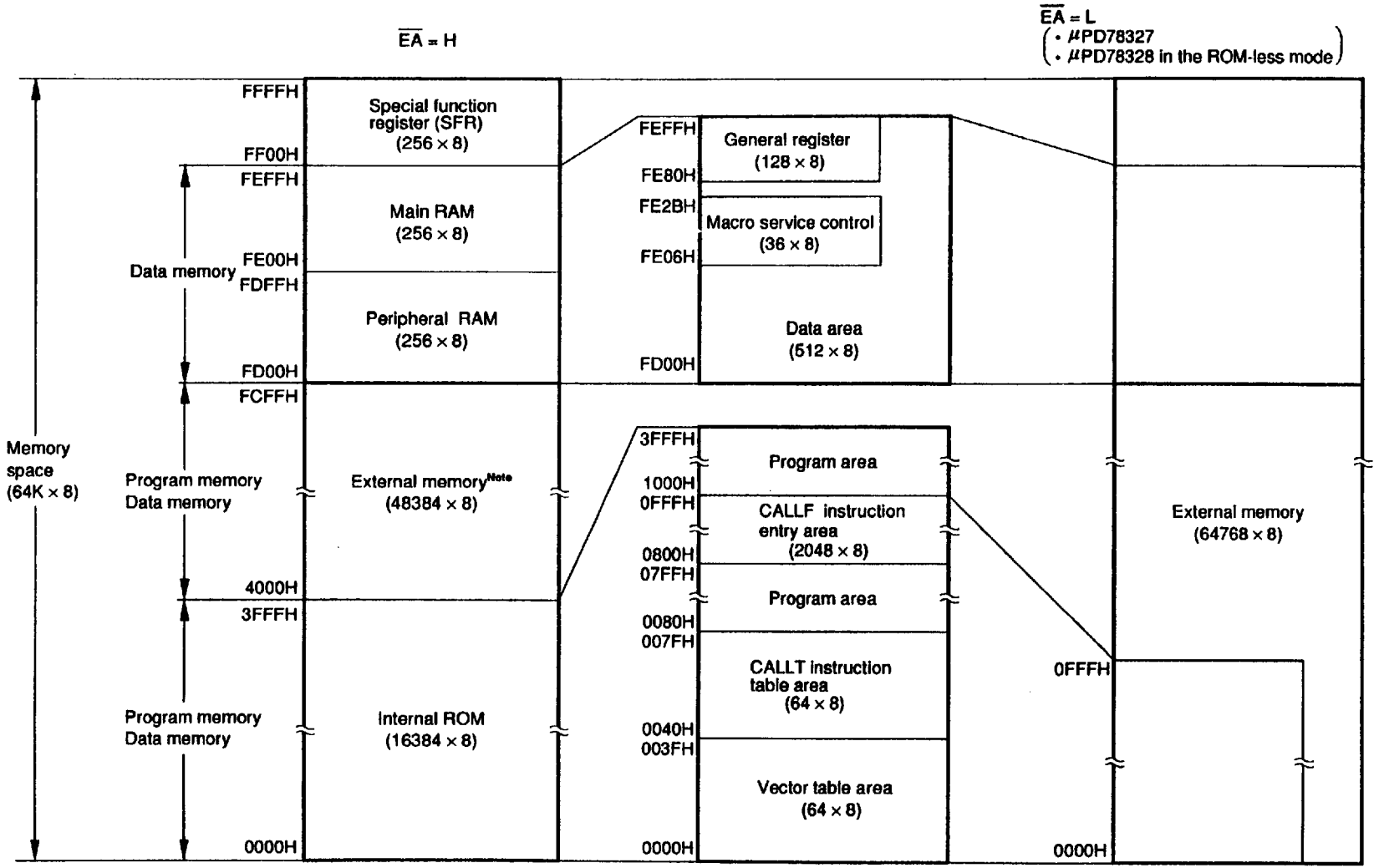
To access an external memory, the pins P40/AD0 to P47/AD7 (multiplexed address/data bus), P50/A8 to P57/A15 (address bus), $\overline{RD}$, $\overline{WR}$, and ASTB are used.

An external access area is mapped in the 16-byte area from FFD0H to FFDFH of the special function register (SFR). Thus, $\overline{SFR}$ addressing can access an external memory.

The $\overline{TAS}$ and TMD pins are provided to exclusively connect the turbo access manager (μPD71P301)**Note**. When the turbo access manager is used, the same program processing speed as the internal ROM can be obtained. ★

Note The turbo access manager (μPD71P301) is maintenance service only.

Fig. 2-1 Memory Map



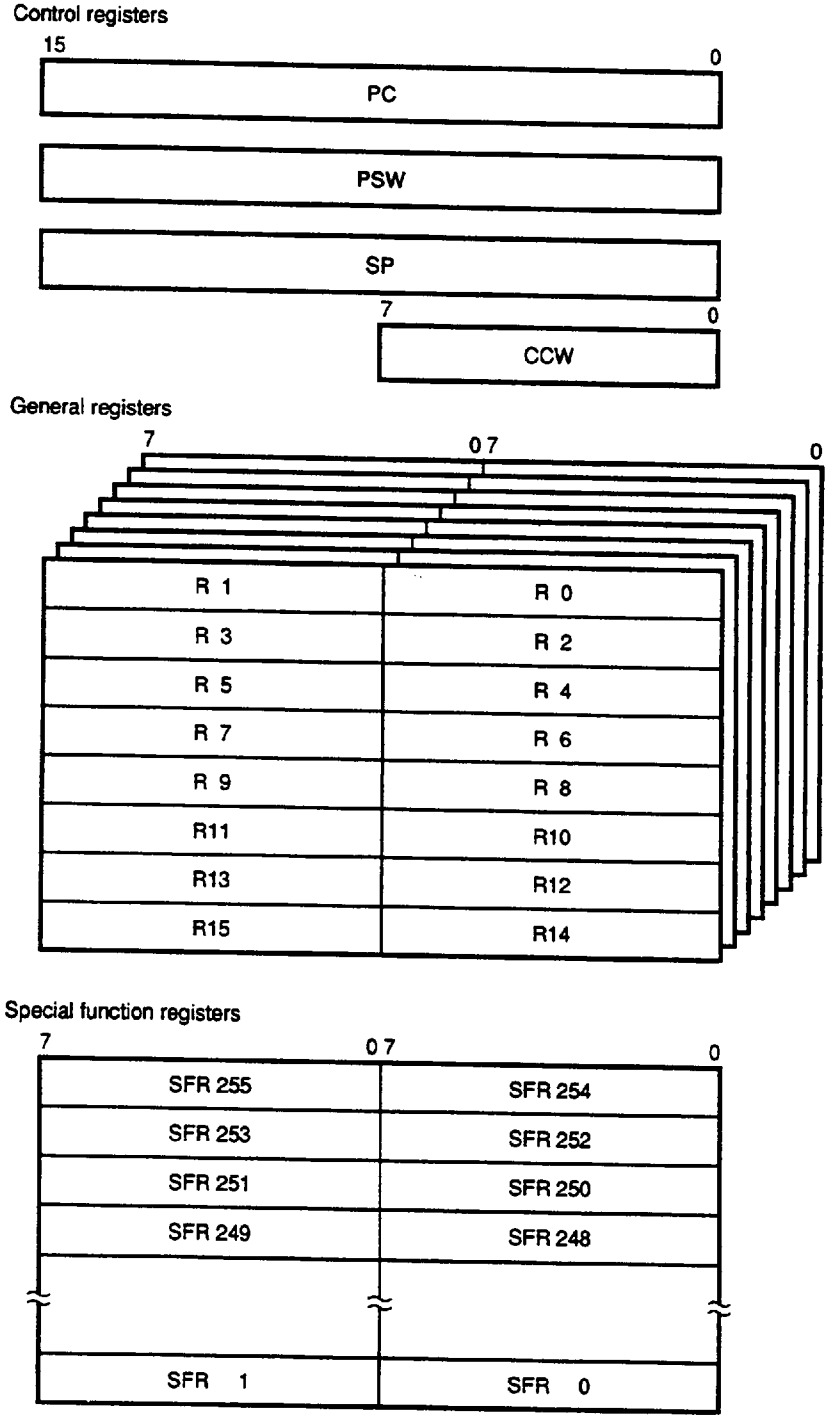
Note Access in the external memory expansion mode.

★ **Caution When word access (including the stack operation) to the main RAM space (FE00H to FEFFH) is executed, the addresses specified in the operand must be even numbers.**

2.2 PROCESSOR REGISTER

There are three groups of registers: a general register group consisting of eight banks, each of which consists of 16 8-bit registers, control register group consisting of an 8-bit register and three 16-bit registers, and a special function register group consisting of registers having special functions such as I/O mode registers for the peripheral hardware.

Fig. 2-2 Register Configuration



Remark The CCW of the control register group is mapped in the special function register (SFR) area.

2.2.1 Control Register

The control register group controls the program sequence, status, and stack memory and modifies operand addressing. This group consists of an 8-bit register and three 16-bit registers.

(1) Program counter (PC)

The program counter is a 16-bit register which holds 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, the contents of the immediate data or register are set in the PC. Input to the $\overline{\text{RESET}}$ pin sets the data in the reset vector table at 0000H and 0001H in the PC and makes a branch.

(2) Program status word (PSW)

The program status word (PSW) is a 16-bit register consisting of flags set or reset according to the result of executing an instruction. Read and write operations are performed by the eight high-order bits (PSWH) or eight low-order bits (PSWL). Flags are operated by the bit manipulation instructions. When an interrupt request is issued and when a BRK instruction is executed, the PSW is automatically saved in the stack and an RETI or RETB instruction is used to return control.

Input to the $\overline{\text{RESET}}$ pin resets all bits.

Fig. 2-3 PSW Configuration

	7	6	5	4	3	2	1	0
PSWH	UF	RBS2	RBS1	RBS0	0	0	0	0
	7	6	5	4	3	2	1	0
PSWL	S	Z	RSS	AC	IE	P/V	LT	CY

(a) Interrupt priority level transition flag (LT)

This flag controls the interrupt priority. To maintain normal operation of the interrupt control circuit, do not operate this flag with the program.

(b) Carry flag (CY)

This flag is set when the result of executing an arithmetic/logical instruction generates a carry of bit 7 or bit 15 or a borrow into bit 7 or bit 15. Otherwise, this flag is reset. This flag can be tested with a conditional branch instruction.

This flag also functions as a bit accumulator when a bit manipulation instruction is executed.

(c) Zero flag (Z)

This flag is set when the arithmetic/logical operation result is zero. Otherwise, this flag is reset. This flag can be tested with a conditional branch instruction.

(d) Sign flag (S)

This flag is set when the most significant bit after arithmetic/logical operation is 1. Otherwise, this flag is reset. This flag can be tested with a conditional branch instruction.

(e) Parity/overflow flag (P/V)

This flag is set if an overflow or underflow occurs in 2's complement when an arithmetic instruction is executed. Otherwise, this flag is reset. (Overflow flag operation)

This flag is set if the number of set bits as the result of executing a logical instruction is an even number. Otherwise, this flag is reset. (Parity flag operation)

This flag can be tested with a conditional branch instruction.

(f) Auxiliary carry flag (AC)

This flag is set when the result of executing an arithmetic/logical instruction generates a carry of bit 3 or a borrow into bit 3. Otherwise, this flag is reset. This flag can be tested with a conditional branch instruction.

(g) Register set selection flag (RSS)

This flag specifies a general register which functions as the X, A, C, or B register. The setting of RSS changes the correspondence between the function register and the absolute register, as shown in Table 2-1. Changing the RSS flag, therefore, can enable another register set (X, A, C, B) to be used.

(h) Interrupt request enable flag (IE)

This flag enables or disables an interrupt request. Executing an EI instruction sets this flag. Executing a DI instruction or receiving an interrupt resets this flag.

(i) Register bank selection flag (RBS0-RBS2)

This 3-bit flag selects one of eight register banks (RBANK0 to RBANK7).

(j) User flag (UF)

This flag controls the program. The flag is set or reset on the user program.

(3) Stack pointer (SP)

The stack pointer is a 16-bit register which holds the first address of the memory stack area (LIFO format). The SP is manipulated by a dedicated instruction.

The stack pointer is decremented before data is saved (written) into the stack memory, and is incremented after data is read from the stack memory.

As input to the $\overline{\text{RESET}}$ pin causes the SP to become undefined, the SP must be set before calling a subroutine.

(4) CPU control word (CCW)

The CPU control word is an 8-bit register which consists of flags related to CPU control. The CCW is mapped in the special function register area and can be controlled by the software. Input to the $\overline{\text{RESET}}$ pin resets all bits.

Fig. 2-4 CCW Configuration

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	TPF	0	CCW

- Table position flag (TPF)

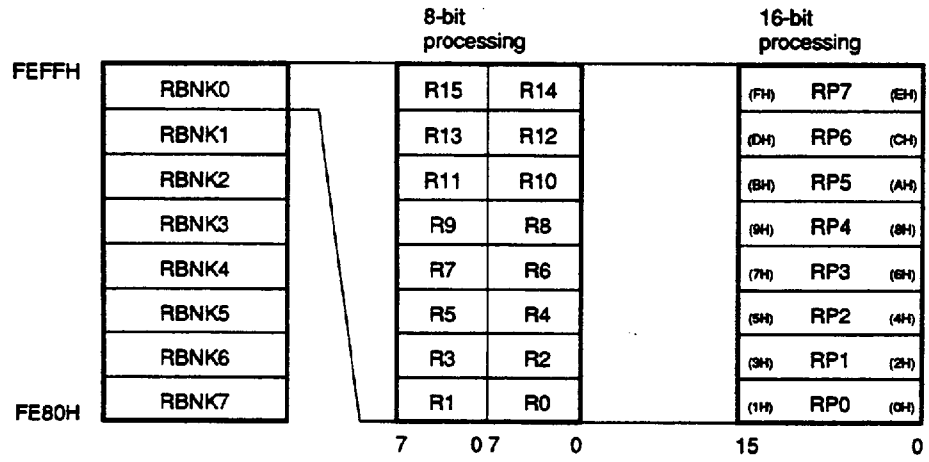
  This flag specifies the memory area used as an interrupt vector table area and CALLT instruction table area.

  After input to the $\overline{\text{RESET}}$ pin, the TPF is reset. The addresses from 0000H to 007FH are used as a table area. Setting the TPF via software can use the addresses from 8002H to 807FH of the external memory area as a table area, instead of the addresses from 0002H to 007FH. However, since the BRK instruction, operation code trap interrupt, and reset input vector table are fixed to 003EH, 003CH, and 0000H, respectively, they are not affected by the TPF.

2.2.2 General Register

The 128-byte general register group which consists of eight banks is mapped in the specific area (FE80H to FEFFH) of the internal RAM space. Each bank consists of 16 8-bit general registers.

Fig. 2-5 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 2-1 is assigned to each 8-bit register (16 registers) to identify each characteristic. 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 a counter. 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 the register having a specific function, as shown in Table 2-1.

When the program is described by a function name, use of the RSS flag causes another X, A, C, or B register to be used.

The μPD78327 and μPD78328 can implement two types of addressing: implied addressing and register addressing. Implied addressing is performed as process data addressing by a function name which places much importance on the specific function of each register. Register addressing is performed by an absolute name which aims to create a program which is easy to describe and which performs less data transfer operations, enabling high-speed data processing.

Table 2-1 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

2.2.3 Special Function Register (SFR)

The special function register group consists of the registers to which special functions are given such as mode and control registers (CCW) for the peripheral hardware.

The special function register group is assigned to the 256-byte space from FF00H to FFFFH. Short direct memory addressing is applicable to the 32-byte area from FF00H to FF1FH, allowing short-word data processing.

Bit manipulation, arithmetic, and move instructions can be executed in all the areas. An external memory from the 16-byte area (FFD0H to FFDFH) is accessed by SFR addressing. Thus, short-word instructions can manipulate bits of an external device and access an external memory.

Table 2-2 lists the special function registers (SFR). The items in Table 2-2 mean:

- Abbreviation : A symbol indicating the address of a built-in special function register
  This can be specified in the operand field of an instruction.

- R/W : Indicates whether data can be read from the special function register and/or data can be written into the register.
  R/W: Can be read and written.
  R : Can be read. (The bits of the register can be tested.)
  W : Can be written.

- Manipulation bit unit : Indicates the unit of bits when manipulating the special function register.
  The SFR which can be manipulated in units of 16 bits can be specified in the sfrp operand. An even address is specified for the address specification.
  The SFR which can be manipulated bit by bit can be specified by a bit manipulation instruction.

- At resetting : Indicates the status of each register for the input to the $\overline{\text{RESET}}$ pin.

Cautions 1. The addresses to which a special function register is not assigned in the area from FF00H to FFFFH cannot be accessed.

2. Do not write data into the register which is only used for data reading. If an attempt is made to write data into such registers, the internal circuit may not operate normally.

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

Address	Special function register (SFR) name	Abbreviation	R/W	Manipulation bit unit			At resetting
				1	8	16	
FF00H	Port 0	P0	R/W	o	o		Undefined
FF02H	Port 2	P2	R		o		Undefined
FF03H	Port 3	P3	R/W	o	o		Undefined
FF04H	Port 4	P4	R/W	o	o		Undefined
FF05H	Port 5	P5	R/W	o	o		Undefined
FF07H	Port 7	P7	R		o		Undefined
FF08H	Port 8	P8	R/W	o	o		Undefined
FF09H	Port 9	P9	R/W	o	o		Undefined
FF0CH	Timer register 2	TM2	R			o	0000H
FF1CH	Capture/compare register 10	CC10	R/W			o	Undefined
FF20H	Port 0 mode register	PM0	W		o		FFH
FF23H	Port 3 mode register	PM3	W		o		xxx1 1111B
FF25H	Port 5 mode register	PM5	W		o		FFH
FF28H	Port 8 mode register	PM8	W		o		FFH
FF29H	Port 9 mode register	PM9	W		o		xxxx 1111B
FF2AH	Timer register 0	TM0	R			o	0000H
FF2CH	Timer register 1	TM1	R			o	0000H
FF43H	Port 3 mode control register	PMC3	W		o		xxx0 0000B
FF48H	Port 8 mode control register	PMC8	W		o		00H
FF4CH	Baud rate generator	BRG	R/W	o	o		Undefined
FF60H	Port 0 buffer register L	P0L	R/W	o	o		Undefined
FF61H	Port 0 buffer register H	P0H	R/W	o	o		Undefined
FF62H	Port read control register	PRDC	R/W		o		00H
FF63H	Real-time output port control register	RTPC	R/W	o	o		00H
FF64H	PWM control register	PWMC	R/W	o	o		00H
FF66H	PWM buffer register	PWMB	R/W	o	o		Undefined
FF68H	A/D converter mode register	ADM	R/W	o	o		00H
FF6AH	A/D conversion result register (for 16-bit access)	ADCR	R			o	Undefined
FF6BH	A/D conversion result register (for high-order 8-bit access)	ADCRH	R		o		Undefined
FF70H	Compare register 00 set	CM00S	R/W			o	Undefined
FF72H	Compare register 01 set	CM01S	R/W			o	Undefined
FF74H	Compare register 02 set	CM02S	R/W			o	Undefined
FF76H	Compare register 03 set	CM03S	R/W			o	Undefined
FF78H	Compare register 04 set	CM04S	R/W			o	Undefined
FF7AH	Compare register 05 set	CM05S	R/W			o	Undefined
FF7CH	Compare register 06	CM06	R/W			o	Undefined

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

Address	Special function register (SFR) name	Abbreviation		R/W	Manipulation bit unit			At resetting
					1	8	16	
FF7EH	Compare register 20	CM20		R/W			o	Undefined
FF80H	Clock synchronous serial interface mode register	CSIM		R/W	o	o		00H
FF82H	Serial bus interface control register	SBIC		R/W	o	o		00H
FF86H	Serial I/O shift register	SIO		R/W	o	o		Undefined
FF88H	Asynchronous serial interface mode register	ASIM		R/W	o	o		80H
FF8AH	Asynchronous serial interface status register	ASIS		R	o	o		00H
FF8CH	Serial reception register (UART)	RXB		R		o		Undefined
FF8EH	Serial transmission shift register (UART)	TXS		W		o		Undefined
FF90H	Compare register 00 reset	CM00R		R/W			o	Undefined
FF92H	Compare register 01 reset	CM01R		R/W			o	Undefined
FF94H	Compare register 02 reset	CM02R		R/W			o	Undefined
FF96H	Compare register 03 reset	CM03R		R/W			o	Undefined
FF98H	Compare register 04 reset	CM04R		R/W			o	Undefined
FF9AH	Compare register 05 reset	CM05R		R/W			o	Undefined
FFB0H	Timer control register 0	TMC0		R/W	o	o		00H
FFB1H	Baud rate generator mode register	BRGM		R/W	o	o		00H
FFB2H	Timer control register 1	TMC1		R/W	o	o		04H
FFB4H	Timer unit mode register	TUM		R/W	o	o		00H
FFBAH	Timer output register	TOUT		R/W	o	o		FFH
FFC0H	Standby control register	STBC[Note]		R/W	o	o		0000 ×000B
FFC1H	CPU control word	CCW		R/W	o	o		00H
FFC2H	Watchdog timer mode register	WDM[Note]		R/W	o	o		00H
FFC4H	Memory expansion mode register	MM		R/W	o	o		00H
FFC6H	Programmable wait control register	PWC		R/W	o	o		22H
FFC9H	Fetch cycle control register	FCC		R/W	o	o		00H
FFD0H to FFDFH	External access area			R/W	o	o		Undefined
FFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	o	o	o	00H
FFE1H	Interrupt request flag register 0H	IF0H		R/W	o	o		00H
FFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	o	o	o	00H
FFE3H	—	—		—	—	—		—
FFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	o	o	o	FFH
FFE5H	Interrupt mask flag register 0H	MK0H		R/W	o	o		FFH

Note Data can be written into this register for special instructions.

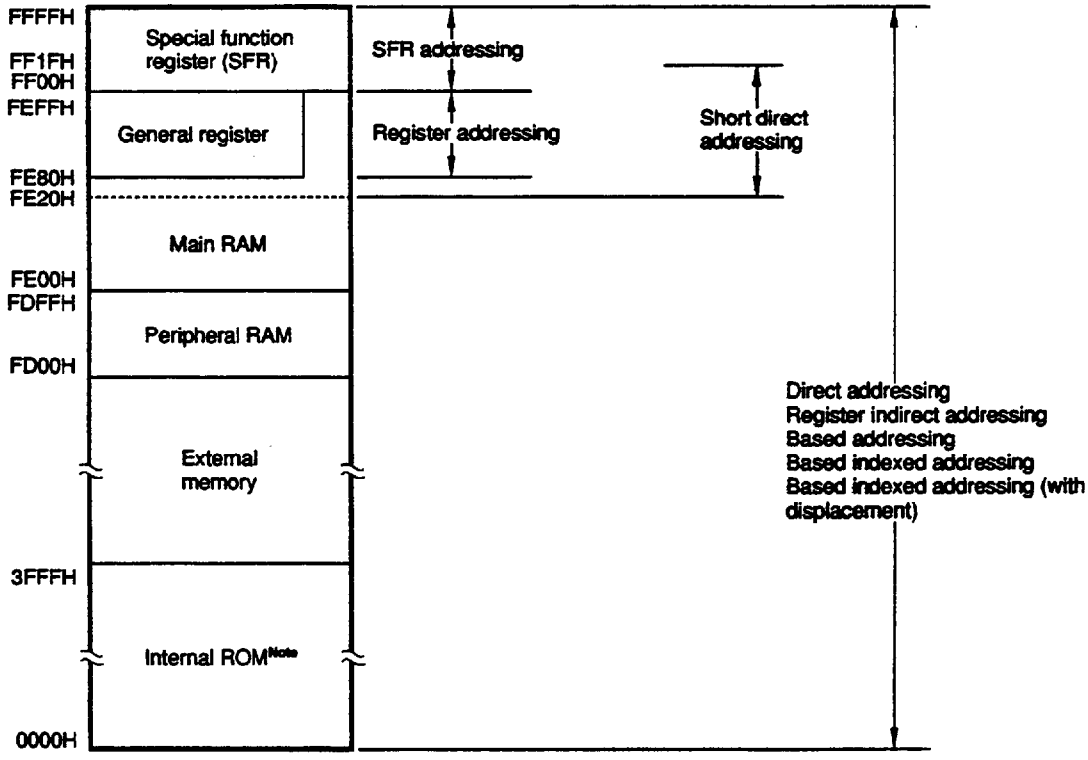
Table 2-2 Special Function Registers (3/3)

Address	Special function register (SFR) name	Abbreviation		R/W	Manipulation bit unit			At resetting
					1	8	16	
FFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	o	o	o	××××1111B
FFE7H	—	—		—	—	—		—
FFE8H	Priority specification buffer register 0L	PB0L	PB0	R/W	o	o	o	00H
FFE9H	Priority specification buffer register 0H	PB0H		R/W	o	o		00H
FFEAH	Priority specification buffer register 1L	PB1L	PB1	R/W	o	o	o	00H
FFEBH	—	—		—	—	—		—
FFECH	Interrupt processing mode specification register 0L	ISM0L	ISM0	R/W	o	o	o	00H
FFEDH	Interrupt processing mode specification register 0H	ISM0H		R/W	o	o		00H
FFEEH	Interrupt processing mode specification register 1L	ISM1L	ISM1	R/W	o	o	o	00H
FFEFH	—	—		—	—	—		—
FFF0H	Context switching enable register 0L	CSE0L	CSE0	R/W	o	o	o	00H
FFF1H	Context switching enable register 0H	CSE0H		R/W	o	o		00H
FFF2H	Context switching enable register 1L	CSE1L	CSE1	R/W	o	o	o	00H
FFF3H	—	—		—	—	—		—
FFF4H	External interrupt mode register 0	INTM0		R/W	o	o		00H
FFF8H	In-service priority register	ISPR		R		o		00H
FFE9H	Priority specification register	PRSL		R/W	o	o		00H

2.3 DATA MEMORY ADDRESSING

For the μPD78327 and μPD78328, the internal RAM space (FD00H to FEFFH) and special function register area (FF00H to FFFFH) are mapped in the area from FD00H to FFFFH. Short direct addressing is applicable to the space (FE20H to FF1FH) of a data memory, and direct high-speed addressing is possible with single-byte data in an instruction word.

Fig. 2-6 Addressing of a Data Memory



Note External memory is assigned for the μPD78327 or when $\overline{EA}$ is low.

★ **Caution When word access (including the stack operation) to the main RAM space (FE00H to FEFFH) is executed, the addresses specified in the operand must be even numbers.**

2.3.1 General-Register Addressing

A general register group consists of eight banks made up of a set of registers, that is, 16 8-bit registers or eight 16-bit registers.

General-register addressing is performed by the register set selection flag (RSS), register bank selection flag (RBS0 to RBS2) in the PSW, and the 3-bit or 4-bit register specification field in the instruction word.

2.3.2 Short Direct Addressing

Short direct addressing, which can directly perform addressing at high speed with single-byte data in an instruction word, is applicable to the space FE20H to FF1FH. A short direct memory is accessed as 8-bit data or 16-bit data. When accessing a short direct memory as 16-bit data, only double-byte data specified by two contiguous addresses is allowed. In this case, do not specify an odd address for the address specification data.

2.3.3 Special Function Register (SFR) Addressing

This addressing is applicable to the case where the special function register (SFR) mapped in the SFR area (FF00H to FFFFH) is manipulated. This addressing is done by single-byte data in the instruction word corresponding to the eight low-order bits of the address of the special function register. When accessing a 16-bit SFR, double-byte data specified by two contiguous addresses is accessed.

3. BLOCK FUNCTION

3.1 BUS CONTROL UNIT (BCU)

The bus control unit (BCU) activates a required bus cycle according to the physical address obtained from the execution unit (EXU). When the EXU does not issue a bus cycle activation request, the BCU generates an address required for prefetching an instruction. The prefetched instruction code is fetched into the instruction queue.

3.2 EXECUTION UNIT (EXU)

The execution unit (EXU) controls address calculation, arithmetic/logical operations, and data transfer by a microprogram. A 256-byte main RAM area is built into the EXU.

The 256-byte main RAM area in the EXU can be accessed at higher speed with an instruction than 256-byte peripheral RAM areas.

3.3 ROM/RAM

This area consists of a 16K-byte ROM area and 256-byte peripheral RAM area.

Access to this ROM area can be inhibited by using the $\overline{EA}$ pin.

3.4 INTERRUPT CONTROLLER

The interrupt controller processes various interrupt requests (NMI, INTP0 to INTP2) issued from peripheral hardware and external device with the context switching, vector interrupt, or macro service function.

The interrupt controller also specifies the 3-level interrupt priority.

3.5 PORT FUNCTION

Table 3-1 lists the digital I/O ports.

Each port allows bit manipulations as well as 8-bit I/O data manipulations, thus enabling a wide variety of control.

Table 3-1 Functions and Features of the Ports

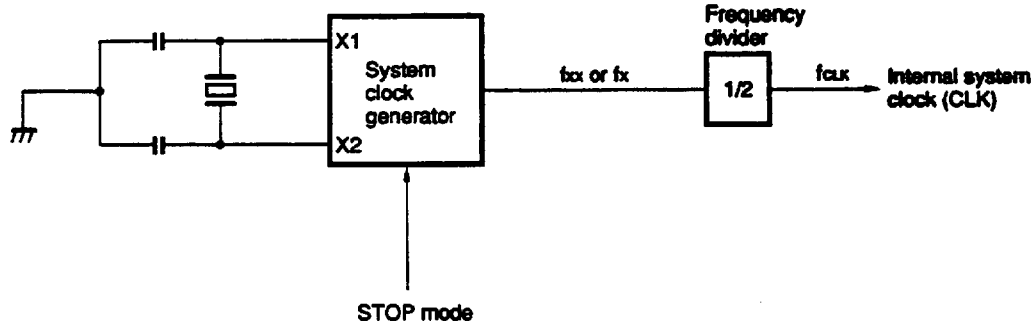
Port name	Function	Feature	Remarks
Port 0	4/8-bit I/O	Specifiable as input or output bit by bit. Specifiable also as real-time output port in units of 4 bits.	Shares pins with RTP0-RTP7.
Port 2	3-bit input	Input port pins. Functions also as external interrupt input.	Shares pins with NMI, INTP0, and INTP1/TI.
Port 3	5-bit I/O	Specifiable as port pin or control pin bit by bit.	Shares pins with TxD, RxD, SO/SB0, SI/SB1, and $\overline{SCK}$.
Port 4	8-bit I/O	Specifiable as input or output in units of 8 bits. Functions as multiplexed address/data bus (AD0-AD7) in external memory expansion mode.	Always functions as multiplexed address/ data bus with μPD78327.
Port 5	8-bit I/O	Specifiable as input or output bit by bit. Functions as address bus (A8-A15) in external memory expansion mode. Pins not used as address bus can be used as port.	Always functions as address bus with μPD78327.
Port 7	8-bit input	Input port pins. Functions also as analog input to A/D converter.	Shares pins with ANI0-ANI7.
Port 8	8-bit I/O	Specifiable as port pin or control pin bit by bit.	Shares pins with TO0-TO5, TO6/ INTP2, and TO7/ PWM.
Port 9	4-bit I/O	Specifiable as input or output bit by bit. P90 functions as $\overline{RD}$ output, and P91 functions as $\overline{WR}$ output in external memory expansion mode. P92 functions as $\overline{TAS}$ output, and P93 functions as TMD output in external memory high-speed fetch mode.	With μPD78327, P90 always functions as $\overline{RD}$ output, and P91 always functions as $\overline{WR}$ output.

3.6 CLOCK GENERATOR

The clock generator generates and controls an internal system clock (CLK) supplied to the CPU.

The clock generator is configured as shown in Fig. 3-1.

Fig. 3-1 Block Diagram of the Clock Generator



Remarks 1. fxx : Crystal oscillator frequency
2. fx : External clock frequency
3. fCLK: Internal system clock frequency

The system clock generator generates a clock signal with a crystal resonator connected to the X1 and X2 pins. The system clock generator stops oscillation when it is set to the standby mode (STOP).

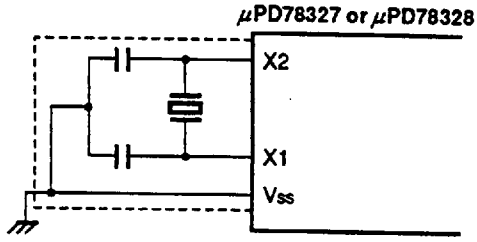
An external clock can be applied. In this case, a clock signal is to be applied to the X1 pin, with the inverted signal to be applied to the X2 pin. However, the X2 pin can be left open.

Caution When using an external clock, do not set the STP bit of STBC.

The frequency divider divides system clock generator output (fxx for the crystal oscillator or fx for an external clock) by two to produce an internal system clock (fCLK).

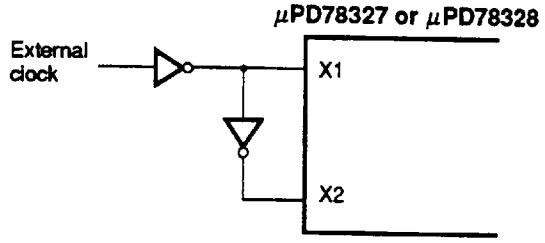
Fig. 3-2 External Circuitry of the System Clock Generator

(a) Crystal oscillator

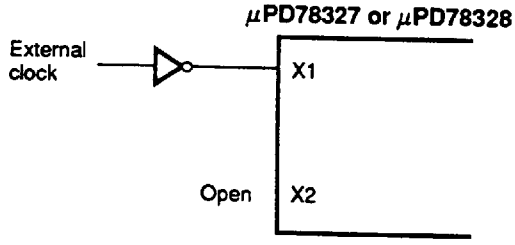


(b) External clock

(i) When an external clock signal input to the X1 pin is inverted and input to the X2 pin



(ii) When the X2 pin is open



Cautions 1. When using the system clock generator, run wires enclosed in broken lines (⬚) in Fig. 3-2 according to the following rules to avoid effects such as stray capacitance:

- Minimize the wiring.
- Never cause the wires to cross other signal lines or run near a line carrying a large varying current.
- Cause the grounding point of the capacitor of the oscillator circuit to have the same potential as Vss. Never connect the capacitor to a ground pattern carrying a large current.
- Never extract a signal from the oscillator.

2. Keep loads such as stray capacitance around wiring away form the X2 pin, when an external clock signal is input to the X1 pin and the X2 pin is open.

3.7 REAL-TIME PULSE UNIT (RPU)

The RPU can measure pulse intervals and frequencies, and can also output programmable pulses (PWM control).

The RPU is built around three timers. One timer has registers suitable for PWM control. In addition, the user can choose from two timer output modes so that the user can match a wide variety of applications in a flexible manner.

The RPU also controls the set or reset timing for port output of the real-time output port (P0).

3.7.1 Configuration

As shown in Fig. 3-3 and 3-4, the RPU mainly consists of timer 0 (TM0), timer 1 (TM1), timer 2 (TM2), and compare registers and compare/capture registers. Timer 0 functions as a 16-bit interval timer/free-running counter. Timer 1 functions as a 16-bit interval timer/free-running counter/event counter. Timer 2 functions as a 16-bit interval timer.

Fig. 3-3 Block Diagram of the RPU in Mode 0

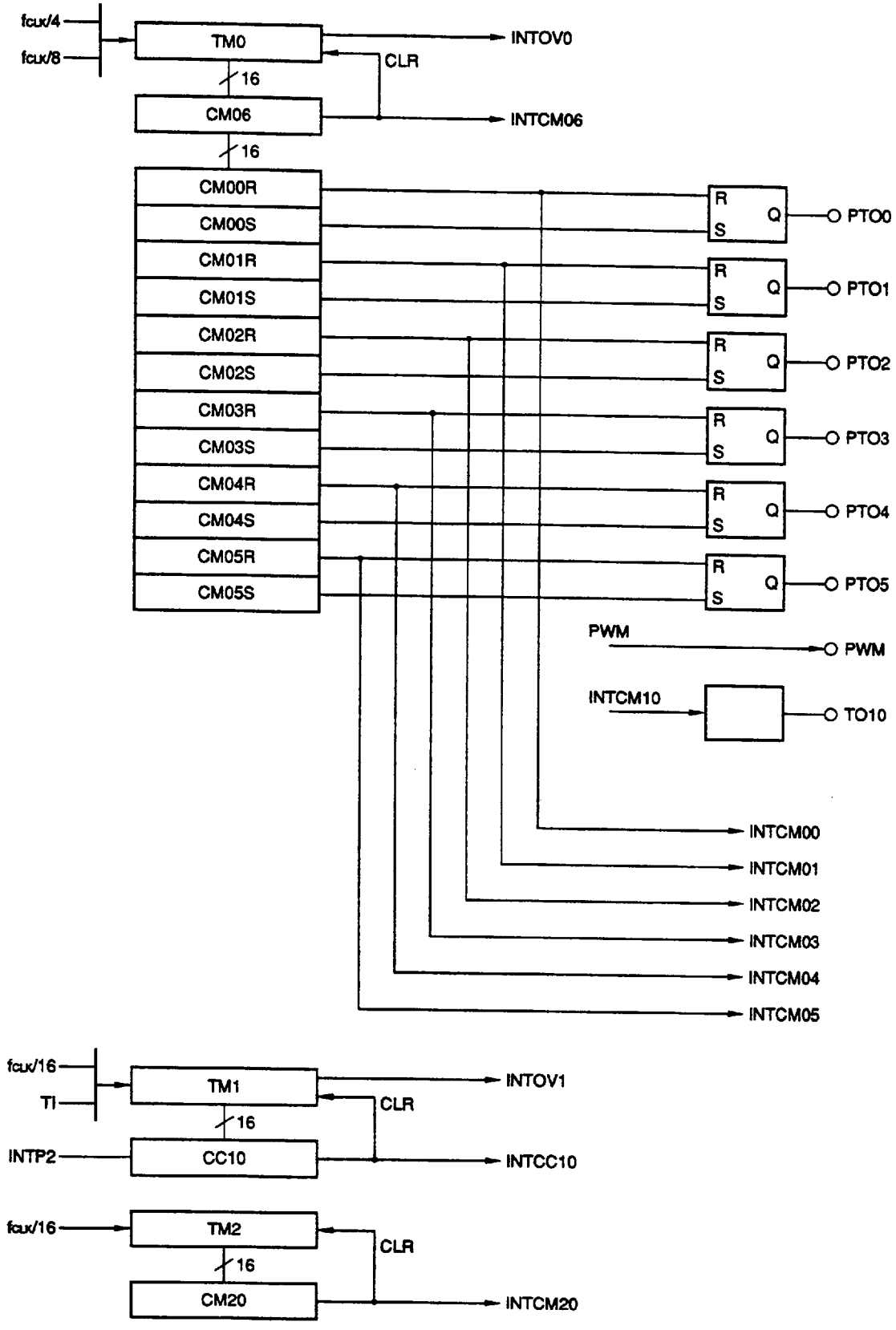
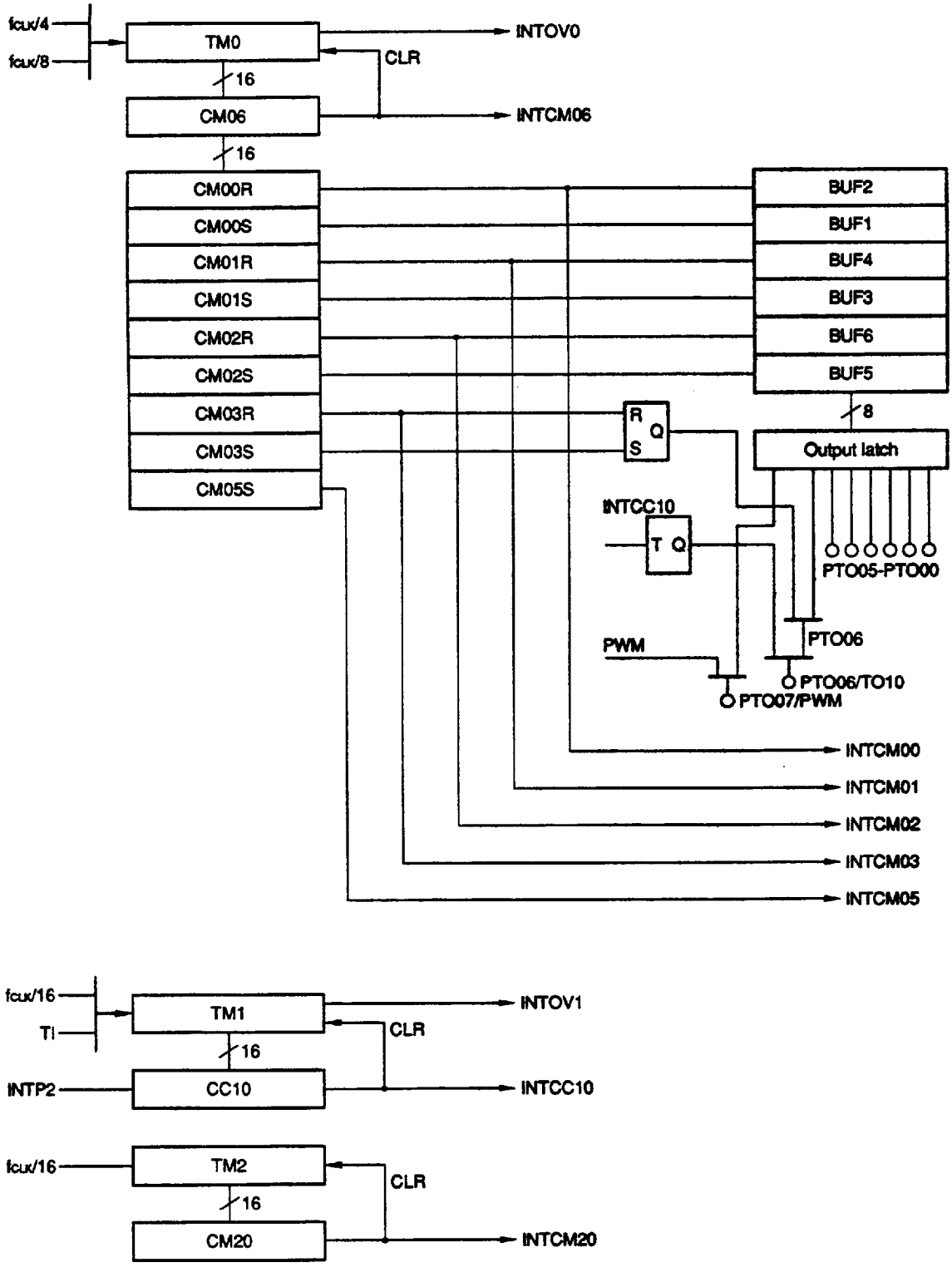


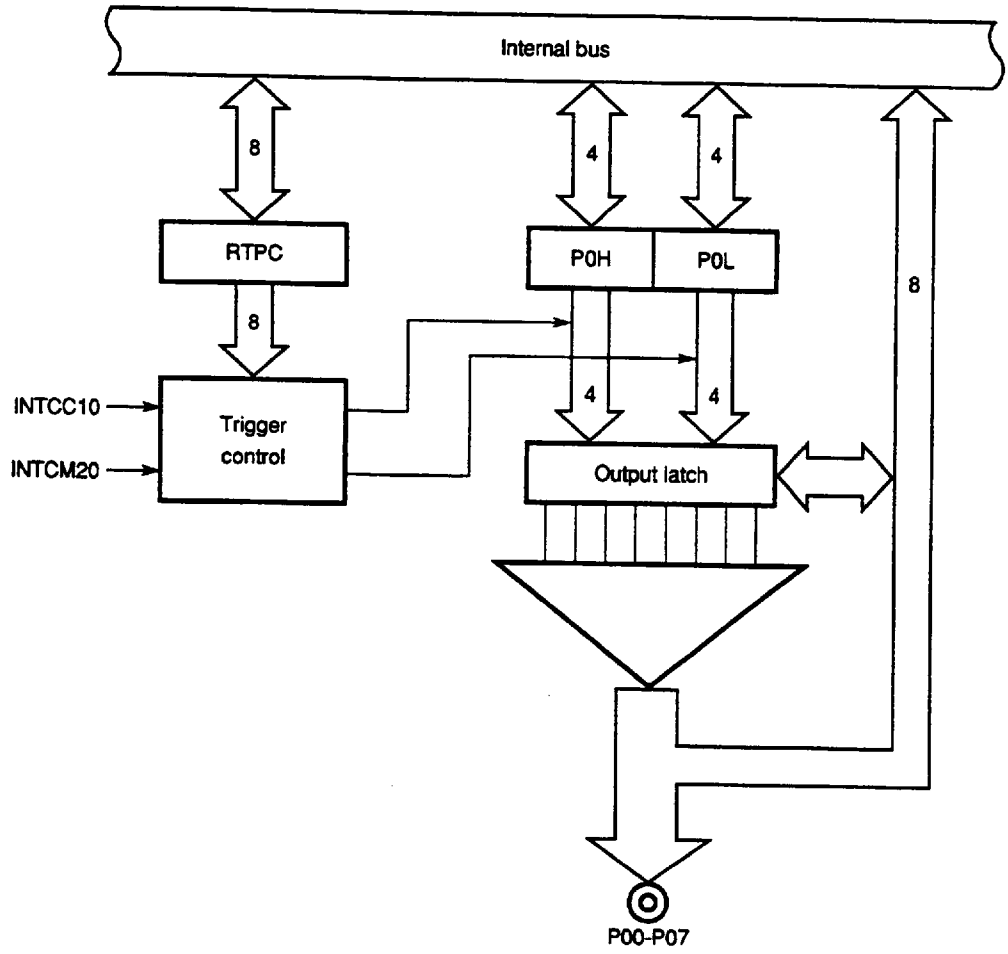
Fig. 3-4 Block Diagram of the RPU in Mode 1



3.7.2 Real-Time Output Function

The real-time output function transfers the contents of a buffer in units of four or eight bits to port 0 in phase with a trigger signal from the real-time pulse unit. The function enables multi-channel synchronous pulse output. Fig. 3-5 shows the block diagram of port 0.

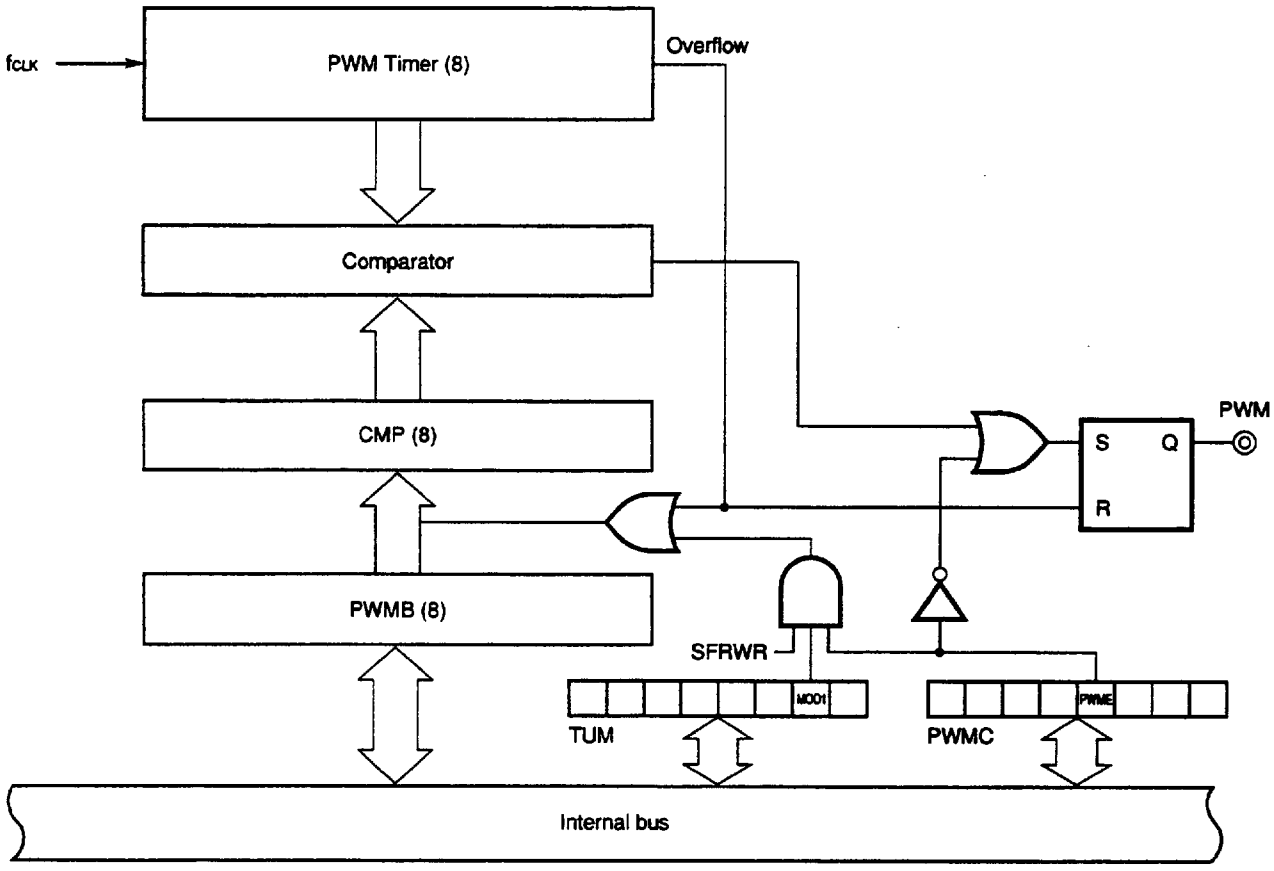
Fig. 3-5 Block Diagram of Port 0



3.7.3 Eight-Bit High-Speed PWM Signal Output Function

The real-time pulse unit contains an 8-bit high speed PWM signal output function, which can change the active level duty within certain cycles. Fig. 3-6 shows the block diagram of PWM signal output.

Fig. 3-6 Block Diagram of PWM Signal Output

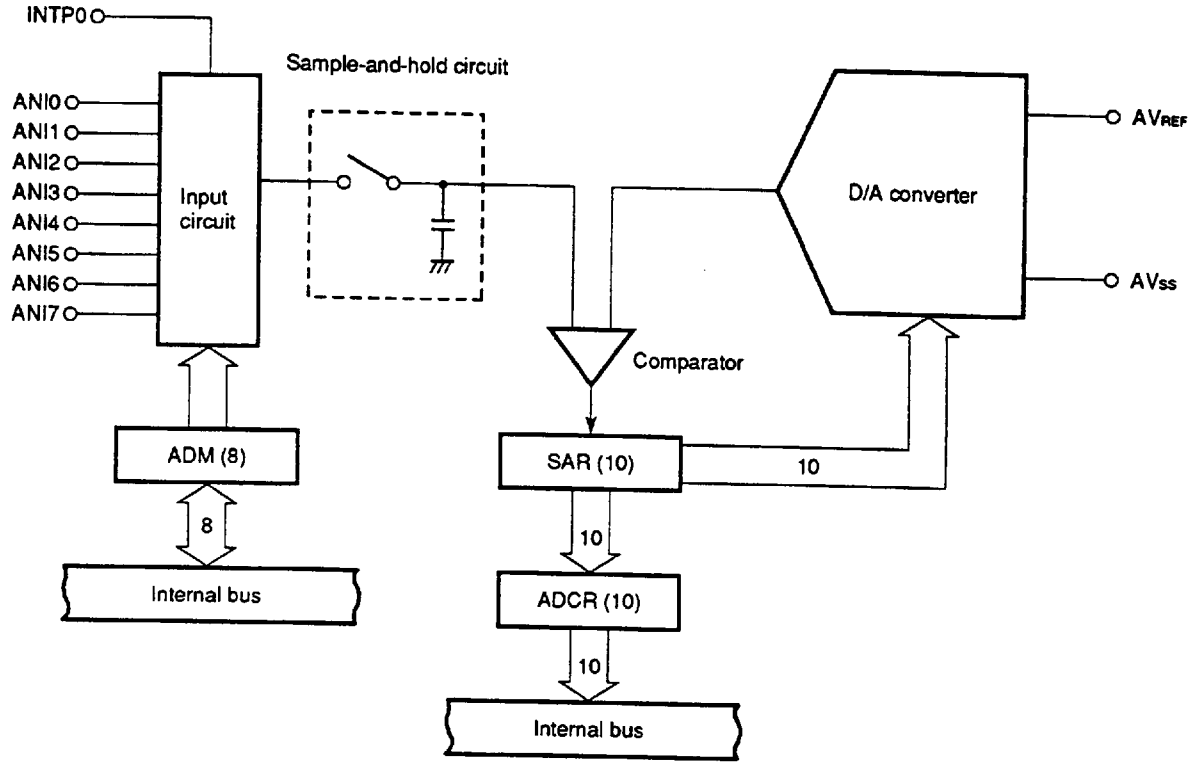


3.8 A/D CONVERTER

The μPD78327 and μPD78328 contain a high-speed, high-resolution 10-bit analog-digital (A/D) converter. The A/D converter has eight analog inputs (ANI0 to ANI7) and an A/D conversion result register (ADCR) that holds conversion results. It can accept an external trigger signal for the start of a conversion.

Upon completion of the conversion, the converter generates an interrupt that can initiate a macro service.

Fig. 3-7 A/D Converter Block Diagram



3.9 SERIAL INTERFACE

The μPD78327 and μPD78328 are provided with the following two separate channels for serial interface functions:

- Asynchronous serial interface
- Synchronous serial interface
  - Three-wire serial I/O mode
  - Serial bus interface mode (SBI mode)

In addition, the μPD78327 and μPD78328 contain a baud rate generator, enabling serial transfer to be performed at a desired rate independently of the operation frequency. The baud rate generator works with both of the two channels of serial interfaces.

The serial transfer rate can be selected between 75 bps and 19.2 kbps according to the setting of the mode register.

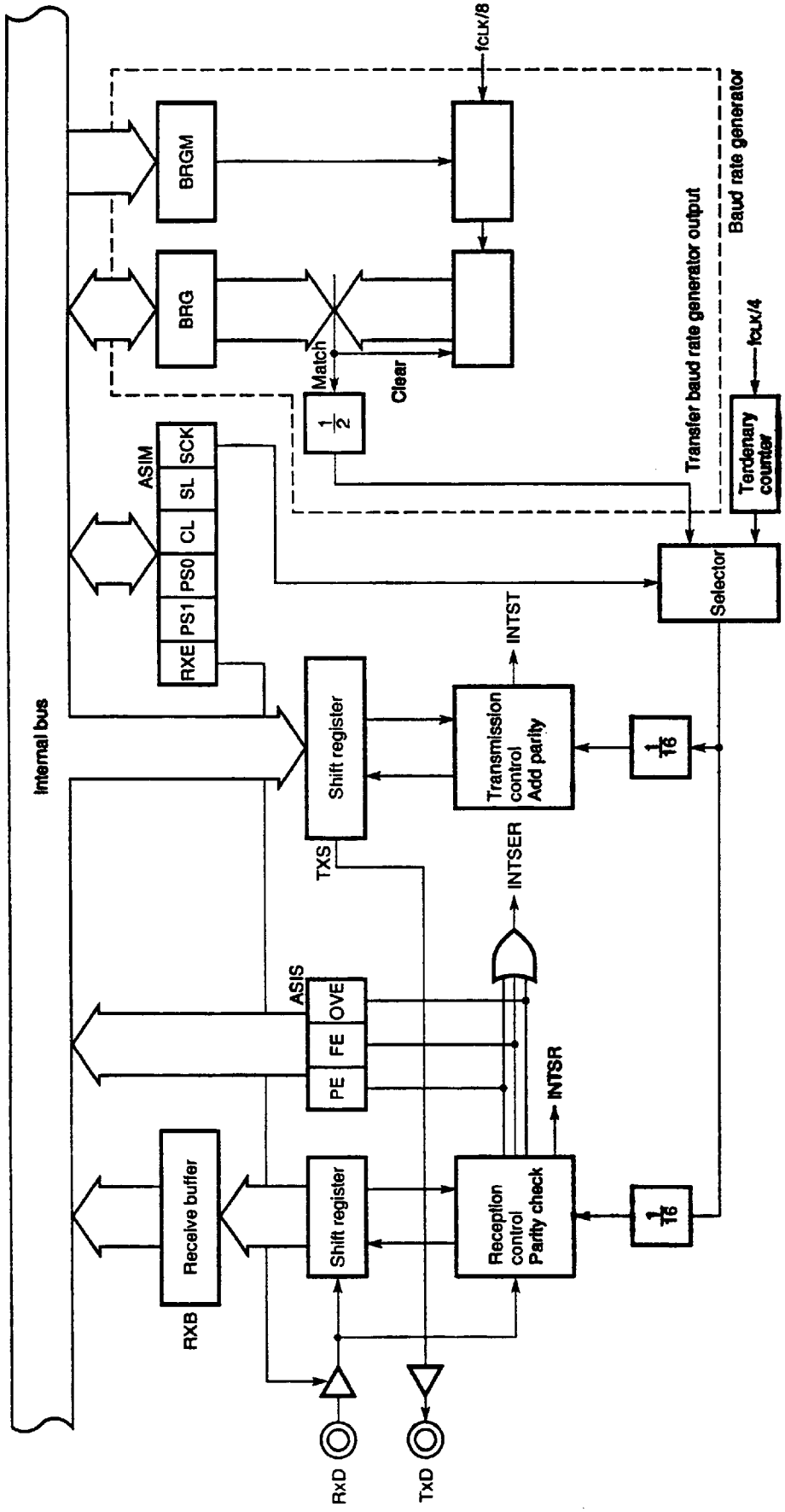


Fig. 3-8 Asynchronous Serial Interface Block Diagram

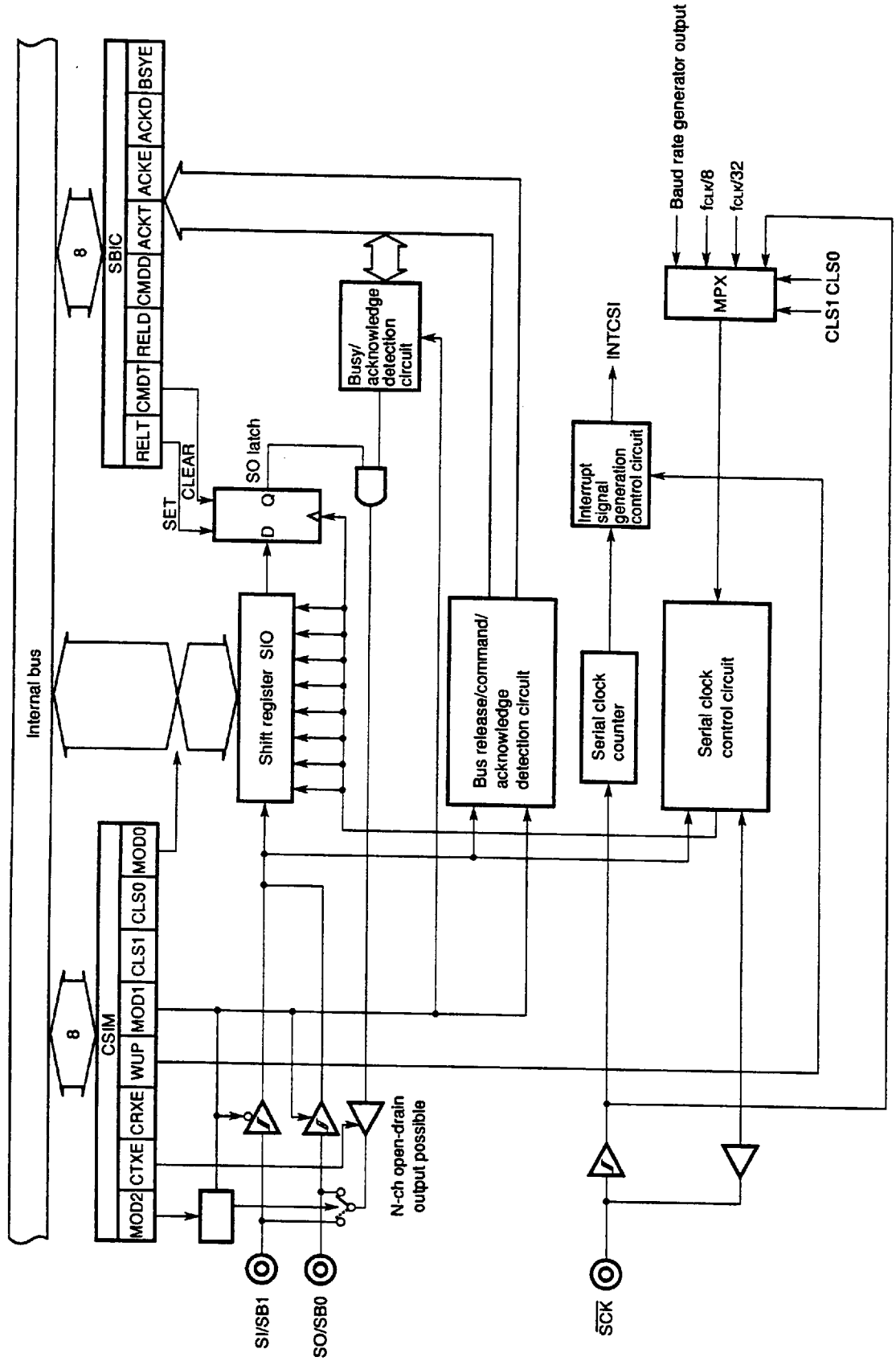


Fig. 3-9 Synchronous Serial Interface Block Diagram

3.10 WATCHDOG TIMER

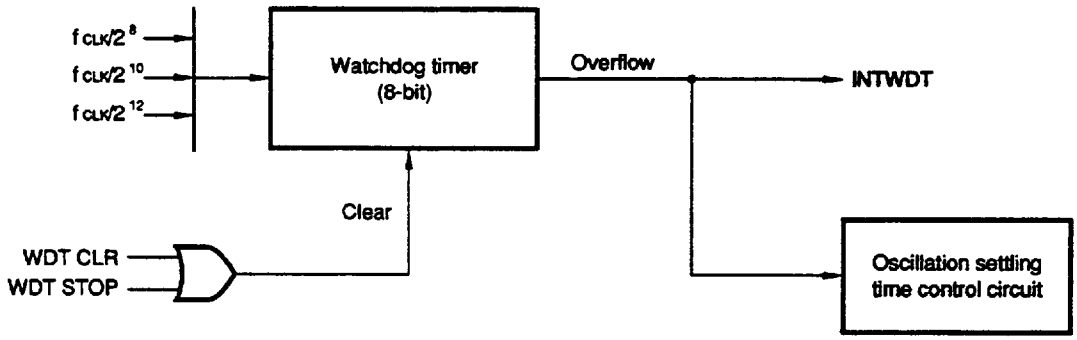
The watchdog timer is designed to prevent program crashes or deadlocks. While no watchdog timer interrupt is observed, the program or system is recognized as operating properly. Each module in a program should contain an instruction to clear (to start) the watchdog timer.

If an instruction to clear the watchdog timer does not clear the watchdog timer within a predetermined time and an overflow occurs in the timer, a watchdog timer interrupt is generated.

The watchdog timer can also be used to guarantee a time required for the oscillator to perform stable oscillation when the stop mode is released.

Fig. 3-10 is a watchdog timer configuration.

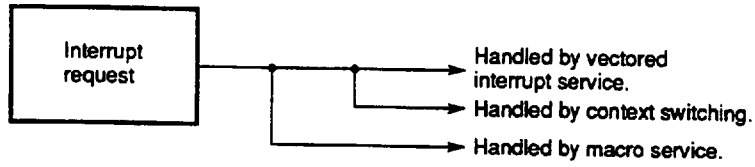
Fig. 3-10 Watchdog Timer Configuration



4. INTERRUPT FUNCTION

4.1 OVERVIEW

The μPD78327 and μPD78328 can handle various interrupt requests issued from on-chip peripheral hardware and the external environment in three processing modes as shown in the following figure.



Interrupt requests are classified into the following three types:

- Nonmaskable interrupt requests
- Maskable interrupt requests
- Interrupt requests by software

Fig. 4-1 shows the processing mode for maskable interrupt requests. Table 4-1 lists processable interrupt sources.

Fig. 4-1 Interrupt Request Processing Mode

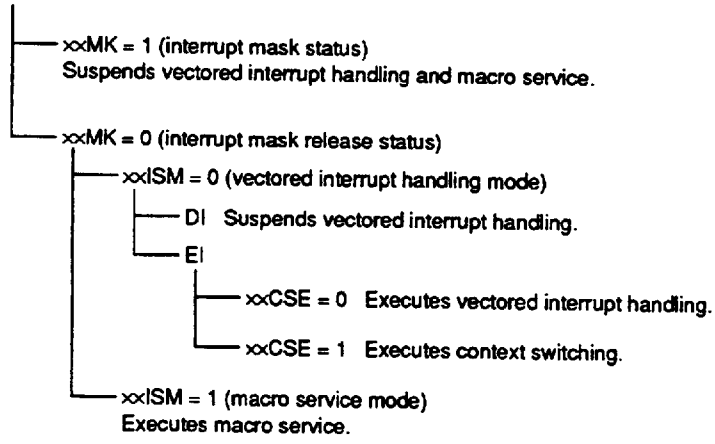


Table 4-1 Interrupt Source List

Interrupt request type	Default priority	Interrupt source		Unit requesting an interrupt	Macro service	Vector table address
		Request signal	Function			
Software	–	–	BRK instruction	–	–	003EH
	–	–	Op-code trap	–	–	003CH
Non-maskable	–	NMI	Input to pin NMI	External interrupt	–	0002H
	–	INTWDT	Watchdog timer	WDT	–	0004H
Maskable	0	INTOV0	Timer 0 overflow	RPU	YES	0006H
	1	INTP0	Input to pin INTP0	External	YES	0008H
	2	INTP1	Input to pin INTP1/TI	External	YES	000AH
	3	INTP2	Input to pin INTP2	External	YES	000CH
	4	INTOV1	Timer 1 overflow	RPU	YES	000EH
	5	INTCM00	CM00R match signal	RPU	YES	0010H
	6	INTCM01	CM01R match signal	RPU	YES	0012H
	7	INTCM02	CM02R match signal	RPU	YES	0014H
	8	INTCM03	CM03R match signal	RPU	YES	0016H
	9	INTCM04	CM04R match signal	RPU	YES	0018H
	10	INTCM05	CM05R/CM05S match signal	RPU	YES	001AH
	11	INTCM06	CM06 match signal	RPU	YES	001CH
	12	INTCC10	CC10 match signal	RPU	YES	001EH
	13	INTCM20	CM20 match signal	RPU	YES	0020H
	14	INTSR	Serial reception end interrupt	UART	YES	0024H
	15	INTST	Serial transmission end interrupt	UART	YES	0026H
	16	INTCSI	Serial transmission and reception end interrupt	CSI	YES	0028H
	17	INTAD	A/D conversion end interrupt	A/D	YES	002AH
–	–	INTSER[Note]	Serial reception error signal	UART	–	_[Note]
Reset	–	RESET	Input to pin $\overline{\text{RESET}}$	–	–	0000H

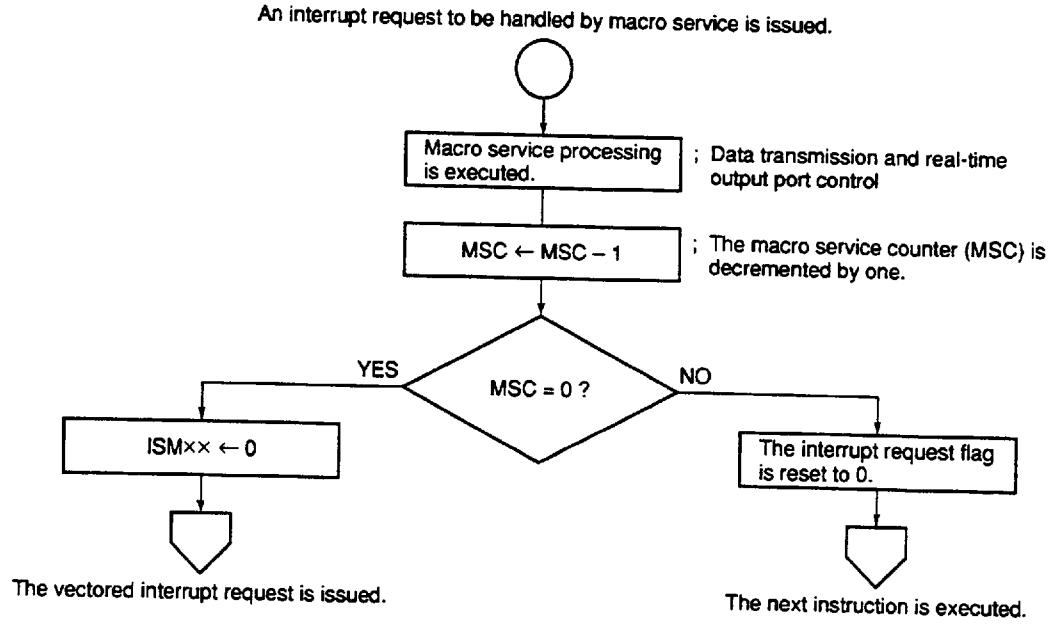
Note Test source. This source does not generate a vectored interrupt.

4.2 MACRO SERVICE

The macro service function is used to operate and transmit data on a hardware basis between the special function register area and the memory space according to an interrupt request.

When the macro service starts, the CPU temporarily stops program execution. One- or two-byte data operation and transmission are automatically performed between the special function register (SFR) and memory. When the macro service ends, the interrupt request flag is reset (0) and the CPU restarts program execution. After the macro service is performed the number of times set in the macro service counter (MSC), a vectored interrupt request is issued.

Fig. 4-2 Example of Macro Service Processing Sequence ★



4.3 CONTEXT SWITCHING FUNCTION

The context switching function selects the specified register bank on a hardware basis. When an interrupt request is issued or when the BRKCS instruction is executed, it makes a branch to the vector address pre-stored in the register bank, and simultaneously stacks the contents of the current PC and PSW in the register bank.

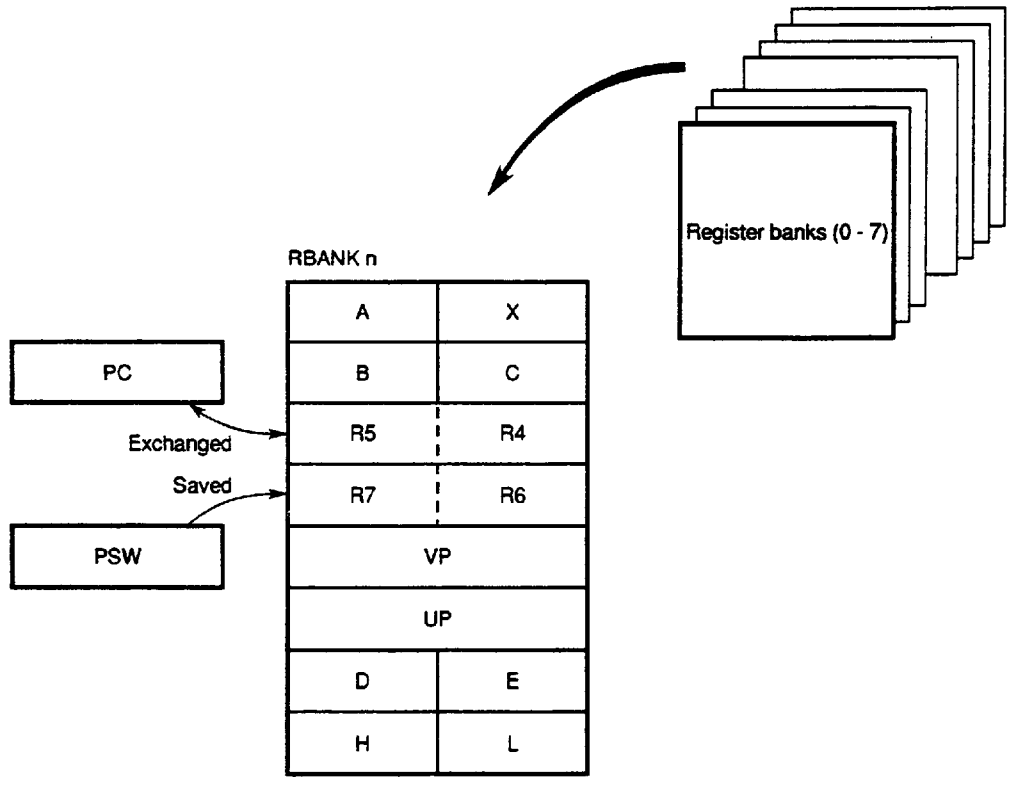
4.3.1 Context Switching Function by an Interrupt Request

Setting the ××CSE bits (1) to be specified corresponding to each interrupt request enables activation of the context switching function.

The register bank which is specified in the three low-order bits of the low address (even address) of the vector table of the interrupt is selected when an interrupt request is issued which is not masked in the EI status and for which the context switching function is enabled. The vector address previously stored in the selected register is transmitted to the PC, and simultaneously the contents of the PC and PSW before transmission are saved in the register bank and the branch is sent to the interrupt service routine.

The RETCS instruction is executed to return.

Fig. 4-3 Context Switching Function by Interrupt Request



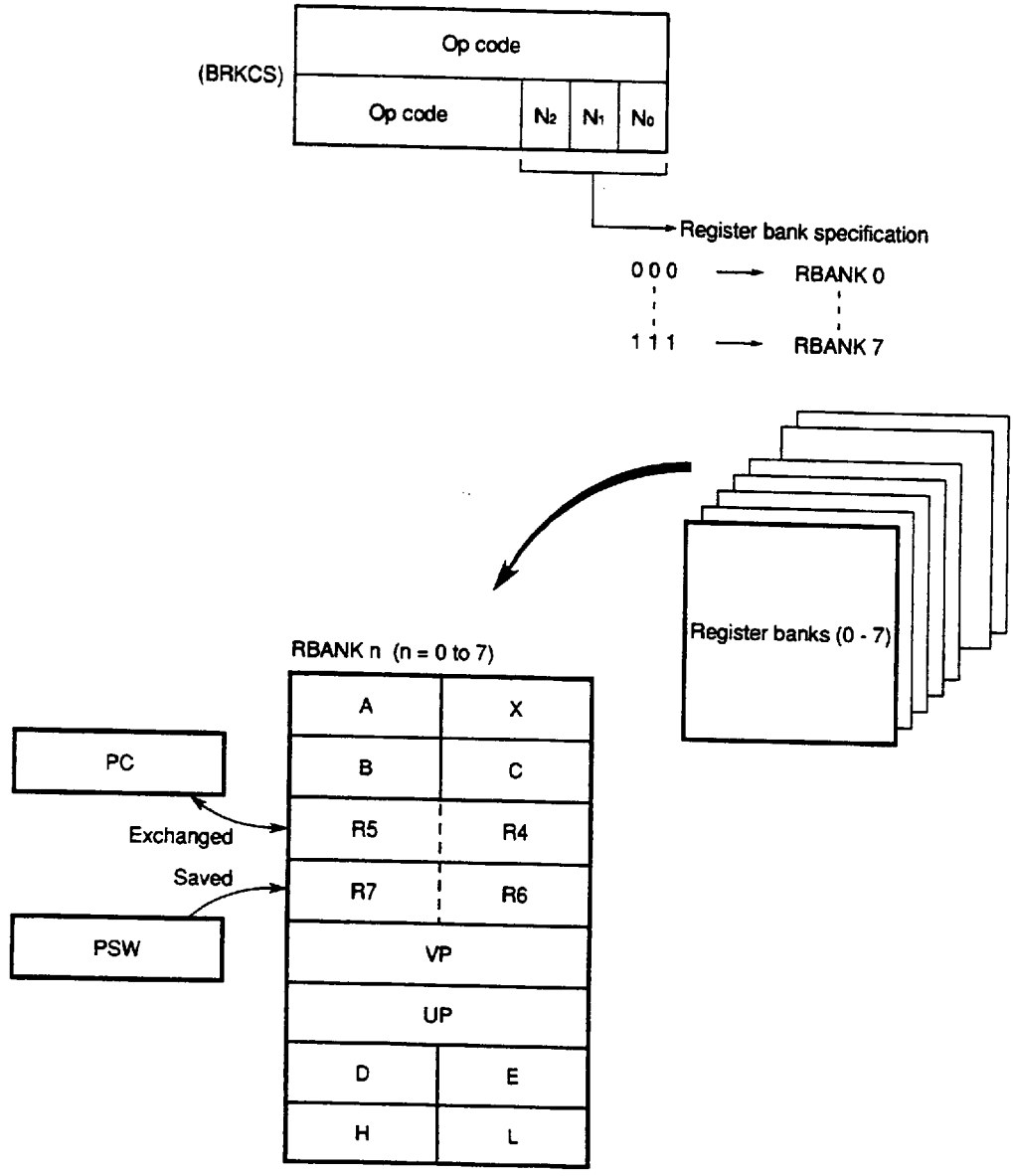
4.3.2 Context Switching Function by a BRKCS Instruction

The context switching function can be activated by executing the BRKCS instruction.

The three low-order bits of immediate data in the second operation code of the BRKCS instruction is used to specify the register bank after context switching. When the BRKCS instruction is executed, the register bank to be specified with three-bit immediate data is selected and the vector address previously stored in the register bank is set in the PC and a branch is made. Simultaneously, the contents of PC and PSW before setting are saved in the register bank.

The RETCSB instruction is executed to return.

Fig. 4-4 Context Switching by a BRKCS Instruction Execution



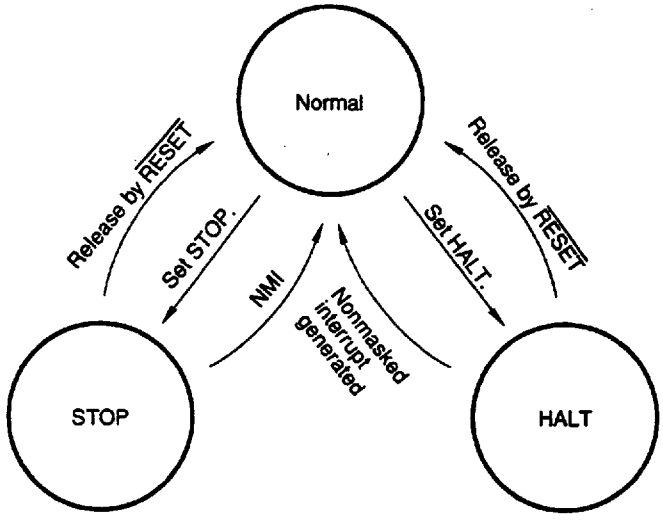
5. STANDBY FUNCTION

The μPD78327 and μPD78328 have a standby function to reduce power consumption of the system. There are the following two types of standby modes.

- HALT mode : in this mode, the CPU operation clock is stopped. Intermittent operation in the normal operation mode can reduce the total system power consumption.
- STOP mode : the oscillator is stopped to stop the entire system.
  Since only a leakage current may flow in this mode, system power consumption can be minimized.

These modes are set by the software. Fig. 5-1 is a transition diagram of the standby modes (STOP and HALT modes).

Fig. 5-1 Transition Diagram of Standby Modes



6. EXTERNAL DEVICE EXPANSION FUNCTION

For the μPD78327 and μPD78328, external devices (data memory, program memory, and peripheral devices) can be expanded in the area (4000H-FCFFH) other than internal ROM or RAM. Tables 6-1 and 6-3 list the pins used for access to an external device and indicate how to assign functions to the pins.

Table 6-1 Assigning Functions to Pins (μPD78328)

$\overline{EA}$ pin	Memory expansion mode register		Fetch cycle control register	Pin function						Remarks
	MM0-MM2	MM7		P40-P47	P50-P57	P90	P91	P92	P93	
1	Port mode	0	00H	General-purpose port						—
		1	Not to be set							
	Expansion mode	0	00H	AD0-AD7	Set to A8-A15 according to the expanded memory size	$\overline{RD}$	$\overline{WR}$	General-purpose port		External device connection mode
		1	Other than 00H					$\overline{TAS}$	TMD	μPD71P301 connection mode

There are four memory expansion modes: 256-, 4K-, 16K-, and 48K-byte expansion modes. The number of bits operating as the address bus can be changed according to the size of memory expanded externally by using pins P50-P57. Pins not used as the address bus can be used as general-purpose I/O ports.

Table 6-2 Port/Address Setting in Port 5 (μPD78328)

P57	P56	P55	P54	P53	P52	P51	P50	External address space
Port	Port	Port	Port	Port	Port	Port	Port	Within 256 bytes
Port	Port	Port	Port	A11	A10	A9	A8	Within 4K bytes
Port	Port	A13	A12	A11	A10	A9	A8	Within 16K bytes
A15	A14	A13	A12	A11	A10	A9	A8	Within about 48K bytes

Table 6-3 Setting Functions to Pins (μPD78327)

$\overline{EA}$ pin	Memory expansion mode register	Fetch cycle control register	Pin function						Remarks
	MM7		P40-P47	P50-P57	P90	P91	P92	P93	
ASTB	—	—	AD0-AD7	A8-A15	$\overline{RD}$	$\overline{WR}$	$\overline{TAS}$	TMD	μPD78328 emulation mode
0	0	00H					General-purpose port		External device connection mode
	1	Other than 00H					$\overline{TAS}$	TMD	μPD71P301 connection mode

7. OPERATION AFTER RESET

When input to pin $\overline{\text{RESET}}$ becomes low, the system is reset and each hardware component is put in the initial status (reset status). When input to pin $\overline{\text{RESET}}$ becomes high, the reset status is released and program execution starts. Initialize the contents of registers in the program as required.

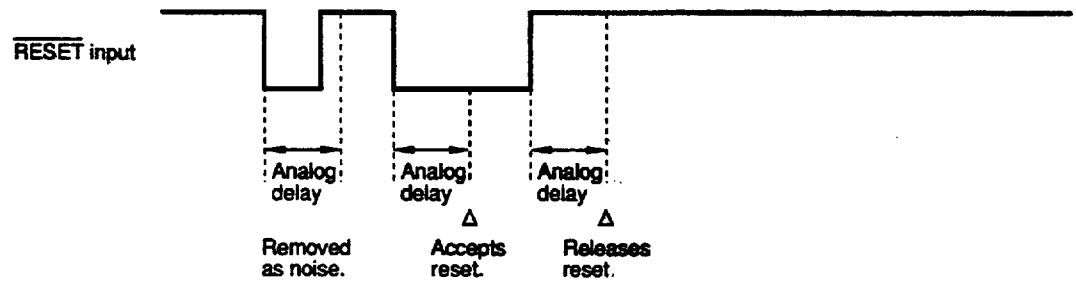
Change the number of cycles specified in the programmable wait control register and fetch cycle control register as required.

A noise eliminator using analog delay is built into the $\overline{\text{RESET}}$ input pin to prevent maloperation due to noise.

Cautions 1. When $\overline{\text{RESET}}$ is active, all pins other than pins AV$_{REF}$, AV$_{CC}$, AV$_{SS}$, V$_{CC}$, V$_{SS}$, X1, and X2 are high-impedance.

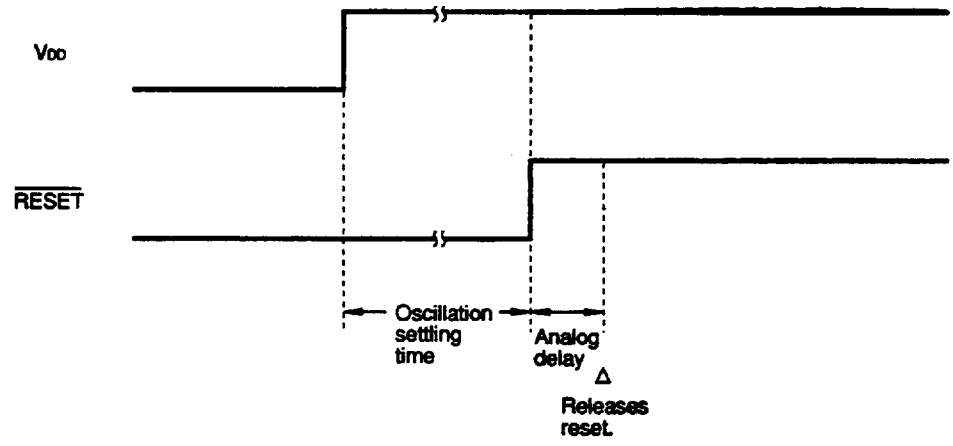
2. When RAM is expanded externally, attach a pull-up resistor to pins P90/$\overline{\text{RD}}$ and P91/$\overline{\text{WR}}$. Otherwise, these pins may go into the high-impedance state, and the contents of the external RAM may be destroyed. Moreover, signals may contend for the address/data bus, resulting in damage to the input/output circuits.

Fig. 7-1 Accepting a Reset Signal



For the reset operation at power-on, reserve 40 ms for the oscillation to stabilize from power-on to reset acceptance as shown in Fig. 7-2.

Fig. 7-2 Reset at Power-On



8. INSTRUCTION SET

This chapter explains only the operations of the instructions. Refer to the *μPD78328 User's Manual* (IEU-1268) for the instruction codes and number of clocks for executing each instruction.

(1) Operand notation and coding format

Operands are coded in the operand field of each instruction as listed in the coding column of Table 9-1. 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. The symbols #, $, !, and [ ] must not be omitted when describing labels.

Table 8-1 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 2-2.**) Special function register abbreviation (16-bit manipulation register. See **Table 2-2.**)
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]: Based indexed mode [DE+byte], [HL+byte], [VP+byte], [UP+byte], [SP+byte]: Based mode word[A], word[B], word[DE], word[HL]: Indexed 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-FDFFH Immediate data or label: Relative addressing 0000H-FDFFH 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, however)[Note] 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)

Note Do not attempt to access word data at an odd-numbered address (bit 0 = 1).

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) as well as absolute names (R0 to R15 and RP0 to RP7) can be specified in r, r1, rp, rp1, and post. See **Table 2-1** for the correspondence between the absolute names and functional names.

3. Immediate addressing is effective for the 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.

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

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

Remark For the meanings of the symbols in flag column, see the following table.

Symbol	Explanation
(Blank)	No change
0	Cleared to zero.
1	Set to 1.
×	Set or reset according to the result.
P	P/V flag operates as a parity flag.
V	P/V flag operates as an overflow flag.
R	Saved values are restored.

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

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
8-bit arithmetic/ logical	SUB	A,#byte	2	A, CY←A-byte	×	×	×	V	×
		saddr,#byte	3	(saddr), CY←(saddr)-byte	×	×	×	V	×
		sfr,#byte	4	sfr, CY←sfr-byte	×	×	×	V	×
		r,r1	2	r, CY←r-r1	×	×	×	V	×
		A,saddr	2	A, CY←A-(saddr)	×	×	×	V	×
		A,sfr	3	A, CY←A-sfr	×	×	×	V	×
		saddr,saddr	3	(saddr), CY←(saddr)-(saddr)	×	×	×	V	×
		A,mem	2-4	A, CY←A-(mem)	×	×	×	V	×
		mem,A	2-4	(mem), CY←(mem)-A	×	×	×	V	×
	SUBC	A,#byte	2	A, CY←A-byte-CY	×	×	×	V	×
		saddr,#byte	3	(saddr), CY←(saddr)-byte-CY	×	×	×	V	×
		sfr,#byte	4	sfr, CY←sfr-byte-CY	×	×	×	V	×
		r,r1	2	r, CY←r-r1-CY	×	×	×	V	×
		A,saddr	2	A, CY←A-(saddr)-CY	×	×	×	V	×
		A,sfr	3	A, CY←A-sfr-CY	×	×	×	V	×
		saddr,saddr	3	(saddr), CY←(saddr)-(saddr)-CY	×	×	×	V	×
		A,mem	2-4	A, CY←A-(mem)-CY	×	×	×	V	×
		mem,A	2-4	(mem), CY←(mem)-A-CY	×	×	×	V	×
	AND	A,#byte	2	A←A∧byte	×	×		P	
		saddr,#byte	3	(saddr)←(saddr)∧byte	×	×		P	
		sfr,#byte	4	sfr←sfr∧byte	×	×		P	
		r,r1	2	r←r∧r1	×	×		P	
		A,saddr	2	A←A∧(saddr)	×	×		P	
		A,sfr	3	A←A∧sfr	×	×		P	
		saddr,saddr	3	(saddr)←(saddr)∧(saddr)	×	×		P	
		A,mem	2-4	A←A∧(mem)	×	×		P	
		mem,A	2-4	(mem)←(mem)∧A	×	×		P	

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
8-bit arithmetic/ logical	OR	A,#byte	2	A←A∨byte	×	×		P	
		saddr,#byte	3	(saddr)←(saddr)∨byte	×	×		P	
		sfr,#byte	4	sfr←sfr∨byte	×	×		P	
		r,r1	2	r←r∨r1	×	×		P	
		A,saddr	2	A←A∨(saddr)	×	×		P	
		A,sfr	3	A←A∨sfr	×	×		P	
		saddr,saddr	3	(saddr)←(saddr)∨(saddr)	×	×		P	
		A,mem	2-4	A←A∨(mem)	×	×		P	
		mem,A	2-4	(mem)←(mem)∨A	×	×		P	
	XOR	A,#byte	2	A←A∀byte	×	×		P	
		saddr,#byte	3	(saddr)←(saddr)∀byte	×	×		P	
		sfr,#byte	4	sfr←sfr∀byte	×	×		P	
		r,r1	2	r←r∀r1	×	×		P	
		A,saddr	2	A←A∀(saddr)	×	×		P	
		A,sfr	3	A←A∀sfr	×	×		P	
		saddr,saddr	3	(saddr)←(saddr)∀(saddr)	×	×		P	
		A,mem	2-4	A←A∀(mem)	×	×		P	
		mem,A	2-4	(mem)←(mem)∀A	×	×		P	
	CMP	A,#byte	2	A-byte	×	×	×	V	×
		saddr,#byte	3	(saddr)-byte	×	×	×	V	×
		sfr,#byte	4	sfr-byte	×	×	×	V	×
		r,r1	2	r-r1	×	×	×	V	×
		A,saddr	2	A-(saddr)	×	×	×	V	×
		A,sfr	3	A-sfr	×	×	×	V	×
		saddr,saddr	3	(saddr)-(saddr)	×	×	×	V	×
		A,mem	2-4	A-(mem)	×	×	×	V	×
		mem,A	2-4	(mem)-A	×	×	×	V	×

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
16-bit arithmetic/ logical	ADDW	AX,#word	3	AX,CY←AX+word	×	×	×	V	×
		saddrp,#word	4	(saddrp),CY←(saddrp)+word	×	×	×	V	×
		sfrp,#word	5	sfrp,CY←sfrp+word	×	×	×	V	×
		rp,rp1	2	rp,CY←rp+rp1	×	×	×	V	×
		AX,saddrp	2	AX,CY←AX+(saddrp)	×	×	×	V	×
		AX,sfrp	3	AX,CY←AX+sfrp	×	×	×	V	×
		saddrp,saddrp	3	(saddrp),CY←(saddrp)+(saddrp)	×	×	×	V	×
	SUBW	AX,#word	3	AX,CY←AX-word	×	×	×	V	×
		saddrp,#word	4	(saddrp),CY←(saddrp)-word	×	×	×	V	×
		sfrp,#word	5	sfrp,CY←sfrp-word	×	×	×	V	×
		rp,rp1	2	rp,CY←rp-rp1	×	×	×	V	×
		AX,saddrp	2	AX,CY←AX-(saddrp)	×	×	×	V	×
		AX,sfrp	3	AX,CY←AX-sfrp	×	×	×	V	×
		saddrp,saddrp	3	(saddrp),CY←(saddrp)-(saddrp)	×	×	×	V	×
	CMPW	AX,#word	3	AX-word	×	×	×	V	×
		saddrp,#word	4	(saddrp)-word	×	×	×	V	×
		sfrp,#word	5	sfrp-word	×	×	×	V	×
		rp,rp1	2	rp-rp1	×	×	×	V	×
		AX,saddrp	2	AX-(saddrp)	×	×	×	V	×
		AX,sfrp	3	AX-sfrp	×	×	×	V	×
		saddrp,saddrp	3	(saddrp)-(saddrp)	×	×	×	V	×
Multiply/ divide	MULU	r1	2	AX←Axr1					
	DIVUW	r1	2	AX(quotient), r1 (remainder)←AX÷r1					
	MULUW	rp1	2	AX (16 high-order bits), rp1 (16 low-order bits)←AX×rp1					
	DIVUX	rp1	2	AXDE (quotient), rp1 (remainder)←AXDE÷rp1					
Signed multiply	MULW	rp1	2	AX (16 high-order bits), rp1 (16 low-order bits)←AX×rp1					

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Increment/ decrement	INC	r1	1	r1←r1+1	×	×	×	V	
		saddr	2	(saddr)←(saddr)+1	×	×	×	V	
	DEC	r1	1	r1←r1-1	×	×	×	V	
		saddr	2	(saddr)←(saddr)-1	×	×	×	V	
	INCW	rp2	1	rp2←rp2+1					
		saddrp	3	(saddrp)←(saddrp)+1					
	DECW	rp2	1	rp2←rp2-1					
		saddrp	3	(saddrp)←(saddrp)-1					
Shift/ rotate	ROR	r1, n	2	(CY, $r1_7$←$r1_0$, $r1_{m-1}$←$r1_m$) × n times				P	×
	ROL	r1, n	2	(CY, $r1_0$←$r1_7$, $r1_{m+1}$←$r1_m$) × n times				P	×
	RORC	r1, n	2	(CY←$r1_0$, $r1_7$←CY, $r1_{m-1}$←$r1_m$) × n times				P	×
	ROLC	r1, n	2	(CY←$r1_7$, $r1_0$←CY, $r1_{m+1}$←$r1_m$) × n times				P	×
	SHR	r1, n	2	(CY←$r1_0$, $r1_7$←0, $r1_{m-1}$←$r1_m$) × n times	×	×	0	P	×
	SHL	r1, n	2	(CY←$r1_7$, $r1_0$←0, $r1_{m+1}$←$r1_m$) × n times	×	×	0	P	×
	SHRW	rp1, n	2	(CY←$rp1_0$, $rp1_{15}$←0, $rp1_{m-1}$←$rp1_m$) × n times	×	×	0	P	×
	SHLW	rp1, n	2	(CY←$rp1_{15}$, $rp1_0$←0, $rp1_{m+1}$←$rp1_m$) × n times	×	×	0	P	×
	ROR4	[rp1]	2	A_{3-0}←$(rp1)_{3-0}$, $(rp1)_{7-4}$←A_{3-0}, $(rp1)_{3-0}$←$(rp1)_{7-4}$					
	ROL4	[rp1]	2	A_{3-0}←$(rp1)_{7-4}$, $(rp1)_{3-0}$←A_{3-0}, $(rp1)_{7-4}$←$(rp1)_{3-0}$					
BCD correction	ADJBA		2	Decimal adjust accumulator	×	×	×	P	×
	ADJBS								
Data conversion	CVTBW		1	When A_7=0, X←A, A←00H When A_7=1, X←A, A←FFH					

Instruction set	Mnemonic	Operand	Byte	Operation	Flag S	Z	AC	P/V	CY
Bit manipulation	MOV1	CY,saddr.bit	3	CY←(saddr.bit)					×
		CY,sfr.bit	3	CY←sfr.bit					×
		CY,A.bit	2	CY←A.bit					×
		CY,X.bit	2	CY←X.bit					×
		CY,PSWH.bit	2	CY←PSW$_H$.bit					×
		CY,PSWL.bit	2	CY←PSW$_L$.bit					×
		saddr.bit,CY	3	(saddr.bit)←CY					
		sfr.bit,CY	3	sfr.bit←CY					
		A.bit,CY	2	A.bit←CY					
		X.bit,CY	2	X.bit←CY					
		PSWH.bit,CY	2	PSW$_H$.bit←CY					
		PSWL.bit,CY	2	PSW$_L$.bit←CY					
	AND1	CY,saddr.bit	3	CY←CY∧(saddr.bit)					×
		CY,/saddr.bit	3	CY←CY∧$\overline{(saddr.bit)}$					×
		CY,sfr.bit	3	CY←CY∧sfr.bit					×
		CY,/sfr.bit	3	CY←CY∧$\overline{sfr.bit}$					×
		CY,A.bit	2	CY←CY∧A.bit					×
		CY,/A.bit	2	CY←CY∧$\overline{A.bit}$					×
		CY,X.bit	2	CY←CY∧X.bit					×
		CY,/X.bit	2	CY←CY∧$\overline{X.bit}$					×
		CY,PSWH.bit	2	CY←CY∧PSW$_H$.bit					×
		CY,/PSWH.bit	2	CY←CY∧$\overline{PSW_H.bit}$					×
		CY,PSWL.bit	2	CY←CY∧PSW$_L$.bit					×
		CY,/PSWL.bit	2	CY←CY∧$\overline{PSW_L.bit}$					×
	OR1	CY,saddr.bit	3	CY←CY∨(saddr.bit)					×
		CY,/saddr.bit	3	CY←CY∨$\overline{(saddr.bit)}$					×
		CY,sfr.bit	3	CY←CY∨sfr.bit					×
		CY,/sfr.bit	3	CY←CY∨$\overline{sfr.bit}$					×
		CY,A.bit	2	CY←CY∨A.bit					×
		CY,/A.bit	2	CY←CY∨$\overline{A.bit}$					×
		CY,X.bit	2	CY←CY∨X.bit					×
		CY,/X.bit	2	CY←CY∨$\overline{X.bit}$					×
		CY,PSWH.bit	2	CY←CY∨PSW$_H$.bit					×
		CY,/PSWH.bit	2	CY←CY∨$\overline{PSW_H.bit}$					×
		CY,PSWL.bit	2	CY←CY∨PSW$_L$.bit					×
		CY,/PSWL.bit	2	CY←CY∨$\overline{PSW_L.bit}$					×

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Bit manipulation	XOR1	CY,saddr.bit	3	CY←CY∀(saddr.bit)					×
		CY,sfr.bit	3	CY←CY∀sfr.bit					×
		CY,A.bit	2	CY←CY∀A.bit					×
		CY,X.bit	2	CY←CY∀X.bit					×
		CY,PSWH.bit	2	CY←CY∀PSW$_H$.bit					×
		CY,PSWL.bit	2	CY←CY∀PSW$_L$.bit					×
	SET1	saddr.bit	2	(saddr.bit)←1					
		sfr.bit	3	sfr.bit←1					
		A.bit	2	A.bit←1					
		X.bit	2	X.bit←1					
		PSWH.bit	2	PSW$_H$.bit←1					
		PSWL.bit	2	PSW$_L$.bit←1	×	×	×	×	×
	CLR1	saddr.bit	2	(saddr.bit)←0					
		sfr.bit	3	sfr.bit←0					
		A.bit	2	A.bit←0					
		X.bit	2	X.bit←0					
		PSWH.bit	2	PSW$_H$.bit←0					
		PSWL.bit	2	PSW$_L$.bit←0	×	×	×	×	×
	NOT1	saddr.bit	3	(saddr.bit)←$\overline{\text{(saddr.bit)}}$					
		sfr.bit	3	sfr.bit←$\overline{\text{sfr.bit}}$					
		A.bit	2	A.bit←$\overline{\text{A.bit}}$					
		X.bit	2	X.bit←$\overline{\text{X.bit}}$					
		PSWH.bit	2	PSW$_H$.bit←$\overline{\text{PSW}_H\text{.bit}}$					
		PSWL.bit	2	PSW$_L$.bit←$\overline{\text{PSW}_L\text{.bit}}$	×	×	×	×	×
	SET1	CY	1	CY←1					1
	CLR1	CY	1	CY←0					0
	NOT1	CY	1	CY←$\overline{\text{CY}}$					×

Instruction set	Mnemonic	Operand	Byte	Operation	Flag S	Z	AC	P/V	CY
Call/ return	CALL	!addr16	3	(SP-1)←(PC+3)H, (SP-2)←(PC+3)L, PC←addr16, SP←SP-2					
	CALLF	!addr11	2	(SP-1)←(PC+2)H, (SP-2)←(PC+2)L, PC15-11←00001, PC10-0←addr11, SP←SP-2					
	CALLT	[addr5]	1	(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	(SP-1)←(PC+2)H, (SP-2)←(PC+2)L, PCH←rp1H,PCL← rp1L,SP←SP-2					
		[rp1]	2	(SP-1)←(PC+2)H, (SP-2)←(PC+2)L, PCH←(rp1+1),PCL←(rp1),SP←SP-2					
	BRK		1	(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	PCL←(SP), PCH←(SP+1), SP←SP+2					
	RETB		1	PCL←(SP), PCH←(SP+1) PSWL←(SP+2), PSWH←(SP+3) SP←SP+4	R	R	R	R	R
	RETI		1	PCL←(SP), PCH←(SP+1) PSWL←(SP+2), PSWH←(SP+3) SP←SP+4	R	R	R	R	R
Stack manipulation	PUSH	sfrp	3	(SP-1)←sfrH, (SP-2)←sfrL, SP←SP-2					
		post	2	{(SP-1)←postH, (SP-2)←postL, SP←SP-2} × n times[Note]					
		PSW	1	(SP-1)←PSWH, (SP-2)←PSWL, SP←SP-2					
	PUSHU	post	2	{(UP-1)←postH, (UP-2)←postL, UP←UP-2} × n times[Note]					
	POP	sfrp	3	sfrL←(SP), sfrH←(SP+1), SP←SP+2					
		post	2	{postL←(SP), postH←(SP+1), SP←SP+2} × n times[Note]					
		PSW	1	PSWL←(SP), PSWH←(SP+1), SP←SP+2	R	R	R	R	R
	POPU	post	2	{postL←(UP), postH←(UP+1), UP←UP+2} × n times[Note]					
	MOVW	SP,#word	4	SP←word					
		SP,AX	2	SP←AX					
		AX,SP	2	AX←SP					
	INCW	SP	2	SP←SP+1					
	DECW	SP	2	SP←SP-1					
Special	CHKL	sfr	3	(Pin level)∀(Signal level before output buffer)	×	×		P	
	CHKLA	sfr	3	A←(Pin level)∀(Signal level before output buffer)	×	×		P	

Note n indicates the number of registers specified in post.

Instruction set	Mnemonic	Operand	Byte	Operation	Flag S	Z	AC	P/V	CY
Unconditional branch	BR	!addr16	3	PC←addr16					
		rp1	2	PC_H←$rp1_H$, PC_L←$rp1_L$					
		[rp1]	2	PC_H←(rp1+1), PC_L←(rp1)					
		$addr16	2	PC←PC+2+jdisp8					
Conditional branch	BC	$addr16	2	PC←PC+2+jdisp8 if CY=1					
	BL								
	BNC	$addr16	2	PC←PC+2+jdisp8 if CY=0					
	BNL								
	BZ	$addr16	2	PC←PC+2+jdisp8 if Z=1					
	BE								
	BNZ	$addr16	2	PC←PC+2+jdisp8 if Z=0					
	BNE								
	BV	$addr16	2	PC←PC+2+jdisp8 if P/V=1					
	BPE								
	BNV	$addr16	2	PC←PC+2+jdisp8 if P/V=0					
	BPO								
	BN	$addr16	2	PC←PC+2+jdisp8 if S=1					
	BP	$addr16	2	PC←PC+2+jdisp8 if S=0					
	BGT	$addr16	3	PC←PC+3+jdisp8 if (P/V⊽S)∨Z=0					
	BGE	$addr16	3	PC←PC+3+jdisp8 if P/V⊽S=0					
	BLT	$addr16	3	PC←PC+3+jdisp8 if P/V⊽S=1					
	BLE	$addr16	3	PC←PC+3+jdisp8 if (P/V⊽S)∨Z=1					
	BH	$addr16	3	PC←PC+3+jdisp8 if Z∨CY=0					
	BNH	$addr16	3	PC←PC+3+jdisp8 if Z∨CY=1					
	BT	saddr.bit, $addr16	3	PC←PC+3+jdisp8 if (saddr.bit)=1					
		sfr.bit, $addr16	4	PC←PC+4+jdisp8 if sfr.bit=1					
		A.bit, $addr16	3	PC←PC+3+jdisp8 if A.bit=1					
		X.bit, $addr16	3	PC←PC+3+jdisp8 if X.bit=1					
		PSWH.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_H.bit=1					
		PSWL.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_L.bit=1					
	BF	saddr.bit, $addr16	4	PC←PC+4+jdisp8 if (saddr.bit)=0					
		sfr.bit, $addr16	4	PC←PC+4+jdisp8 if sfr.bit=0					
		A.bit, $addr16	3	PC←PC+3+jdisp8 if A.bit=0					
		X.bit, $addr16	3	PC←PC+3+jdisp8 if X.bit=0					
		PSWH.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_H.bit=0					
		PSWL.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_L.bit=0					

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Conditional branch	BTCLR	saddr.bit, $addr16	4	PC←PC+4+jdisp8 if (saddr.bit)=1 then reset (saddr.bit)					
		sfr.bit, $addr16	4	PC←PC+4+jdisp8 if sfr.bit=1 then reset sfr.bit					
		A.bit, $addr16	3	PC←PC+3+jdisp8 if A.bit=1 then reset A.bit					
		X.bit, $addr16	3	PC←PC+3+jdisp8 if X.bit=1 then reset X.bit					
		PSWH.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_H.bit=1 then reset PSW_H.bit					
		PSWL.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_L.bit=1 then reset PSW_L.bit	×	×	×	×	×
	BFSET	saddr.bit, $addr16	4	PC←PC+4+jdisp8 if (saddr.bit)=0 then set (saddr.bit)					
		sfr.bit, $addr16	4	PC←PC+4+jdisp8 if sfr.bit=0 then set sfr.bit					
		A.bit, $addr16	3	PC←PC+3+jdisp8 if A.bit=0 then set A.bit					
		X.bit, $addr16	3	PC←PC+3+jdisp8 if X.bit=0 then set X.bit					
		PSWH.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_H.bit=0 then set PSW_H.bit					
		PSWL.bit, $addr16	3	PC←PC+3+jdisp8 if PSW_L.bit=0 then set PSW_L.bit	×	×	×	×	×
	DBNZ	r2, $addr16	2	r2←r2-1, then PC←PC+2+jdisp8 if r2≠0					
		saddr, $addr16	3	(saddr)←(saddr)-1, then PC←PC+3+jdisp8 if (saddr)≠0					
Context switching	BRKCS	RBn	2	PC_H↔R5, PC_L↔R4, R7←PSW_H, R6←PSW_L, RBS2-0←n, RSS←0, IE←0					
	RETCS	!addr16	3	PC_H←R5, PC_L←R4, R5, R4←addr16, PSW_H←R7, PSW_L←R6	R	R	R	R	R
	RETCSB	!addr16	4	PC_H←R5, PC_L←R4, R5, R4←addr16, PSW_H←R7, PSW_L←R6	R	R	R	R	R

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
String	MOVM	[DE+], A	2	(DE+)←A, C←C-1 End if C=0					
		[DE-], A	2	(DE-)←A, C←C-1 End if C=0					
	MOVBK	[DE+], [HL+]	2	(DE+)←(HL+), C←C-1 End if C=0					
		[DE-], [HL-]	2	(DE-)←(HL-), C←C-1 End if C=0					
	XCHM	[DE+], A	2	(DE+)↔A, C←C-1 End if C=0					
		[DE-], A	2	(DE-)↔A, C←C-1 End if C=0					
	XCHBK	[DE+], [HL+]	2	(DE+)↔(HL+), C←C-1 End if C=0					
		[DE-], [HL-]	2	(DE-)↔(HL-), C←C-1 End if C=0					
	CMPME	[DE+], A	2	(DE+)-A, C←C-1 End if C=0 or Z=0	×	×	×	V	×
		[DE-], A	2	(DE-)-A, C←C-1 End if C=0 or Z=0	×	×	×	V	×
	CMPBKE	[DE+], [HL+]	2	(DE+)-(HL+), C←C-1 End if C=0 or Z=0	×	×	×	V	×
		[DE-], [HL-]	2	(DE-)-(HL-), C←C-1 End if C=0 or Z=0	×	×	×	V	×
	CMPMNE	[DE+], A	2	(DE+)-A, C←C-1 End if C=0 or Z=1	×	×	×	V	×
		[DE-], A	2	(DE-)-A, C←C-1 End if C=0 or Z=1	×	×	×	V	×
	CMPBKNE	[DE+], [HL+]	2	(DE+)-(HL+), C←C-1 End if C=0 or Z=1	×	×	×	V	×
		[DE-], [HL-]	2	(DE-)-(HL-), C←C-1 End if C=0 or Z=1	×	×	×	V	×
	CMPMC	[DE+], A	2	(DE+)-A, C←C-1 End if C=0 or CY=0	×	×	×	V	×
		[DE-], A	2	(DE-)-A, C←C-1 End if C=0 or CY=0	×	×	×	V	×
	CMPBKC	[DE+], [HL+]	2	(DE+)-(HL+), C←C-1 End if C=0 or CY=0	×	×	×	V	×
		[DE-], [HL-]	2	(DE-)-(HL-), C←C-1 End if C=0 or CY=0	×	×	×	V	×

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

Note An op-code trap interrupt occurs if an invalid op-code is specified in an STBC or WDM register manipulation instruction.

Trap operation: (SP-1)←PSW_H, (SP-2)←PSW_L,
(SP-3)←$(PC-4)_H$, (SP-4)←$(PC-4)_L$,
PC_L←(003CH), PC_H←(003DH),
SP←SP-4, IE←0

9. ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (T_A = 25 °C)

Parameter	Symbol	Conditions		Rating	Unit
Supply voltage	V_{DD}			−0.5 to +7.0	V
	AV_{DD}			−0.5 to V_{DD} + 0.5	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	Note 1		−0.5 to V_{DD} + 0.5	V
Low-level output current	I_{OL}	Each pin		4.0	mA
		Total of all output pins		90	mA
High-level output current	I_{OH}	Each pin		−1.0	mA
		Total of all output pins		−20	mA
Analog input voltage	V_{IAN}	Note 2	AV_{DD} > V_{DD}	−0.5 to V_{DD} + 0.5	V
			V_{DD} ≥ AV_{DD}	−0.5 to AV_{DD} + 0.5	
A/D converter reference input voltage	AV_{REF}		AV_{DD} > V_{DD}	−0.5 to V_{DD} + 0.3	V
			V_{DD} ≥ AV_{DD}	−0.5 to AV_{DD} + 0.3	
Operating ambient temperature	T_A			−10 to +70	°C
Storage temperature	T_{stg}			−65 to +150	°C

Notes 1. Pins other than P70/ANI0 to P77/ANI7

2. Pins P70/ANI0 to P77/ANI7

Caution Absolute maximum ratings are rated values beyond which some physical damages may be caused to the product; if any of the parameters in the table above exceeds its rated value even for a moment, the quality of the product may deteriorate. Be sure to use the product within the rated values. ★

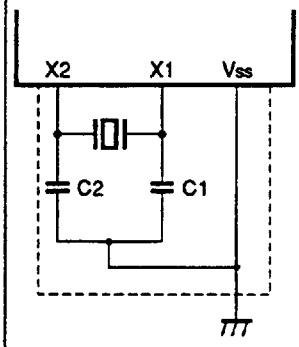
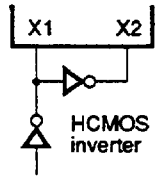
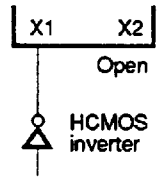
RECOMMENDED OPERATING CONDITIONS

Oscillator frequency	T_A	V_{DD}
8 MHz ≤ f_{XX} ≤ 16 MHz	−10 to +70 °C	+5.0 V ±10 %

CAPACITANCE (T_A = 25 °C, V_{SS} = V_{DD} = 0 V)

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

OSCILLATOR CHARACTERISTICS (T_A = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

Resonator	Recommended circuit	Parameter	Min.	Max.	Unit
Ceramic resonator or crystal		Oscillator frequency (f_{XX})	8	16	MHz
External clock		X1 input frequency (f_X)	8	16	MHz
		X1 input rise and fall times (t_{XR}, t_{XF})	0	20	ns
		X1 input high- and low-level widths (t_{WXH}, t_{WXL})	25	80	ns

★ **Caution When using the system clock generator, run wires enclosed in broken lines ( ) according to the following rules to avoid effects such as stray capacitance:**

- **Minimize the wiring.**
- **Never cause the wires to cross other signal lines or run near a line carrying a large varying current.**
- **Cause the grounding point of the capacitor of the oscillator circuit to have the same potential as Vss. Never connect the capacitor to a ground pattern carrying a large current.**
- **Never extract a signal from the oscillator.**

RECOMMENDED OSCILLATOR CONSTANTS

Ceramic oscillator

Manufacturer	Model name	Frequency [MHz]	Recommended constants	
			C1 [pF]	C2 [pF]
Murata Mfg. Co., Ltd.	CSA8.00MT	8.0	30	30
	CSA12.0MT	12.0		
	CSA16.00MX040	16.0	15	15
	CST8.00MTW	8.0	Contained	Contained
	CST12.0MTW	12.0		
	CST16.00MXW0C3	16.0		

DC CHARACTERISTICS (T_A = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Low-level input voltage	V_{IL}			0		0.8	V
High-level input voltage	V_{IH1}	Note 1		2.2			V
	V_{IH2}	Note 2		0.8V_{DD}			
Low-level output voltage	V_{OL}	I_{OL} = 2.0 mA				0.45	V
High-level output voltage	V_{OH}	I_{OH} = −400 μA		V_{DD} − 1.0			V
Input leakage current	I_{LI}	0 V ≤ V_I ≤ V_{DD}				±10	μA
Output leakage current	I_{LO}	0 V ≤ V_O ≤ V_{DD}				±10	μA
V_{DD} supply current	I_{DD1}	Operation mode			45	75	mA
	I_{DD2}	HALT mode			25	45	mA
Data retention voltage	V_{DDDR}	STOP mode		2.5			V
Data retention current	I_{DDDR}	STOP mode	V_{DDDR} = 2.5 V		3	15	μA
			V_{DDDR} = 5.0 V ±10 %		10	50	μA

Notes 1. Other than the pins listed in **Note 2**

2. $\overline{\text{RESET}}$, X1, X2, P20/NMI, P21/INTP0, P22/INTP1/TI, P86/INTP2/TO0, P32/SO/SB0, P33/SI/SB1, and P34/$\overline{\text{SCK}}$

AC CHARACTERISTICS (T_A = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

Discontinuous Read/Write Operation (When the General Memory is Connected)

Parameter	Symbol	Conditions	Min.	Max.	Unit
System clock cycle time	t_{CYK}		125	250	ns
Address setup time (referenced to ASTB ↓)	t_{SAST}		22		ns
Address hold time (referenced to ASTB ↓)	t_{HSTA}		32		ns
Delay from address to $\overline{RD}$ ↓	t_{DAR}		85		ns
Address float time (referenced to $\overline{RD}$ ↓)	t_{FRA}			8	ns
Delay from address to data input	t_{DAID}			222	ns
Delay from $\overline{RD}$ ↓ to data input	t_{DRID}			112	ns
Delay from ASTB Ø to $\overline{RD}$ ↓	t_{DSTR}		42		ns
Data hold time (referenced to $\overline{RD}$ ↑)	t_{HRID}		0		ns
Delay from $\overline{RD}$ ↑ to address active	t_{DRA}		50		ns
$\overline{RD}$ low-level width	t_{WRL}		147		ns
ASTB high-level width	t_{WSTH}		37		ns
Delay from address to $\overline{WR}$ ↓	t_{DAW}		85		ns
Delay from ASTB ↓ to data output	t_{DSTOD}			102	ns
Delay from $\overline{WR}$ ↓ to data output	t_{DWOD}			40	ns
Delay from ASTB ↓ to $\overline{WR}$ ↓	t_{DSTW}		42		ns
Data setup time (referenced to $\overline{WR}$ ↑)	t_{SODW}		137		ns
Data hold time (referenced to $\overline{WR}$ ↑)	t_{HWOD}		32		ns
Delay from $\overline{WR}$ ↑ to ASTB ↑	t_{DWST}		42		ns
$\overline{WR}$ low-level width	t_{WWL}		147		ns

t_{CYK}-DEPENDENT BUS TIMING DEFINITION

Parameter	Calculation formula	Min./Max.	Unit
t_{SAST}	0.5T – 40	Min.	ns
t_{HSTA}	0.5T – 30	Min.	ns
t_{DAR}	T – 40	Min.	ns
t_{DAID}	(2.5 + n)T – 90	Max.	ns
t_{DRID}	(1.5 + n)T – 75	Max.	ns
t_{DSTR}	0.5T – 20	Min.	ns
t_{DRA}	0.5T – 12	Min.	ns
t_{WRL}	(1.5 + n)T – 40	Min.	ns
t_{WSTH}	0.5T – 25	Min.	ns
t_{DAW}	T – 40	Min.	ns
t_{DSTOD}	0.5T + 40	Max.	ns
t_{DSTW}	0.5T – 20	Min.	ns
t_{SODW}	1.5T – 50	Min.	ns
t_{HWOD}	0.5T – 30	Min.	ns
t_{DWST}	0.5T – 20	Min.	ns
t_{WWL}	(1.5 + n)T – 40	Min.	ns

Remarks 1. T = t_{CYK} = 1/f_{CLK} (f_{CLK} is the internal system clock frequency obtained by dividing fxx or fx by 2.)

2. n represents the number of wait cycles to be defined by user software.

3. Of the bus timings, only items listed above are dependent on t_{CYK}.

SERIAL OPERATION (T_A = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions		Min.	Max.	Unit
Serial clock cycle time	t_{CYSK}	Input	External clock	1		μs
		Output	Internal clock divided by 8	8T		t_{CYK}
			Internal clock divided by 32	32T		t_{CYK}
Serial clock low-level width	t_{WSKL}	Input	External clock	420		ns
		Output	Internal clock divided by 8	4T − 80		ns
			Internal clock divided by 32	16T − 100		ns
Serial clock high-level width	t_{WSKH}	Input	External clock	420		ns
		Output	Internal clock divided by 8	4T − 80		ns
			Internal clock divided by 32	16T − 100		ns
SI setup time (referenced to $\overline{SCK}$ ↑)	t_{SRXSK}			80		ns
SI hold time (referenced to $\overline{SCK}$ ↑)	t_{HSKRX}			80		ns
SO/SB0, SI/SB1 output delay time (referenced to $\overline{SCK}$ ↓)	t_{DSBSK1}	CMOS push-pull output (3-wire serial I/O mode)		0	210	ns
	t_{DSBSK2}	Open drain output (SBI mode), R_L = 1 kΩ		0	600	ns
SB0, SB1 high hold time (referenced to $\overline{SCK}$ ↑)	t_{HSBSK}	SBI mode		4T		t_{CYK}
SB0, SB1 low setup time (referenced to $\overline{SCK}$ ↓)	t_{SSBSK}			4T		t_{CYK}
SB0, SB1 low-level width	t_{WSBL}			4T − 20		ns
SB0, SB1 high-level width	t_{WBSH}			4T − 20		ns

Remark T = t_{CYK} = 1/f_{CLK} (f_{CLK} is the internal system clock frequency obtained by dividing fxx or fx by 2.)

OTHER OPERATIONS (T_A = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Max.	Unit
NMI high/low level width	t_{WNIH}, t_{WNIL}		5		μs
INTP0 high/low level width	t_{WI0H}, t_{WI0L}		8T		t_{CYK}
INTP1 high/low level width	t_{WI1H}, t_{WI1L}		8T		t_{CYK}
INTP2 high/low level width	t_{WI2H}, t_{WI2L}		8T		t_{CYK}
$\overline{\text{RESET}}$ high/low level width	t_{WRSH}, t_{WRSL}		5		μs
TI high/low level width	t_{WTIH}, t_{WTIL}	In the TM1 event counter mode	8T		t_{CYK}

Remark T = t_{CYK} = 1/f_{CLK} (f_{CLK} is the internal system clock frequency obtained by dividing fxx or fx by 2.)

EXTERNAL CLOCK TIMING (T_A = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Max.	Unit
X1 input high-/low-level width	t_{WXH}, t_{WXL}		25	80	ns
X1 input rising/falling time	t_{XR}, t_{XF}		0	20	ns
X1 input cycle time	t_{CYX}		62	125	ns

A/D CONVERTER CHARACTERISTICS

(T_A = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = AV_{SS} = 0 V, V_{DD} − 0.5 V ≤ AV_{DD} ≤ V_{DD})

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Resolution				10			bit
Total error[Note 1]		4.5 V ≤ AV_{REF} ≤ AV_{DD}				±0.4	%FSR
		3.4 V ≤ AV_{REF} ≤ AV_{DD}				±0.7	%FSR
Quantization error						±1/2	LSB
Conversion time	t_{CONV}			144			t_{CYK}
Sampling time	t_{SAMP}			24			t_{CYK}
Zero-scale calibration[Note 1]		4.5 V ≤ AV_{REF} ≤ AV_{DD}			+1.5	±2.5	LSB
		3.4 V ≤ AV_{REF} ≤ AV_{DD}			+1.5	±4.5	LSB
Full scale calibration[Note 1]		4.5 V ≤ AV_{REF} ≤ AV_{DD}			+1.5	±2.5	LSB
		3.4 V ≤ AV_{REF} ≤ AV_{DD}			+1.5	±4.5	LSB
Nonlinearity calibration[Note 1]		4.5 V ≤ AV_{REF} ≤ AV_{DD}			+1.5	±2.5	LSB
		3.4 V ≤ AV_{REF} ≤ AV_{DD}			+1.5	±4.5	LSB
★ Analog input voltage[Note 2]	V_{IAN}			-0.3		AV_{DD}	V
Reference voltage	AV_{REF}			3.4		AV_{DD}	V
AV_{REF} current	AI_{REF}				1.0	3.0	mA
AV_{DD} supply current	AI_{DD}				2.0	6.0	mA
A/D converter data retention current	AI_{DDDR}	STOP mode	AV_{DDDR} = 2.5 V		2.0	10	μA
			AV_{DDDR} = 5 V ±10 %		10	50	μA

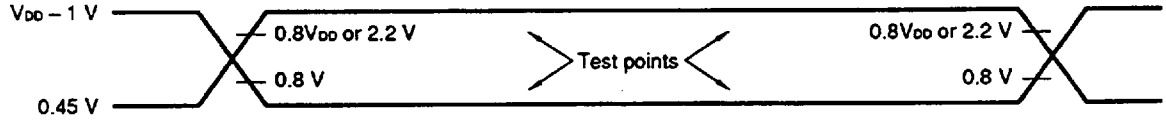
Notes 1. Quantization error is excluded.

2. When −0.3 V ≤ V_{IAN} ≤ 0 V, the conversion result is 00H.
When 0 V < V_{IAN} < AV_{REF}, the voltage is converted with a 10-bit resolution.
When AV_{REF} ≤ V_{IAN} ≤ AV_{DD}, the conversion result is 3FFH.

STANDBY FLAG RETENTION CHARACTERISTICS (T_A = −10 to +70 °C)

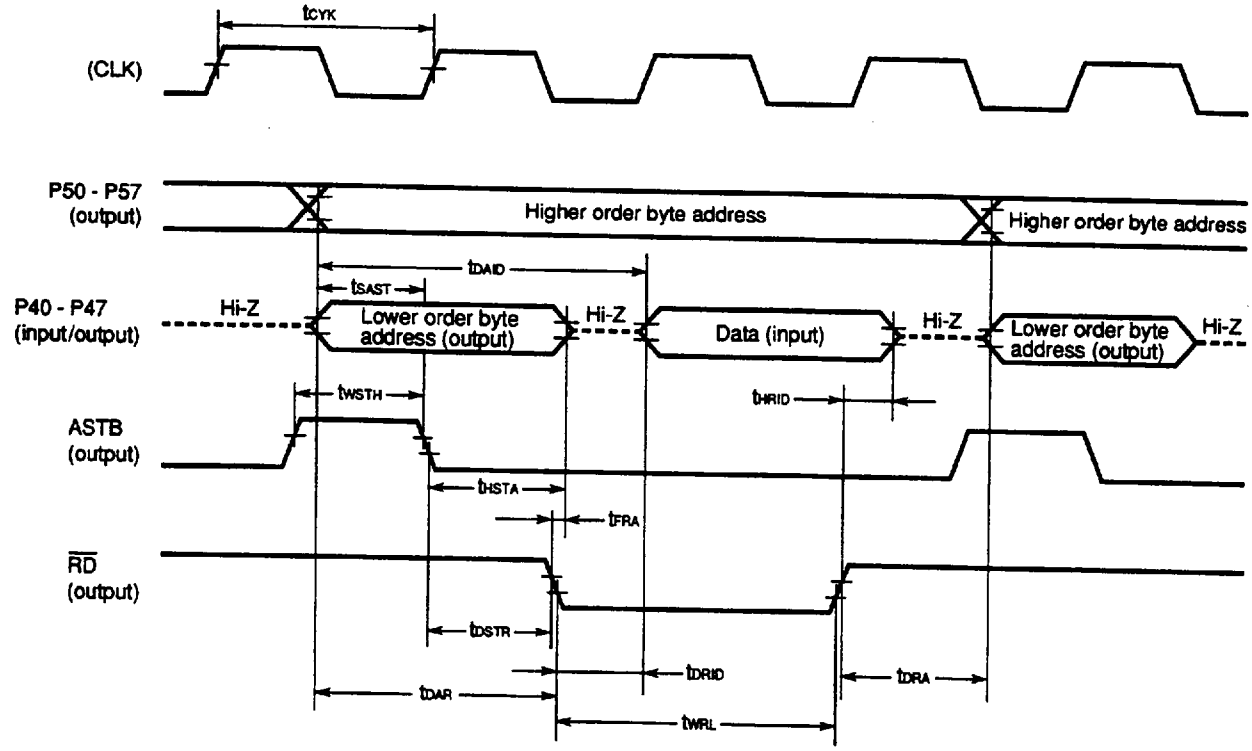
Parameter	Symbol	Conditions	Min.	Max.	Unit
Standby flag retention supply voltage	V_{DDDR}		2.5	5.5	V
V_{DD} rising/falling time	t_{RVD}, t_{FVD}		200		ns

AC Timing Test Points

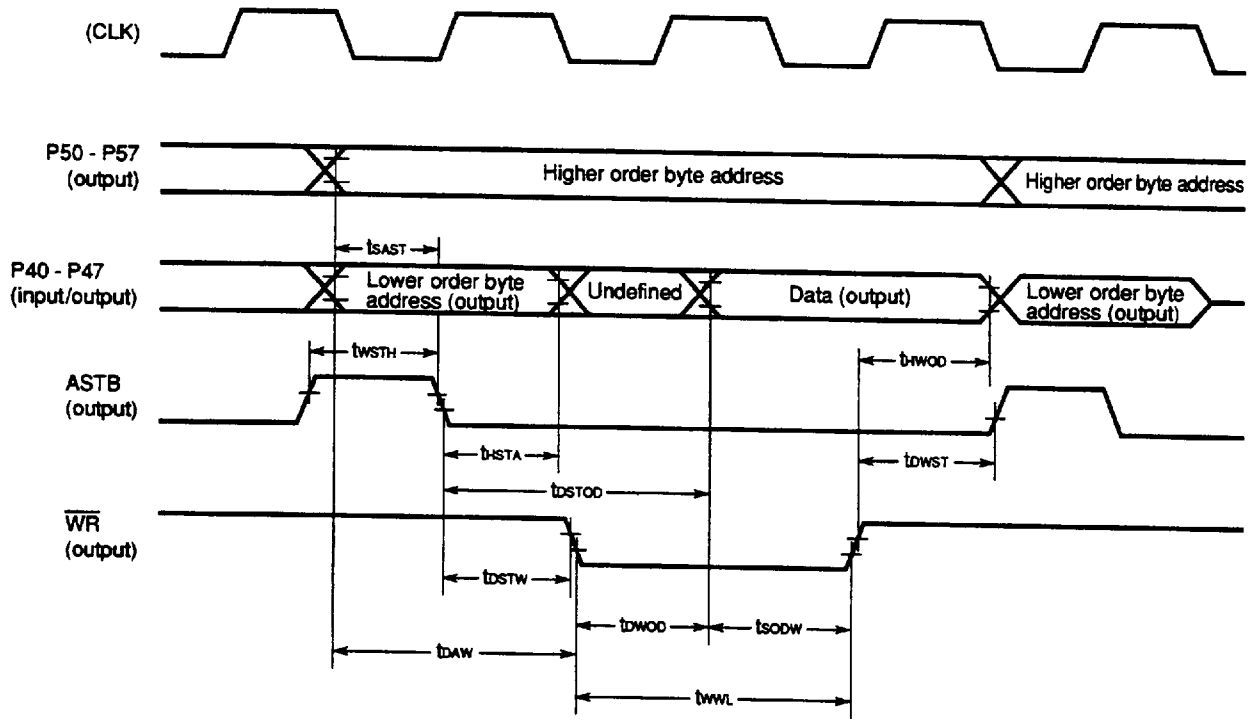


Timing Waveform

Discontinuous read operation

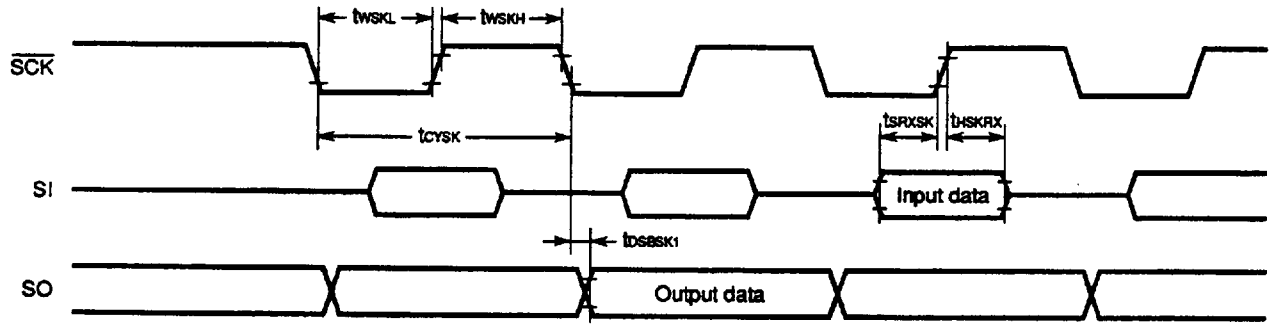


Discontinuous write operation



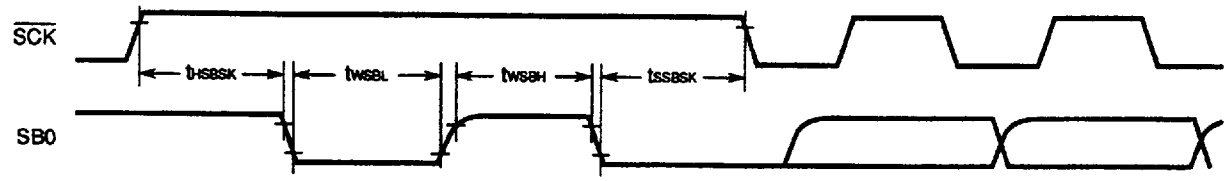
Serial operation

Three-wire serial I/O mode:

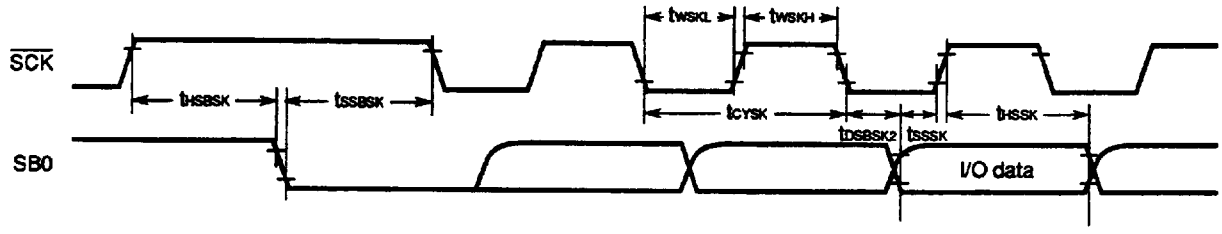


SBI mode:

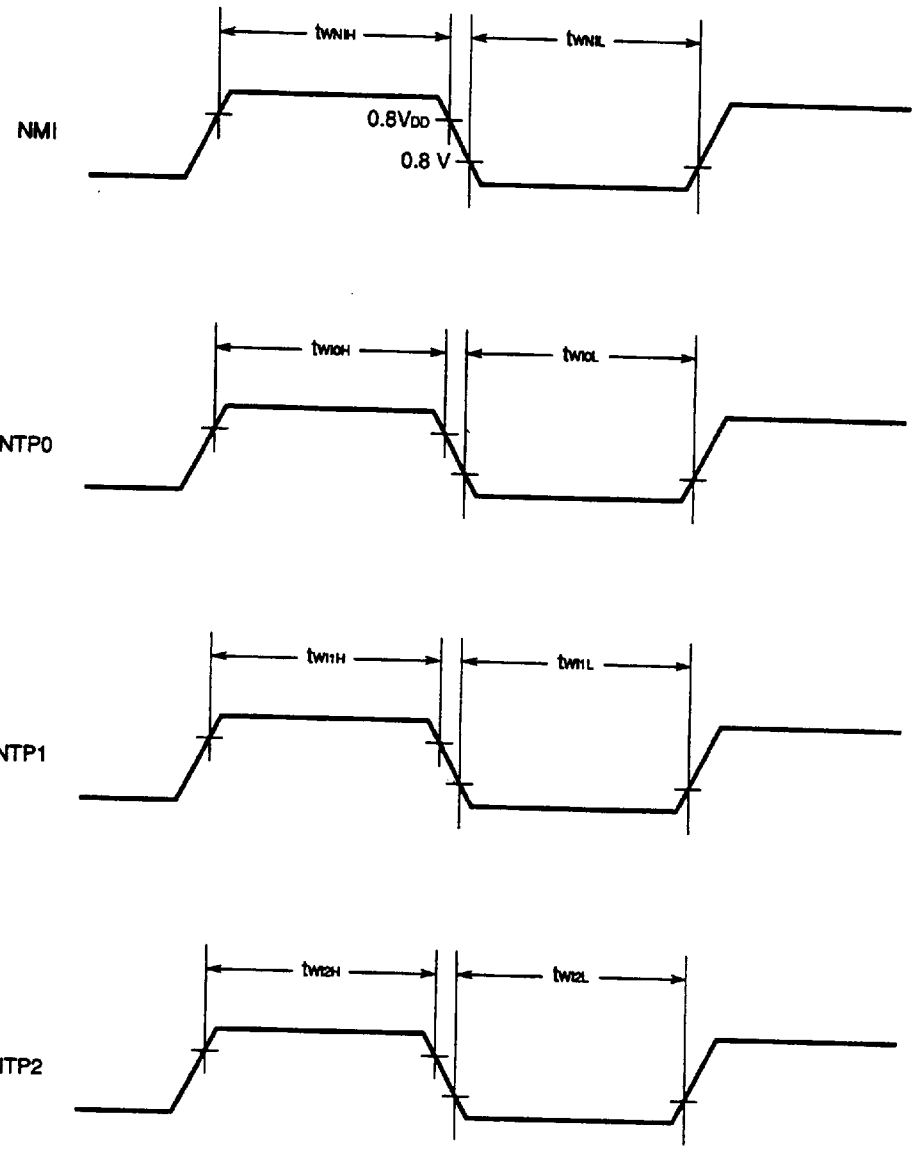
Bus release signal transfer



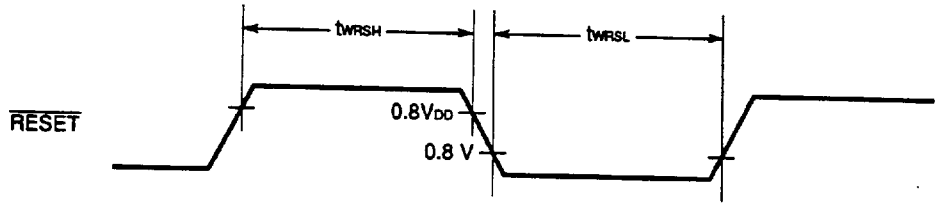
Command signal transfer



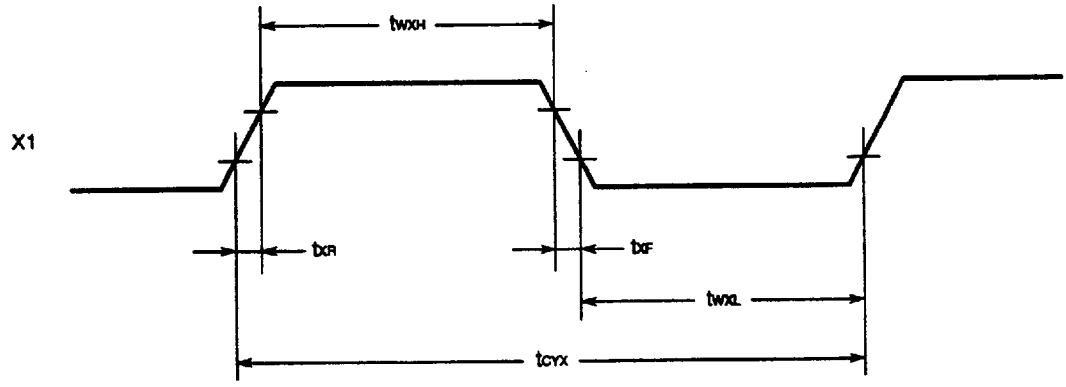
Interrupt input timing



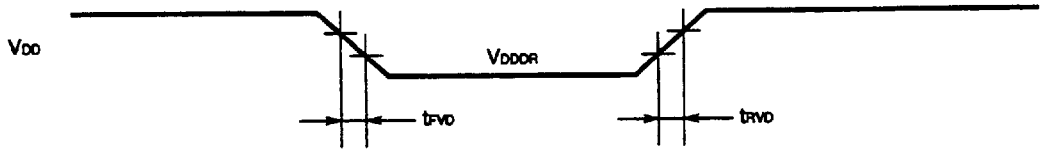
Reset input timing



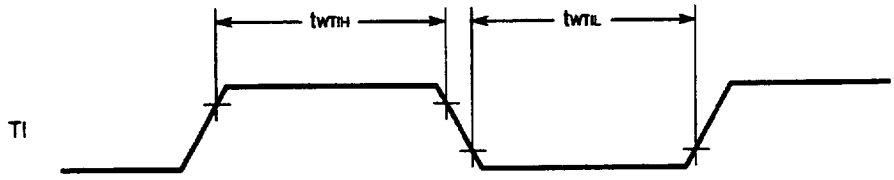
External clock timing



Standby flag retention timing

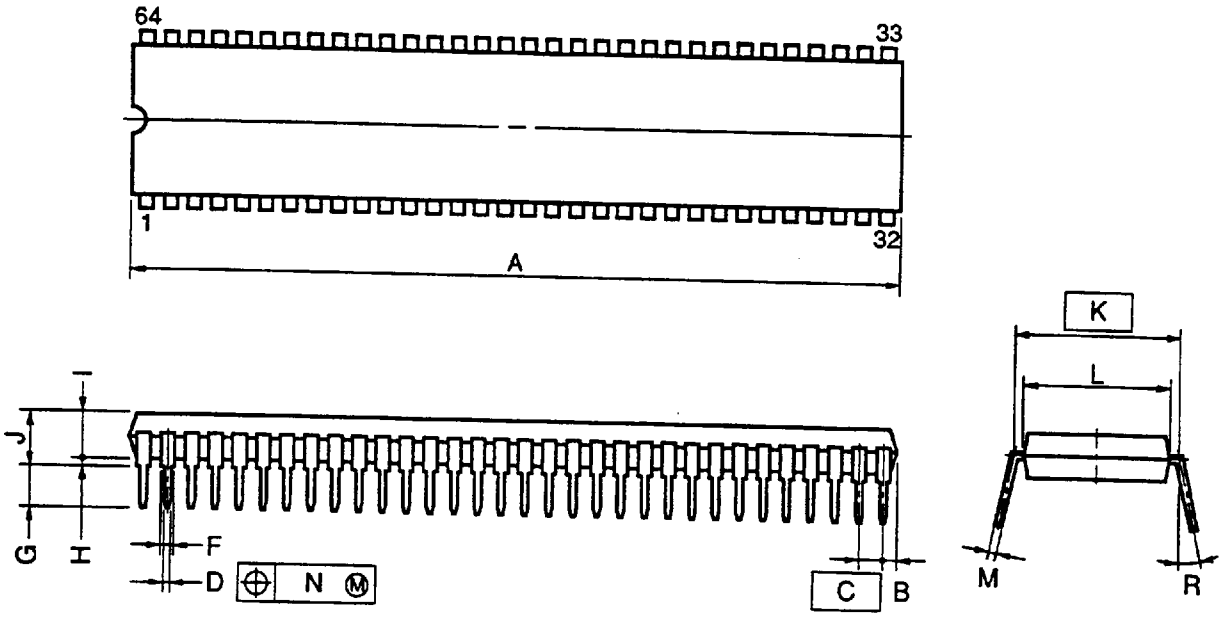


TI pin input timing



10. PACKAGE DRAWINGS

64 PIN PLASTIC SHRINK DIP (750 mil)



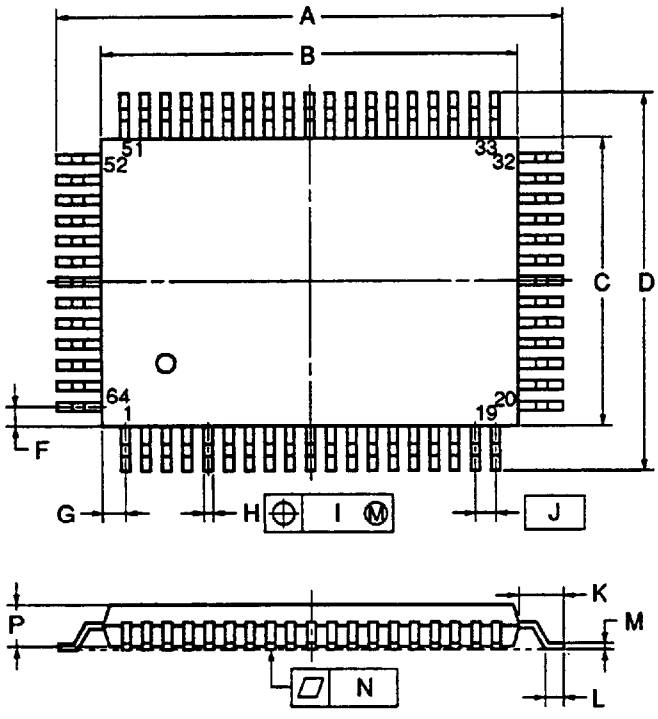
NOTE

1) Each lead centerline is located within 0.17 mm (0.007 inch) of its true position (T.P.) at maximum material condition.
2) Item "K" to center of leads when formed parallel.

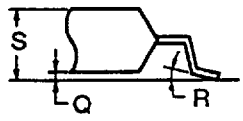
ITEM	MILLIMETERS	INCHES
A	58.68 MAX.	2.311 MAX.
B	1.78 MAX.	0.070 MAX.
C	1.778 (T.P.)	0.070 (T.P.)
D	0.50 ± 0.10	$0.020^{+0.004}_{-0.005}$
F	0.9 MIN.	0.035 MIN.
G	3.2 ± 0.3	0.126 ± 0.012
H	0.51 MIN.	0.020 MIN.
I	4.31 MAX.	0.170 MAX.
J	5.08 MAX.	0.200 MAX.
K	19.05 (T.P.)	0.750 (T.P.)
L	17.0	0.669
M	$0.25^{+0.10}_{-0.05}$	$0.010^{+0.004}_{-0.003}$
N	0.17	0.007
R	0~15°	0~15°

P64C-70-750A,C-1

64 PIN PLASTIC QFP (14×20)



detail of lead end



NOTE

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

ITEM	MILLIMETERS	INCHES
A	23.6±0.4	0.929±0.016
B	20.0±0.2	$0.795^{+0.008}_{-0.009}$
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	$0.016^{+0.004}_{-0.005}$
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	0.10	0.004
P	2.7	0.106
Q	0.1±0.1	0.004±0.004
R	5°±5°	5°±5°
S	3.0 MAX.	0.119 MAX.

P64GF-100-3B8,3BE,3BR-2

11. 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* (IE-1207).

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

Table 11-1 Soldering Conditions for Surface-Mount Devices

μPD78327GF-3BF : 64-pin plastic QFP (14 × 20 mm)
μPD78328GF-xxx-3BF: 64-pin plastic QFP (14 × 20 mm)

Soldering process	Soldering conditions	Recommended conditions
Infrared ray reflow	Peak package's surface temperature: 235 °C Reflow time: 30 seconds or less (210 °C or more) Maximum allowable number of reflow processes: 2 **<Cautions>** (1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering. (2) Do not use water for flux cleaning before a second reflow soldering.	IR35-00-2
VPS	Peak package's surface temperature: 215 °C Reflow time: 40 seconds or less (200 °C or more) Maximum allowable number of reflow processes: 2 **<Cautions>** (1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering. (2) Do not use water for flux cleaning before a second reflow soldering.	VP15-00-2
Wave soldering	Solder temperature: 260 °C or less Flow time: 10 seconds or less Number of flow process: 1 Preheating temperature: 120 °C max. (measured on the package surface)	WS60-00-1
Partial heating method	Terminal temperature: 300 °C or less Heat time: 3 seconds or less (for one side of a device)	–

Table 11-2 Soldering Conditions for Inserted Devices

μPD78327CW : 64-pin plastic shrink DIP (750 mil)
μPD78328CW-xxx: 64-pin plastic shrink DIP (750 mil)

Soldering process	Soldering conditions
Wave soldering (Only for terminals)	Solder temperature: 260 °C or less Flow time: 10 seconds or less
Partial heating method	Terminal temperature: 300 °C or less Heat time: 3 seconds or less (for each terminal)

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

APPENDIX A DIFFERENCE BETWEEN μPD78328 AND μPD78322

		μPD78328	μPD78322
Package		68-pin shrink DIP 64-pin QFP	68-pin plastic QFJ 74-pin QFP 80-pin QFP
RAM capacity		512 bytes	640 bytes
RPU	Number of timers	3 (16 bits × 3)	2 (18/16 bits, 16 bits)
	Number of compare registers	14 (16 bits × 14)	6 (16 bits × 6)
	Number of capture registers	0	4 (18/16 bits × 4)
	Number of capture/ compare registers	1 (16 bits)	2 (18/16 bits × 2)
	Timer output	• Mode 0: 7 channels (Set/reset output: 6) (Toggle output: 1) • Mode 1: 8 channels (Buffer output: 7) (Buffer output, set/reset output, toggle output : 1)	6 channels (Set/reset output, toggle output: 4) (Toggle output: 2)
Real-time output port		4/8 (Buffer output in 4 or 8 bits)	8 (Set/reset output in bits)
Port	Port 0	4-/8-bit input/output (Specified for real-time output port in 4 bits.)	8-bit input/output (Specified for real-time output port in bits.)
	Port 2	3-bit input	8-bit input
	Port 8	8-bit input/output	6-bit input/output
Interrupt request source		• External: 4 (including NMI) • Internal : 16	• External: 8 (including NMI) • Internal : 14 (2 pin are also used for external interrupts.)
Test source		Internal: 1	
Macro service function		Provided (more suitable for PWM inverter control than μPD78322)	Provided
Watchdog timer output pin		Not provided	Provided
8-bit high-speed PWM signal output function		1 channel	Not provided
UART Maximum baud rate when external clock frequency is 16 MHz		9600 bps	2400 bps

APPENDIX B TOOLS

★

B.1 DEVELOPMENT TOOLS

The following tools are provided for developing a system that uses the μPD78327 or μPD78328:

Language processor

<table>
<tr><td>78K/III series relocatable assembler (RA78K/III)</td><td colspan="4">This relocatable program can be used for all 78K/III series emulators. With its macro functions, it allows the user to improve program development efficiency. A structured-programming assembler is also provided, which enables explicit description of program control structures. This assembler could improve productivity in program production and maintenance.

<table>
<tr><th rowspan="2">Host machine</th><th colspan="2" rowspan="2"></th><th rowspan="2">Part number</th></tr>
<tr></tr>
<tr><td>PC-9800 series</td><td>MS-DOS™</td><td>3.5-inch 2HD</td><td>μS5A13RA78K3</td></tr>
<tr><td></td><td></td><td>5.25-inch 2HD</td><td>μS5A10RA78K3</td></tr>
<tr><td>IBM PC/AT™ or compatibles</td><td>PC DOS™</td><td>3.5-inch 2HC</td><td>μS7B13RA78K3</td></tr>
<tr><td></td><td></td><td>5.25-inch 2HC</td><td>μS7B10RA78K3</td></tr>
<tr><td>HP9000 series 700™</td><td>HP-UX™</td><td>DAT</td><td>μS3P16RA78K3</td></tr>
<tr><td>SPARCstation™</td><td>SunOS™</td><td rowspan="2">Cartridge tape (QIC-24)</td><td>μS3K15RA78K3</td></tr>
<tr><td>NEWS™</td><td>NEWS-OS™</td><td>μS3R15RA78K3</td></tr>
</table></td></tr>
<tr><td>78K/III series C compiler (CC78K/III)</td><td colspan="4">This C compiler can be used for all 78K/III series emulators. The compiler converts programs written in C language into object codes executable on the microcomputer. When the compiler is used, the 78K/III series relocatable assembler package (RA78K/III) is needed.

<table>
<tr><th rowspan="2">Host machine</th><th colspan="2" rowspan="2"></th><th rowspan="2">Part number</th></tr>
<tr></tr>
<tr><td>PC-9800 series</td><td>MS-DOS</td><td>3.5-inch 2HD</td><td>μS5A13CC78K3</td></tr>
<tr><td></td><td></td><td>5.25-inch 2HD</td><td>μS5A10CC78K3</td></tr>
<tr><td>IBM PC/AT or compatibles</td><td>PC DOS</td><td>3.5-inch 2HC</td><td>μS7B13CC78K3</td></tr>
<tr><td></td><td></td><td>5.25-inch 2HC</td><td>μS7B10CC78K3</td></tr>
<tr><td>HP9000 series 700</td><td>HP-UX</td><td>DAT</td><td>μS3P16CC78K3</td></tr>
<tr><td>SPARCstation</td><td>SunOS</td><td rowspan="2">Cartridge tape (QIC-24)</td><td>μS3K15CC78K3</td></tr>
<tr><td>NEWS</td><td>NEWS-OS</td><td>μS3R15CC78K3</td></tr>
</table></td></tr>
</table>

Remark It is guaranteed that the relocatable assembler and C compiler run only under the OSs on the corresponding host machines described above.

PROM programming tools

Hardware	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.			
	UNISITE 2900	PROM programmer manufactured by Data I/O Japan.			
	PA-78P328CW PA-78P328GF	Programmer adapter for writing programs to the μPD78P328. Used with a PROM programmer such as the PG-1500. PA-78P328CW: For μPD78P328CW and μPD78P328DW PA-78P328GF : For μPD78P328GF			
Software	PG-1500 controller	This program enables the host machine to control the PG-1500 through the serial and parallel interfaces.			
		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 or compatibles	PC DOS	3.5-inch 2HC	μS7B13PG1500
				5.25-inch 2HC	μS7B10PG1500

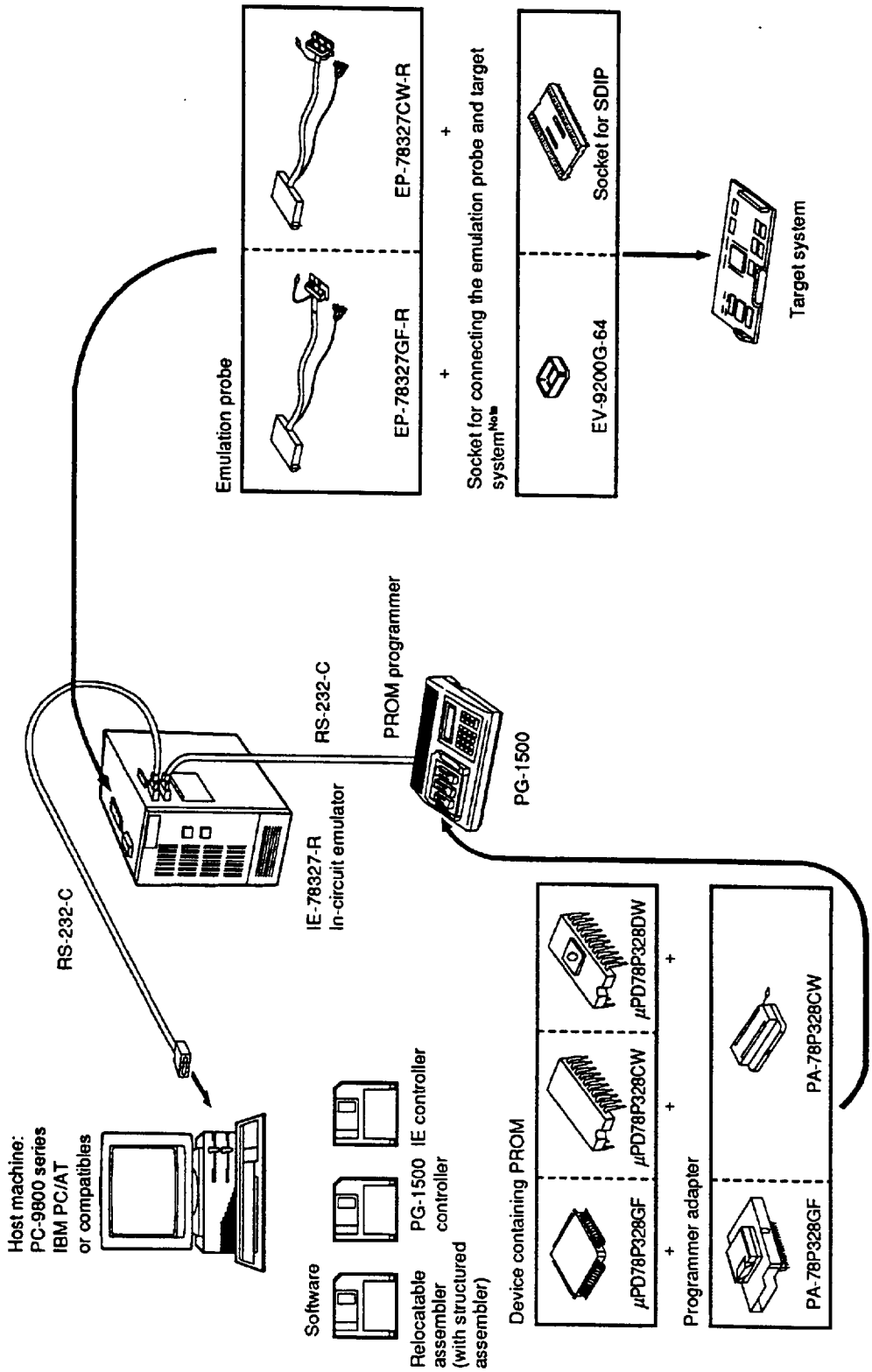
Remark It is only guaranteed that the software operates under the OSs on the corresponding host machines described above.

Debugging tools

Hardware	IE-78327-R	In-circuit emulator for developing and debugging an application system. For debugging, connect the emulator to the host machine.			
	EP-78327CW-R	Emulation probe for connecting the IE-78327-R to target systems with a 64-pin plastic shrink DIP.			
	EP-78327GF-R EV-9200G-64	Emulation probe for connecting the IE-78327-R to target systems with a 64-pin plastic QFP. One EV-9200G-64 conversion socket is provided for connection to the target system.			
Software	IE-78327-R control program (IE-controller)	This control program allows the user to control the IE-78327-R from the host machine. Its automatic command execution function ensures more efficient debugging.			
		Host machine	OS	Distribution media	Part number
		PC-9800 series	MS-DOS	3.5-inch 2HD	μS5A13IE78327
				5.25-inch 2HD	μS5A10IE78327
		IBM PC/AT or compatibles	PC DOS	3.5-inch 2HC	μS7B13IE78327
				5.25-inch 2HC	μS7B10IE78327

Remark It is only guaranteed that the software operates under the OSs on the corresponding host machines described above.

Configuration of development tools



Note The socket is supplied with the emulation probe.

Remarks 1. The PG-1500 can be directly connected to the host machine via the RS-232-C interface.

2. In this figure, the distribution media of software is represented by the 3.5-inch floppy disk.

B.2 EVALUATION TOOLS

The following evaluation tools are provided for evaluating the functions of the μPD78327 and μPD78328:

Part number	Host machine	Function
EB-78327-98	PC-9800 series	When the evaluation tool is connected to the host machine, the functions of the μPD78327 and μPD78328 can easily be evaluated. As the command system of the EB-78327-98/PC conforms to that of the IE-78327-R, the migration can easily be made to the development of the application system with the IE-78327-R. The turbo access manager (μPD71P301)[Note] can be mounted on the EB-78327-98/PC.
EB-78327-PC	IBM PC/AT or compatibles	

Note The turbo access manager is maintenance service only.

Cautions 1. These products are not development tools for the application system that uses the μPD78327 or μPD78328.

2. These products do not have the emulation function for executing the program in the internal ROM of the μPD78328.

B.3 EMBEDDED SOFTWARE

To improve the efficiency of program development and simplify the maintenance of systems incorporating this microcontroller, the following embedded software is provided.

Real-time OS

Real-time OS (RX78K/III)	This operating system was designed to provide a multitasking environment for control applications that require real-time processing. System performance is improved by using the idling CPU for other processing. RX78K/III provides system calls that conform to μITRON specifications. The RX78K/III package provides the RX78K/III nucleus and a tool (Configurator) that is used for creating multiple information tables.			
	Host machine	OS	Distribution media	Part number
	PC-9800 series	MS-DOS	3.5-inch 2HD	μS5A13RX78320
			5.25-inch 2HD	μS5A10RX78320
	IBM PC/AT or compatibles	PC DOS	3.5-inch 2HC	μS7B13RX78320
			5.25-inch 2HC	μS7B10RX78320

Caution Before purchasing this software, complete the purchase application sheet and sign the software license agreement.

Remark To use the RX78K/III real-time operating system, the optional RA78K/III assembler package is required.

Fuzzy inference development support system

Tool for creating fuzzy knowledge data (FE9000, FE9200)

This program supports the input/editing and simulation of fuzzy knowledge data (fuzzy rules and membership functions).

Host machine	OS		Distribution media	Part number
PC-9800 series	MS-DOS		3.5-inch 2HD	μS5A13FE9000
			5.25-inch 2HD	μS5A10FE9000
IBM PC/AT or compatibles	PC DOS	Windows™	3.5-inch 2HC	μS7B13FE9200
			5.25-inch 2HC	μS7B10FE9200

Translator (FT78K3)Note

This program converts fuzzy knowledge data, obtained using the tool for creating fuzzy knowledge data, into an assembler source program for RA78K/III.

Host machine	OS	Distribution media	Part number
PC-9800 series	MS-DOS	3.5-inch 2HD	μS5A13FT78K3
		5.25-inch 2HD	μS5A10FT78K3
IBM PC/AT or compatibles	PC DOS	3.5-inch 2HC	μS7B13FT78K3
		5.25-inch 2HC	μS7B10FT78K3

Fuzzy inference module (FI78K/III)Note

This program performs fuzzy inference by linking the fuzzy knowledge data converted by Translator.

Host machine	OS	Distribution media	Part number
PC-9800 series	MS-DOS	3.5-inch 2HD	μS5A13FI78K3
		5.25-inch 2HD	μS5A10FI78K3
IBM PC/AT or compatibles	PC DOS	3.5-inch 2HC	μS7B13FI78K3
		5.25-inch 2HC	μS7B10FI78K3

Fuzzy inference debugger (FD78K/III)

This software supports the evaluation and adjustment of fuzzy knowledge data at the hardware level, by using an in-circuit emulator.

Host machine	OS	Distribution media	Part number
PC-9800 series	MS-DOS	3.5-inch 2HD	μS5A13FD78K3
		5.25-inch 2HD	μS5A10FD78K3
IBM PC/AT or compatibles	PC DOS	3.5-inch 2HC	μS7B13FD78K3
		5.25-inch 2HC	μS7B10FD78K3

Note Under development

[MEMO]

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.

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**

U10208EJ5V0DS00