DATA SHEET

NEC

MOS INTEGRATED CIRCUIT

μPD78350A, 78352A

16/8 BIT SINGLE-CHIP MICROCOMPUTER

The μPD78352A is a product of the 16/8-bit single-chip microcomputer 78K/III series. It contains a 16-bit high-performance CPU.

The μPD78352A contains only hardware necessary for operating as an ASIC controller so that a unique application system with the ASIC connected can be developed. And, since the sum-of-products instruction is added to enhance operation functions, the μPD78352A can be used in many fields as high-speed, simple CPU.

The μPD78350A is a ROM-less version of the μPD78352A. The μPD78P352 is a PROM version of the μPD78352A.

The following user's manuals completely describe the functions of the μPD78350 and μPD78352A. Be sure to read them before designing an application system.

μPD78352A User's Manual, Hardware: IEU-781

μPD78356 User's Manual, Instruction: IEU-853

FEATURES

- 16-bit internal architecture, 8-bit external data bus
- High-speed data processing using the pipeline control system and high-speed operation clock
  - Minimum instruction execution time: 125 ns (in operation at an internal clock of 16 MHz or an external clock of 32 MHz)
- Internal memory: ROM: Not provided (μPD78350A)
  32K bytes (μPD78352A)
  RAM: 640 bytes
- An instruction set suited for control applications (μPD78322 upward compatible)
  - Multiply/divide instruction (16 bits × 16 bits, 32 bits ÷ 16 bits)
  - Sum-of-products operation instruction (16 bits × 16 bits + 32 bits)
  - Bit manipulation instruction and so on
- Built-in high-speed interrupt controller
  - A 4-level priority can be specified.
  - One interrupt processing mode can be selected out of three types: vectored interrupt function, macro service function, and context switching function.
- 8-bit PWM signal output function: 2 channels

APPLICATIONS

- Office automation (OA) field such as for hard disk drive or floppy disk drive control
- Factory automation (FA) field

Unless otherwise specified, the description of the μPD78352A applies to the μPD78350A.

Document No. IC-3391
(O.D.No. IC-8823)
Date Published April 1994 P
Printed in Japan

ORDERING INFORMATION

Part number	Package	Internal ROM
μPD78350AG-22	64-pin plastic QFP (14 × 14 mm)	Not provided
μPD78352AG-xxx-22	64-pin plastic QFP (14 × 14 mm)	Mask ROM

Remark xxx indicates a ROM code.

QUALITY GRADE

Standard

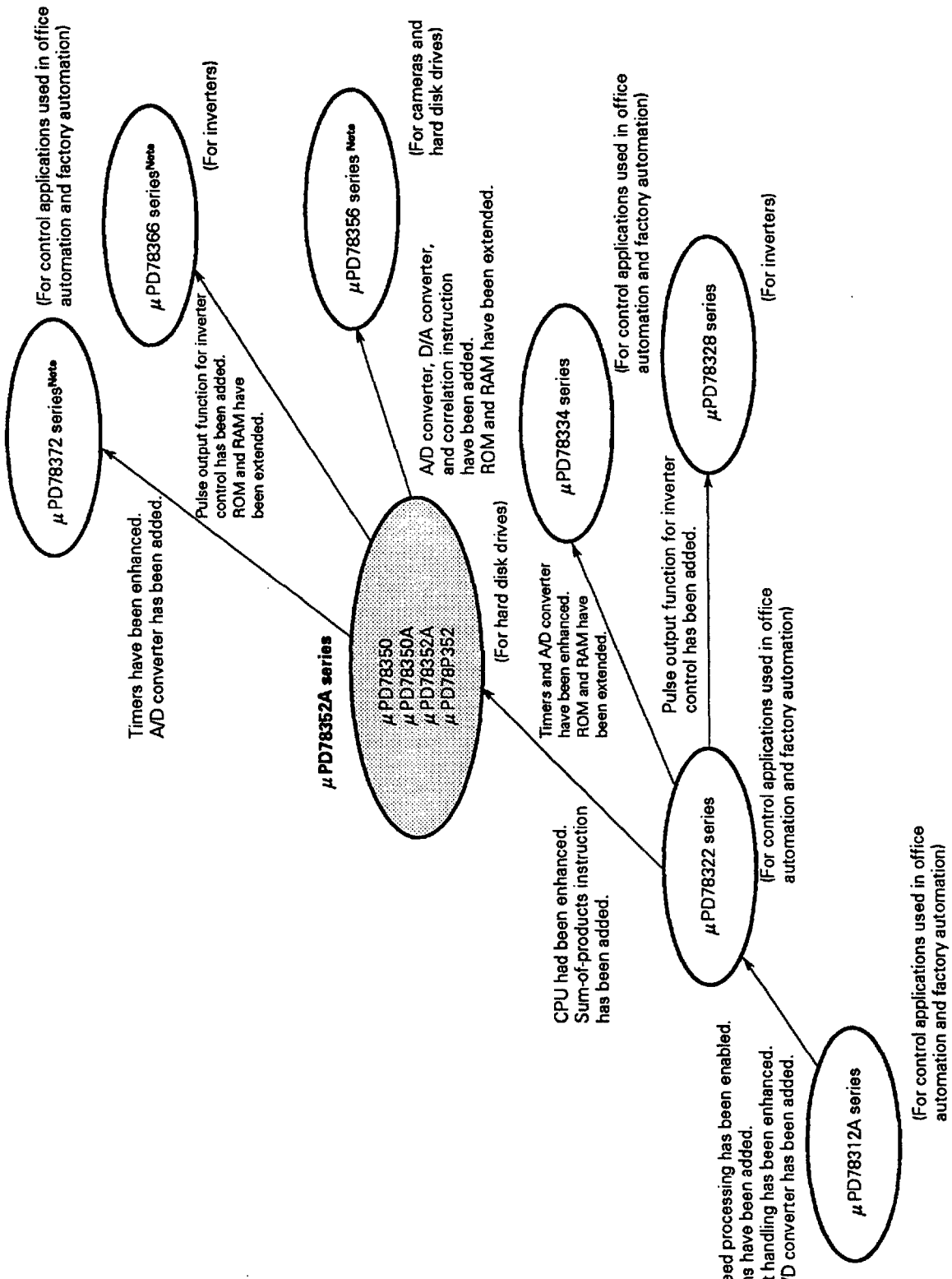
Please refer to *Quality Grades on NEC Semiconductor Devices* (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

DIFFERENCES BETWEEN THE μPD78352A AND μPD78350A

Item \ Part number		μPD78352A	μPD78350A
Internal ROM		32K bytes	Not provided
I/O pins	Input pins	6	
	I/O pins	44	24
	Total	50	30
P40/AD0 - P47/AD7		8-bit I/O port (P40 to P47) These pins function as an address/data bus (AD0 to AD7) when the external memory is expanded. **Note**	P40 to P47 are not provided. These pins function only as an address data bus (AD0 to AD7).
P50/A8 - P57/A15		8-bit I/O port (P50 to P57) These pins function as an address bus (A8 to A15) when the external memory is expanded. **Note**	P50 to P57 are not provided. These pins function only as an address bus (A8 to A15).
P90/$\overline{RD}$, P91/$\overline{WR}$		4-bit I/O port (P90 to P93) When the external memory is expanded, P90 functions as the $\overline{RD}$ output pin and P91 functions as the $\overline{WR}$ output pin. P92 functions as the IC pin.	P90 and P91 are not provided. These pins function only as $\overline{RD}$ and $\overline{WR}$ strobe signal output pins.
P92/IC•Open, P93			P92 and P93 are not provided.
MODE0, MODE1		• Normal operating mode Set both MODE0 and MODE1 low. • ROM-less mode Set MODE0 high and MODE1 low.	Always set MODE0 high and MODE1 low.
Access to external memory		Setting the memory expansion mode register (MM) allows external memory to be expanded sequentially to the following size: 256 bytes, 4K bytes, 16K bytes, and 32K bytes.	An external memory of 64K bytes can be accessed independently of the setting of the memory expansion mode register (MM).

Note Which pins are used as an address bus depends upon the size of external memory (set by the memory expansion mode register).

78K/III SERIES PRODUCT DEVELOPMENT

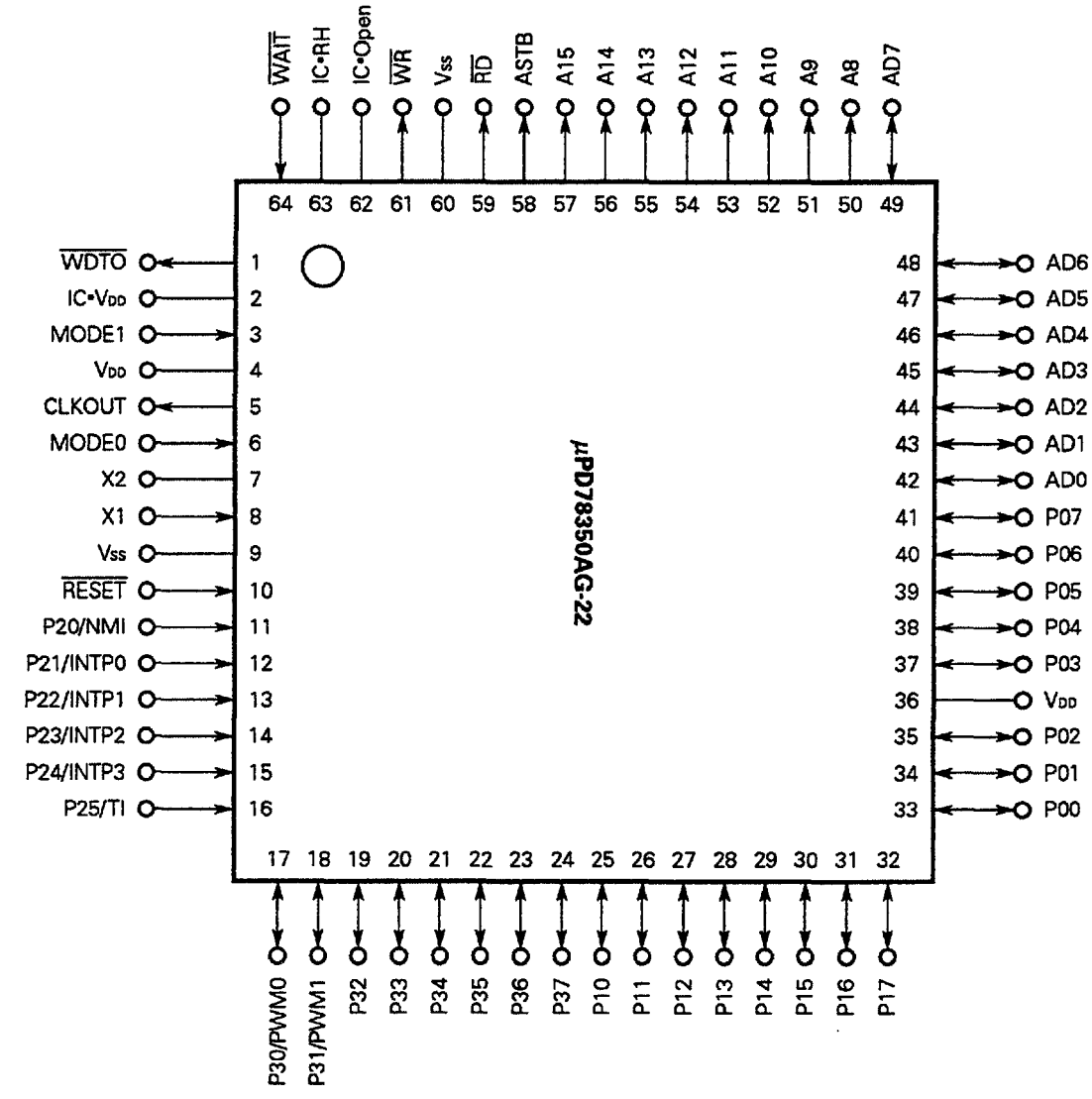


Note Under development

PIN CONFIGURATION (TOP VIEW)

(1) Pin configuration of the μPD78350A

64-pin plastic QFP (14 × 14 mm)



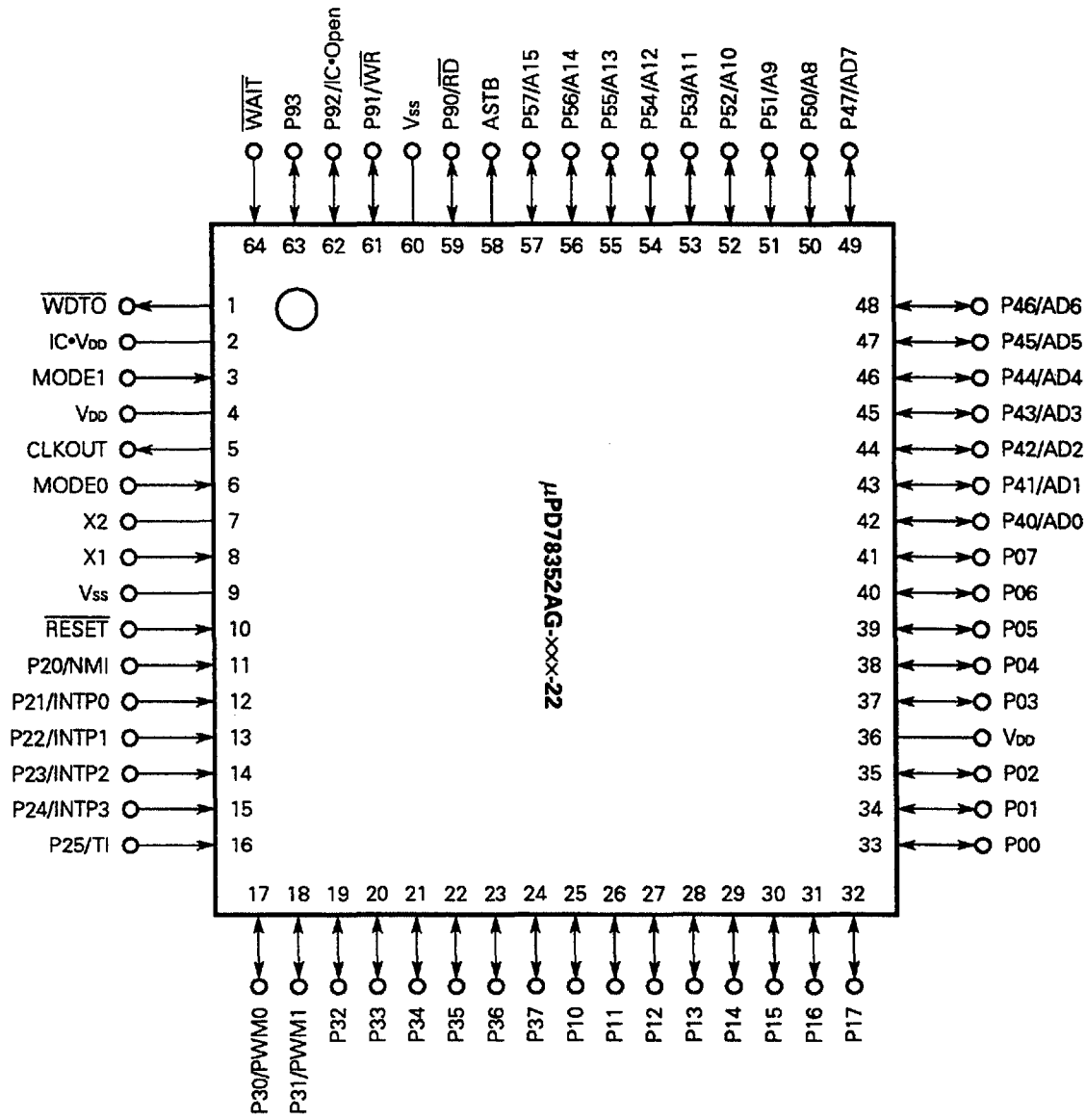
Cautions 1. Connect the IC•VDD pin directly to the VDD pin.

2. Connect the IC•RH pin to the VDD pin through a resistor.

3. Leave the IC•Open pin open.

(2) Pin configuration of the μPD78352A

64-pin plastic QFP (14 × 14 mm)



Cautions 1. Connect the IC•VDD pin directly to the VDD pin.

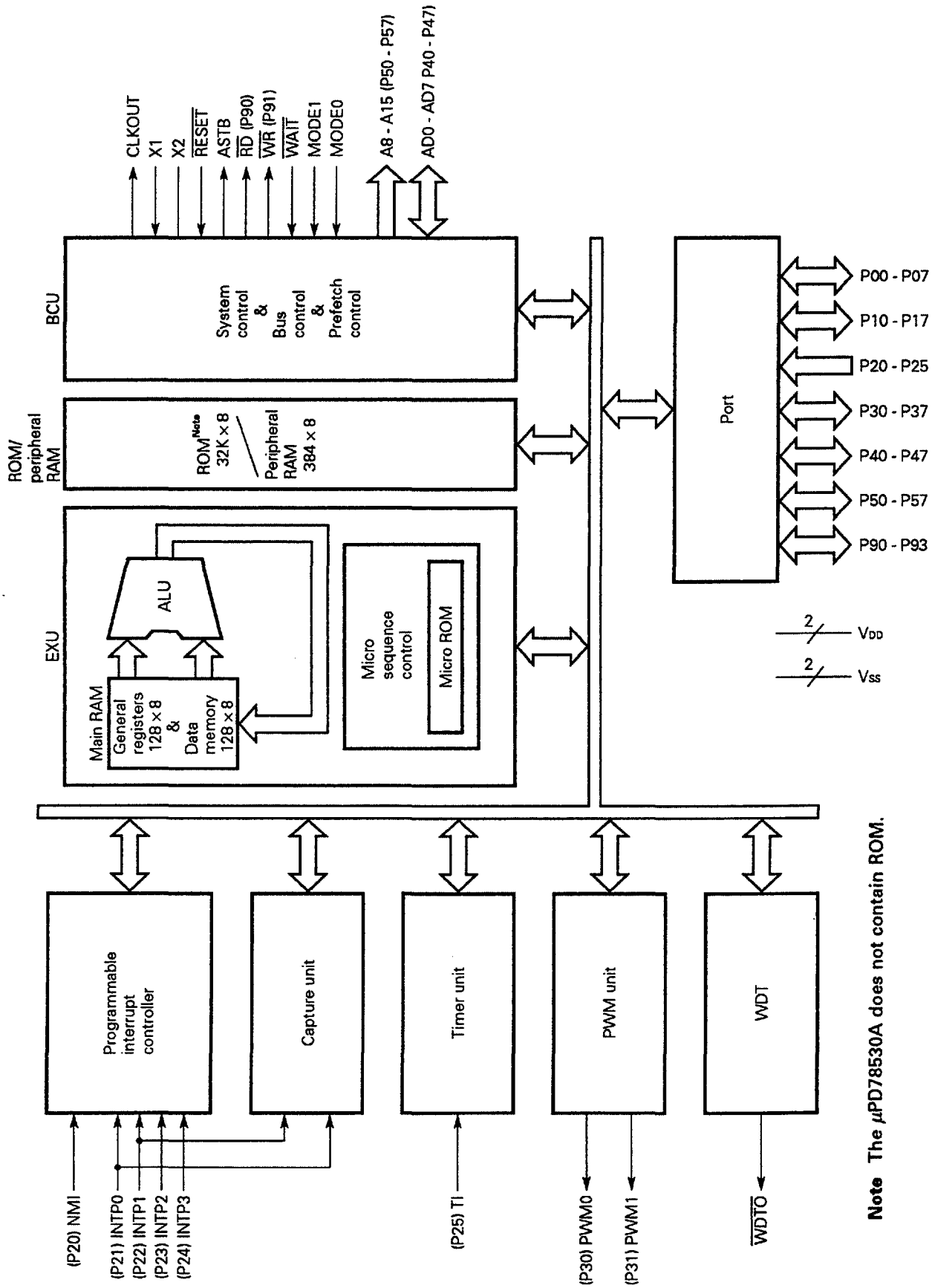
2. Leave the P92/IC•Open pin open when using an external expansion memory.

P00 - P07	: Port 0
P10 - P17	: Port 1
P20 - P25	: Port 2
P30 - P37	: Port 3
P40 - P47	: Port 4
P50 - P57	: Port 5
P90 - P93	: Port 9
NMI	: Nonmaskable interrupt
INTP0 - INTP3	: Interrupt from peripherals
TI	: Timer input
PWM0, PWM1	: Pulse width modulation output
$\overline{\text{WDTO}}$	: Watchdog timer output
MODE0, MODE1	: Mode
AD0 - AD7	: Address/data bus
A8 - A15	: Address bus
ASTB	: Address strobe
$\overline{\text{RD}}$	: Read strobe
$\overline{\text{WR}}$	: Write strobe
CLKOUT	: Clock output
$\overline{\text{WAIT}}$	: Wait
$\overline{\text{RESET}}$	: Reset
X1, X2	: Crystal
V_{DD}	: Power supply
Vss	: Ground
IC•×××	: Internally connected

FUNCTION OVERVIEW

Item	Description
Number of basic instructions	113
Minimum instruction execution time	125 ns (in operation at an internal clock of 16 MHz or an external clock of 32 MHz)
Internal memory	• ROM: 32K bytes (μPD78352A) Not provided (μPD78350A) • RAM: 640 bytes
Memory space	64K bytes (can externally be expanded)
General register	8 bits × 16 × 8 banks
Instruction set	• 16-bit transfer or arithmetic/logical instruction • Multiply/divide instruction (16 bits × 16 bits, 32 bits 16 bits) • Bit manipulation instruction • String instruction • Sum-of-products instruction (16 bits × 16 bits + 32 bits)
Capture/timer unit	• One 16-bit free running timer • One 16-bit timer/event counter • One 16-bit interval timer • Two 16-bit capture registers • Two 16-bit compare registers
Interrupt function	• Five external and four internal sources • A 4-level priority can be specified by software. • One interrupt processing mode can be selected out of three types: vectored interrupt function, macro service function, and context switching function.
I/O line	• Six input ports • 44 I/O ports (μPD78352A) 24 I/O ports (μPD78350A)
PWM unit	Two-channel 8-bit PWM outputs
Package	64-pin plastic QFP (14 × 14 mm)
Others	• Watchdog timer function • Standby function (HALT mode, STOP mode) • Wait control pin

BLOCK DIAGRAM



Note The μPD78530A does not contain ROM.

CONTENTS

1. PIN FUNCTIONS

1.1 PIN FUNCTIONS OF THE μPD78350A

(1) Port pins

Pin	I/O	Function	Dual-function pin
P00 - P07	I/O	Port 0 8-bit I/O port Can be specified as input or output bit by bit.	–
P10 - P17	I/O	Port 1 8-bit I/O port Can be specified as input or output bit by bit.	–
P20	I	Port 2 6-bit input dedicated port	NMI
P21			INTP0
P22			INTP1
P23			INTP2
P24			INTP3
P25			TI
P30	I/O	Port 3 8-bit I/O port Can be specified as input or output bit by bit.	PWM0
P31			PWM1
P32 - P37			–

(2) Non-port pins

Pin	I/O	Function	Dual-function pin
NMI	I	Nonmaskable interrupt request input	P20
INTP0		External interrupt request input	P21
INTP1			P22
INTP2			P23
INTP3			P24
TI		External count input to timer 1 (TM1)	P25
PWM0	O	PWM signal output	P30
PWM1			P31
$\overline{\text{WDTO}}$		Signal output which indicates the occurrence of a watchdog timer interrupt	–
MODE0	I	Control signal input to set an operation mode. Normally, connect the MODE0 pin to the V_{DD} pin and the MODE1 pin to the V_{SS} pin.	–
MODE1			
AD0 - AD7	I/O	Multiplexed address/data bus when memory is expanded externally	–
A8 - A15	O	Address bus when memory is expanded externally	–
ASTB	O	Address strobe signal output	–
$\overline{\text{RD}}$		Read strobe signal output to the external device	–
$\overline{\text{WR}}$		Write strobe signal output to the external memory	–
CLKOUT		System clock output	–
$\overline{\text{WAIT}}$	I	Control signal input to set a bus cycle to the wait state	–
$\overline{\text{RESET}}$	I	System reset input	–
X1	I	Connected to a crystal used for system clock oscillation: An external signal (if used) is input to the X1 pin. Leave the X2 pin open.	–
X2	–		
V_{DD}	–	Positive power supply	–
V_{SS}	–	Ground	–
IC•V_{DD}	–	Internally connected pin. Connect this pin directly to the V_{DD} pin.	–
IC•RH	–	Internally connected pin. Connect this pin to the V_{DD} pin through a resistor.	
IC•Open	–	Internally connected pin. Leave this pin open.	–

1.2 PIN FUNCTIONS OF THE μPD78352A

(1) Port pins

Pin	I/O	Function	Dual-function pin
P00 - P07	I/O	Port 0 8-bit I/O port Can be specified as input or output bit by bit.	–
P10 - P17	I/O	Port 1 8-bit I/O port Can be specified as input or output bit by bit.	–
P20	I	Port 2 6-bit input dedicated port	NMI
P21			INTP0
P22			INTP1
P23			INTP2
P24			INTP3
P25			TI
P30	I/O	Port 3 8-bit I/O port Can be specified as input or output bit by bit.	PWM0
P31			PWM1
P32 - P37			–
P40 - P47	I/O	Port 4 8-bit I/O port Can be specified as input or output in units of 8 bits.	AD0 - AD7
P50 - P57	I/O	Port 5 8-bit I/O port Can be specified as input or output bit by bit.	A8 - A15
P90	I/O	Port 9 4-bit I/O port Can be specified as input or output bit by bit.	$\overline{RD}$
P91			$\overline{WR}$
P92			IC•Open
P93			–

(2) Non-port pins

Pin	I/O	Function	Dual-function pin
NMI	I	Nonmaskable interrupt request input	P20
INTP0		External interrupt request input	P21
INTP1			P22
INTP2			P23
INTP3			P24
TI		External count input to timer 1 (TM1)	P25
PWM0	O	PWM signal output	P30
PWM1			P31
$\overline{\text{WDTO}}$		Signal output which indicates the occurrence of a watchdog timer interrupt	–
MODE0	I	Control signal input to set an operation mode. Normally, connect the MODE0 and MODE1 pins to the Vss pin.	–
MODE1			
AD0 - AD7	I/O	Multiplexed address/data bus when an external memory is expanded	P40 - P47
A8 - A15	O	Address bus when an external memory is expanded	P50 - P57
ASTB	O	Address strobe signal output	–
$\overline{\text{RD}}$		Read strobe signal output to the external device	P90
$\overline{\text{WR}}$		Write strobe signal output to the external memory	P91
CLKOUT		System clock output	–
$\overline{\text{WAIT}}$	I	Control signal input to set a bus cycle to the wait state	–
$\overline{\text{RESET}}$	I	System reset input	–
X1	I	Crystal input pin for system clock oscillation: An external signal (if used) is input to the X1 pin. Leave the X2 pin open.	–
X2	–		
V_{DD}	–	Positive power supply	–
Vss	–	Ground	–
IC•V_{DD}	–	Internal connected pin. Connect this pin directly to the V_{DD} pin.	–
IC•Open	–	Internal connected pin. Leave this pin open.	P92

1.3 INPUT/OUTPUT CIRCUITS OF EACH PIN AND CONNECTION OF UNUSED PINS

Tables 1-1, 1-2, and Fig. 1-1 show the input and output circuits of each pin in a simplified format.

Table 1-1 Input/Output Circuits of Each Pin and Recommended Connection of Unused Pins (μPD78350A)

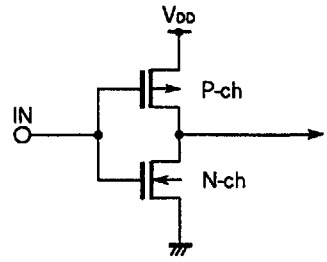
Pin	I/O circuit type	Recommended connection
P00 - P07	5	Input status: To be connected to the V_{DD} or V_{SS} pin via a resistor Output status: Open
P10 - P17		
P30/PWM0 P31/PWM1 P32 - P37		
AD0 - AD7		
A8 - A15		
P20/NMI P21/INTP0 - P24/INTP3 P25/TI	2	To be connected to the V_{SS} pin
$\overline{RD}$ $\overline{WR}$	5	Open
ASTB	4	
CLKOUT	3	
$\overline{WDTO}$	19	To be connected to the V_{SS} pin
$\overline{WAIT}$	1	To be connected to the V_{DD} pin
MODE0 MODE1		–
$\overline{RESET}$	2	
IC•V_{DD}	–	To be directly connected to the V_{DD} pin
IC•RH		To be connected to the V_{DD} pin via a resistor
IC•Open		Open

Table 1-2 Input/Output Circuits of Each Pin and Recommended Connection of Unused Pins (μPD78352A)

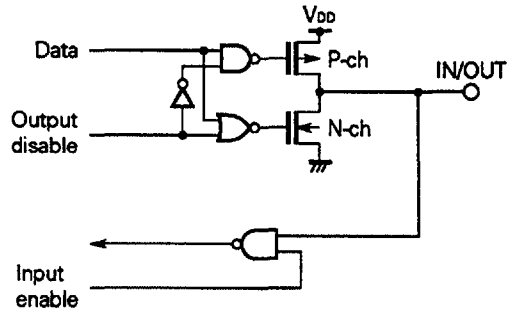
Pin	I/O circuit type	Recommended connection
P00 - P07	5	Input status: To be connected to the V_{DD} or V_{SS} pin via a resistor Output status: Open
P10 - P17		
P30/PWM0 P31/PWM1 P32 - P37		
P40/AD0 - P47/AD7		
P50/A8 - P57/A15		
P20/NMI P21/INTP0 - P24/INTP3 P25/TI	2	To be connected to the V_{SS} pin
P90/$\overline{RD}$ P91/$\overline{WR}$ P92/IC•Open P93	5	Input status: To be connected to the V_{DD} or V_{SS} pin via a resistor Output status: Open
ASTB	4	Open
CLKOUT	3	
$\overline{WDTO}$	19	To be connected to the V_{SS} pin
$\overline{WAIT}$	1	To be connected to the V_{DD} pin
MODE0 MODE1		–
$\overline{RESET}$	2	
IC•V_{DD}	–	To be directly connected to the V_{DD} pin

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

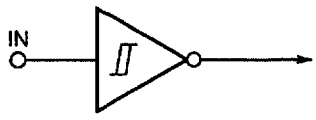
Type 1



Type 5

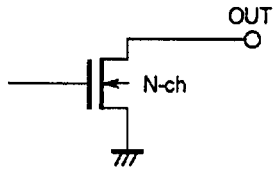


Type 2

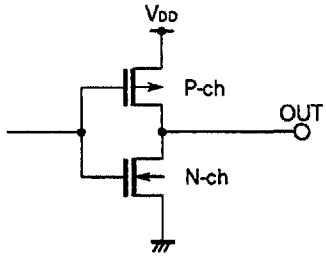


Schmitt trigger input with hysteresis characteristics

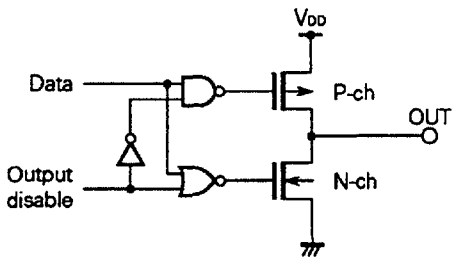
Type 19



Type 3



Type 4



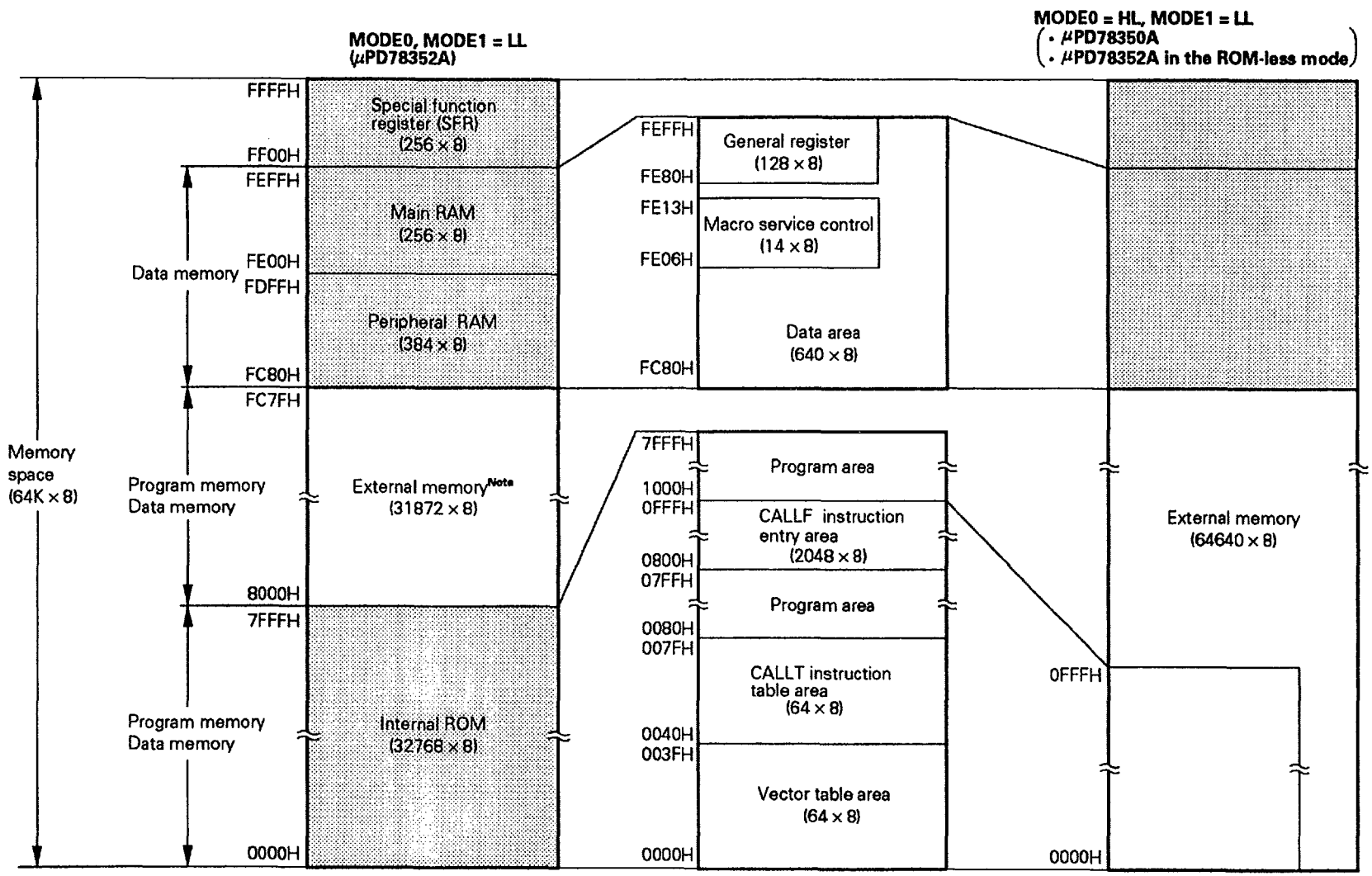
Push-pull output which can have high impedance
(Both the positive and negative channels are off.)

2. CPU ARCHITECTURE

2.1 MEMORY SPACE

The μPD78352A can access memory of up to 64K bytes. Fig. 2-1 shows the memory map.

Fig. 2-1 Memory Map



Note Access in the external memory expansion mode.

Remark Shaded portions indicate internal memory.

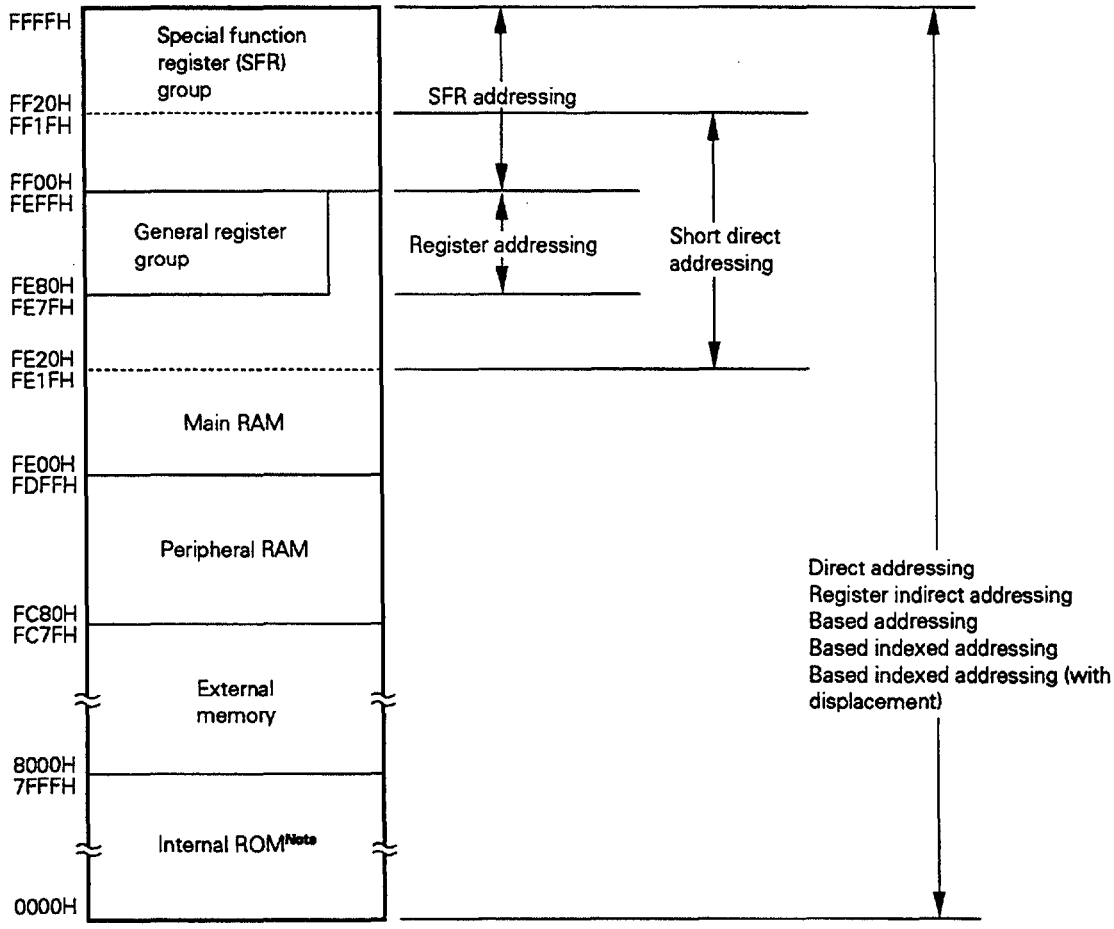
Caution **When word access (including the stack operation) to the main RAM space (FE00H to FEFFH) is executed, the addresses specified in the operand must be even numbers.**

2.2 DATA MEMORY ADDRESSING

Various addressing modes are provided for the μPD78352A to improve memory operability or to enable the use of a high-level language. Special addressing is applicable, in particular, to the space of data memory from FC80H to FFFFH according to each function of the special function register (SFR) group and general register group.

Fig. 2-2 shows the addressing space of data memory.

Fig. 2-2 Addressing Space of Data Memory



Note External memory is assigned for the μPD78350A or in the ROM-less mode of the μPD78352A.

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 PROCESSOR REGISTERS

The μPD78352A contains three processor register groups.

2.3.1 Control Registers

(1) Program counter (PC)

The program counter is a 16-bit register which contains the address of the next instruction to be executed.

(2) Program status word (PSW)

The program status word is a 16-bit register which contains the status of the CPU according to the instruction execution result.

(3) Stack pointer (SP)

The stack pointer is a register which contains the first address of the stack area (LIFO type) in memory.

(4) CPU control word (CCW)

The CPU control word is an 8-bit register which is related to CPU control.

Fig. 2-3 Control Register Configuration

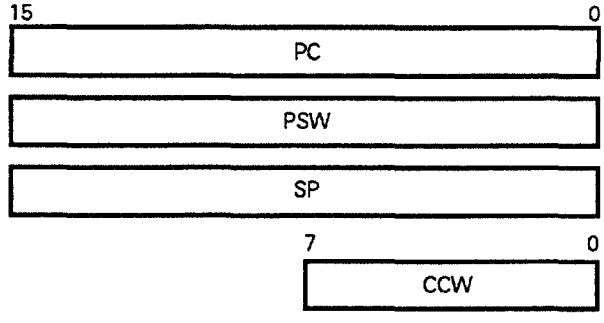
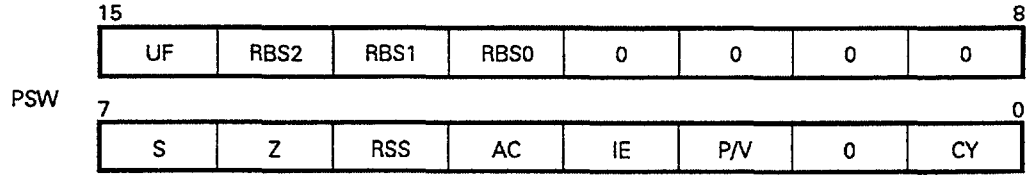
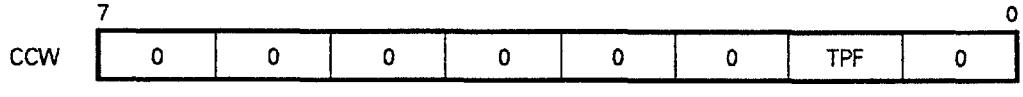


Fig. 2-4 PSW Configuration



- UF : User flag
- RBS0-RBS2: Register bank selection flag
- S : Sign flag (MSB after arithmetic/logical operation)
- Z : Zero flag
- RSS : Register set selection flag
- AC : Auxiliary carry flag
- IE : Interrupt request enable flag
- P/V : Parity/overflow flag
- CY : Carry flag

Fig. 2-5 CCW Configuration



TPF: Table position flag

2.3.2 General Register

The general register group consists of eight banks (one bank: 8 words × 16 bits). Fig. 2-6 shows general register configuration. The general register group is mapped into addresses from FE80H to FFEFH, and functions as a 16-bit register as well as an 8-bit register (see Fig. 2-7). The use of this register enables easy control of complicated multitask processing.

Fig. 2-6 General Register Configuration

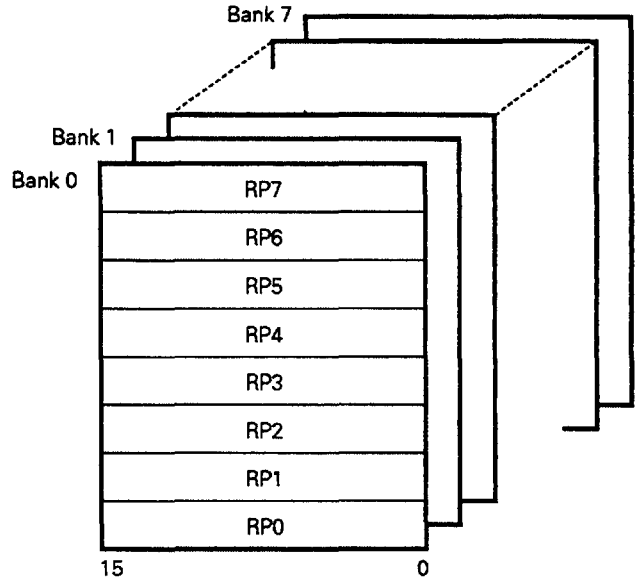
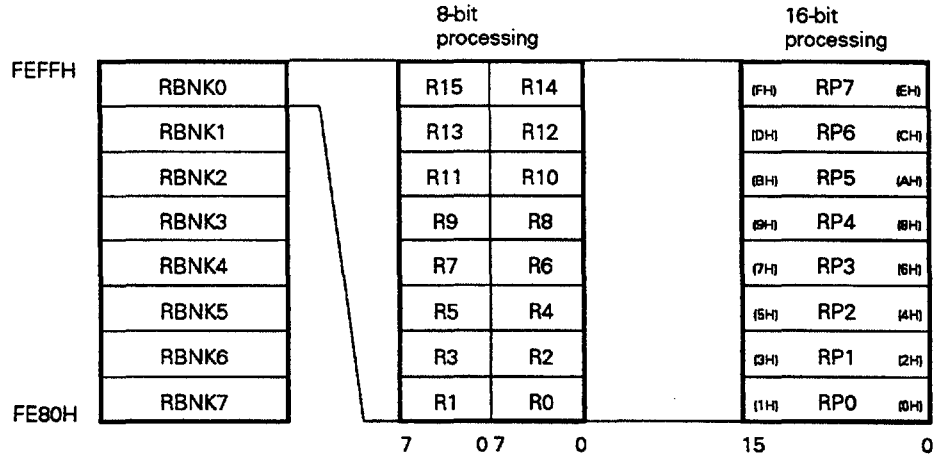


Fig. 2-7 Bit Processing for General Register



2.3.3 Special Function Registers (SFR)

The special function register group consists of the registers for control of the peripheral hardware the μPD78352A contains. This register group is mapped into addresses from FF00H to FFFFH. The operation of these registers enables control of ports, a timer, and PWM unit.

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

Address	Special function register (SFR) name	Abbreviation	R/W	Unit for manipulation (bit)			When reset
				1	8	16	
FF00H	Port 0	P0	R/W	O	O	–	Undefined
FF01H	Port 1	P1		O	O	–	
FF02H	Port 2	P2	R	O	O	–	
FF03H	Port 3	P3	R/W	O	O	–	
FF04H	Port 4	P4		O	O	–	
FF05H	Port 5	P5		O	O	–	
FF09H	Port 9	P9		O	O	–	
FF10H	Capture register 00	CT00	R/W	–	–	O	
FF11H							
FF12H	Capture register 01	CT01		–	–	O	
FF13H							
FF14H	Compare register 10	CM10		–	–	O	
FF15H							
FF1EH	Compare register 20	CM20		–	–	O	
FF1FH							
FF20H	Port 0 mode register	PM0	R/W	O	O	–	FFH
FF21H	Port 1 mode register	PM1		O	O	–	
FF23H	Port 3 mode register	PM3		O	O	–	
FF25H	Port 5 mode register	PM5		O	O	–	
FF29H	Port 9 mode register	PM9		O	O	–	×FX
FF30H	Timer register 0	TM0	R	–	–	O	0000H
FF31H							
FF32H	Timer register 1	TM1		–	–	O	
FF33H							
FF34H	Timer register 2	TM2		–	–	O	
FF35H							
FF38H	Timer control register 0	TMC0	R/W	O	O	–	00H
FF39H	Timer control register 1	TMC1		O	O	–	
FF3CH	External interrupt mode register 0	INTM0		O	O	–	
FF3DH	External interrupt mode register 1	INTM1		O	O	–	

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

Address	Special function register (SFR) name	Abbreviation	R/W	Unit for manipulation (bit)			When reset
				1	8	16	
FF43H	Port 3 mode control register	PMC3	R/W	O	O	–	00H
FF62H	Port read control register	PRDC		O	O	–	
FF64H	PWM control register	PWMC		O	O	–	
FF66H	PWM buffer register 0	PWM0		O	O	–	Undefined
FF6EH	PWM buffer register 1	PWM1		O	O	–	
FFA8H	In-service priority register	ISPR	R	O	O	–	00H
FFAAH	Interrupt mode control register	IMC	R/W	O	O	–	80H
FFACH	Interrupt mask flag register	MKL		O	O	–	7FH
FFACH FFADH	Interrupt mask flag register	MK[Note 1]		–	–	O	××7FH
FFC0H	Standby control register	STBC[Note 2]		–	O	–	0000 ×000B
FFC1H	CPU control word	CCW		O	O	–	00H
FFC2H	Watchdog timer mode register	WDM[Note 2]		–	O	–	
FFC4H	Memory expansion mode register	MM		O	O	–	
FFC6H FFC7H	Programmable wait control register	PWC		–	–	O	C0AAH
FFD0H to FFDFH	External SFR area	–		O	O	–	Undefined
FFE0H	Interrupt control register (INTOV)	OVIC		O	O	–	43H
FFE1H	Interrupt control register (INTP0)	PIC0		O	O	–	
FFE2H	Interrupt control register (INTP1)	PIC1		O	O	–	
FFE3H	Interrupt control register (INTCM10)	CMIC10		O	O	–	
FFE4H	Interrupt control register (INTCM20)	CMIC20		O	O	–	
FFE5H	Interrupt control register (INTP2)	PIC2		O	O	–	
FFE6H	Interrupt control register (INTP3)	PIC3		O	O	–	

Notes 1. Used only when a word is accessed by an instruction with the sfrp operand.

2. Data can be written in the registers with special instructions.

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. The EXU contains 256-byte main RAM.

The 256-byte main RAM in the EXU can be accessed at higher speed with an instruction than 384-byte peripheral RAM.

3.3 ROM/RAM

This area consists of a 32K-byte ROM area and a 384-byte peripheral RAM area. No internal ROM is provided in the μPD78350A.

The MODE0 and MODE1 pins inhibit access to ROM and enable access to 64K-byte external memory.

3.4 INTERRUPT CONTROLLER

The interrupt controller processes various interrupt requests (NMI and INTP0 to INTP3) issued from peripheral hardware and external device with the vectored interrupt, macro service, or context switching.

The interrupt controller also specifies the 4-level interrupt priority.

3.5 CAPTURE/TIMER UNIT

The capture/timer unit consists of the following hardware.

- Three 16-bit timers/counters
- Two 16-bit capture registers
- Two 16-bit compare registers

The capture/timer unit can output a programmable pulse and measure a pulse width and frequency.

3.6 PWM UNIT

The μPD78350 has two channels of 8-bit PWM signal outputs. By connecting an external low-pass filter, a PWM output can be used as an analog voltage output.

3.7 WATCHDOG TIMER (WDT)

The 8-bit watchdog timer is built into the CPU to detect a program crash and system error. This microcomputer has the $\overline{\text{WDTO}}$ pin to notify the external device that a watchdog timer interrupt occurs.

3.8 PORT

The following ports are provided:

- Five 8-bit I/O ports : P0, P1, P3, P4, P5
- One 4-bit I/O port : P9
- One 8-bit input port : P2

Each port functions as a digital port and also functions as I/O pins for internal hardware.

4. PERIPHERAL HARDWARE FUNCTIONS

4.1 PORT FUNCTIONS

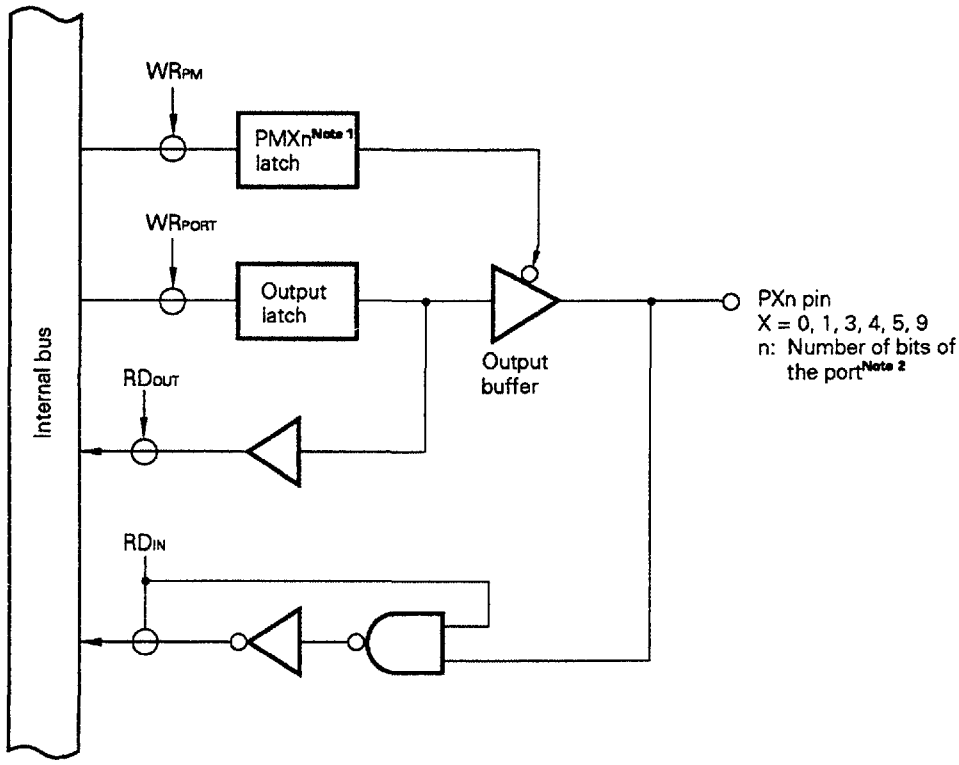
4.1.1 Hardware Configuration

As shown in Fig. 4-1, three-state bidirectional ports are basically used for the ports of the μPD78352A.

A $\overline{\text{RESET}}$ input signal sets each bit of a port mode register to 1, specifying the port as an input port. All port pins go into the high-impedance state. A $\overline{\text{RESET}}$ input signal makes the contents of the output latch undefined.

Fig. 4-2 shows the port configuration.

Fig. 4-1 Basic I/O Port Configuration

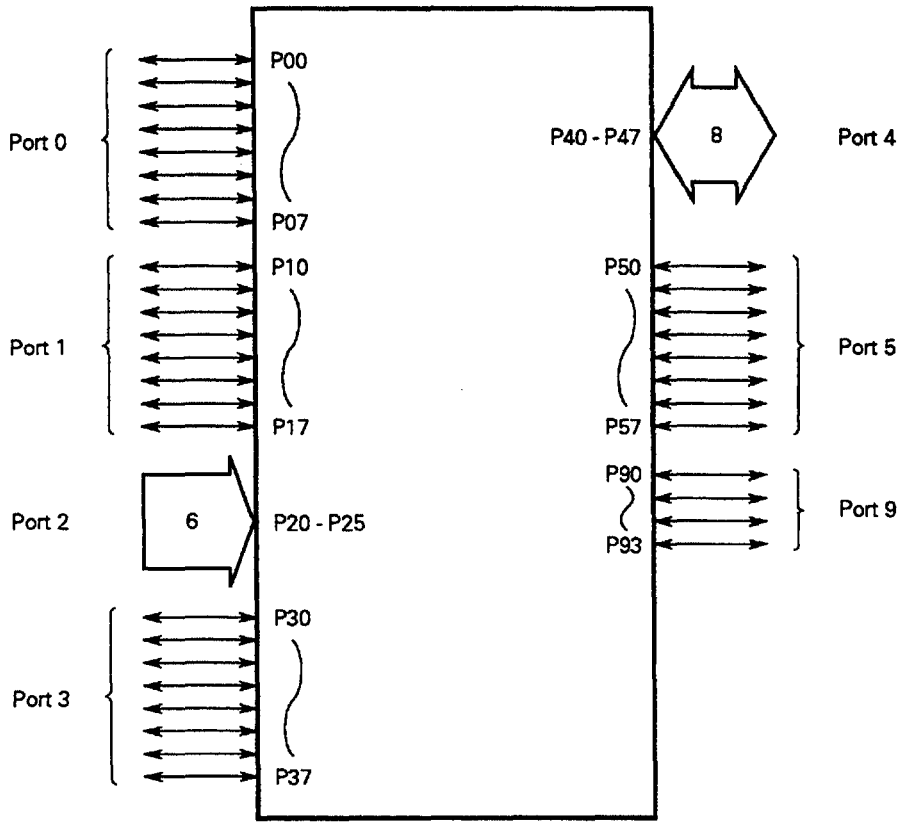


Notes 1. PMXn latch: Bit n of port mode register PMX (X = 0, 1, 3, 4, 5, 9)

2. When X = 0, 1, 3, 4, 5, n = 0 to 7.
When X = 9, n= 0 to 3.

Remark Port 2 is used only for 6-bit input.

Fig. 4-2 Port Configuration



4.1.2 Functions of the Digital I/O Ports

Table 4-1 lists the ports of the μPD78352A.

Each port allows bit manipulations as well as 8-bit data manipulations, thus enabling a wide variety of control. Each port functions as a digital port and also functions as I/O pins for internal hardware.

Table 4-1 Port Functions and Additional Functions of the Ports

Port name	Port function	Additional function
Port 0	8-bit I/O port. Specifiable as input or output bit by bit.	–
Port 1	8-bit I/O port. Specifiable as input or output bit by bit.	–
Port 2	Port used only for 6-bit input	Capture trigger input and count pulse input for capture/timer unit, and external interrupt input in control mode
Port 3	8-bit I/O port. Specifiable as input or output bit by bit.	PWM signal output in control mode
Port 4	8-bit I/O port. Specifiable as input or output in units of 8 bits.	Address/data bus for memory expansion (AD0 - AD7)
Port 5	8-bit I/O port. Specifiable for input or output bit by bit.	Address bus for memory expansion (A8 - A15)
Port 9	4-bit I/O port. Specifiable as input or output bit by bit.	Control signal output for memory expansion. The P90, P91, and P92 pins function as the $\overline{RD}$ output, $\overline{WR}$ output, and IC•Open pins respectively.

4.1.3 Port Output Check Function

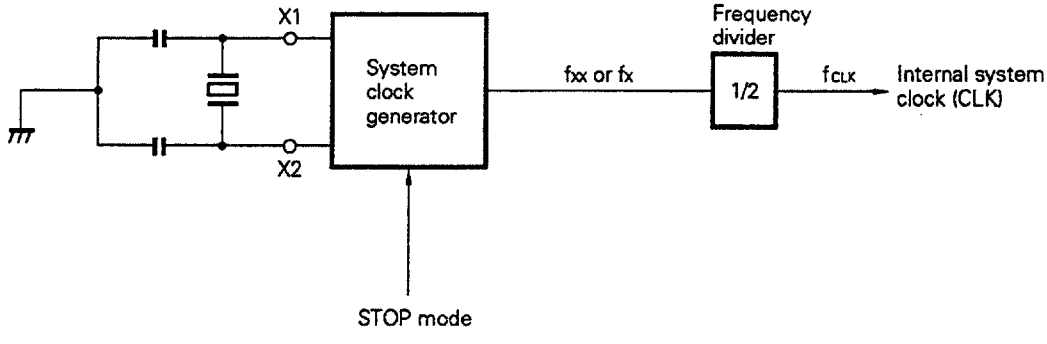
The μPD78352A has a function of reading pin state (pin access mode) to improve system application reliability in port output mode. With this function, output data (output latch data) and actual pin state can be checked as required. For frequent port state checking, special instructions (CHKL and CHKLA) are available.

4.2 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. 4-3.

Fig. 4-3 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 the standby mode (STOP) is set.

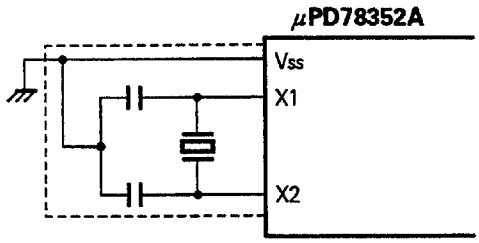
An external clock can be applied. In this case, a clock signal is to be applied to the X1 pin. The X2 pin must be left open.

Caution When using an external clock, do not set the STOP mode.

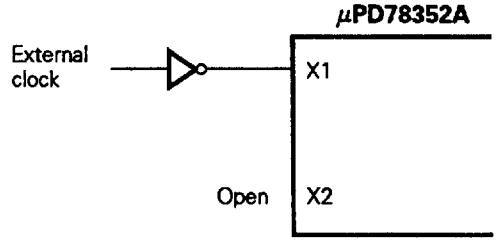
The frequency divider divides system clock generator output (fxx for the crystal oscillator or fx for an external clock) by two to produce an internal system clock (fCLK).

Fig. 4-4 External Circuitry of the System Clock Generator

(a) Crystal oscillator



(b) External clock



Cautions 1. When using the system clock generator, run wires in the portion surrounded by dotted lines in Fig. 4-4 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.

4.3 CAPTURE/TIMER UNIT

The capture/timer unit can output programmable pulses and can also measure pulse intervals and frequencies.

The capture/timer unit mainly consists of three timers and four registers.

4.3.1 Configuration of the Capture/Timer Unit

The capture/timer unit consists of the hardware components listed in Table 4-2. Fig. 4-5 shows the configuration of the capture/timer unit.

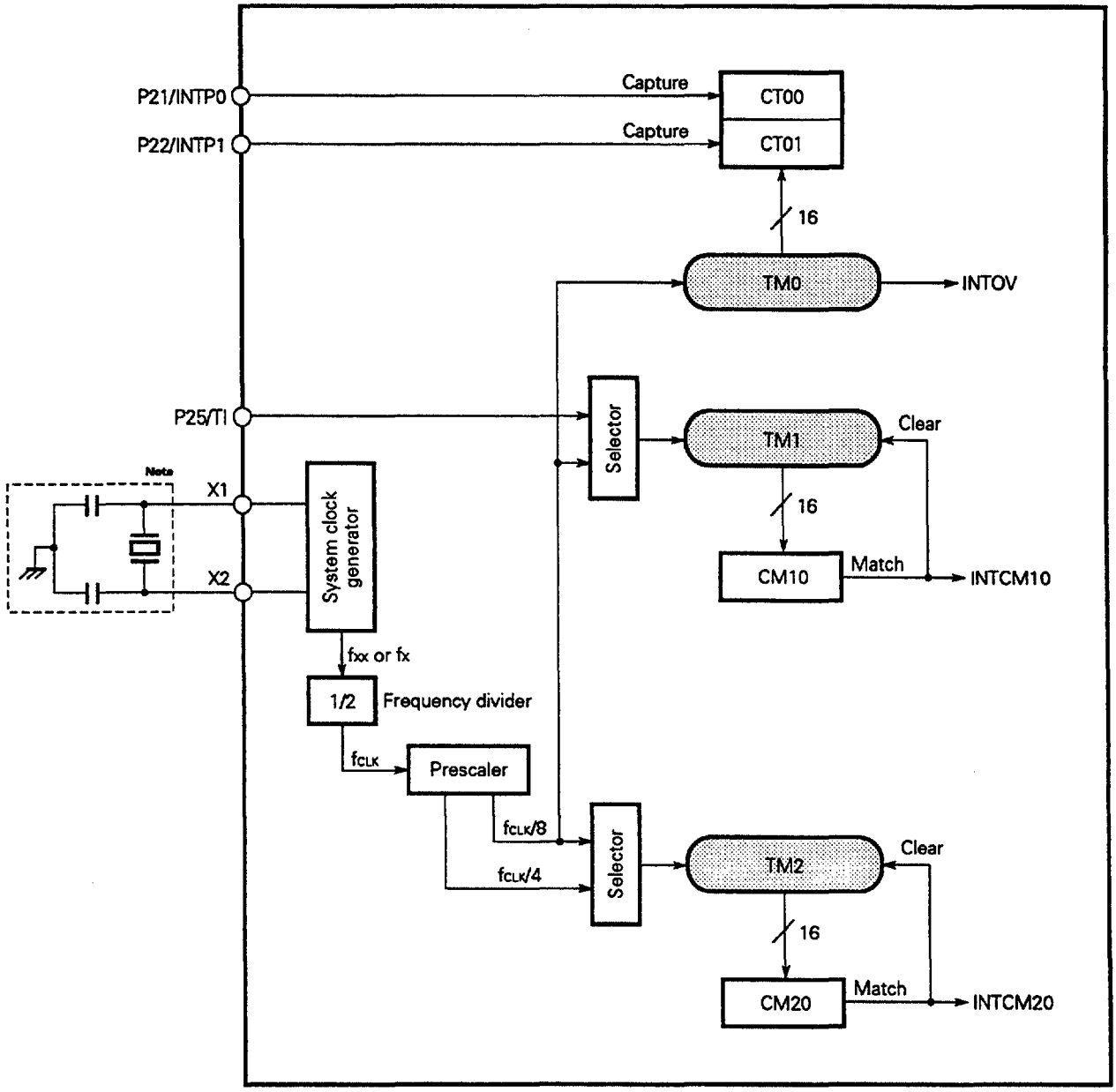
Table 4-2 Components of the Capture/Timer Unit

Timer	Count clock	Register	Compare register match interrupt	Capture trigger
16-bit timer (TM0)	$f_{CLK}/8$	16-bit capture register (CT00) 16-bit capture register (CT01)	– –	INTP0 INTP1
16-bit timer (TM1)	$f_{CLK}/8$ TI pin input	16-bit compare register (CM10)	INTCM10	–
16-bit timer (TM2)	$f_{CLK}/4$ $f_{CLK}/8$	16-bit compare register (CM20)	INTCM20	–

Remarks 1. f_{CLK}: Internal system clock

2. INTP0, INTP1: External interrupt
3. Timer 0 has an overflow interrupt function.
4. Timer 1 is cleared by INTCM10.
5. Timer 2 is cleared by INTCM20.

Fig. 4-5 Configuration of the Capture/Timer Unit



Note An external clock can be used instead of the circuit surrounded by dotted lines. Apply a clock signal to the X1 pin and leave the X2 pin open.

4.3.2 Function

(1) Timer 0 (TM0)

Timer 0 is a 16-bit free-running timer.

Timer 0 counts an internal clock, and generates an overflow interrupt (INTOV) when a timer overflow occurs.

(2) Timer 1 (TM1)

Timer 1 is a 16-bit timer/event counter. Timer 1 can count an internal clock or external event applied to the TI pin.

Timer 1 can be cleared by a match interrupt (INTCM10) from the compare register CM10.

(3) Timer 2 (TM2)

Timer 2 is a 16-bit interval timer. Timer 2 counts an internal clock. Timer 2 is cleared by a match interrupt (INTCM20) from the compare register CM20.

(4) 16-bit compare registers (CM10 and CM20)

A 16-bit compare register compares the contents of each timer with the data held in the compare register at all times, and generates a match signal when a match is found.

See Table 4-2 for detailed information about the configuration of the timers and compare registers, and the correspondence between the compare registers and interrupt sources.

(5) 16-bit capture registers (CT00 and CT01)

A 16-bit capture register takes in (captures) the contents of timer 0 when a capture trigger signal occurs. As a capture trigger, an external interrupt (INTP0 or INTP1) can be used.

See Table 4-2 for the correspondence between the registers and capture triggers.

The occurrence of a capture trigger also means the occurrence of an interrupt. By using a capture register, the pulse width and period of an externally applied pulses can be easily measured.

4.4 PWM UNIT

The μPD78352A has two PWM signal outputs of 8-bit resolution. By externally connecting a low-pass filter, a PWM output can be used as a digital-to-analog conversion output. The PWM outputs are most suitable, for example, for a control signal for the actuator of a motor.

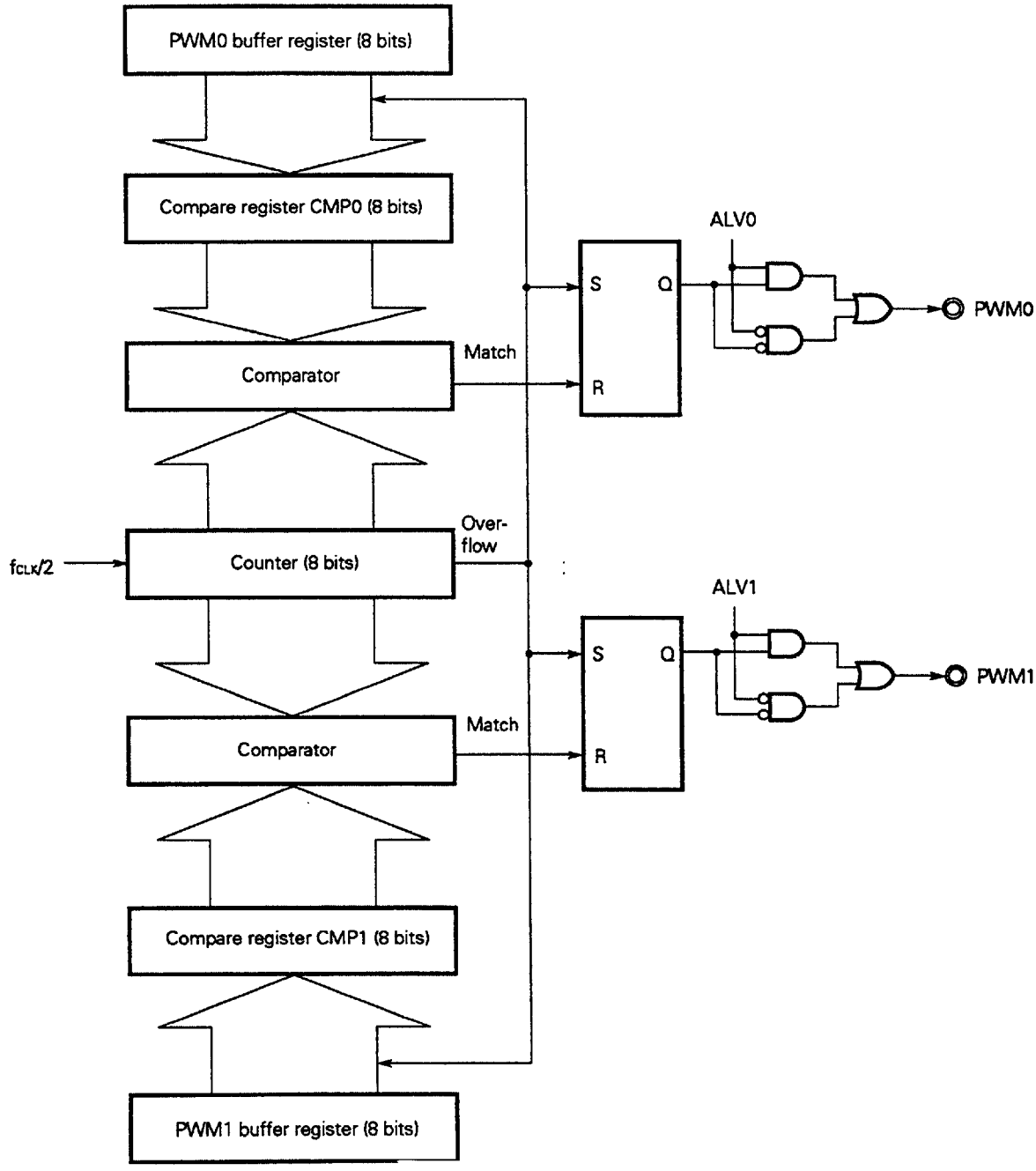
Table 4-3 lists PWM signal output repetition frequencies. Fig. 4-6 shows the configuration of the PWM output function.

Table 4-3 PWM Signal Repetition Frequencies

Resolution per bit	Repetition frequency
$2/f_{CLK}$ (0.125 μs)	$f_{CLK}/2^9$ (31.25 kHz)

Remark The values in parentheses are for f_{CLK} = 16 MHz.

Fig. 4-6 Configuration of the PWM Output Function



4.5 WATCHDOG TIMER (WDT)

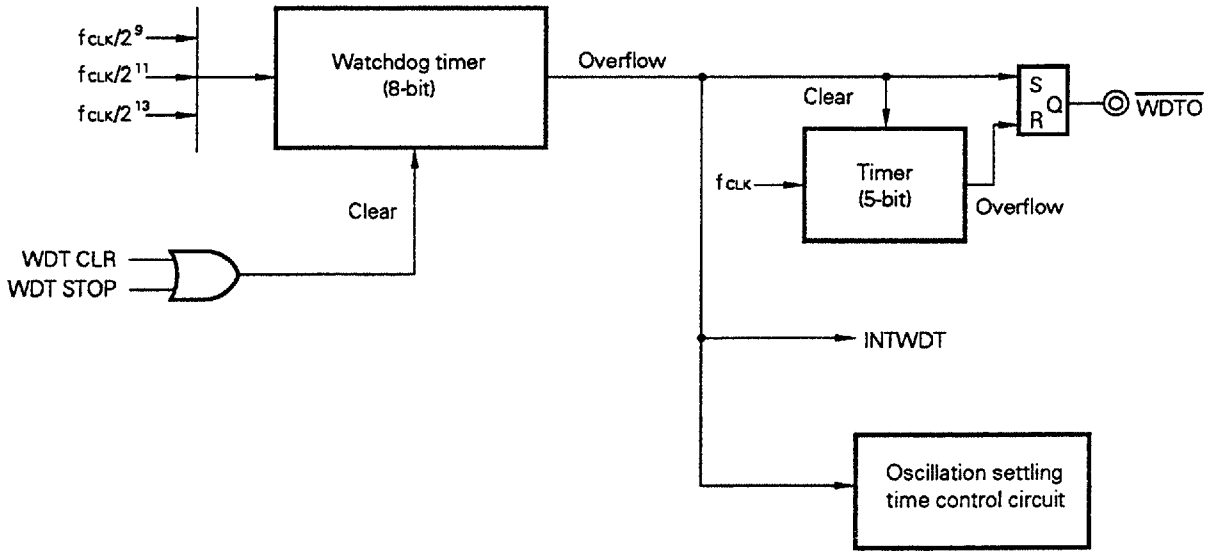
The watchdog timer is a free-running counter with a nonmaskable interrupt function designed to prevent crashes or deadlocks. A program error can be detected when a watchdog timer overflow interrupt (INTWDT) is generated or when the watchdog timer output pin ($\overline{WDTO}$) goes low. By connecting this output to the $\overline{RESET}$ pin, abnormal application system operation caused by a program error can be prevented.

The watchdog timer detects any program error by hardware. So it ensures the detection of crashes and deadlocks for restarting the program. The watchdog timer can also be used to guarantee a time required for the oscillator to perform stable operation when the stop mode is released.

4.5.1 WDT Configuration

Fig. 4-7 shows the configuration of the watchdog timer.

Fig. 4-7 Configuration of the Watchdog Timer



4.5.2 WDT Operation

The watchdog timer generates an interrupt at specified time intervals to detect a program error. So a program should be divided into modules so that the processing of each module can be completed within the WDT interval. Each module should contain an instruction to clear and restart the watchdog timer. For this control, the watchdog timer mode register (WDM) is used.

Once the watchdog timer is started after $\overline{RESET}$ signal input, it cannot be stopped with an instruction. This is intended to prevent a program error from stopping the watchdog timer. Only a $\overline{RESET}$ input signal can stop the watchdog timer. As another means to prevent an error, a special instruction is used to write data into the watchdog timer.

When a WDT overflow occurs, the watchdog timer output pin ($\overline{WDTO}$) allows the low level to be output for the period of 32 f_{CLK}. This pin is externally connected with the $\overline{RESET}$ pin, and is used to reset the system automatically when a program error occurs.

Cautions 1. $\overline{WDTO}$ is designed to output the low level for the period of 32 f_{CLK} even after $\overline{RESET}$ input considering its direct connection to the $\overline{RESET}$ pin.

2. $\overline{WDTO}$ may go low for a maximum of 32 f_{CLK} immediately after power-on.

Remark f_{CLK}: Internal system clock (oscillator frequency/2)

5. INTERRUPT FUNCTION

The μPD78352A has a powerful interrupt function that can handle interrupt requests from the peripheral hardware or other external devices. Three interrupt handling modes are available:

- Vectored interrupt handling
- Macro service
- Context switching

With this interrupt function, complex multitask processing can be efficiently performed at high speed.

Table 5-1 Types of Interrupt Requests and Handling Modes

Handling mode / Interrupt request	Vectored interrupt handling	Macro service	Context switching
Nonmaskable interrupt	○	–	–
Maskable interrupt	○	○	○
Software interrupt	○	–	○
Exception trap	○	–	–

5.1 TYPES OF INTERRUPT REQUESTS

With the μPD78352A, four types of interrupt requests are used:

- Nonmaskable interrupt
- Maskable interrupt
- Software interrupt
- Exception trap

Each type of interrupt request is explained below.

(1) Nonmaskable interrupt

The nonmaskable interrupt is a type of interrupt whose acceptance cannot be disabled with an instruction. A nonmaskable interrupt can be accepted at all times. Nonmaskable interrupt requests are classified into the following two types:

- NMI pin input (NMI)
- Watchdog timer output (WDT)

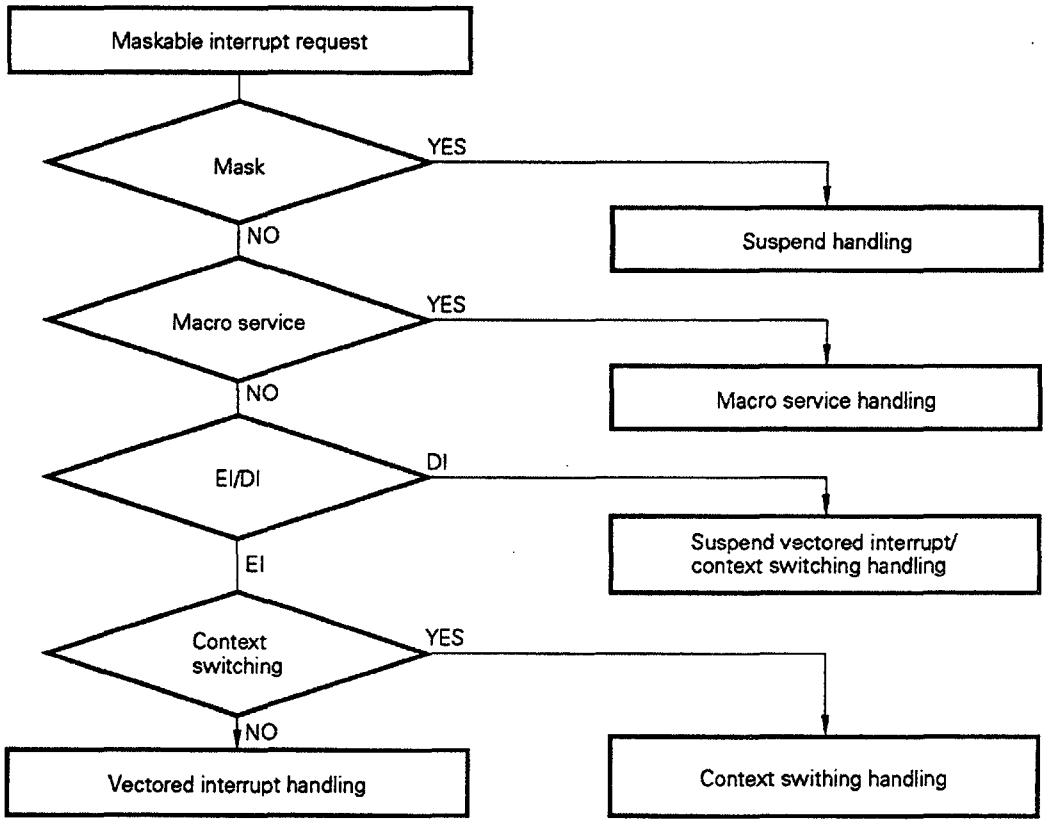
For a nonmaskable interrupt, vectored interrupt handling can be performed.

(2) Maskable interrupt

The maskable interrupt is a type of interrupt whose acceptance can be masked with a control register. Seven interrupt sources are available. For a maskable interrupt, one of the following three handling modes can be selected:

- Vectored interrupt handling
- Macro service
- Context switching

Fig. 5-1 Maskable Interrupt Handling



If multiple maskable interrupts occur at the same time, their priorities are determined according to the default priorities. Besides the default priorities, four priority levels can be set by software.

(3) Software interrupt

The software interrupt request is an interrupt request made by executing a CPU break instruction, and can be accepted at all times. For a software interrupt, vectored interrupt handling is performed. The following two instructions can generate a software interrupt:

- BRK : Causes a branch to the address indicated by the contents of memory addresses 003EH and 003FH.
- BRKCS: Causes a branch by context switching processing for switching to the register bank specified in the instruction.

(4) Exception trap

For an exception trap, vectored interrupt handling can be performed. An exception trap occurs in the following case:

- Invalid op code (TRAP): Occurs when an instruction for writing to the standby control register and watchdog timer mode register is not executed normally.

5.2 INTERRUPT HANDLING MODES

With the μPD78352A, three interrupt handling modes are available:

- Vectored interrupt handling
- Macro service
- Context switching

(1) Vectored interrupt handling

When an interrupt is accepted, the contents of PC and PSW are saved automatically. Then a branch is made to the address indicated by the data contained in the vector address table to execute the interrupt service routine.

(2) Macro service

When an interrupt is accepted, CPU execution is terminated temporarily to execute the service set by firmware. The macro service is performed without CPU involvement, so that the CPU statuses such as PC and PSW need not be saved or restored. Thus the macro service much increases CPU service time.

(3) Context switching

When an interrupt is accepted, a specified register bank is selected by hardware. Then a branch is made to the already selected vector address in the register bank, and the current contents of PC and PSW are saved in the register bank at the same time.

Remark The context means CPU registers that can be accessed from a program being executed. The registers include general registers, PC, PSW, and SP.

Table 5-2 lists the interrupt sources.

Table 5-2 Interrupt Source List

Type	Note	Interrupt source		Unit requesting interrupt	Vector table address	Macro service	Context switch
		Name	Trigger				
Non-maskable	–	NMI	NMI pin input	External	0002H	No	No
	–	WDT	Watchdog timer	WDT	0004H		
Maskable	0	INTOV	Timer 0 overflow	Capture/timer unit	0006H	Yes	Yes
	1	INTP0	INTP0 pin input	External	0008H		
	2	INTP1	INTP1 pin input	External	000AH		
	3	INTCM10	CM10 match signal	Capture/timer unit	000CH		
	4	INTCM20	CM20 match signal	Capture/timer unit	000EH		
	5	INTP2	INTP2 pin input	External	0010H		
	6	INTP3	INTP3 pin input	External	0012H		
Software	–	BRK	BRK instruction	–	003EH	No	No
	–	BRKCS	BRKCS instruction	–	–		Yes
Exception	–	TRAP	Invalid op code trap	–	003CH		No
Reset	–	RESET	RESET input	–	0000H		

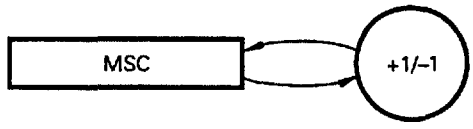
Note Default priority: Priority used when multiple maskable interrupts occur at the same time, with 0 for the highest priority and 6 for the lowest priority

5.3 MACRO SERVICE

The μD78352A has five types of macro services. Each macro service is explained below.

(1) Counter mode: EVTCNT

- **Operation**
  - (a) This mode increments or decrements the 8-bit macro service counter (MSC).
  - (b) When the MSC reaches 0, a vectored interrupt request occurs.

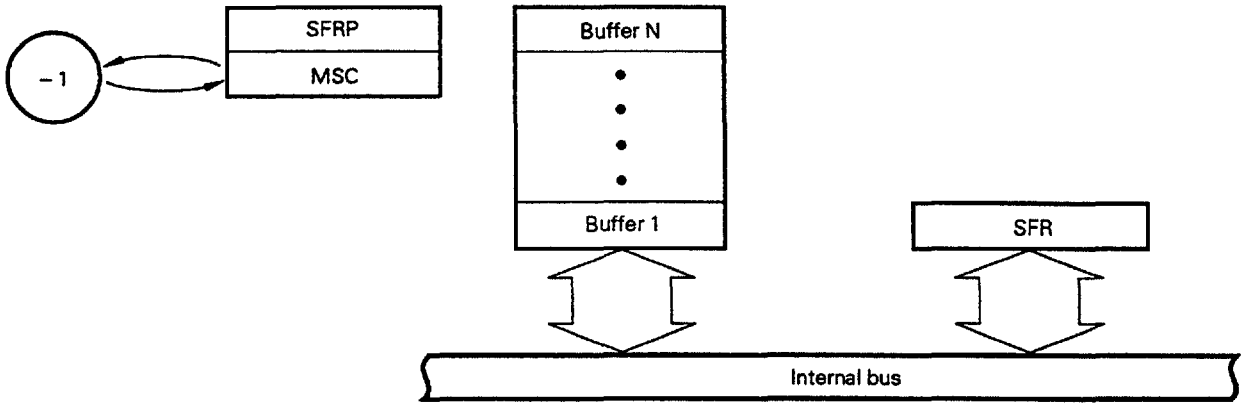


- **Sample application**

  This mode can be used as the event counter or capture counter.

(2) Block transfer mode: BLKTRS

- **Operation**
  - (a) This mode transfers a data block between the buffer and the SFR pointed to by the SFR pointer (SFRP).
  - (b) Either an SFR or buffer area can be specified as a transfer source or transfer destination. In addition, either the byte or word can be selected as the length of transfer data.
  - (c) The MSC is used to specify the number of data transfers (block size).
  - (d) Each time the macro service is executed, the MSC is automatically decremented by one.
  - (e) When the MSC reaches 0, vectored interrupt handling is activated.



- **Sample application**

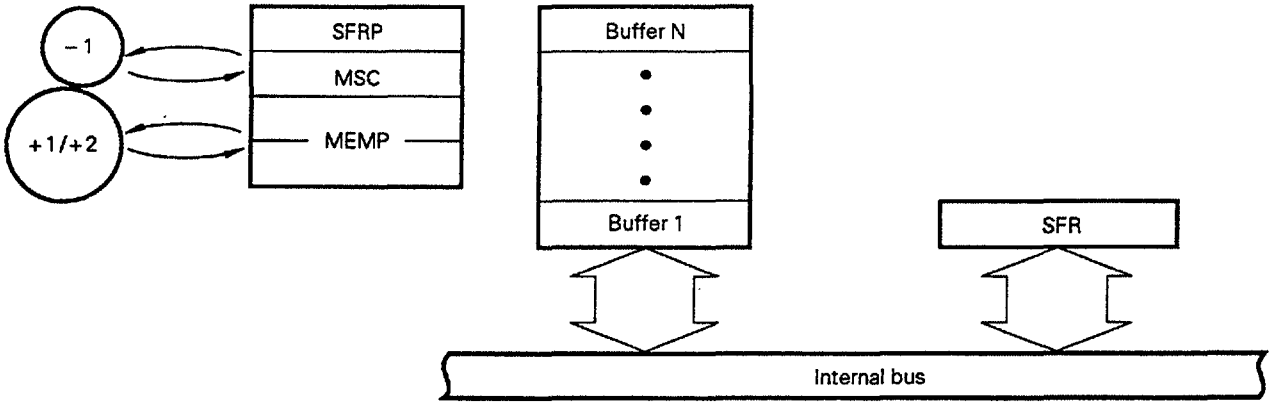
  This mode can be used to read port data in response to an external interrupt request.

(3) Block transfer mode (with memory pointer): BLKTRS-P

- **Operation**

  This mode is the block transfer mode with a memory pointer (MEMP) added. The additional buffer area for MEMP can be freely set in memory space.

  Remark Each time the macro service is executed, the MEMP is automatically incremented (by one for a byte-data transfer or by two for a word-data transfer).



- **Sample application**

  Same as (2) above

(4) Data difference mode: DTADIF

- **Operation**

  (a) This mode calculates the difference between the contents (current value) of the SFR pointed to by the SFRP and the contents of the SFR already held in the last data buffer (LDB).

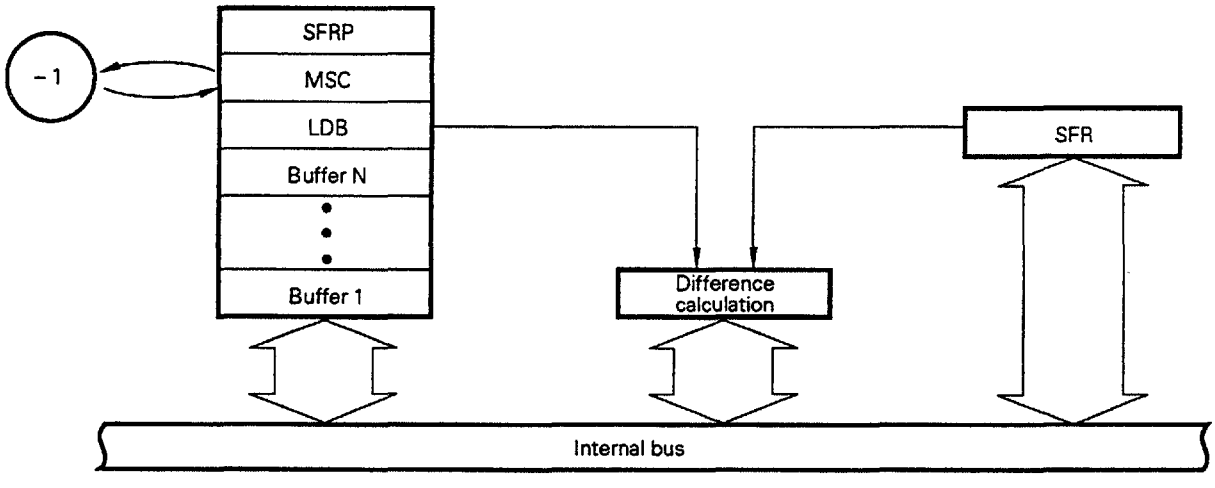
  (b) The result of calculation is stored in a buffer area specified beforehand.

  (c) The current value of the SFR is loaded into the LDB.

  (d) The MSC is used to specify the number of data transfers (block size). Each time the macro service is executed, the MSC is automatically decremented by one.

  (e) When the MSC reaches 0, vectored interrupt handling is activated.

  Remark The difference can be calculated only for a 16-bit SFR.



- **Sample application**

  This mode can be used to measure periods or pulse widths of the capture/timer unit.

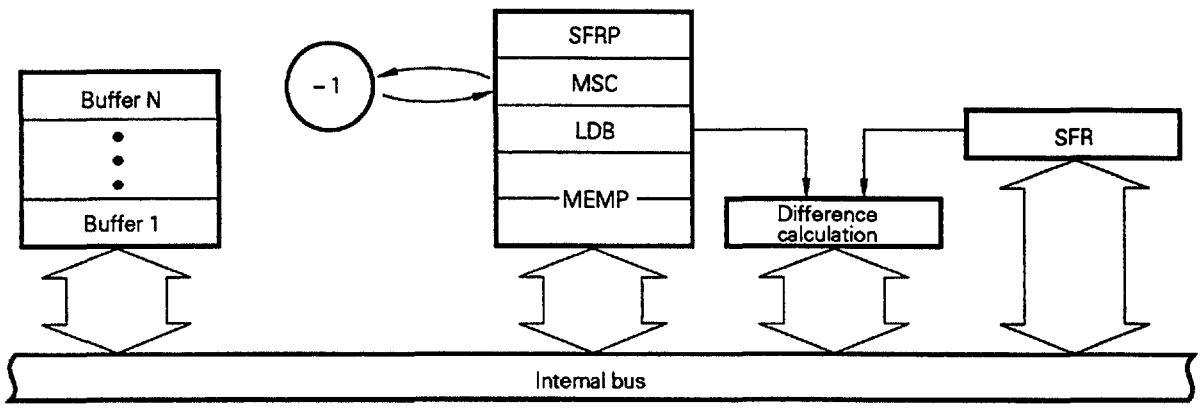
(5) Data difference mode (with memory pointer): DTADIF-P

- **Operation**

  This mode is the data difference mode with a memory pointer (MEMP) added. With this MEMP addition, a buffer area for storing difference data can be freely set in memory space.

  Remark A buffer is specified by the result of operation on the MEMP and MSC[Note]. The MEMP is not updated after data transfer.

  Note MEMP - (MSC × 2) + 2



- **Sample application**

  Same as (4) above

5.4 CONTEXT SWITCHING

The context switching is a function that selects a specified register bank by hardware when an interrupt occurs or a BRKCS instruction is executed, then causes a branch to the vector address set beforehand in the register bank and saves the current contents of PC and PSW in the register bank at the same time.

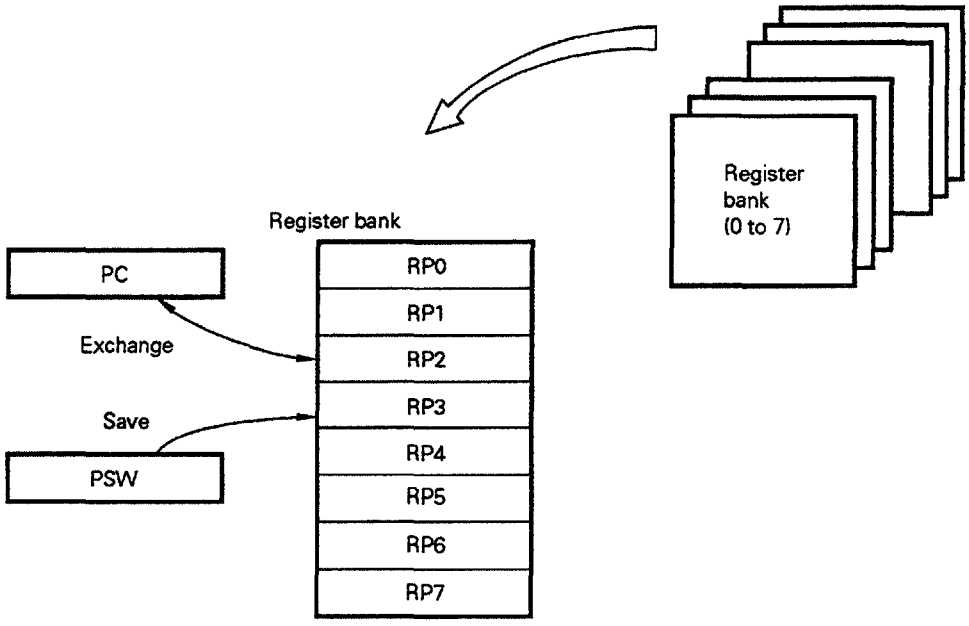
5.4.1 Context Switching Function Based on an Interrupt Request

The context switching function can be activated when the context switching enable register corresponding to each maskable interrupt request is set to 1 in the EI (interrupt enable) state.

Context switching operation based on an interrupt request is performed as described below.

(1) When an interrupt request occurs, a register bank subject to context switching is specified from the contents of the lower three bits of the row address (even address) of the corresponding vector table.
(2) The vector address set beforehand in the register bank subject to context switching is transferred to PC, and the contents of PC and PSW present immediately before switching operation are saved in the register bank.
(3) A branch is made to the address pointed to by the newly set contents of PC.

Fig. 5-2 Context Switching Operation



5.4.2 Context Switching Function Based on the BRKCS Instruction

The context switching function can be activated with the BRKCS instruction.

Context switching operation based on an interrupt request is performed as described below.

(1) An 8-bit register is specified in an operand of the BRKCS instruction. The contents of the register determine a register bank subject to context switching. (Only the low-order three bits of the eight bits are used.)
(2) The vector address set beforehand in the register bank subject to context switching is transferred to PC, and the contents of PC and PSW present immediately before switching operation are saved in the register bank at the same time.
(3) A branch is made to the address pointed to by the newly set contents of PC.

5.4.3 Return from Context Switching

To return from context switching, one of the following two instructions is used. The source of context switching activation determines which instruction to use.

Table 5-3 Return from Context Switching

Return instruction	Context switching activation source
RETCS	Interrupt occurrence
RETCSB	BRKCS instruction

6. EXTERNAL DEVICE EXPANSION FUNCTION

Besides its internal ROM and RAM areas, the μPD78352A can have external devices (data memory, program memory, and peripheral devices) expanded. When an external device is connected, port 4, 5, or 9 is used for address/data, $\overline{RD}$, and $\overline{WR}$ control.

Table 6-1 Pin Functions Assigned When External Devices Are Connected

Pin	Pin function when external device is connected	
	Function	Name
P40 - P47	Multiplexed address/data bus	AD0 - AD7
P50 - P57	Address bus	A8 - A15
P90	Read strobe	$\overline{RD}$
P91	Write strobe	$\overline{WR}$
CLKOUT	System clock output	CLKOUT
ASTB	Address strobe	ASTB

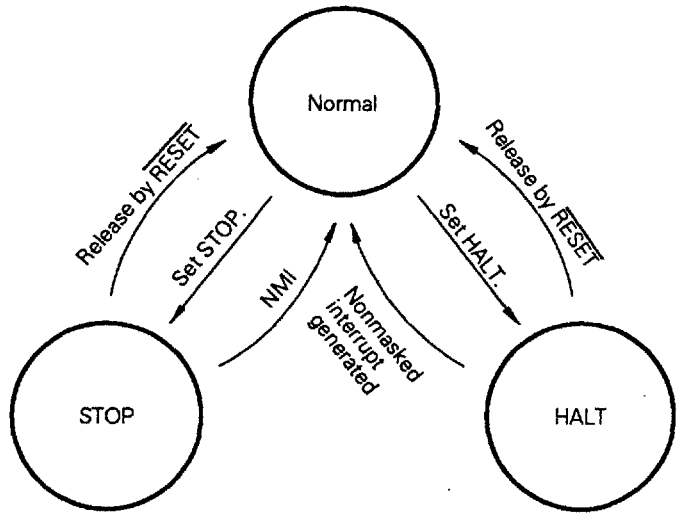
7. STANDBY FUNCTION

The μPD78352A has a standby function to reduce power consumption of the system. With the standby function, two modes are available:

- HALT mode: In this mode, the CPU operation clock is stopped. Intermittent operation, when combined with the normal operation mode, can reduce overall system power consumption.
- STOP mode: In this mode, the oscillator is stopped to stop the entire system.
  Since only leakage currents may flow in this mode, system power consumption can be minimized.

Each mode is set by software. Fig. 7-1 is the transition diagram of the standby modes (STOP and HALT modes).

Fig. 7-1 Transition Diagram of the Standby Modes



8. RESET FUNCTION

When the signal applied to the $\overline{\text{RESET}}$ input pin is low, the system is reset, and each hardware component is placed in the status indicated in Table 8-1. When the signal applied to the $\overline{\text{RESET}}$ input port goes high, the reset status is released, and program execution starts. The contents of registers must be initialized in the program as required.

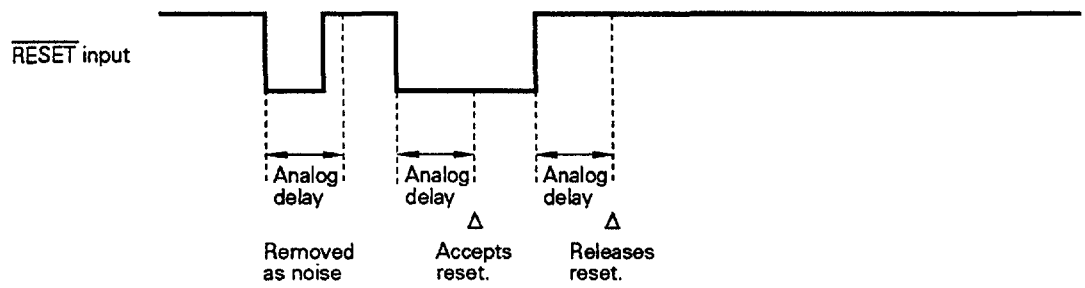
In particular, the number of cycles specified in the programmable wait control register must be changed as required.

The $\overline{\text{RESET}}$ input pin contains a noise eliminator based on analog delays to prevent abnormal operation due to noise.

Cautions 1. When $\overline{\text{RESET}}$ is active (low level), all pins except $\overline{\text{WDTO}}$, CLKOUT, V_{DD}, V_{SS}, X1, and X2 go into the high-impedance state.

2. When RAM is expanded externally, connect pull-up resistors to the P90/$\overline{\text{RD}}$ pin and P91/$\overline{\text{WR}}$ pin. If these pins go into the high-impedance state, the contents of the external RAM may be lost. Signals may collide with each other at the address/data bus, resulting in damage to the input/output circuits.

Fig. 8-1 Acceptance of the $\overline{\text{RESET}}$ Signal



A time to settle oscillation is required from when the microcomputer is turned on to when it receives a power-on reset signal as shown in Fig. 8-2.

Fig. 8-2 Reset Operation at Power-On

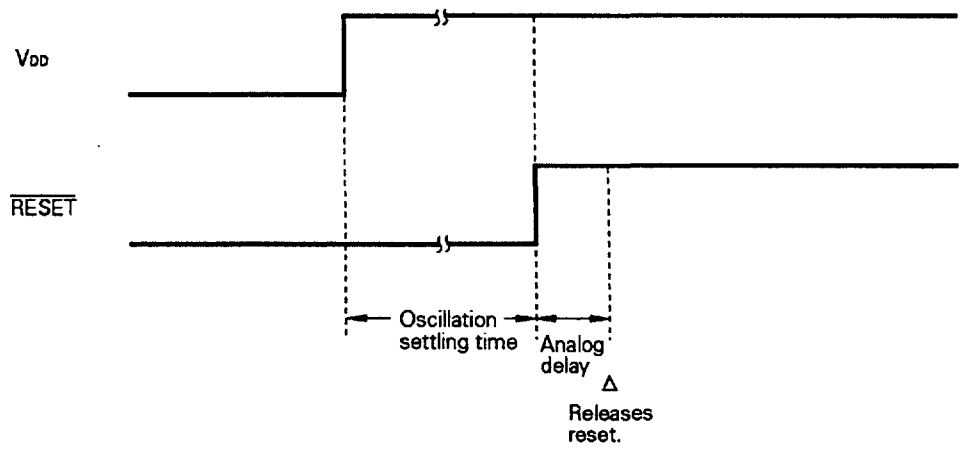


Table 8-1 Hardware Statuses after Reset

Hardware			Status after reset
Control registers	Program counter (PC)		The contents of a reset vector table (0000H, 0001H) are set.
	Stack pointer (SP)		Undefined[Note]
	Program status word (PSW)		0000H
	CPU control word (CCW)		00H
Internal RAM	Data memory		Undefined[Note]
	General registers (R0-R15)		
Ports	Output latches (P0, P1, P3-P5, P9)		Undefined
	Mode registers	PM0, PM1, PM3, PM5	FFH
		PM9	×FH
	Mode control register (PMC3)		00H
	Port read control register (PRDC)		00H
Capture/timer unit	Timers (TM0, TM1, TM2)		0000H
	Timer control registers (TMC0, TMC1)		00H
	Capture registers (CT00, CT01)		Undefined
	Compare registers (CM10, CM20)		Undefined
PWM output function	PWM control register (PWMC)		00H
	PWM buffer registers (PWM0, PWM1)		Undefined
External expansion function	Memory expansion mode register (MM)		00H
	Programmable wait control register (PWC)		C0AAH
Watchdog timer	Watchdog timer mode register (WDM)		00H
Interrupt function	External interrupt mode registers (INTM0, INTM1)		00H
	Interrupt mode control register (IMC)		80H
	Interrupt mask flag registers	MKL	7FH
		MK	××7FH
	Interrupt control registers (OVIC, PIC0, PIC1, CMIC10, CMIC20, PIC2, PIC3)		43H
	In-service priority register (ISPR)		00H
CPU control	Standby control register (STBC)		0000 ×000B

Note When the stop mode is released by a $\overline{\text{RESET}}$ input, the values which had been held immediately before the stop mode was set are restored.

9. INSTRUCTION SET

(1) Operand identifier and description

Operands are coded in the operand field of each instruction as listed in the description 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 9-1 Operand Identifier and Description

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 2-1.**) Special function register abbreviation (16-bit manipulation register. See **Table 2-1.**)
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 must be 0) 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. Only data FE00H to FEFFH can be coded in an MOVTBLW instruction.) 800H-FFFH Immediate data or label 40H-7EH Immediate data (bit 0 must be 0)[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) can be specified in r, r1, rp, rp1, and post, as well as absolute names (R0 to R15 and RP0 to RP7).

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

(2) Legend

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
R0-R15	:	Register 0 to register 15 (absolute name)
AX	:	Register pair (AX); 16 bit accumulator
BC	:	Register pair (BC)
DE	:	Register pair (DE)
HL	:	Register pair (HL)
RP0-RP7	:	Register pair 0 to register pair 7 (absolute name)
PC	:	Program counter
SP	:	Stack pointer
UP	:	User stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
P/V	:	Parity/overflow flag
S	:	Sign flag
TPF	:	Table position flag
RBS	:	Register bank selecting flag
RSS	:	Register set selecting flag
IE	:	Interrupt enable flag
STBC	:	Standby control register
WDM	:	Watchdog timer mode register
jdisp8	:	Signed 8-bit data (displacement value: -128 to +127)
( )	:	Contents of memory at an address enclosed in parentheses or at the address specified by a register enclosed in parentheses. (+) and (-) indicate that an address or the contents of a register enclosed in parentheses are incremented and decremented by one after execution of the instruction, respectively.
(( ))	:	Contents of memory at the address specified by the contents of memory at an address enclosed in parentheses (( )).
xxH	:	Hexadecimal number
xH, xL	:	High-order 8 bits and low-order 8 bits of 16-bit register

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 The table below shows the symbols used in the flag column and their meanings.

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←AX×r1					
	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					
Sum-of-products	MACW	n	3	AXDE←(B)×(C)+AXDE B←B+2, C←C+2, n←n-1 End if n=0 or P/V=1	×	×	×	V	×
Table shift	MOVTBLW	!addr16,n	4	(addr16+2)←(addr16), n←n-1 addr16←addr16-2, End if n=0					

Remark The addressing range of the table shift instruction is FE00H to FEFFH.

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					

Remark n in the shift/rotate instructions indicates the number of shifts or rotations.

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)←$\overline{(saddr.bit)}$					
		sfr.bit	3	sfr.bit←$\overline{sfr.bit}$					
		A.bit	2	A.bit←$\overline{A.bit}$					
		X.bit	2	X.bit←$\overline{X.bit}$					
		PSWH.bit	2	PSW$_H$.bit←$\overline{PSW_H.bit}$					
		PSWL.bit	2	PSW$_L$.bit←$\overline{PSW_L.bit}$	×	×	×	×	×
	SET1	CY	1	CY←1					1
	CLR1	CY	1	CY←0					0
	NOT1	CY	1	CY←$\overline{CY}$					×

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					
		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					
	POP	sfrp	3	sfrL←(SP), sfrH←(SP+1), SP←SP+2					
		post	2	{postL←(SP), postH←(SP+1), SP←SP+2} × n times					
		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					
	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					

Remark n in the stack manipulation instructions indicates the number of registers specified in post.

Instruction set	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Special	CHKL	sfr	3	(Pin level)∀(Signal level before output buffer)	×	×		P	
	CHKLA	sfr	3	A←{(Pin level)∀(Signal level before output buffer)}	×	×		P	
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←1 (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:

$(SP-1) \leftarrow PSW_H$, $(SP-2) \leftarrow PSW_L$,

$(SP-3) \leftarrow (PC-4)_H$, $(SP-4) \leftarrow (PC-4)_L$,

$PC_L \leftarrow (003CH)$, $PC_H \leftarrow (003DH)$,

$SP \leftarrow SP-4$, $IE \leftarrow 0$

10. ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (T_a = 25 °C)

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{DD}		−0.5 to +7.0	V
Input voltage	V_I		−0.5 to V_{DD} + 0.5	V
Output voltage	V_O		−0.5 to V_{DD} + 0.5	V
Low-level output current	I_{OL}	Each pin	4.0	mA
		Total of all output pins	100	mA
High-level output current	I_{OH}	Each pin	−1.0	mA
		Total of all output pins	−20	mA
Operating temperature	T_{opt}		−10 to +70	°C
Storage temperature	T_{stg}		−65 to +150	°C

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.

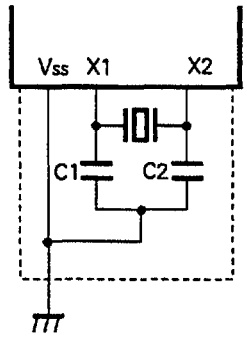
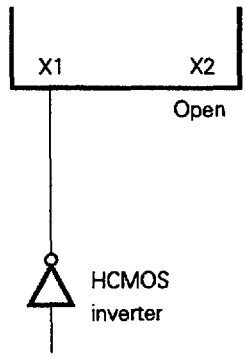
RECOMMENDED OPERATING CONDITIONS

Oscillation frequency	T_a	V_{DD}
8 MHz ≤ fxx ≤ 32MHz	−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 Pins other than those measured must be 0 V.			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	32	MHz
External clock		X1 input frequency (f_X)	8	32	MHz
		X1 input rising and falling times (t_{XR}, t_{XF})	0	10	ns
		X1 input high-level and low-level widths (t_{WXH}, t_{WXL})	10	115	ns

Caution When using the system clock generator, run wires in the portion surrounded by dotted 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 V_{SS}. Never connect the capacitor to a ground pattern carrying a large current.**
- **Never extract a signal from the oscillator.**

Recommended Capacitors in an Oscillation Circuit

Ceramic resonator

Manufacturer	Part number	Frequency [MHz]	Recommended constant	
			C1 [pF]	C2 [pF]
Murata Mfg.	CSA16.00MXZ040	16.0	22	22
	CSA25.00MXZ040	25.0	5	5
	CSA32.00MXZ040	32.0	Open	Open

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			60	87	mA
	I_{DD2}	HALT mode			20	30	mA
Data retention voltage	V_{DDDR}	STOP mode		2.5			V
Data retention current	I_{DDDR}	STOP mode	V_{DDDR} = 2.5 V		2	10	μA
			V_{DDDR} = 5.0 V ±10 %		10	50	μA

Notes 1. For pins other than those described in Note 2

2. $\overline{\text{RESET}}$, X1, X2, P20/NMI, P21/INTP0, P22/INTP1, P23/INTP2, P24/INTP3, and P25/TI

AC CHARACTERISTICS (T_a = -10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V, C_L = 100 pF, f_{XX} = 32 MHz)

Read/Write Operation (When the General Memory Is Connected)

Parameter	Symbol	Conditions	Min.	Max.	Unit
System clock cycle time	t_{CYK}	C_L = 50 pF	62.5	250	ns
Address setup time (to ASTB ↓)	t_{SAST}		7		ns
Address hold time (to ASTB ↓)	t_{HSTA}		11		ns
Delay from $\overline{RD}$ ↓ to address float	t_{FRA}			0	ns
Delay from address to data input	t_{DAID}			100	ns
Delay from $\overline{RD}$ ↓ to data input	t_{DRID}			49	ns
Delay from ASTB ↓ to $\overline{RD}$ ↓	t_{DSTR}		15		ns
Data hold time (to $\overline{RD}$ ↑)	t_{HRID}		0		ns
Delay from $\overline{RD}$ ↑ to address active	t_{DRA}		25		ns
$\overline{RD}$ low-level width	t_{WRL}		63		ns
ASTB high-level width	t_{WSTH}		14		ns
Delay from $\overline{WR}$ ↓ to data output	t_{DWOD}			21	ns
Delay from ASTB ↓ to $\overline{WR}$ ↓	t_{DSTW}		15		ns
Delay from $\overline{WR}$ ↑ to ASTB ↑	t_{DWST}		78		ns
Data setup time (to $\overline{WR}$ ↑)	t_{SODW}		57		ns
Data hold time (to $\overline{WR}$ ↑)	t_{HWOD}		8		ns
$\overline{WR}$ low-level width	t_{WWL}		57		ns
$\overline{WAIT}$ setup time (to address)	t_{SAWT}			44	ns
$\overline{WAIT}$ hold time (to address)	t_{HAWT}		86		ns
$\overline{WAIT}$ setup time (to $\overline{RD}$ ↓)	t_{SRRY}			-25	ns
$\overline{WAIT}$ setup time (to $\overline{WR}$ ↓)	t_{SWRY}			-25	ns
$\overline{WAIT}$ hold time (to $\overline{RD}$ ↓)	t_{HRRY}		17		ns
$\overline{WAIT}$ hold time (to $\overline{WR}$ ↓)	t_{HWRY}		17		ns
Delay from address to $\overline{RD}$ ↓	t_{DAR}			69	ns
Delay from address to $\overline{WR}$ ↓	t_{DAW}			69	ns

t_{CYK}-Dependent Bus Timing Definition

Symbol	Calculation formula	Min./Max.	Unit
t_{SAST}	(0.5 + a)T – 24	Min.	ns
t_{HSTA}	0.5T – 20	Min.	ns
t_{WSTH}	(0.5 + a)T – 17	Min.	ns
t_{DSTR}	0.5T – 16	Min.	ns
t_{WRL}	(1.5 + n)T – 30	Min.	ns
t_{DAID}	(2.5 + a + n)T – 56	Max.	ns
t_{DRID}	(1.5 + n)T – 44	Max.	ns
t_{DRA}	0.5T – 6	Min.	ns
t_{DSTW}	0.5T – 16	Min.	ns
t_{DWST}	1.5T – 15	Min.	ns
t_{WWL}	(1.5 + n)T – 36	Min.	ns
t_{DWOD}	0.5T – 10	Max.	ns
t_{SODW}	(1 + n)T – 5	Min.	ns
t_{SAWT}	(a + n)T – 18	Max.	ns
t_{HAWT}	(0.5 + a + n)T – 7	Min.	ns
t_{SRRY}	(n – 1)T – 25	Max.	ns
t_{SWRY}	(n – 1)T – 25	Max.	ns
t_{HRRY}	(n – 0.5)T – 14	Min.	ns
t_{HWRY}	(n – 0.5)T – 14	Min.	ns
t_{DAR}	(a + 1)T + 7	Max.	ns
t_{DAW}	(a + 1)T + 7	Max.	ns

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

2. When an address wait is inserted, the value of a is 1. Otherwise, it is 0.
3. The number n represents the number of wait cycles specified by the external wait pin ($\overline{WAIT}$) or PWC register.
4. Only the bus timing items listed above are dependent on t_{CYK}.

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}		2		μs
INTP0 high/low level width	t_{WI0H}, t_{WI0L}		500		ns
INTP1 high/low level width	t_{WI1H}, t_{WI1L}		500		ns
INTP2 high/low level width	t_{WI2H}, t_{WI2L}		500		ns
INTP3 high/low level width	t_{WI3H}, t_{WI3L}		500		ns
$\overline{\text{RESET}}$ high/low level width	t_{WRSH}, t_{WRSL}		2		μs
TI high/low level width	t_{WTIH}, t_{WTIL}		500		ns

Other t_{CYK}-Dependent Operations

Symbol	Calculation formula	Min./Max.	Unit
t_{WI0H}	8T	Min.	ns
t_{WI0L}	8T	Min.	ns
t_{WI1H}	8T	Min.	ns
t_{WI1L}	8T	Min.	ns
t_{WI2H}	8T	Min.	ns
t_{WI2L}	8T	Min.	ns
t_{WI3H}	8T	Min.	ns
t_{WI3L}	8T	Min.	ns
t_{WTIH}	8T	Min.	ns
t_{WTIL}	8T	Min.	ns

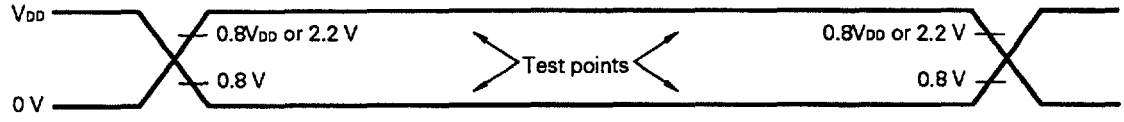
Remarks 1. T = t_{CYK} = 1/f_{CLK} (f_{CLK} is the internal system clock frequency.)

2. The bus timing items listed above are dependent on t_{CYK}.

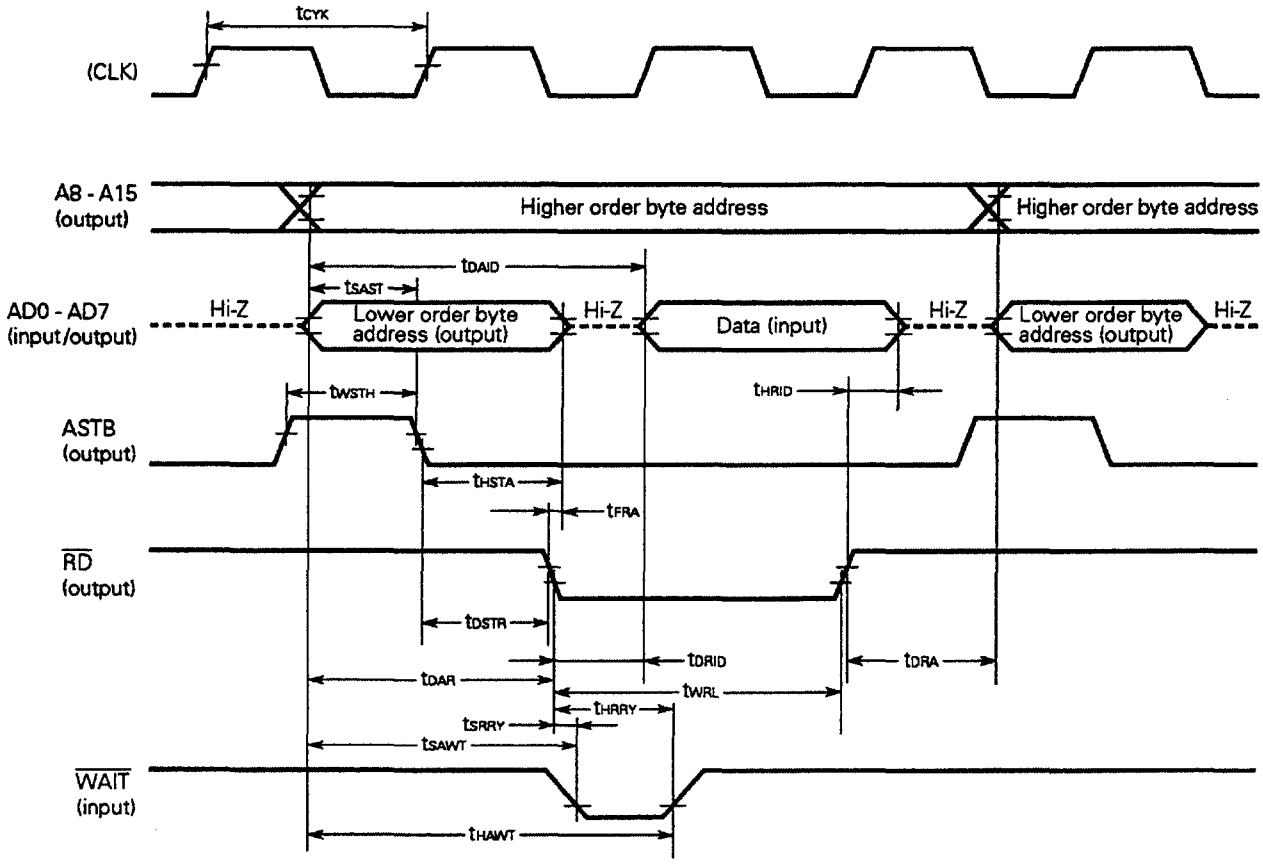
External Clock Timing (T_a = −10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min.	Max.	Unit
X1 input high/low level width	t_{WXH}, t_{WXL}		10	115	ns
X1 input rising/falling time	t_{XR}, t_{XF}		0	10	ns
X1 input clock cycle time	t_{CYX}		31.25	125	ns

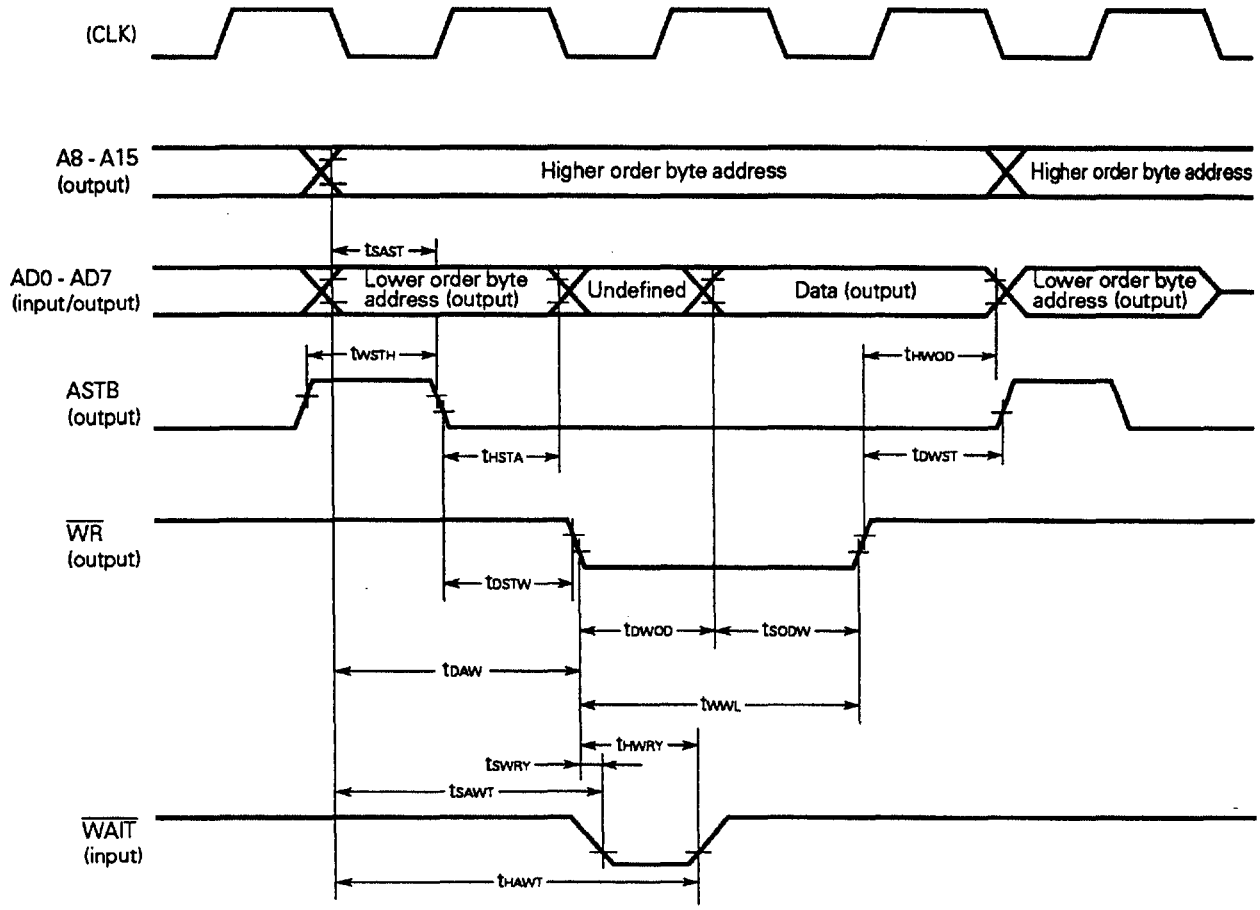
AC Timing Test Points



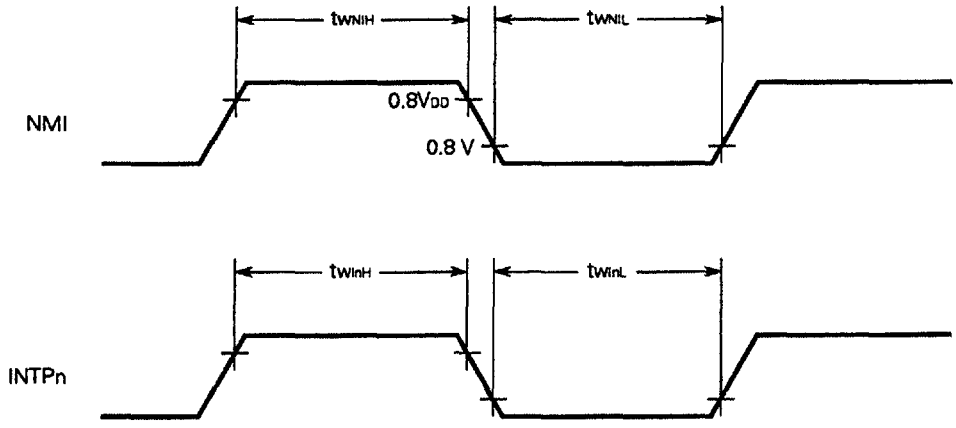
Read Operation



Write Operation

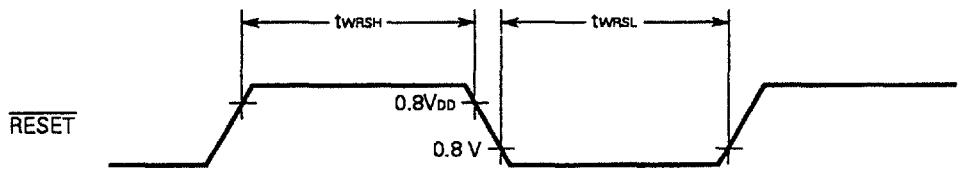


Interrupt Input Timing

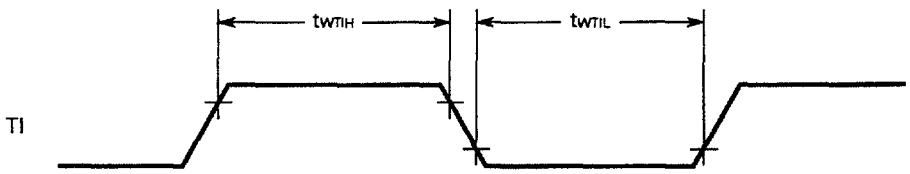


Remark n = 0 to 3

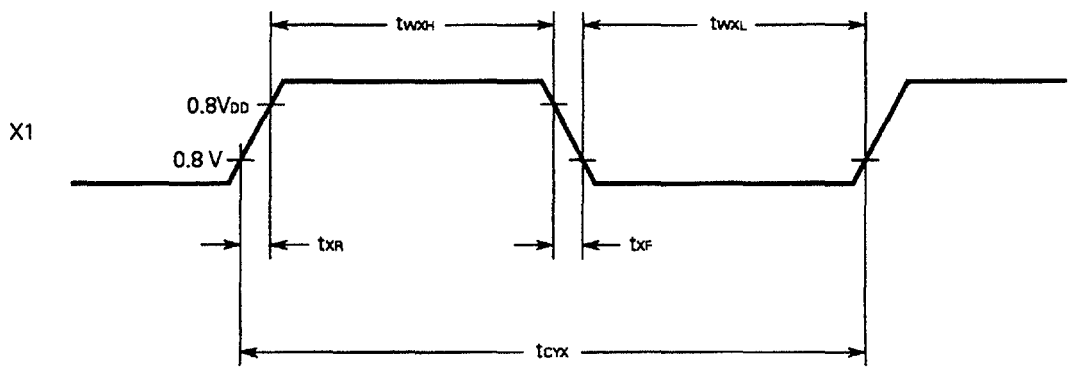
Reset Input Timing



TI Pin Input Timing

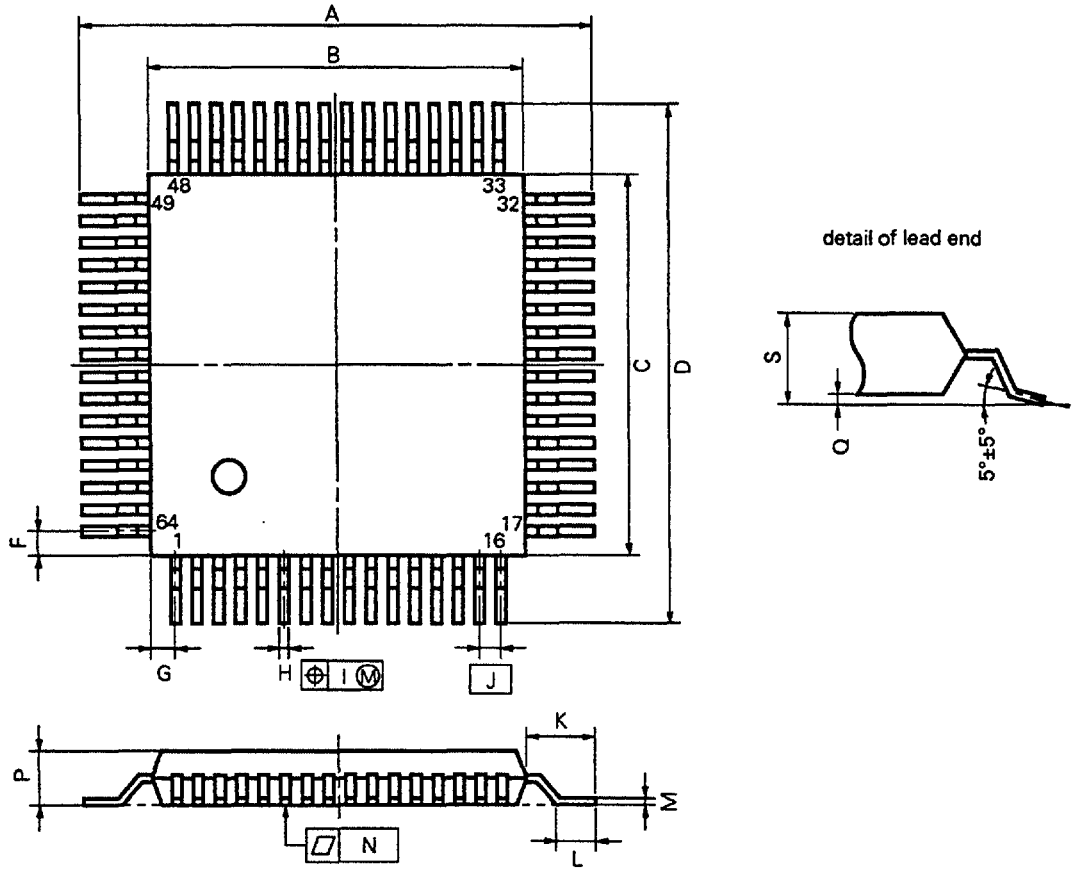


External Clock Timing



11. PACKAGE DRAWINGS

64 PIN PLASTIC QFP (□14)



P64G-80-22-2

NOTE

Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	18.4±0.4	$0.724^{+0.017}_{-0.016}$
B	14.0±0.2	$0.551^{+0.009}_{-0.008}$
C	14.0±0.2	$0.551^{+0.009}_{-0.008}$
D	18.4±0.4	$0.724^{+0.017}_{-0.016}$
F	1.0	0.039
G	1.0	0.039
H	0.35±0.10	$0.014^{+0.004}_{-0.005}$
I	0.15	0.006
J	0.8 (T.P.)	0.031 (T.P.)
K	2.2±0.2	$0.087^{+0.008}_{-0.009}$
L	1.0±0.2	$0.039^{+0.009}_{-0.008}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
N	0.15	0.006
P	1.5±0.1	0.059±0.004
Q	0.1±0.1	0.004±0.004
S	1.7 MAX.	0.067 MAX.

12. RECOMMENDED SOLDERING CONDITIONS

The conditions listed below shall be met when soldering the μPD78352A.

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

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

Table 12-1 Soldering Conditions for Surface-Mount Devices

μPD78350AG-22 : 64-pin plastic QF (14 × 14 mm)
μPD78352AG-×××-22 : 64-pin plastic QF (14 × 14 mm)

Soldering process	Soldering conditions	Symbol
Infrared ray reflow	Peak package's surface temperature: 235 °C Reflow time: 30 seconds or less (at 210 °C or more) Maximum allowable number of reflow processes: 2 Exposure limit Note: 7 days (10 hours of pre-baking is required at 125 °C afterward.) <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.	IR35-107-2
VPS	Peak package's surface temperature: 215 °C Reflow time: 40 seconds or less (at 200 °C or more) Maximum allowable number of reflow processes: 2 Exposure limit Note: 7 days (10 hours of pre-baking is required at 125 °C afterward.) <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.	VP15-107-2
Partial heating method	Terminal temperature: 300 °C or less Flow time: 3 seconds or less (for each side of device)	–

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

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

APPENDIX A DIFFERENCES BETWEEN THE μPD78352A, μPD78350A AND THE μPD78322, μPD78320

Item \ Product		μPD78352A	μPD78350A	μPD78322	μPD78320
Number of basic instructions		113		111	
Minimum instruction execution time		125 ns (in operation at an internal clock of 16 MHz or an external clock of 32 MHz)		250 ns (in operation at an internal clock of 8 MHz or an external clock of 16 MHz)	
Internal memory	ROM	32768 × 8 bits	—	16384 × 8 bits	—
	RAM	640 × 8 bits			
Memory space		64K bytes			
I/O line	Input	6		16 (analog input: 8)	
	I/O	44	24	39	21
Real-time pulse unit (Capture/timer unit)		• One 16-bit free running timer • One 16-bit timer/event counter • One 16-bit interval timer • Two 16-bit capture registers • Two 16-bit compare registers		• One 18/16-bit free running timer • One 16-bit timer/event counter • Six 16-bit compare registers • Four 18-bit capture registers • Two 18-bit capture/compare registers • Eight real-time output ports	
Serial interface		—		• Serial interface with dedicated band rate generator One-channel UART One-channel clock synchronous serial interface/SBI	
A/D converter		—		10-bit resolution, 8 inputs	
Interrupt		• Five external and four internal sources • 4-level programmable priority		• Eight external and 14 internal sources (Two pins are used for both sources.) • 3-level programmable priority	
		• Three processing modes: Vectored interrupt function, macro service function, and context switching function			
Test source		—		One internal test source	
Instruction set		Sum-of-products instruction is added to those used with the μPD78320 and μPD78322.		Much more instructions are added to those used with the μPD78310A and μPD78312A.	
Other specifications		• Watchdog timer: Provided • Standby function (STOP mode, HALT mode)			
		Wait control pin		—	
		—		Can be directly connected to the turbo access manager (μPD71P301).	
Package		• 64-pin plastic QFP (14 × 14 mm)		• 68-pin plastic QFJ (950 × 950 mil) • 74-pin plastic QFP (20 × 20 mm) • 80-pin plastic QFP (14 × 20 mm)	

APPENDIX B TOOLS

B.1 DEVELOPMENT TOOLS

The following tools are provided for developing a system that uses the μPD78352A:

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>Host machine</th><th>OS</th><th>Distribution media</th><th>Part number</th></tr><tr><td rowspan="2">PC-9800 series</td><td rowspan="2">MS-DOS™
Ver. 3.30
to
Ver. 5.00A Note</td><td>3.5-inch 2HD</td><td>μS5A13RA78K3</td></tr><tr><td>5.25-inch 2HD</td><td>μS5A10RA78K3</td></tr><tr><td>IBM PC/AT™</td><td>PC DOS™
(Ver. 3.1)</td><td>5.25-inch 2HC</td><td>μS7B10RA78K3</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>Host machine</th><th>OS</th><th>Distribution media</th><th>Part number</th></tr><tr><td rowspan="2">PC-9800 series</td><td rowspan="2">MS-DOS
Ver. 3.30
to
Ver. 5.00A Note</td><td>3.5-inch 2HD</td><td>μS5A13CC78K3</td></tr><tr><td>5.25-inch 2HD</td><td>μS5A10CC78K3</td></tr><tr><td>IBM PC/AT</td><td>PC DOS
(Ver. 3.1)</td><td>5.25-inch 2HC</td><td>μS7B10CC78K3</td></tr></table></td></tr>
</table>

Note This software cannot use the task swap function, which is available in MS-DOS Ver. 5.00 and Ver. 5.00A.

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

<table>
<tr><td rowspan="2">Hardware</td><td>PG-1500</td><td colspan="3">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.</td></tr>
<tr><td>PA-78P352G[Note 1]
PA-78P352KK[Note 1]</td><td colspan="3">Programmer adapter for writing programs to the μPD78P352. Used with a PROM programmer such as the PG-1500.
PA-78P352G : For μPD78P352G
PA-78P352KK : For μPD78P352KK[Note 1]</td></tr>
<tr><td rowspan="4">Software</td><td rowspan="4">PG-1500 controller</td><td colspan="3">This program enables the host machine to control the PG-1500 through the serial and parallel interfaces.</td></tr>
<tr><td rowspan="2">Host machine</td><td rowspan="2">OS</td><td rowspan="2">Distribution media</td><td rowspan="2">Part number</td></tr>
<tr></tr>
<tr><td>PC-9800 series</td><td>MS-DOS Ver. 3.30 to Ver. 5.00A[Note 2]</td><td>3.5-inch 2HD</td><td>μS5A13PG1500</td></tr>
<tr><td>PC-9800 series</td><td>MS-DOS Ver. 3.30 to Ver. 5.00A[Note 2]</td><td>5.25-inch 2HD</td><td>μS5A10PG1500</td></tr>
<tr><td>IBM PC/AT</td><td>PC DOS (Ver. 3.1)</td><td>5.25-inch 2HC</td><td>μS7B10PG1500</td></tr>
</table>

Notes 1. Under development

2. This software cannot use the task swap function, which is available in MS-DOS Ver. 5.00 and Ver. 5.00A.

Remark It is guaranteed that the PG controller runs only under the OSs on the corresponding host machines described above.

Debugging tools

Hardware	IE-78350-R	In-circuit emulator for developing and debugging an application system. For debugging, connect the emulator to the host machine.
	IE-78350-R-EM1	I/O emulation board for emulating peripheral hardware such as the I/O ports of the μPD78352A.
	EP-78240GC-R EV-9200GC-64	Emulation probe for connecting the IE-78350-R to the target system. Used with the IE-78350-R-EM1. One EV-9200GC-64 conversion socket is provided for connection to the target system.
Software	IE-78350-R control program (IE controller)	This control program allows the user to control the IE-78350-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 Ver. 3.30 to Ver. 5.00A[Note]	3.5-inch 2HD	μS5A13IE78350
		5.25-inch 2HD	μS5A10IE78350
IBM PC/AT	PC DOS (Ver. 3.1)	5.25-inch 2HC	μS7B10IE78350

Note This software cannot use the task swap function, which is available in MS-DOS Ver. 5.00 and Ver. 5.00A.

Remark It is guaranteed that the IE controller runs only under the OSs on the corresponding host machines described above.

B.2 EVALUATION TOOLS

The following evaluation tools are provided for evaluating the function of the μPD78352A:

Part number	Host machine	Function
EB-78350-98	PC-9800 series	When the evaluation tool is connected to the host machine, the functions of the μPD78352A can easily be evaluated. As the command system of the EB-78350-98/PC conforms to that of the IE-78350-R, the migration can easily be made to the development of the application system with the IE-78350-R.
EB-78350-PC	IBM PC/AT	

Cautions 1. These products are not development tools for the application system that uses the μPD78352A.

2. These products do not have an emulation function used during the execution of the program in internal ROM for the μPD78352A.

B.3 EMBEDDED SOFTWARE

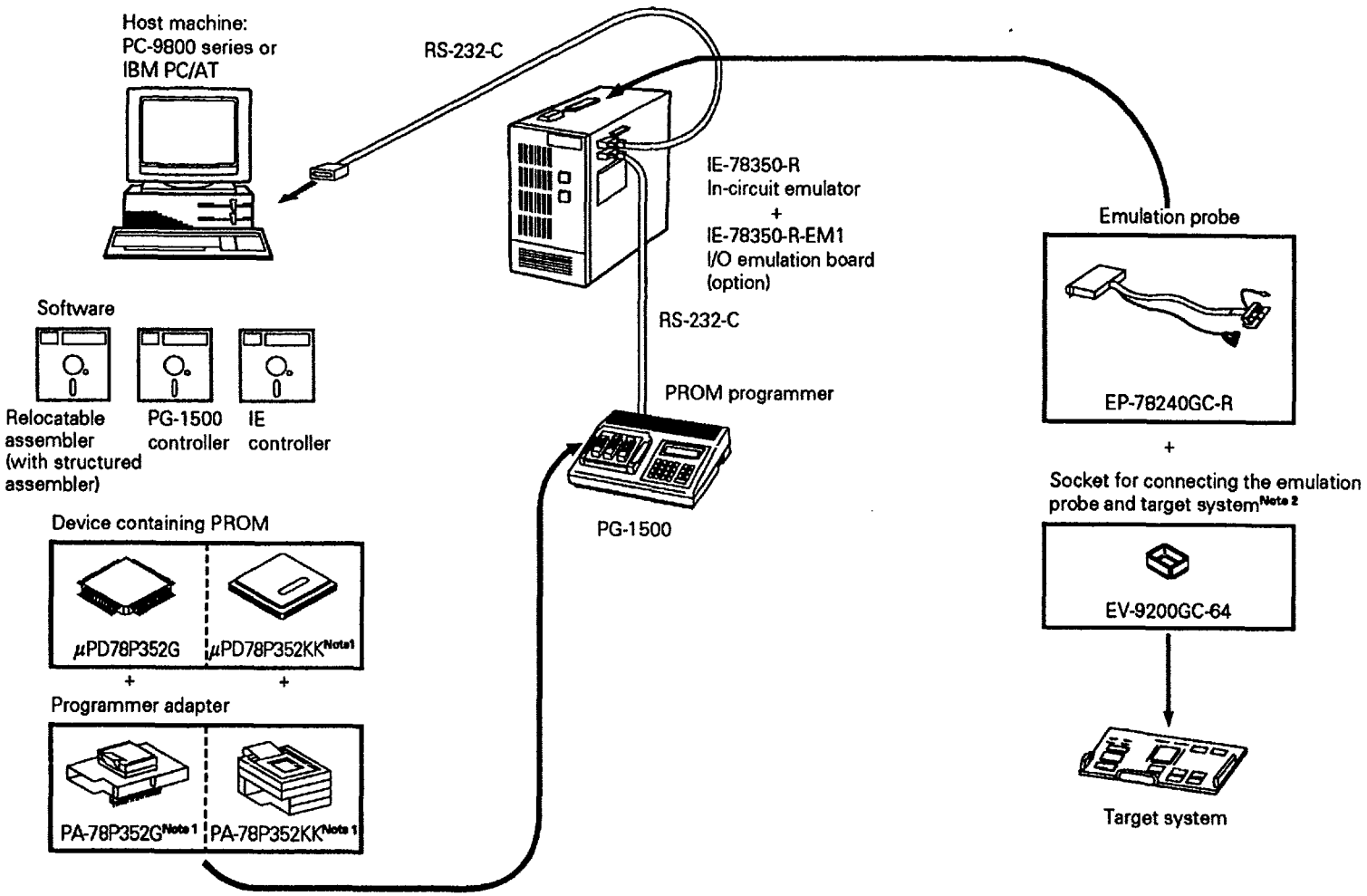
To improve the efficiency of program development and simplify the maintenance of systems incorporating this microcontroller, the following embedded software is provided.

Fuzzy inference development support system

<table>
<tr><td>Tool for creating fuzzy knowledge data (FE9000, FE9200)</td><td colspan="3">This program supports the input/editing and simulation of fuzzy knowledge data (fuzzy rules and membership functions).
<table><tr><th>Host machine</th><th>OS</th><th>Distribution media</th><th>Part number</th></tr><tr><td>PC-9800 series</td><td rowspan="2">MS-DOS (Ver. 3.30 to Ver. 5.00A[Note 1])</td><td>3.5-inch 2HD</td><td>μS5A13FE9000</td></tr><tr><td>5.25-inch 2HD</td><td>μS5A10FE9000</td></tr><tr><td>IBM PC/AT</td><td>PC DOS (Ver. 3.1)</td><td>5.25-inch 2HC</td><td>μS7B10FE9200</td></tr></table></td></tr>
<tr><td>Translator (FT78K3)[Note 2]</td><td colspan="3">This program converts fuzzy knowledge data, obtained using the tool for creating fuzzy knowledge data, into an assembler source program for RA78K/III.
<table><tr><th>Host machine</th><th>OS</th><th>Distribution media</th><th>Part number</th></tr><tr><td>PC-9800 series</td><td rowspan="2">MS-DOS (Ver. 3.30 to Ver. 5.00A[Note 1])</td><td>3.5-inch 2HD</td><td>μS5A13FT78K3</td></tr><tr><td>5.25-inch 2HD</td><td>μS5A10FT78K3</td></tr><tr><td>IBM PC/AT</td><td>PC DOS (Ver. 3.1)</td><td>5.25-inch 2HC</td><td>μS7B10FT78K3</td></tr></table></td></tr>
<tr><td>Fuzzy inference module (FI78K/III)[Note 2]</td><td colspan="3">This program performs fuzzy inference by linking the fuzzy knowledge data converted by Translator.
<table><tr><th>Host machine</th><th>OS</th><th>Distribution media</th><th>Part number</th></tr><tr><td>PC-9800 series</td><td rowspan="2">MS-DOS (Ver. 3.30 to Ver. 5.00A[Note 1])</td><td>3.5-inch 2HD</td><td>μS5A13FI78K3</td></tr><tr><td>5.25-inch 2HD</td><td>μS5A10FI78K3</td></tr><tr><td>IBM PC/AT</td><td>PC DOS (Ver. 3.1)</td><td>5.25-inch 2HC</td><td>μS7B10FI78K3</td></tr></table></td></tr>
<tr><td>Fuzzy inference debugger (FD78K/III)</td><td colspan="3">This software supports the evaluation and adjustment of fuzzy knowledge data at the hardware level, by using an in-circuit emulator.
<table><tr><th>Host machine</th><th>OS</th><th>Distribution media</th><th>Part number</th></tr><tr><td>PC-9800 series</td><td rowspan="2">MS-DOS (Ver. 3.30 to Ver. 5.00A[Note 1])</td><td>3.5-inch 2HD</td><td>μS5A13FD78K3</td></tr><tr><td>5.25-inch 2HD</td><td>μS5A10FD78K3</td></tr><tr><td>IBM PC/AT</td><td>PC DOS (Ver. 3.1)</td><td>5.25-inch 2HC</td><td>μS7B10FD78K3</td></tr></table></td></tr>
</table>

Notes 1. This software cannot use the task swap function, which is available in MS-DOS Ver. 5.00 and Ver. 5.00A.

2. Under development



Notes 1. Under development

2. The socket is supplied with the emulation probe.

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

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**

IC-3391

70646

80