NEC
NEC Electronics Inc.

μPD78352 Family (μPD78350/352A/P352) 16-/8-Bit, K-Series Microcontrollers With Real-Time Output Ports

September 1993

Description

The μPD78350, μPD78352A, and μPD78P352 are members of the K-Series® of microcontrollers. These 16-/8-bit microcontrollers—with a minimum instruction time of 125 ns at 32 MHz (160 ns at 25 MHz for the μPD78350)—are designed for high-speed, real-time process control. They feature a 16-bit CPU, an 8-bit external data bus, eight banks of main registers, an advanced interrupt handling facility, and a powerful set of memory-mapped on-chip peripherals. A 16-bit multiply and accumulate instruction provides hardware convolution capability. On-board memory includes 640 bytes of RAM, 32K bytes of mask ROM in the μPD78352A, and 32K bytes of UV EPROM or one-time programmable (OTP) ROM in the μPD78P352.

The advanced interrupt handling facility has four levels of programmable hardware-priority control and three methods of servicing interrupt requests, including vectoring, hardware context switching, and macro service. The macro service facility reduces the CPU overhead involved in servicing peripheral interrupts by transferring data between the memory-mapped special function registers (SFRs) and memory without the use of time consuming interrupt service routines. In addition, the macro service facility can be used to perform certain CPU functions, such as event counting and math-oriented data alterations.

The combination of high-speed hardware convolution capability and context switching, eight register banks, and the macro service facility makes these devices ideal for applications in the hard-disk drive and tape drive markets as well as the automotive and industrial control/robotics markets.

K-Series is a registered trademark of NEC Electronics Inc.

Features

- Complete single-chip microcontroller
  - 16-bit ALU
  - 640 bytes of RAM
  - 32K bytes of mask ROM (μPD78352A)
  - 32K bytes of UV EPROM or OTP ROM (μPD78P352)
- Powerful instruction set
  - 16-bit unsigned and signed multiply
  - 16-bit unsigned divide
  - 16-bit multiply and accumulate instruction
  - 1-bit and 8-bit logic instructions
  - String instructions
- Minimum instruction time
  - 160 ns at 25 MHz (μPD78350)
  - 125 ns at 32 MHz (μPD78352A/P352)
- 5-byte instruction prefetch queue
- Memory expansion
  - 8085 bus-compatible
  - 64K-byte address space
- Large I/O capacity
  - Up to 30 I/O port lines (μPD78350)
  - Up to 50 I/O port lines (μPD78352A/P352)
- Memory-mapped, on-chip peripherals (special function registers)
- Timer/counter unit
  - 16-bit free-running timer:
    Two 16-bit capture registers;
    Two external interrupt/capture lines
  - 16-bit timer/event counter:
    One 16-bit compare register;
    One external event counter line
  - 16-bit interval timer:
    One 16-bit compare register
- Two 8-bit precision pulse-width modulated (PWM) output lines
- Programmable priority interrupt controller (four levels)
- Three methods of interrupt service
  - Vectored interrupts
  - Context switching with hardware register bank switch
  - Macro service mode with choice of five different functions
- Watchdog timer with dedicated output
- STOP and HALT standby functions
- Single 5-volt power supply

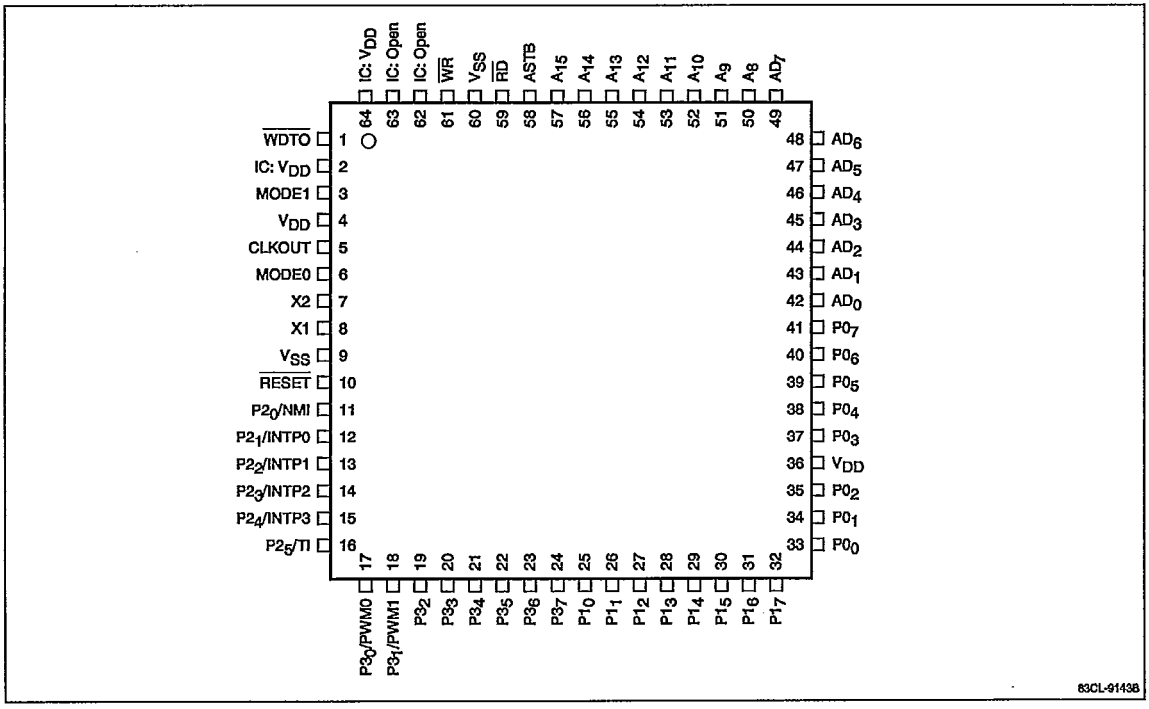
Ordering Information

Part Number	ROM	Package	Package Drawing
μPD78350GC-3BE	ROMless	64-pin plastic QFP (3.0-mm height)	P64GC-80-3BE
μPD78352AG-xxx-22	32K mask ROM	64-pin plastic QFP (1.7-mm height)	P64G-80-22-1
μPD78P352G-22	32K OTP ROM		
μPD78P352KK	32K UV EPROM	64-pin ceramic LCC with window	X80KW-80B

xxx indicates ROM code suffix.

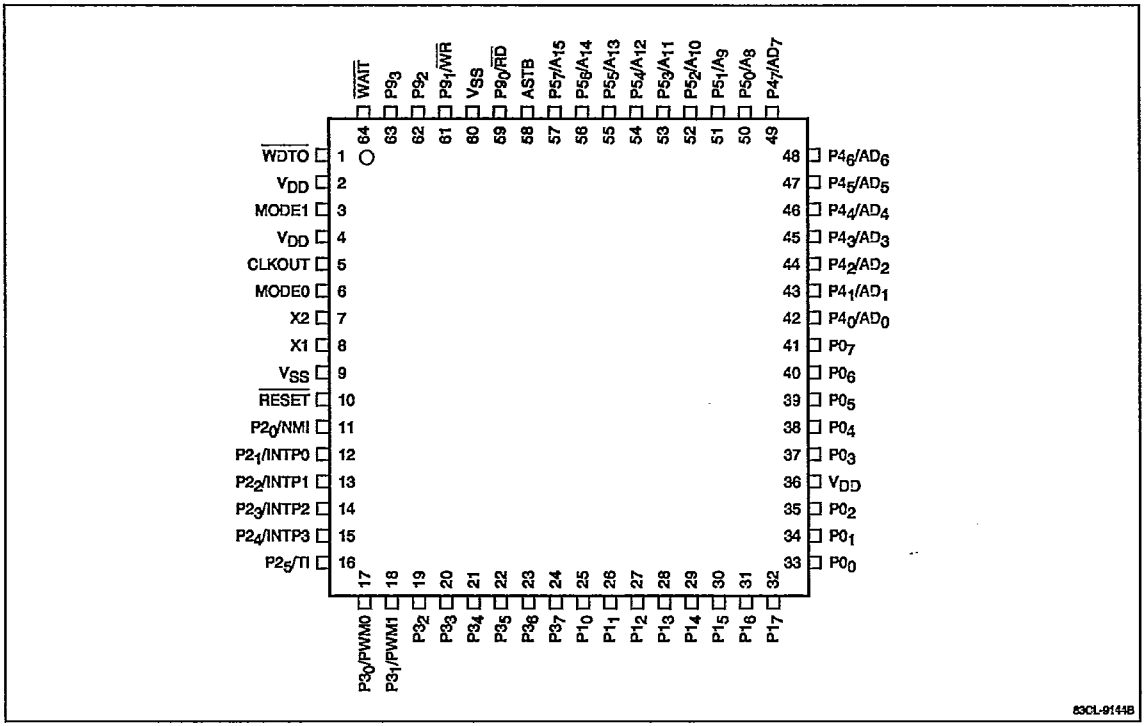
Pin Configurations

64-Pin Plastic QFP (μPD78350)



Pin Configurations (cont)

64-Pin Plastic QFP and Ceramic LCC (μPD78352A/P352)

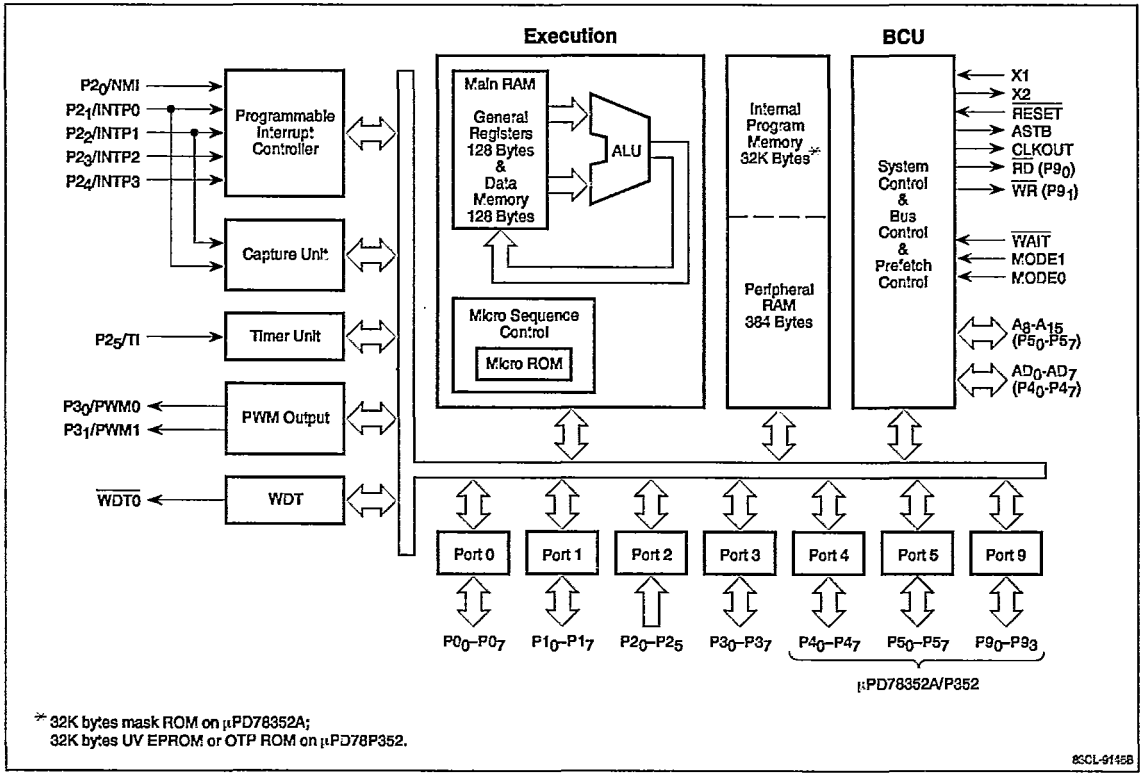


Pin Functions; Normal Operating Mode

Symbol	Function	Alternate Symbol	Alternate Function
$P0_0$ - $P0_7$	Port 0; 8-bit, bit-selectable I/O port		
$P1_0$ - $P1_7$	Port 1; 8-bit, bit-selectable I/O port		
$P2_0$	Port 2; 6-bit input port	NMI	External nonmaskable interrupt
$P2_1$		INTP0	Maskable external interrupts
$P2_2$		INTP1	
$P2_3$		INTP2	
$P2_4$		INTP3	
$P2_5$		TI	External input for timer 1
$P3_0$	Port 3; 8-bit, bit-selectable I/O port	PWM0	Pulse-width modulated outputs
$P3_1$		PWM1	
$P3_2$ - $P3_7$			
$P4_0$ - $P4_7$	Port 4; byte-selectable I/O port (μPD78352A/P352)	AD_0 - AD_7	Low-order 8 bits of the multiplexed external address/data bus
$P5_0$ - $P5_7$	Port 5; bit-selectable I/O port (μPD78352A/P352)	A_8 - A_{15}	High-order 8 bits of the external address bus
$P9_0$	Port 9; 4-bit, bit-selectable I/O port (μPD78352A/P352). For 78350, $P9_0$ functions as $\overline{RD}$ and $P9_1$ functions as $\overline{WR}$ signals only. $P9_2$ and $P9_3$ are not provided for 78350.	$\overline{RD}$	External read strobe output
$P9_1$		$\overline{WR}$	External write strobe output
$P9_2$		IC	Internally connected; must be left open (μPD78350).
$P9_3$		IC	
ASTB	Address strobe output; used to latch address for external memory.		
CLKOUT	Output of the system clock		
IC	Internally connected; must be left open.		
MODE0	Connect to V_{DD} for μPD78350 and μPD78P352 in programing mode. Connect to V_{SS} for normal operation of μPD78352A/P352. The level of this pin cannot be changed during normal operation.		
MODE1	Always connect to V_{SS}. The level of this pin cannot be changed during normal operation.		
$\overline{RESET}$	External system reset input		
$\overline{WAIT}$	A low-level input adds wait states to the external bus cycle; used by very-slow memory and/or peripherals (only for 78352A/P352).		
$\overline{WDTO}$	Open-drain output from the watchdog timer		
X1	Crystal connection or external clock input		
X2	Crystal connection or open for external clock		
V_{DD}	+5-volt power input		
V_{SS}	Ground		

NEC

Block Diagram



* 32K bytes mask ROM on μPD78352A;
32K bytes UV EPROM or OTP ROM on μPD78P352.

83CL-9145B

FUNCTIONAL DESCRIPTION

Central Processing Unit

The central processing unit (CPU) of the μPD78352 family features 16-bit arithmetic including 16 x 16-bit multiply, both unsigned and signed, and 32 x 16-bit unsigned divide (producing a 32-bit quotient and a 16-bit remainder). The signed multiply executes in 1.12 μs and the divide in 3.44 μs at 25 MHz (0.875 and 2.69 μs, respectively, for μPD78352A/P352 at 32 MHz).

Also, a multiply-and-accumulate instruction, "MACW n," performs a signed multiply on factors from a pair of tables and sums the results in the 32-bit register AXDE. The total execution time for 10 terms is 17.2 μs at 25 MHz for the μPD78350 and 13.44 μs at 32 MHz for the μPD78352A/P352.

A CALLT vector table and a CALLF area decrease the number of bytes in the call instructions for commonly used subroutines. A 1-byte call instruction can access up to 32 subroutines through their addresses in the CALLT vector table. A 2-byte call instruction can access any routine beginning in a specific CALLF area.

The internal system clock (f_{CLK}) is generated by dividing the oscillator frequency by 2. Therefore, at the maximum oscillator frequency of 25 MHz for the μPD78350, the clock is 12.5 MHz. Since instructions execute in two or more cycles, the minimum instruction time is 160 ns. For the μPD78352A/P352 running at 32 MHz, the clock is 16 MHz and the minimum instruction time is 125 ns.

Internal RAM

The μPD78352 family has total of 640 bytes of internal RAM. The upper 256-byte area (FE00H-FEFFH) features high-speed access of one or two internal system clocks per word of data depending on the addressing mode and is known as "Main RAM." The remainder (FC80H-FDFFH) is accessed at the same speed as external memory (1 byte per three internal system clocks) and is known as "Peripheral RAM." The general register banks and the macro service control words are stored in Main RAM. The remainder of Main RAM and any unused register bank locations are available for general storage.

Main RAM Access Speed

Access Mode	Internal System Clocks (f_{CLK})
Memory access	2
Saddr access	1
Register access	1

Internal Program Memory

The μPD78352A contains 32K bytes of mask ROM; μPD78P352 contains 32K bytes of UV EPROM or one-time programmable ROM. Instructions are fetched from this program memory at a maximum rate of 1 byte every two internal system clocks. The μPD78350 does not have internal program memory.

External Memory

The μPD78352 family has a 64K-byte address space. The μPD78352A/P352 can access 0, 256, 4K, 16K, or 32K bytes of external memory in the area from 8000H to FDFFH. External memory can be either ROM, RAM, or peripheral as required. The μPD78352A/P352 has an 8-bit wide external data bus and a 16-bit wide external address bus. The low-order 8 bits of the address bus are multiplexed to provide the 8-bit data bus at I/O port 4.

High-order address bits are taken from port 5 as required. Address latch, read, and write strobes are also provided. In the μPD78352A/P352, the memory mode register (MM) controls the size of the external memory It can be programmed to use 0, 4, 6, or 8 bits from port 5 for the high-order address. Any remaining port 5 bits can be used for I/O.

The μPD78350 does not have ports 4 and 5. It has eight dedicated high-order address lines and eight dedicated address/data lines. All memory below address FC80H must be external, and the MM register is not used.

The programmable wait control register (PWC) allows the programmer to specify one or two additional wait states if they are required for slow-speed memory or external peripheral devices. These wait states for internal and external memory are specified independently in 16K-blocks. If additional wait states are required, an external $\overline{WAIT}$ pin is provided.

In addition, by using the AW0 and AW1 bits of the PWC register, the width of the ASTB signal can be increased by one cycle to allow more precharge time for dynamic RAMs or more address decoding time. This address wait signal can be enabled in 32K-byte blocks. See figure 1.

Figure 1. Programmed Wait Control Register

15							8	7							0
AW1	AW0	0	0	0	0	0	0	PWC7	PWC6	PWC5	PWC4	PWC3	PWC2	PWC1	PWC0

16K Memory Block	Wait Control Register Bits		Wait States	Data Access Clocks	Fetch Cycle Mode	Fetch Clocks
0000H-3FFFH	PWC1, PWC0	0 0	0	3	Normal	3
		0 1	1	4		4
		1 0	2	5		5
		1 1	0	4	High-speed	2
4000H-7FFFH	PWC3, PWC2	0 0	0	3	Normal	3
		0 1	1	4		4
		1 0	2	5		5
		1 1	0	4	High-speed	2
8000H-BFFFH	PWC5, PWC4	0 0	0	3	Normal	3
		0 1	1	4		4
		1 0	2	5		5
		1 1	—	N/A	N/A	N/A
C000H-FC7FH, *FFD0H-FFDFH	PWC7, PWC6	0 0	0	3	Normal	3
		0 1	1	4		4
		1 0	2	5		5
		1 1	—	N/A	N/A	N/A
FC80H-FDFFH	PWC7, PWC6	0 0	0	3	Normal	3
		0 1	0	3		3
		1 0	0	3		3
		1 1	—	N/A	N/A	N/A

* Data in the SFR external access area, FFD0H-FFDFH, cannot be fetched.

32K Memory Block	Wait Control Register Bits		Address Wait
0000H-7FFFH	AW0	0	Disabled
		1	Enabled
8000H-FC7FH	AW1	0	Disabled
		1	Enabled

Program Fetch

The μPD78352 family devices allow opcode fetch in the area between 0000H and FDFFH; they contain a 5-byte instruction prefetch queue.The bus control unit can fetch an instruction byte from memory during cycles in which the execution unit is not using the memory bus. If the instruction byte is fetched from on-chip memory, two internal system clocks are required for each byte, and the queue can hold 5 bytes. If the instruction is fetched from external memory, three internal system clocks are required for each byte, and the queue can hold 3 bytes. For programs located in internal memory, the PWC register also can be programmed to allow 1 byte to be fetched every two, three, four, or five internal system clocks.

CPU Control Registers

Program Counter. The program counter is a 16-bit register that holds the address of the next instruction to be executed. After reset line goes high, the program counter is loaded with the address stored in locations 0000H and 0001H.

Stack Pointer. The stack pointer is a 16-bit register that holds the address of the last item pushed onto the stack. It is decremented before new data is pushed onto the stack and incremented after data is popped off the stack.

CPU Control Word. The CPU control word (CCW) selects the origin of the interrupt vector and CALLT tables. If the TPF bit (bit 1) is zero, the origin is 0000H; if the TPF bit is one, the origin is 8000H. The CCW is a special function register located at address FFC1H. The addresses of the vectors for the RESET input, operation-code trap, and BRK instruction are fixed at 0000H, 003CH, and 003EH, respectively, and are not altered by the TPF bit.

Program Status Word. The program status word (PSW) is a 16-bit register containing flags that are set or reset depending on the results of an instruction. This register can be written to or read from 8 bits at a time. The high-order 8 bits are called the PSWH and the low-order 8 bits are called the PSWL. The individual flags can also be manipulated on a bit-by-bit basis. The assignment of PSW bits follows.

	7	6	5	4	3	2	1	0
PSWH	UF	RBS2	RBS1	RBS0	0	0	0	0

	7	6	5	4	3	2	1	0
PSWL	S	Z	RSS	AC	IE	P/V	0	CY

UF	User flag
RBS2-RBS0	Active register bank number
S	Sign flag (1 if last result was negative)
Z	Zero flag (1 if last result was zero)
RSS	Register set selection flag
AC	Auxiliary carry flag (carry out of 3 bit)
IE	Interrupt enable flag
P/V	Parity or arithmetic overflow flag
CY	Carry bit (or 1-bit accumulator for logic)

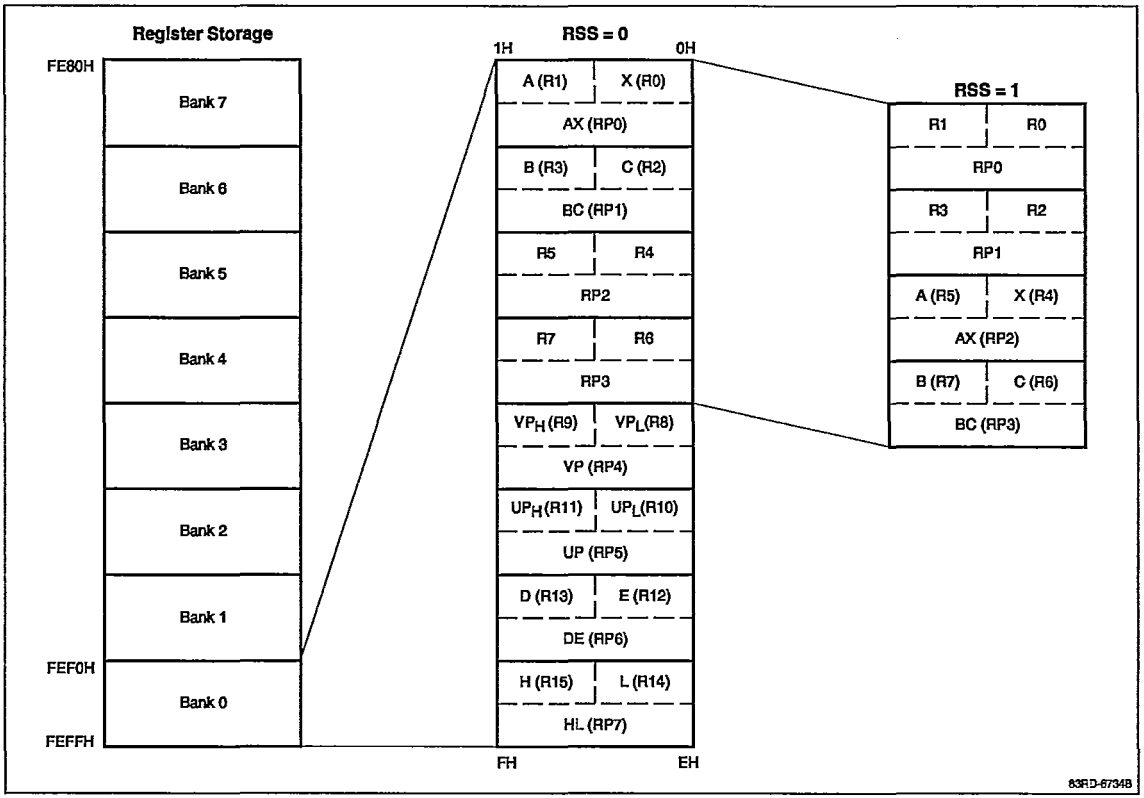
General Registers

There are sixteen 8-bit general registers, which can also be paired to function as 16-bit registers. A complete set of 16 registers is mapped into each of eight program-selectable register banks stored in Main RAM. Three bits in the PSW specify the active register bank.

Registers have functional names (like A, X, B, C for 8-bit registers and AX, BC for 16-bit registers) and absolute names (like R1, R0, R3, R2 for 8-bit registers and RP0, RP1 for 16-bit registers). Each instruction determines whether a register is referred to by functional or absolute name and whether it is 8 or 16 bits.

Two possible relationships may exist between the absolute and functional names of the first four register pairs. The RSS bit in the PSW determines which of these is active at any time. The effect is that the accumulator and counter registers can be saved, and a new set can be specified by toggling the RSS bit. Figure 2 illustrates the general register configuration.

Figure 2. General Registers



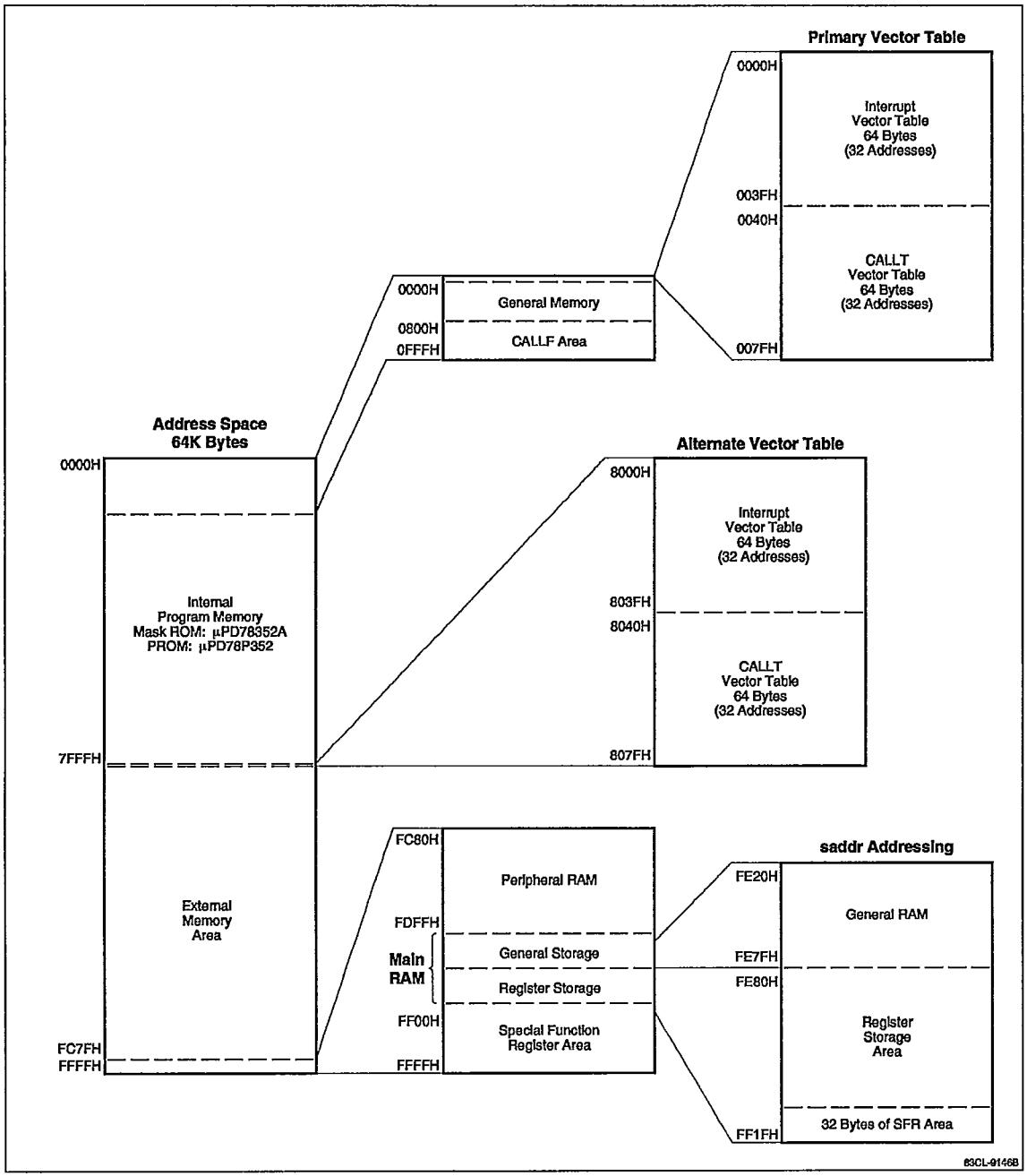
Addressing

The μPD78352 family features 1-byte addressing of both the special function registers and the portion of on-chip RAM from FE20H to FEFFH. The 1-byte sfr addressing accesses the entire SFR area, while the 1-byte saddr addressing accesses the first 32 bytes of the SFR area and 224 bytes of the Main RAM.

The 16-bit SFRs and words of memory in these areas can be addressed by 1-byte saddrp addressing, which is valid for even addresses only. Since many instructions use 1-byte addressing, access to these locations is almost as fast and as versatile as access to the general registers.

There are nine addressing modes for data in main memory: direct, register, register indirect with autoincrement or autodecrement, saddr, saddr indirect, SFR, based, indexed, and based indexed. There are also 8-bit and 16-bit immediate operands. Figure 3 is the memory map of the μPD78352 family.

Figure 3. Memory Map



Special Function Registers

The input/output ports, timers, capture and compare registers, and mode and control registers for both the peripherals and the CPU are collectively known as special function registers. They are all memory-mapped between FF00H and FFFFH and can be accessed either by main memory addressing or by 1-byte SFR addressing. All can be read under program control, and most can also be written. They are either 8 or 16 bits, as required, and many of the 8-bit registers are capable of single-bit access as well.

Locations FFD0H through FFDFH are known as the external access area. Registers in external circuitry, interfaced and mapped to these addresses, can be addressed with SFR addressing. Table 1 lists the special function registers.

Table 1. Special Function Registers

				Access Units (Bits)			
Address	**Register (SFR)**	**Symbol**	**R/W**	**1**	**8**	**16**	**State After Reset**
FF00H	Port 0	P0	R/W	x	x	—	Undefined
FF01H	Port 1	P1	R/W	x	x	—	Undefined
FF02H	Port 2	P2	R	x	x	—	Undefined
FF03H	Port 3	P3	R/W	x	x	—	Undefined
FF04H	Port 4 (Note 1)	P4	R/W	x	x	—	Undefined
FF05H	Port 5 (Note 1)	P5	R/W	x	x	—	Undefined
FF09H	Port 9 (Note 1)	P9	R/W	x	x	—	Undefined
FF10H-FF11H	Compare register 00	CT00	R/W	—	—	x	Undefined
FF12H-FF13H	Compare register 01	CT01	R/W	—	—	x	Undefined
FF14H-FF15H	Compare register 10	CM10	R/W	—	—	x	Undefined
FF1EH-FF1FH	Compare register 20	CM20	R/W	—	—	x	Undefined
FF20H	Port 0 mode register	PM0	R/W	x	x	—	FFH
FF21H	Port 1 mode register	PM1	R/W	x	x	—	FFH
FF23H	Port 3 mode register	PM3	R/W	x	x	—	FFH
FF25H	Port 5 mode register (Note 1)	PM5	R/W	x	x	—	FFH
FF29H	Port 9 mode register (Note 1)	PM9	R/W	x	x	—	xFH
FF30H-FF31H	Timer register 0	TM0	R	—	—	x	00H
FF32H-FF33H	Timer register 1	TM1	R	—	—	x	00H
FF34H-FF35H	Timer register 2	TM2	R	—	—	x	00H
FF38H	Timer control register 0	TMC0	R/W	x	x	—	00H
FF39H	Timer control register 1	TMC1	R/W	x	x	—	00H
FF3CH	External interrupt mode register 0	INTM0	R/W	x	x	—	00H
FF3DH	External interrupt mode register 1	INTM1	R/W	x	x	—	00H
FF43H	Port 3 mode control register 0	PMC3	R/W	x	x	—	00H
FF62H	Port read control register	PRDC	R/W	x	x	—	00H
FF64H	PWM control register	PWMC	R/W	x	x	—	00H
FF66H	PWM buffer register 0	PWM0	R/W	x	x	—	Undefined
FF6EH	PWM buffer register 1	PWM1	R/W	x	x	—	Undefined
FFA8H	In-service priority register	ISPR	R	x	x	—	00H
FFAAH	Interrupt mode control register	IMC	R/W	x	x	—	80H
FFACH	Interrupt mask flag register	MKL	R/W	x	x	—	7FH

Table 1. Special Function Registers (cont)

Address	Register (SFR)	Symbol	R/W	Access Units (Bits) 1	8	16	State After Reset
FFACH-FFADH	Interrupt mask flag register (Note 2)	MK	R/W	—	—	x	xx7FH
FFCOH	Standby control register (Note 3)	STBC	R/W	—	x	—	0000 x000B
FFC1H	CPU control word	CCW	R/W	x	x	—	00H
FFC2H	Watchdog timer mode register (Note 3)	WDM	R/W	—	x	—	00H
FFC4H	Memory expansion mode register	MM	R/W	x	x	—	00H
FFC6H-FFC7H	Programmable wait control register	PWC	R/W	—	—	x	C0AAH
FFD0H-FFDFH	External access area	—	R/W	x	x	—	Undefined
FFE0H	Interrupt control register (INTOV)	OVIC	R/W	x	x	—	43H
FFE1H	Interrupt control register (INTP0)	PIC0	R/W	x	x	—	43H
FFE2H	Interrupt control register (INTP1)	PIC1	R/W	x	x	—	43H
FFE3H	Interrupt control register (INTCM10)	CMIC10	R/W	x	x	—	43H
FFE4H	Interrupt control register (INTCM20)	CMIC20	R/W	x	x	—	43H
FFE5H	Interrupt control register (INTP2)	PIC2	R/W	x	x	—	43H
FFE6H	Interrupt control register (INTP3)	PIC3	R/W	x	x	—	43H

Notes:

(1) μPD78352A/P352 only.

(2) Used only when a word is accessed by an instruction with the sfrp operand.

(3) These are protected registers, which can be written by a special instruction only.

Input/Output Ports

The μPD78350 has four I/O ports providing a total of 30 I/O lines. The μPD78352A/P352 have an additional three I/O ports for a total of 50 I/O lines.

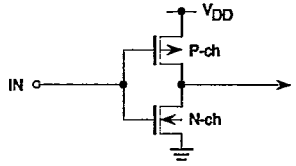
Ports P0, P1, and P3 are 8-bit input/output ports and P2 is a 6-bit input port. All the bits in P0, P1, and P3 can be individually selected for either input or output using port mode registers PM0, PM1, and PM3. Bits $P3_0$ and $P3_1$ can also be programmed for use as PWM outputs PWM0 and PWM1 by using port 3 mode control register PMC3.

Port P2 functions only in the control mode as input pins for the NMI signal, the INTP0 to INTP3 interrupt signals, and the external count clock for timer 1 (TI). However, any masked interrupt automatically becomes an input line and the state of all the pins can be read by the program using a read instruction to port 2. Each pin of P2 can be programmed for rising, falling, or both rising and falling edge detection.

The output level of the P0, P1, and P3 I/O pins can be tested to determine whether they agree with the contents of the output latch. When the low-order bit of port read control register PRDC is set to 1, the output level of the I/O pins can be read with the port still in the output mode. These data values can be compared with the data known to be in the output latch to determine if the port is functioning correctly. Figure 4 shows the structure of each port pin.

Figure 4. I/O Circuits

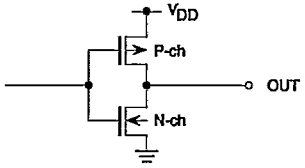
Type 1. $\overline{WAIT}$, MODE0, MODE1



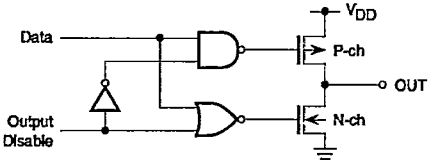
Type 2. Port P2, $\overline{RESET}$

Schmitt trigger input with hysteresis characteristics.

Type 3. CLKOUT

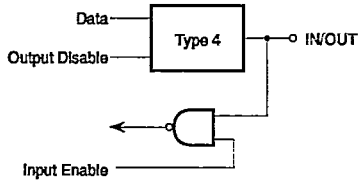


Type 4. Port P9, $\overline{RD}$, $\overline{WR}$, ASTB

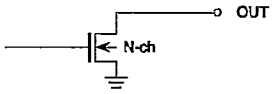


With both P-ch and N-ch off, the output is high impedance.

Type 5. Ports P0, P1, P3, P4, P5; AD_0-AD_7; A_8-A_{15}



Type 14. $\overline{WDTO}$



83CL-9147B

The three additional input/output ports in the μPD78352A/P352 are ports P4, P5, and P9. These ports are available if external memory or memory-mapped external circuitry is not being used. Port 4 is shared with the low-order address/data bus (AD_0 to AD_7) and is byte-selectable for input or output. Port 5 is shared with the high-order address bus (A_8 to A_{15}). Depending on the amount of external memory used, either 8, 6, 4, or 0 bits are available for bit-selectable I/O. Port 9 is a 4-bit, bit-selectable I/O port. Two of its pins are shared with the read and write strobes.

Timers

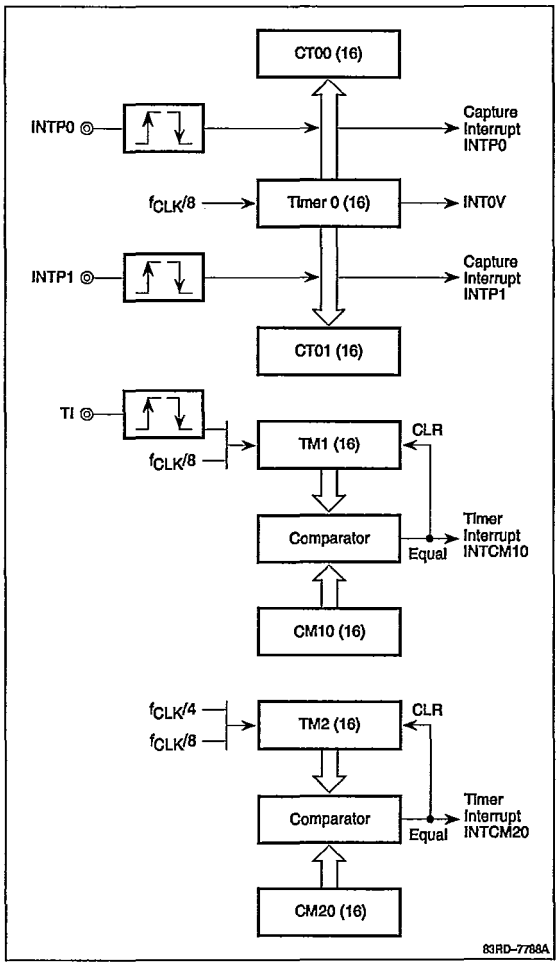
The μPD78352 family has three 16-bit timers. Two of them count only the internal system clock; the third counts either the internal system clock or external events. Refer to the block diagram, figure 5.

Timer 0 is a 16-bit, free-running counter that counts the internal system clock ($f_{CLK}/8$) and generates an interrupt request (INTOV) when it overflows. It also has two associated capture registers, CT00 and CT01. The timer value can be captured in synchronization with external interrupt lines INTP0 and INTP1, respectively. These lines can be programmed to trigger interrupts as well.

Timer 1 is a 16-bit counter serving as an interval timer or an event counter. It can count either the internal system clock ($f_{CLK}/8$) or external events sensed on the TI line. It has an associated comparator register, CM10. When the counter contents match the CM10 contents, the counter is cleared to 0, and an interrupt request (INTCM10) is generated. The counter continues to count until disabled by software.

Timer 2 is a 16-bit counter that serves as an interval timer. It can be programmed to count the internal system clock ($f_{CLK}/4$ or $f_{CLK}/8$). It also has an associated comparator register, CM20. When the counter contents match the CM20 contents, the counter is cleared to 0 and an interrupt request (INTCM20) is generated. The counter continues to count until disabled by software.

Figure 5. Timers Block Diagram

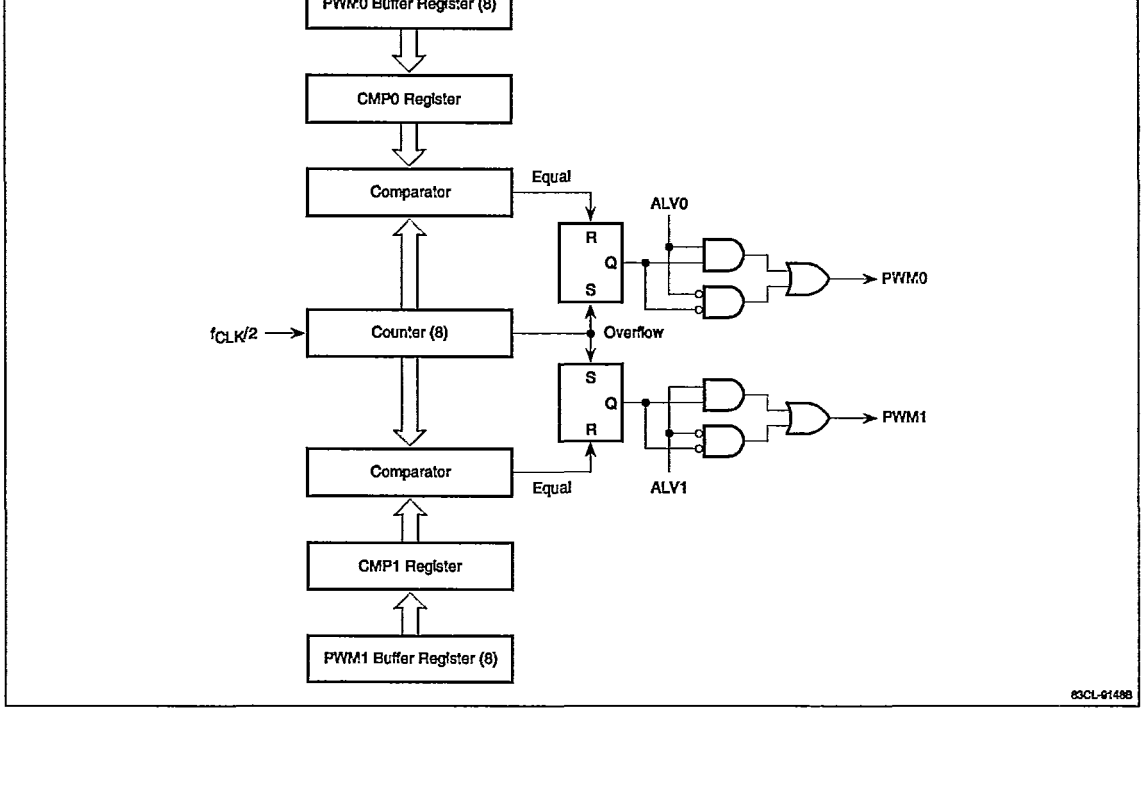


Pulse-Width Modulated Outputs

The μPD78352 family has two high-speed, pulse-width modulated (PWM) outputs. A single 8-bit, free-running counter counts the internal system clock $f_{CLK}/2$ and serves both outputs. For the μPD78350 running at 25 MHz (f_{CLK} = 12.5 MHz), the resolution is 160 ns and the repetition rate is 24.4 kHz. For the μPD78352A/P352 running at 32 MHz (f_{CLK} = 16 MHz), the resolution is 125 ns and the repetition rate is 31.25 kHz.

The polarity of each output can be selected under program control. Whenever the counter overflows, the CMP0 and CMP1 registers are loaded from their respective PWM buffer registers and each output becomes active. When the counter value matches the value in the associated compare register, that output goes inactive. The two PWM outputs, PWM0 and PWM1, share pins with port 3 bits 0 and 1, respectively.

Figure 6. Pulse-Width Modulated Outputs



Interrupts

The μPD78352 family has seven maskable hardware interrupt sources: four external and three internal. The four external maskable interrupts share pins with port 2. Two of them, INTP0 and INTP1, can also be used to trigger capture events in registers CT00 and CT01 associated with timer 0. In addition, there are two nonmaskable interrupts, three software interrupts, and reset. The software interrupts, generated by the BRK or BRKCS instruction and the operation code trap, are not maskable. See table 2.

Table 2. Interrupt Sources

Type of Request	Default Priority	Signal Name	Source	Location	Macro Service Control Word	Vector Address TPF = 0	Vector Address TPF = 1
Software	—	—	Operation code trap	CPU	—	003CH	
	—	—	BRK instruction	CPU	—	003EH	
	—	—	BRKCS instruction (Note 1)	CPU	—	—	
Nonmaskable	—	NMI	NMI input pin	External	—	0002H	8002H
	—	INTWDT	Watchdog timer overflow	Internal	—	0004H	8004H
Maskable	0	INTOV	Timer 0 overflow	Internal	FE06H	0006H	8006H
	1	INTP0	INTP0 pin	External	FE08H	0008H	8008H
	2	INTP1	INTP1 pin	External	FE0AH	000AH	800AH
	3	INTCM10	CM10 coincidence	Internal	FE0CH	000CH	800CH
	4	INTCM20	CM20 coincidence	Internal	FE0EH	000EH	800EH
	5	INTP2	INTP2 pin	External	FE10H	0010H	8010H
	6	INTP3	INTP3 pin	External	FE12H	0012H	8012H
Reset	—	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$ pin	External	—	0000H	

Note:

(1) Initiates context switch

Interrupt Servicing

The μPD78352 family provides four levels of programmable hardware priority control and three different methods of handling maskable interrupt requests: standard vectoring, context switching, and macro service. The programmer can choose the priority and mode of servicing each maskable interrupt by using the interrupt control registers.

Interrupt Control Registers

The μPD78352 family has 10 interrupt control registers. Each maskable interrupt request has its own control register, which includes bits to specify interrupt request, interrupt mask, macro service enable, context switch enable, and priority. Priorities range from 0 (highest) to 3. See figure 7.

There is also a mask flag register, MKL, with a bit for each maskable interrupt. Since each interrupt has two mask bits, the masking of the interrupt is the "or" function of those two bits.

Interrupt mode control register IMC can be used to enable or disable nesting of interrupts set to the lowest priority level (level 3). Inservice priority register ISPR is used by the hardware to hold the priority level of the interrupt request currently being serviced. It is manipulated by hardware only, but it can be read by software.

Finally, the IE bit of the program status word also is used to control the interrupts. If the IE bit is 0, all maskable interrupts, but not macro service, are disabled. The IE bit can be set or cleared by the EI or DI instruction, respectively, or by direct writing to the PSW. The IE bit is cleared each time an interrupt is accepted.

Figure 7. Interrupt Control Register (xxICx)

7	6	5	4
xxIFxx	xxMKxx	xxISMxx	xxCSExx

3	2	1	0
0	0	xxPRx1	xxPRx0

xxIFxx	Interrupt Request Flag
0	No interrupt request
1	Interrupt request received

xxMKxx	Interrupt Mask Flag
0	Interrupt request enabled
1	Interrupt will be pending

xxISMxx	Macro Service Enable
0	Software service
1	Macro service

xxCSExx	Context Switch Enable
0	Vector service
1	Context switch

xxPRx1	xxPRx0	Priority Specification
0	0	Priority 0 (highest)
0	1	Priority 1
1	0	Priority 2
1	1	Priority 3

Interrupt Priority

The two nonmaskable interrupts, NMI and INTWDT, have priority over all others. Their priority relative to each other is under program control.

Four hardware-controlled priority levels are available for the maskable interrupts. Any one of the four levels can be assigned by software to each of the maskable interrupt lines. Interrupt requests of a priority higher that the processor's current priority level are accepted; requests of the same or lower priority are held pending until the processor's priority state is lowered by a return instruction from the current service routine.

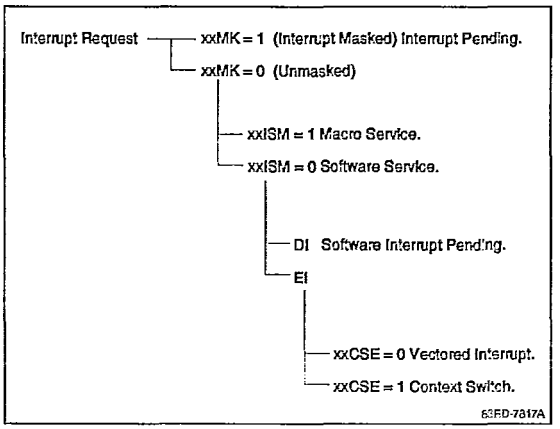
By setting the PRSL bit of the IMC register to zero, it is possible to specify in software that level 3 interrupts (the lowest level) can be accepted when the processor is operating at level 3. This nesting within a level applies to level 3 only.

Interrupt requests programmed to be handled by macro service have priority over all software interrupt service regardless of the assigned priority level, and macro service requests are accepted even when the interrupt enable bit in the PSW is set to the disable state. See figure 8.

The "Default Priorities" listed in table 2 are fixed by hardware; they are effective only when it is necessary to choose between two interrupt requests of the same software-assigned priority. For example, the default priorities would be used after the completion of a high-priority routine if two interrupts of the same lower priority were pending.

Software interrupts, the BRK and BRKCS instructions, and the operation code trap are executed regardless of the processor's priority level and the state of the IE bit. They do not alter the processor's priority level.

Figure 8. Interrupt Service Sequence



Vectored Interrupt

When vectored interrupt is specified for a given interrupt request, (1) the program status word and the program counter are saved on the stack, (2) the processor's priority is raised to that specified for the interrupt, (3) the IE bit in the PSW is set to zero, and (4) the routine whose address is in the interrupt vector table is entered. At completion of the service routine, the RETI instruction (or RETB instruction for software interrupts) reverses the process, and the μPD78352 family device resumes the interrupted routine.

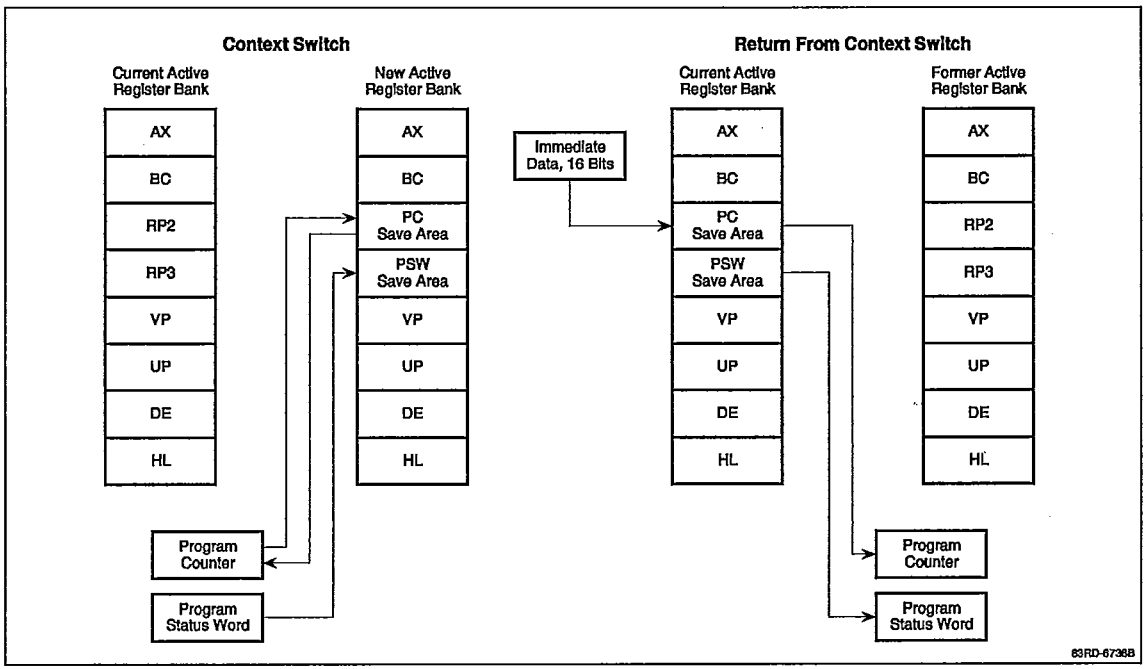
Context Switch

When context switching (figure 9) is specified for a given interrupt, the active register bank is changed to the register bank specified by the three low-order bits of the word in the interrupt vector table. The program counter is loaded from RP2 of the new register bank,

the old program counter and program status word are saved in RP2 and RP3 of the new register bank, and the IE bit in the PSW is set to zero.

At completion of the service routine, the RETCS instruction for routines entered from hardware requests, or the RETCSB instruction for routines entered from the BRKCS instruction, reverse the process. The old program counter and program status word are restored from RP2 and RP3 of the new register bank. The entry address of the service routine, which must be specified in the 16-bit immediate operand of these return instructions, is stored again in RP2.

Figure 9. Context Switching and Return



Macro Service

When macro service is specified for a given interrupt, the macro service hardware temporarily stops the executing program and transfers data between the special function register area and the memory space. Control is then returned to the executing program, providing a completely transparent method of interrupt service. Macro service significantly improves response time and makes it unnecessary to save any registers.

For each request on the interrupt line, one operation is performed, and an 8-bit counter is decremented. When the counter reaches 0, a software service routine is entered according to its specified priority. Either vectored interrupt or context switch can be specified for entry to this routine, which is known as the macro service completion routine.

Macro service is provided for all of the maskable interrupt requests, and each has a specific macro service control word stored in Main RAM. The function to be performed is specified in the control word.

The μPD78352 family provides five different macro service functions.

Function	Description
EVTCNT	Event counter. Counts up to 256 events by incrementing or decrementing the macro service counter. When the counter reaches 00H, the software service routine is entered.
BLKTRS	Block transfer. Transfers a byte or word of data in either direction between a specified special function register and a buffer in Main RAM (FExx).
BLKTRS-P	Block transfer with memory pointer. Transfers a byte or word of data in either direction between a specified special function register and a buffer anywhere in the 64K-byte address space.
DTADIF	Data difference. Stores the difference between the current value of a specified 16-bit special function register and its previous value in a word buffer in Main RAM (FExx).
DTADIF-P	Data difference with memory pointer. Stores the difference between the current value of a specified 16-bit special function register and its previous value in a word buffer anywhere in the 64K-byte address space.

Standby Modes

The standby modes, HALT and STOP, reduce power consumption when CPU action is not required. In HALT mode, the CPU is stopped but the system clock continues to run. The HALT mode is released by any unmasked interrupt, an external NMI, or an external reset pulse. In STOP mode, both the CPU and the system clock are stopped, further minimizing power consumption. The STOP mode is released by either an external reset pulse or an external NMI.

The HALT and STOP modes are entered by programming the standby control register STBC. This register is a protected location and can be written to only by a special instruction. If the third and fourth bytes of the instruction are not complements of each other, the data is not written and an operation code trap interrupt occurs.

Watchdog Timer

The watchdog timer protects against inadvertent program loops. A nonmaskable interrupt occurs if the timer is not reset by the program before it overflows. At the same time, the watchdog timer output pin, $\overline{\text{WDTO}}$, goes active low for a period of 32 system clocks. The $\overline{\text{WDTO}}$ can be connected to the $\overline{\text{RESET}}$ pin or used to control external circuitry. Three program-selectable intervals are available: 10.5, 41.9, and 167.8 ms at 25 MHz; 8.2, 32.8, and 131.1 ms at 32 MHz.

Once started, the timer can be stopped only by an external reset. Watchdog timer mode register WDM is used to select the time interval, to set the relative priority of the watchdog timer interrupt and NMI, and to clear the timer. This register is a protected location and can be written to only by a special instruction. If the third and fourth bytes of the instruction are not complements of each other, the data is not written and an operation code trap interrupt occurs.

External Reset

The μPD78352 family is reset by taking the $\overline{\text{RESET}}$ pin low. The reset circuit contains a noise filter to protect against spurious system resets caused by noise. On power-up, the $\overline{\text{RESET}}$ pin must remain low until the power supply reaches its operating voltage and the oscillator has stabilized. During reset, the program counter is loaded with the address contained in the reset vector table (addresses 0000H, 0001H); program execution starts at that address upon the $\overline{\text{RESET}}$ pin going high. While $\overline{\text{RESET}}$ is low, all external lines except $\overline{\text{WDTO}}$, CLKOUT, V_{SS}, V_{DD}, X1, and X2 are in the high-impedance state.

ELECTRICAL SPECIFICATIONS

Note: Specifications are preliminary for μPD78352A and final for μPD78350/P352.

Absolute Maximum Ratings

T_A = 25°C

Supply voltage, V_{DD}	−0.5 to +7.0 V
Supply voltage, V_{PP}	−0.5 to +13.5 V
Input voltage, V_I	
Except $P2_0$/NMI (A9) of μPD78P352	−0.5 to V_{DD} + 0.5 V
$P2_0$/NMI (A9) of μPD78P352	−0.5 to +13.5 V
Output voltage, V_O	−0.5 to V_{DD} + 0.5 V
Output current, low; I_{OL}	
Each output pin	4.0 mA
Total	100 mA
Output current, high; I_{OH}	
Each output pin	−1.0 mA
Total	−20 mA
Operating temperature, T_{OPT}	−10 to +70°C
Storage temperature, T_{STG}	−65 to +150°C

Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The device should be operated within the limits specified under DC and AC characteristics.

Capacitance

T_A = 25°C; V_{DD} = V_{SS} = 0 V

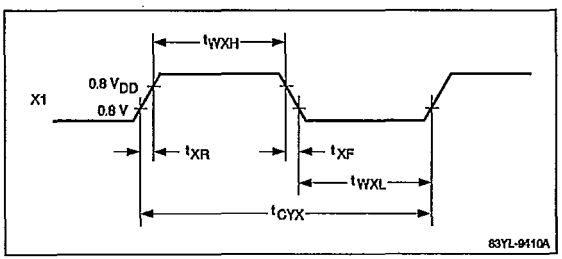
Parameter	Symbol	Max	Unit	Conditions
Input pin capacitance	C_I	20	pF	f = 1 MHz; unmeasured pins returned to 0 V
Output pin capacitance	C_O	20	pF	
I/O pin capacitance	C_{IO}	20	pF	

Oscillator Conditions

T_A = −10 to 70°C; V_{DD} = +5 V ±10%

			μPD78350		μPD78352A/P352		
Oscillator	Parameter	Symbol	Min	Max	Min	Max	Unit
Ceramic resonator or crystal	Oscillation frequency	f_{XX}	8	25	8	32	MHz
External clock	X1 input frequency	f_X	8	25	8	32	MHz
	X1 clock cycle time	t_{CYX}	40	125	31.25	125	ns
	X1 input rise/fall time	t_{XR}, t_{XF}	0	10	0	10	ns
	X1 input high/low level width	t_{WXH}, t_{WXL}	15	60	10	60	ns

External Clock



DC Characteristics

T_A = −10 to +70°C; V_{DD} = +5.0 V ±10%; V_{SS} = 0 V

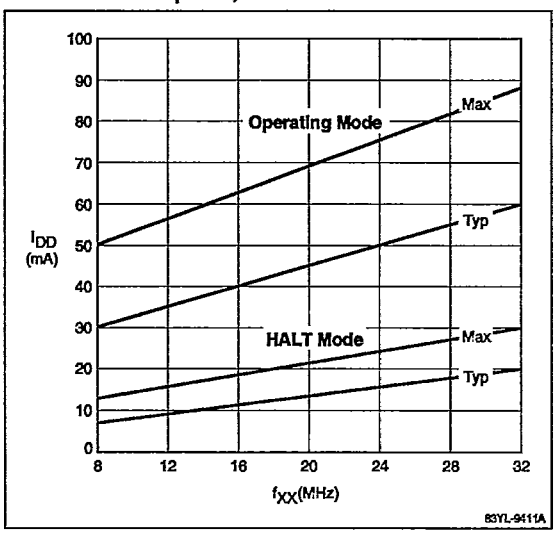
Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Input voltage, low	V_{IL}	0		0.8	V	
Input voltage, high	V_{IH1}	2.2			V	(Note 1)
	V_{IH2}	0.8 V_{DD}			V	(Note 2)
Output voltage, low	V_{OL}			0.45	V	I_{OL} = 2.0 mA
Output voltage, high	V_{OH}	V_{DD} − 1.0			V	I_{OH} = −400 μA
Input leakage current	I_{LI}			±10	μA	V_I = 0 to V_{DD}
Output leakage current	I_{LO}			±10	μA	V_O = 0 to V_{DD}
V_{DD} supply current	I_{DD1}		50	90	mA	Operating mode, μPD78350
			60	87	mA	Operating mode, μPD78352A; f_{XX} = 32 MHz
			80	120	mA	Operating mode, μPD78P352; f_{XX} = 32 MHz
	I_{DD2}		25	40	mA	HALT mode, μPD78350; f_{XX} = 25 MHz
			20	30	mA	HALT mode, μPD78352A; f_{XX} = 32 MHz
			35	50	mA	HALT mode, μPD78P352; f_{XX} = 32 MHz
Data retention voltage	V_{DDDR}	2.5			V	STOP mode
Data retention current	I_{DDDR}		2	10	μA	STOP mode; V_{DDDR} = 2.5 V
			10	50	μA	STOP mode; V_{DDDR} = 5.0 V ±10%

Notes:

(1) All except pins in Note 2.

(2) Pins $\overline{\text{RESET}}$, X1, X2, INTPn, NMI, and TI.

Power Consumption, 78352A



AC Characteristics

T_A = −10 to +70°C; V_{DD} = +5.0 V ±10%; V_{SS} = 0 V

Parameter	Symbol	μPD78350 f_{XX} = 25 MHz Min	μPD78350 f_{XX} = 25 MHz Max	μPD78352A/P352 f_{XX} = 32 MHz Min	μPD78352A/P352 f_{XX} = 32 MHz Max	Unit	Conditions
External Memory Read/Write Operation							
System clock cycle time (Note 1)	t_{CYK}	80	250	62.5	250	ns	C_L = 50 pF
Address setup time to ASTB ↓	t_{SAST}	16		7		ns	C_L = 100 pF (Note 2)
Address hold after ASTB ↓	t_{HSTA}	26		11		ns	C_L = 100 pF
$\overline{RD}$ ↓ to address floating	t_{FRA}		0		0	ns	C_L = 100 pF
Address to data input valid	t_{DAID}		144		100	ns	C_L = 100 pF (Notes 2, 3)
$\overline{RD}$ ↓ to data input valid	t_{DRID}		76		49	ns	C_L = 100 pF (Note 3)
ASTB ↓ to $\overline{RD}$ ↓ delay time	t_{DSTR}	24		15		ns	C_L = 100 pF
Data hold time from $\overline{RD}$ ↑	t_{HRID}	0		0		ns	C_L = 100 pF
$\overline{RD}$ ↑ to address active	t_{DRA}	26		25		ns	C_L = 100 pF
$\overline{RD}$ width low	t_{WRL}	90		63		ns	C_L = 100 pF (Note 3)
ASTB width, high	t_{WSTH}	23		14		ns	C_L = 100 pF (Note 2)
$\overline{WR}$ to data output	t_{DWOD}		29		21	ns	C_L = 100 pF
ASTB ↓ to $\overline{WR}$ ↓ delay	t_{DSTW}	24		15		ns	C_L = 100 pF
Data setup time to $\overline{WR}$ ↑	t_{SODW}	75		57		ns	C_L = 100 pF (Note 3)
Data hold time after $\overline{WR}$ ↑	t_{HWOD}	8		8		ns	C_L = 100 pF
$\overline{WR}$ width, low	t_{WWL}	90		57		ns	C_L = 100 pF (Note 3)
$\overline{WAIT}$ setup time from address	t_{SAWT}		—		107	ns	C_L = 100 pF (Note 2, 4)
$\overline{WAIT}$ setup time from $\overline{RD}$ ↓ or $\overline{WR}$ ↓	t_{SRWRY}		—		37	ns	C_L = 100 pF (Note 4)
$\overline{WAIT}$ hold time from address	t_{HAWT}	—		149		ns	C_L = 100 pF (Note 2, 4)
$\overline{WAIT}$ hold time from $\overline{RD}$ ↓ or $\overline{WR}$ ↓	t_{HRWRY}	—		80		ns	C_L = 100 pF (Note 4)
ASTB ↑ delay time from $\overline{WR}$ ↑	t_{DWST}	110		78		ns	C_L = 100 pF
Address to $\overline{RD}$ ↓ or $\overline{WR}$ ↓ delay	t_{DARW}	89			69	ns	C_L = 100 pF
Other Operations							
NMI high/low level width	t_{WNIH}, t_{WNIL}	2.5		2.0		μs	
INTP0 high/low level width	t_{WI0H}, t_{WI0L}	640		500		ns	
INTP1 high/low level width	t_{WI1H}, t_{WI1L}	640		500		ns	
INTP2 high/low level width	t_{W12H}, t_{W12L}	640		500		ns	
INTP3 high/low level width	t_{WI3H}, t_{W13L}	640		500		ns	
$\overline{RESET}$ high/low level width	t_{WRSH}, t_{WRSL}	2.5		2.0		μs	
TI high/low level width	t_{WTIH}, t_{WTIL}	640		500		ns	

Notes:

(1) t_{CYK} equals twice the period of the crystal or external clock input.

(2) No address wait

(3) No wait states

(4) One external wait state and one internal wait state

Timing Dependent on t_{CYK}

Symbol	Calculation Formula	Min/Max	Unit
t_{SAST}	(0.5 + a)T – 24	Min	ns
t_{HSTA}	0.5T – 14 0.5T – 20 (Note 1)	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 – 14 0.5T – 6 (Note 1)	Min	ns
t_{DSTW}	0.5T – 16	Min	ns
t_{WWL}	(1.5 + n)T – 30 (1.5 + n)T – 36 (Note 1)	Min	ns
t_{DWOD}	0.5T – 10	Max	ns
t_{SODW}	(1 + n)T – 5	Min	ns
t_{SAWT}	(a + n)T – 18 (Note 1)	Max	ns
t_{HAWT}	(0.5 + a + n)T –7 (Note 1)	Min	ns
t_{SRWRY}	(n – 1)T – 25 (Note 1)	Max	ns
t_{HRWRY}	(n – 0.5)T – 14 (Note 1)	Min	ns
t_{DARW}	(a + 1)T+9 (a +1)T+7 (Note 1)	Max	ns
t_{DWST}	1.5T – 10 1.5T – 15 (Note 1)	Min	ns
t_{WIOH}	8T	Min	ns
t_{WIOL}	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

Notes:

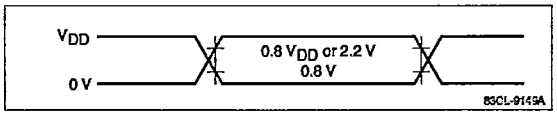
(1) 78352A/P352 only

(2) T = t_{CYK} (ns)

(3) When an address wait is inserted, the value of letter "a" is 1. Otherwise, it is 0.

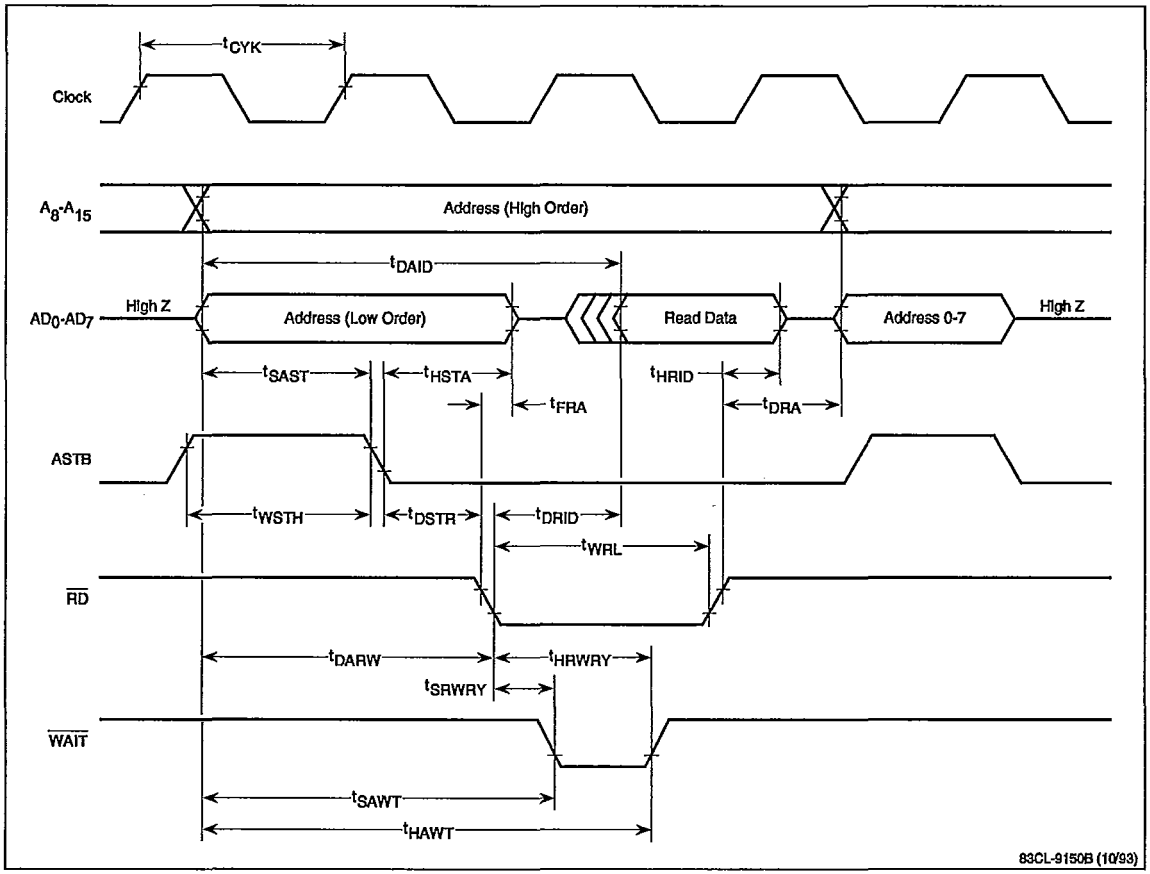
(4) Letter "n" is the number of wait cycles specified by the external wait pin $\overline{WAIT}$ and the PWC register.

AC Timing Test Points



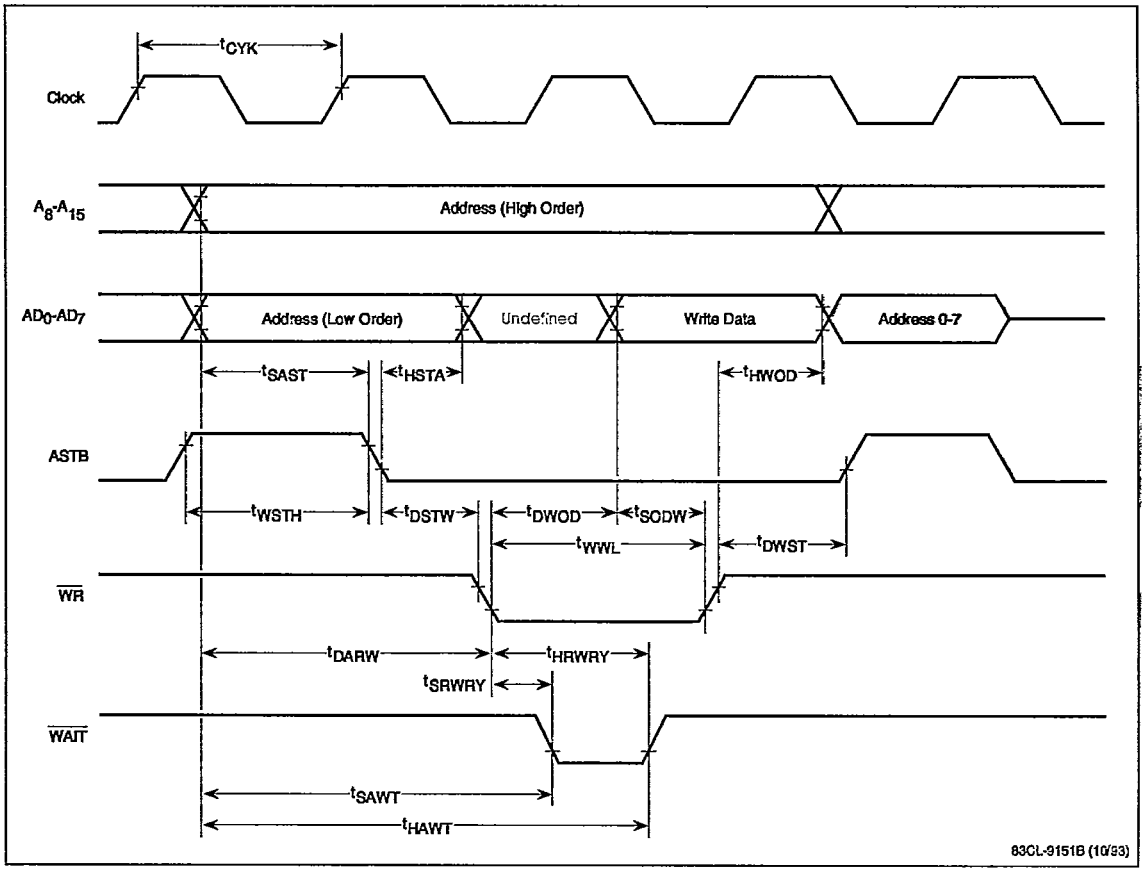
Timing Waveforms

Read Operation



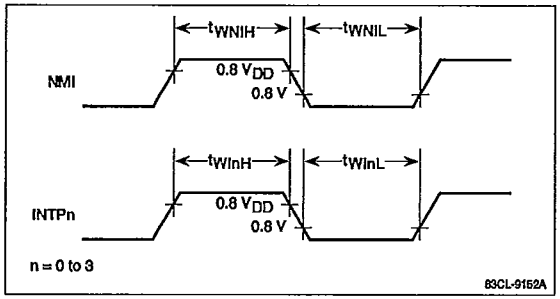
Timing Waveforms (cont)

Write Operation

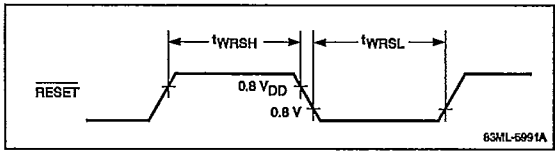


Timing Waveforms (cont)

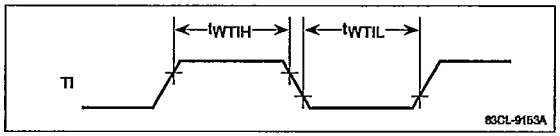
Interrupt Input



Reset Input



TI Input



PROM PROGRAMMING

The PROM in the μPD78P352 is one-time programmable (OTP) or ultraviolet erasable (UV EPROM). The 32,758 x 8-bit PROM has the programming characteristics of an NEC μPD27C1001A, including both page and byte programming modes. The MODE0/V_{PP}, MODE1, $P2_1$, and $\overline{RESET}$ pins are used to place the μPD78P352 into the PROM programmming mode. Table 3 shows the functions of the μPD78P352 pins in normal operating mode and PROM programming mode.

Table 3. Pin Functions During PROM Programming

Function	Normal Operating Mode	Programming Mode
Address input	$P0_0$ - $P0_7$, $P5_0$, $P2_0$, $P5_1$ - $P5_7$	A_0 - A_{16}
Data input	$P4_0$ - $P4_7$	D_0 - D_7
Program pulse	$P1_2$	$\overline{PGM}$
Chip enable	$P1_1$	$\overline{CE}$
Output enable	$P1_0$	$\overline{OE}$
Program voltage	MODE0/V_{PP}	MODE0/V_{PP}
Mode voltage	MODE1, $P2_1$, $\overline{RESET}$	MODE1, $P2_1$, $\overline{RESET}$

PROM Programming Mode

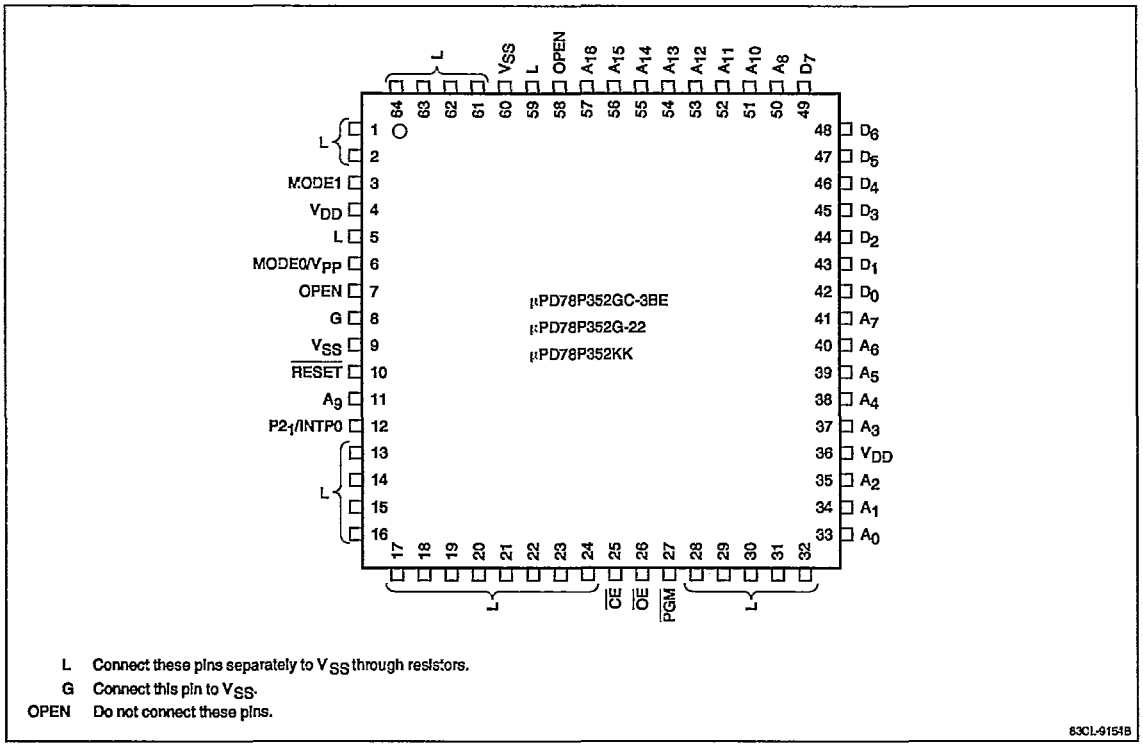
When +6.5 V is applied to the V_{DD} pin and +12.5 V to the MODE0/V_{PP} pin, the μPD78P352 enters the PROM programming mode. Operation in this mode is determined by the setting of $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins as indicated in table 4.

Table 4. Operation Modes For Programming

Mode	MODE1	$P2_1$	$\overline{RESET}$	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	MODE0/V_{PP}	V_{DD}	D_0 - D_7
Page data latch	L	L	L	H	L	H	+12.5 V	+6.5 V	Data input
Page program	L	L	L	H	H	L	+12.5 V	+6.5 V	High impedance
Byte program	L	L	L	L	H	L	+12.5 V	+6.5 V	Data input
Program verify	L	L	L	L	L	H	+12.5 V	+6.5 V	Data output
Program inhibit	L	L	L	X X	L H	L H	+12.5 V	+6.5 V	High impedance
Read	L	L	L	L	L	H	+5.0 V	+5.0 V	Data output
Ouput disable	L	L	L	L	H	X	+5.0 V	+5.0 V	High impedance
Standby	L	L	L	H	X	X	+5.0 V	+5.0 V	High impedance

X can be either H or L.

Figure 10. Pin Functions in μPD78P352 PROM Programming Mode



L Connect these pins separately to V_{SS} through resistors.
G Connect this pin to V_{SS}.
OPEN Do not connect these pins.

83CL-9154B

PROM Byte Programming Procedure

Data can be written to the PROM one byte at a time by the following procedure.

(1) Set the pins not used for programming as indicated in figure 10. Set MODE0/V_{PP} and V_{DD} pins to +5 V and MODE1, $P2_1$, and $\overline{RESET}$ pins to 0 V. The $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins should be high.

(2) Supply +6.5 V to V_{DD} pin and +12.5 V to MODE0/V_{PP} pin. Set $\overline{CE}$ pin low and $\overline{OE}$ pin high.

(3) Provide initial address to pins A_0 - A_{16}.

(4) Provide write data.

(5) Input a 0.1-ms program pulse (active low) to $\overline{PGM}$ pin.

(6) Use verify mode (pulse $\overline{OE}$ low) to test data. If data has been written, proceed to step 8; if not, repeat steps 4–6. If data cannot be written in 10 attempts, go to step 7.

(7) Classify PROM as defective and cease write operation.

(8) Increment address.

(9) Repeat steps 4–8 until last address is programmed.

PROM Page Programming Procedure

Data can be written to the PROM four bytes at a time (page programming) by the following procedure.

(1) Set the pins not used for programming as indicated in figure 10. Set MODE0/V_{PP} and V_{DD} pins to +5 V and MODE1, $P2_1$, and $\overline{RESET}$ pins to 0 V. The $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins should be high.

(2) Supply +6.5 V to V_{DD} pin and +12.5 V to MODE0/V_{PP} pin. Set $\overline{CE}$ pin low.

(3) Provide initial page address to pins A_0 - A_{16}.

(4) Provide first byte of data and latch it into PROM by pulsing $\overline{OE}$ low. Continue incrementing address and latching in data until four bytes have been loaded.

(5) Input a 0.1-ms program pulse (active low) to $\overline{PGM}$ pin. Data bus D_0 - D_7 is in a high-impedance state.

(6) Use verify mode (pulse $\overline{OE}$ low four times) to test four bytes of data. If all four bytes of data have been written, proceed to step 8; if not, repeat steps 4–6. If data cannot be written in 10 attempts, go to step 7.

(7) Classify PROM as defective and cease write operation.

(8) Increment address.

(9) Repeat steps 4–8 until last address is programmed.

PROM Read Procedure

The contents of the PROM can be read out to the external data bus (D_0 - D_7) by the following procedure.

(1) Set the pins not used for programming as indicated in figure 10. Set MODE0/V_{PP} and V_{DD} pins to +5 V and MODE1, $P2_1$, and $\overline{RESET}$ pins to 0 V. The $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins should be high.

(2) Supply +5 V to V_{DD} pin and MODE0/V_{PP} pin.

(3) Input address of data to be read to pins A_0 - A_{16}.

(4) Put an active-low pulse on $\overline{CE}$ and $\overline{OE}$ pins.

(5) Data is output to pins D_0 - D_7.

Program Erasure

The UV EPROM can be erased by exposing the window to light having a wavelength shorter than 400 nm, including ultraviolet, direct sunlight, and fluoresecent light. To prevent unintentional erasure, mask the window.

Typically, data is erased by 254-nm ultraviolet rays. A minimum lighting level of 15 Ws/cm² (ultraviolet ray intensity x exposure time) is required to completely erase the written data. Erasure by an ultraviolet lamp rated at 12,000 μW/cm² takes 15 to 20 minutes. Remove any filter on the lamp and place the device within 2.5 cm of the lamp tubes.

DC Programming Characteristics

T_A = 25°C ±5°C; V_{SS} = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Condition
High-level input voltage	V_{IH1}	2.2		V_{DD}	V	(Note 1)
	V_{IH2}	0.8 V_{DD}		V_{DD}	V	(Note 2)
V_{DDP} power supply voltage	V_{DDP}	6.25	6.5	6.75	V	Memory program mode
		4.5	5.0	5.5	V	Memory read mode
V_{PP} power supply voltage	V_{PP}	12.2	12.5	12.8	V	Memory program mode
			V_{PP} = V_{DDP}		V	Memory read mode
V_{DDP} power supply current	I_{DDP}			30	mA	Memory program mode
				100	mA	Memory read mode
V_{PP} power supply current	I_{PP}			50	mA	Memory program mode
			1	100	μA	Memory read mode

Notes:

(1) All except pins in Note 2.

(2) Pins $\overline{RESET}$, X1, X2, $P2_n$, INTPn, NMI, and TI.

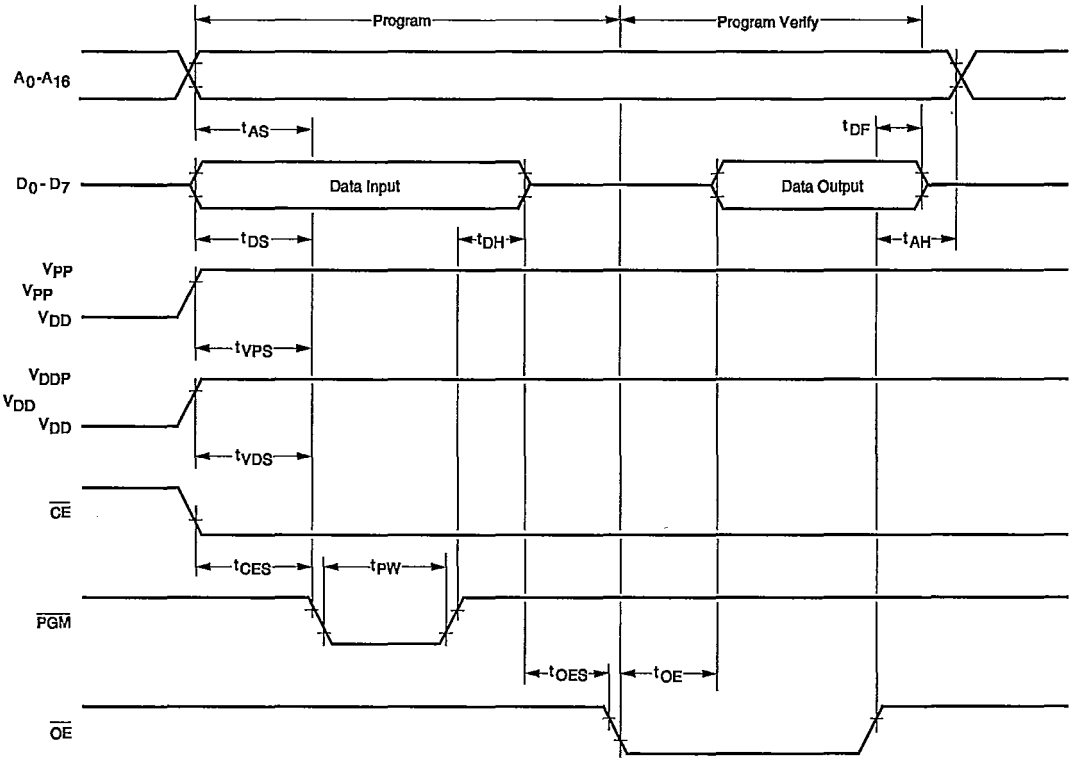
AC Programming Characteristics

T_A = 25°C ±5°C; V_{DD} = 6.5 ±0.25 V; V_{PP} = 12.5 ±0.3 V

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Byte Programming Mode						
Address setup time to $\overline{PGM}$ ↓	t_{AS}	2			μs	
$\overline{CE}$ setup time to $\overline{PGM}$ ↓	t_{CES}	2			μs	
Input data setup time to $\overline{PGM}$ ↓	t_{DS}	2			μs	
Address hold time after $\overline{OE}$ ↑	t_{AH}	2			μs	
Input data hold time after $\overline{PGM}$ ↑	t_{DH}	2			μs	
Output data hold time after $\overline{OE}$ ↑	t_{DF}	0		130	ns	
V_{PP} setup time before $\overline{PGM}$ ↓	t_{VPS}	2			μs	
V_{DD} setup time before $\overline{PGM}$ ↓	t_{VDS}	2			μs	
Program pulse width	t_{PW}	0.095	0.1	0.105	ms	
Data to $\overline{OE}$ ↓ delay time	t_{OES}	2			μs	
$\overline{OE}$ ↓ to data output time	t_{OE}			150	ns	
Page Programming Mode						
Address setup time to $\overline{OE}$ ↓	t_{AS}	2			μs	
$\overline{CE}$ setup time to $\overline{OE}$ ↓	t_{CES}	2			μs	
Input data setup time to $\overline{OE}$ ↓	t_{DS}	2			μs	
Address hold time from $\overline{OE}$ ↑	t_{AH}	2			μs	
	t_{AHL}	2			μs	
	t_{AHV}	0			μs	
Input data hold time after $\overline{OE}$ ↑	t_{DH}	2			μs	
Output data hold time after $\overline{OE}$ ↑	t_{DF}	0		130	ns	
V_{PP} setup time to $\overline{OE}$ ↓	t_{VPS}	2			μs	
V_{DD} setup time to $\overline{OE}$ ↓	t_{VDS}	2			μs	
Program pulse width	t_{PW}	0.095	0.1	0.105	ms	
Address to $\overline{OE}$ ↓ delay time	t_{OES}	2			μs	
$\overline{OE}$ ↓ to data output time	t_{OE}			150	ns	
$\overline{OE}$ pulse width during data latch	t_{LW}	1			μs	
Data to $\overline{PGM}$ ↓ delay time	t_{PGMS}	2			μs	
$\overline{CE}$ hold time from $\overline{PGM}$ ↑	t_{CEH}	2			μs	
$\overline{CE}$ hold time from $\overline{OE}$ ↑	t_{OEH}	2			μs	
Read Mode						
Address to data output time	t_{ACC}			200	ns	$\overline{CE}$ = $\overline{OE}$ = V_{IL}
$\overline{CE}$ ↓ to data output time	t_{CE}			200	ns	$\overline{OE}$ = V_{IL}
$\overline{OE}$ ↓ to data output time	t_{OE}			75	ns	$\overline{CE}$ = V_{IL}
Data hold time from $\overline{OE}$ ↑	t_{DF}	0		60	ns	$\overline{CE}$ = V_{IL}
Data hold time from address	t_{OH}	0			ns	$\overline{CE}$ = $\overline{OE}$ = V_{IL}

PROM Timing Diagrams

Byte Programming Mode



Notes:

[1] V_{DD} must be applied before V_{PP} and removed after V_{PP}.

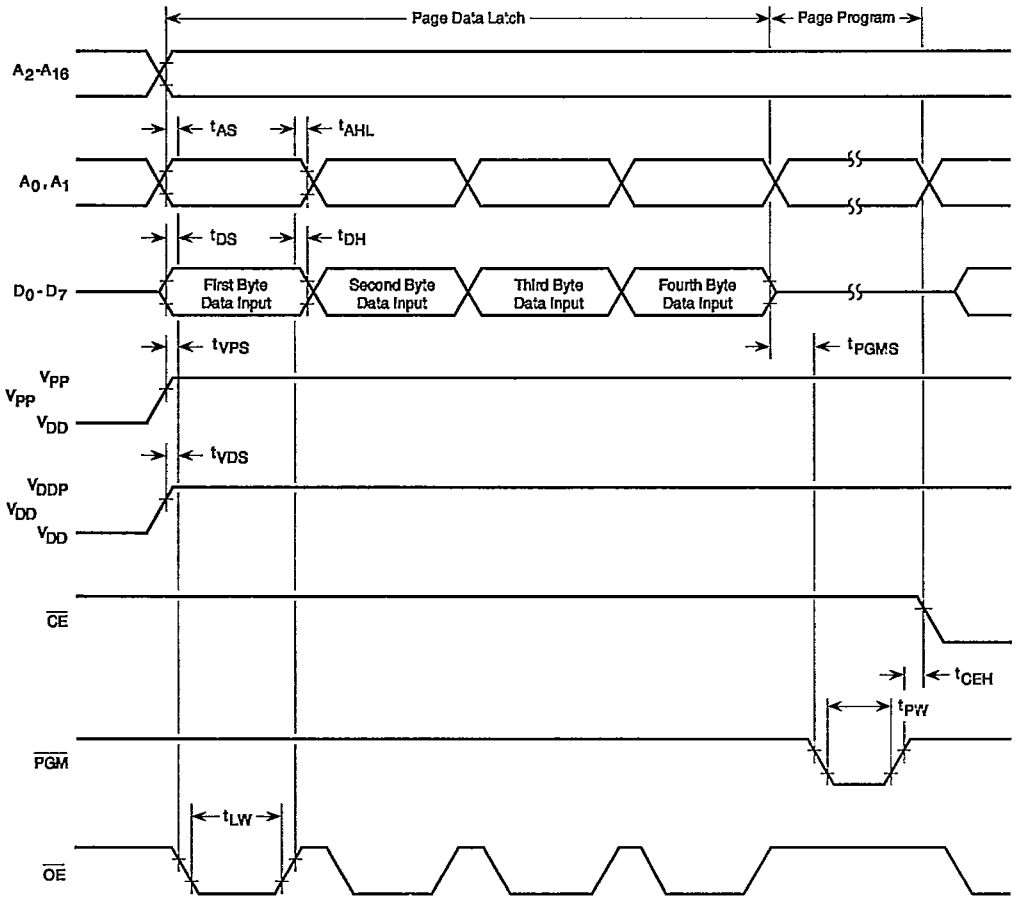
[2] V_{PP} must not be greater than +13.5 V, including overshoot.

[3] Removing and reinserting the device while a voltage of +12.5 V is applied to pin V_{PP} may affect device reliability.

8SCL-91568

PROM Timing Diagrams (cont)

Page Programming Mode; Page Data Latch → Page Program



Notes:

[1] V_{DD} must be applied before V_{PP} and removed after V_{PP}.

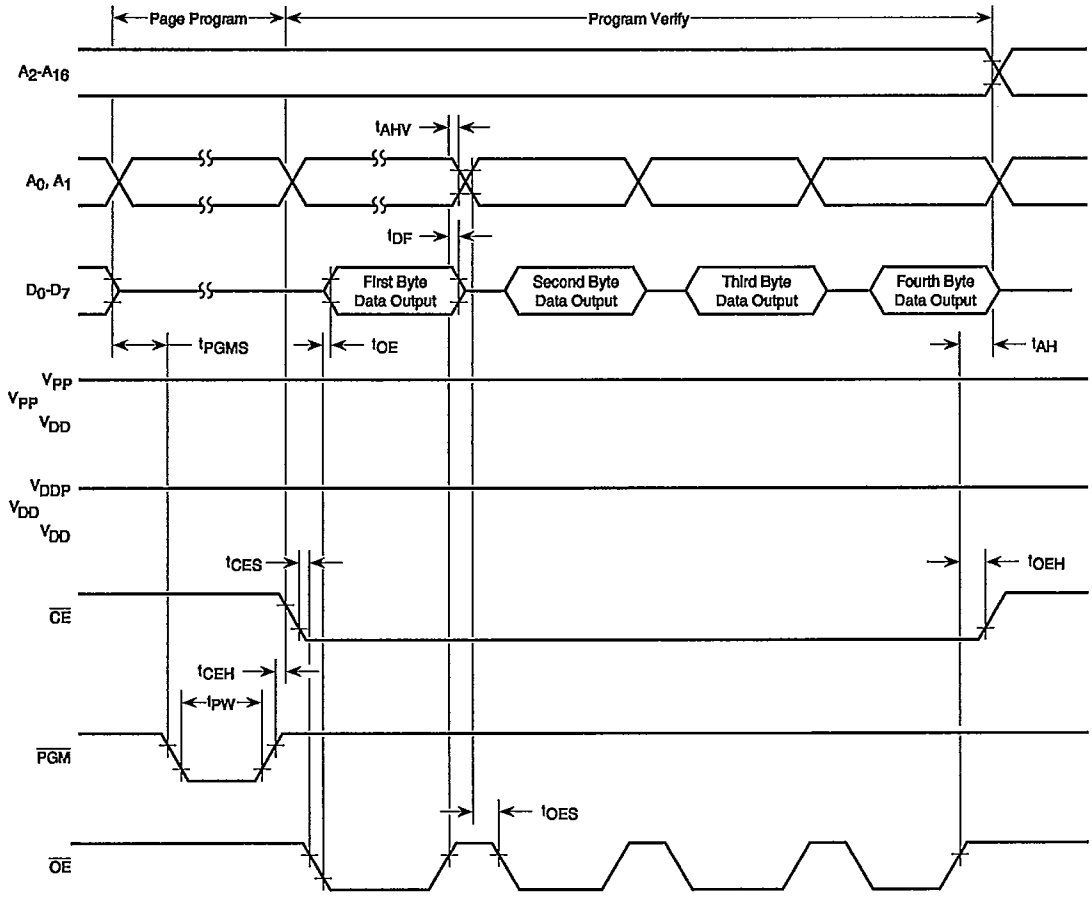
[2] V_{PP} must not be greater than +13.5 V, including overshoot.

[3] Removing and reinserting the device while a voltage of +12.5 V is applied to pin V_{PP} may affect device reliability.

83CL-915EB

PROM Timing Diagrams (cont)

Page Programming Mode; Page Program → Program Verify



Notes:

[1] V_{DD} must be applied before V_{PP} and removed after V_{PP}.

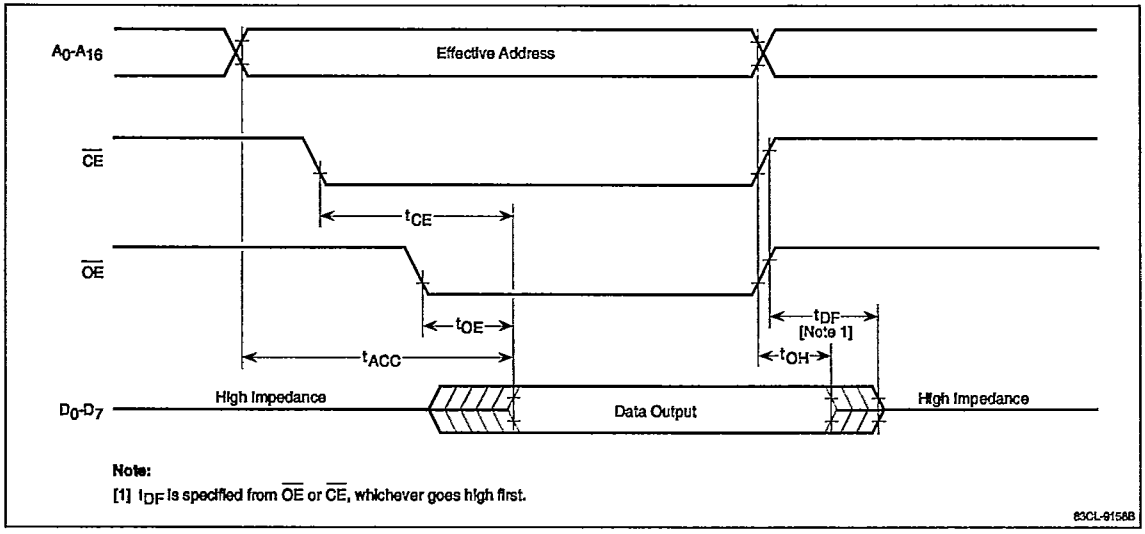
[2] V_{PP} must not be greater than +13.5 V, including overshoot.

[3] Removing and reinserting the device while a voltage of +12.5 V is applied to pin V_{PP} may affect device reliability.

83CL-9157B

PROM Timing Diagrams (cont)

Read Mode



Note:
[1] t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever goes high first.

83CL-91588

INSTRUCTION SET

The instruction set of the μPD78350/P352 is upward compatible with the μPD78322 and μPD78312A families. Two instructions have been added to facilitate digital signal processing. The convolution instruction, MACW, calculates the sum of the products of "n" pairs of terms stored in Main RAM. The value of "n" is limited only by the amount of Main RAM available. The MOVTBL instruction displaces a data table by one 16-bit word to make room for a new data word.

The instruction set features both 8- and 16-bit data transfer, arithmetic, and logic instructions and single-bit manipulation instructions. String manipulation instructions are also included. Branch instructions exist to test individual bits in the program status word, the 16-bit accumulator, the special function registers, and the saddr portion of on-chip RAM. Instructions range in length from 1 to 6 bytes depending on the instruction and addressing mode.

Flag Column Indicators

Symbol	Action
(blank)	No change
0	Set to 0
1	Set to 1
X	Set or cleared according to result
P	P/V indicates parity of result
V	P/V indicates arithmetic overflow
R	Restored from saved PSW

Instruction Set Symbols

Symbol	Definition
r	R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15
r1	R0, R1, R2, R3, R4, R5, R6, R7
r2	C, B
rp	RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7*
rp1	RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7*
rp2	DE, HL, VP, UP
sfr	Special function register, 8 bits
sfrp	Special function register, 16 bits
post	RP0, RP1, RP2, RP3, RP4, RP5/PSW, RP6, RP7. Bits set to 1 indicate register pairs to be pushed/popped to/from stack; RP5 pushed/popped by PUSH/POP, SP is stack pointer; PSW pushed/popped by PUSHU/POPU, RP5 is stack pointer.

Instruction Set Symbols (cont)

Symbol	Definition
mem	Register indirect: [DE], [HL], [DE+], [HL+], [DE–], [HL–], [VP], [UP] Base Index Mode: [DE+A], [HL+A], [DE+B], [HL+B], [VP+DE], [VP+HL] Base Mode: [DE+byte], [HL+byte], [VP+byte], [UP+byte], [SP+byte] Index Mode: word [A], word [B], word [DE], word [HL]
saddr	FE20-FF1FH: Immediate byte addresses one byte in RAM, or label
saddrp	FE20-FF1FH: Immediate byte (bit 0=0) addresses one word in RAM, or label
word	16 bits of immediate data, or label
byte	8 bits of immediate data, or label
jdisp8	8-bit two's complement displacement (immediate data, displacement value –128 to +127)
bit	3 bits of immediate data (bit position in byte), or label
n	3 bits of immediate data
!addr16	16-bit absolute address specified by an immediate address or label
$addr16	Relative branch address or label
addr16	16-bit address
!addr11	11-bit immediate address or label
addr11	0800H-0FFFH: 0800H + (11-bit immediate address), or label
addr5	0040H-007EH: 0040H + 2 X (5-bit immediate address), or label
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
AX	Register pair AX (16-bit accumulator)
BC	Register pair BC
DE	Register pair DE
HL	Register pair HL

Instruction Set Symbols (cont)

Symbol	Definition
RP0-RP7	Register pair 0 to register pair 7
PC	Program counter
SP	Stack pointer
UP	User stack pointer (RP5)
PSW	Program status word
PSWH	High-order 8 bits of PSW
PSWL	Low-order 8 bits of PSW
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 select flag
RSS	Register set select flag
IE	Interrupt enable flag
STBC	Standby control register
WDM	Watchdog timer mode register
( )	Contents of the location whose address is within parentheses; (+) and (–) indicate that the address is incremented after or decremented after it is used
(( ))	Contents of the memory location defined by the quantity within the sets of parentheses
xxH	Hexadecimal quantity
X_H, X_L	High-order 8 bits and low-order 8 bits of X
∧	Logical product (AND)
∨	Logical sum (OR)
∀	Exclusive logical sum (exclusive OR)
—	Inverted data

* rp and rp1 describe the same registers but generate different machine code.

Instruction Set

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
8-Bit Data Transfer								
MOV	r1, #byte	r1 ← byte	2					
	saddr, #byte	(saddr) ← byte	3					
	sfr, #byte (Note 1)	sfr ← byte	3					
	r, r1	r ← r1	2					
	A, r1	A ← r1	1					
	A, saddr	A ← (saddr)	2					
	saddr, A	(saddr) ← A	2					
	saddr, saddr	(saddr) ← (saddr)	3					
	A, sfr	A ← sfr	2					
	sfr, A	sfr ← A	2					
	A, mem (Note 2)	A ← (mem)	1					
	A, mem	A ← (mem)	2-4					
	mem, A (Note 2)	(mem) ← A	1					
	mem, A	(mem) ← A	2-4					
	A, [saddrp]	A ← ((saddrp))	2					
	[saddrp], A	((saddrp)) ← A	2					
	A, !addr16	A ← (addr16)	4					
	!addr16, A	(addr16) ← A	4					
	PSWL, #byte	PSWL ← byte	3	X	X	X	X	X
	PSWH, #byte	PSWH ← byte	3					
	PSWL, A	PSWL ← A	2	X	X	X	X	X
	PSWH, A	PSWH ← A	2					
	A, PSWL	A ← PSWL	2					
	A, PSWH	A ← PSWH	2					
XCH	A, r1	A ↔ r1	1					
	r, r1	r ↔ r1	2					
	A, mem	A ↔ (mem)	2-4					
	A, saddr	A ↔ (saddr)	2					
	A, sfr	A ↔ sfr	3					
	A, [saddrp]	A ↔ ((saddrp))	2					
	saddr, saddr	(saddr) ↔ (saddr)	3					
16-Bit Data Transfer								
MOVW	rp1, #word	rp1 ← word	3					
	saddrp, #word	(saddrp) ← word	4					
	sfrp, #word	sfrp ← word	4					
	rp, rp1	rp ← rp1	2					
	AX, saddrp	AX ← (saddrp)	2					
	saddrp, AX	(saddrp) ← AX	2					
	saddrp, saddrp	(saddrp) ← (saddrp)	3					

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
16-Bit Data Transfer (cont)								
MOVW (cont)	AX, sfrp	AX ← sfrp	2					
	sfrp, AX	sfrp ← AX	2					
	rp1, !addr16	rp1 ← (addr16)	4					
	!addr16, rp1	(addr16) ← rp1	4					
	AX, mem	AX ← (mem)	2-4					
	mem, AX	(mem) ← AX	2-4					
XCHW	AX, saddrp	AX ↔ (saddrp)	2					
	AX, sfrp	AX ↔ sfrp	3					
	saddrp, saddrp	(saddrp) ↔ (saddrp)	3					
	rp, rp1	rp ↔ rp1	2					
	AX, mem	AX ↔ (mem)	2-4					
8-Bit Arithmetic								
ADD	A, #byte	A, CY ← A + byte	2	X	X	X	V	X
	saddr, #byte	(saddr), CY ← (saddr) + byte	3	X	X	X	V	X
	sfr, #byte	sfr, CY ← sfr + byte	4	X	X	X	V	X
	r, r1	r, CY ← r + r1	2	X	X	X	V	X
	A, saddr	A, CY ← A + (saddr)	2	X	X	X	V	X
	A, sfr	A, CY ← A + sfr	3	X	X	X	V	X
	saddr, saddr	(saddr), CY ← (saddr) + (saddr)	3	X	X	X	V	X
	A, mem	A, CY ← A + (mem)	2-4	X	X	X	V	X
	mem, A	(mem), CY ← (mem) + A	2-4	X	X	X	V	X
ADDC	A, #byte	A, CY ← A + byte + CY	2	X	X	X	V	X
	saddr, #byte	(saddr), CY ← (saddr) + byte + CY	3	X	X	X	V	X
	sfr, #byte	sfr, CY ← sfr + byte + CY	4	X	X	X	V	X
	r, r1	r, CY ← r + r1 + CY	2	X	X	X	V	X
	A, saddr	A, CY ← A + (saddr) + CY	2	X	X	X	V	X
	A, sfr	A, CY ← A + sfr + CY	3	X	X	X	V	X
	saddr, saddr	(saddr), CY ← (saddr) + (saddr) + CY	3	X	X	X	V	X
	A, mem	A, CY ← A + (mem) + CY	2-4	X	X	X	V	X
	mem, A	(mem), CY ← (mem) + A + CY	2-4	X	X	X	V	X
SUB	A, #byte	A, CY ← A – byte	2	X	X	X	V	X
	saddr, #byte	(saddr), CY ← (saddr) – byte	3	X	X	X	V	X
	sfr, #byte	sfr, CY ← sfr – byte	4	X	X	X	V	X
	r, r1	r, CY ← r – r1	2	X	X	X	V	X
	A, saddr	A, CY ← A – (saddr)	2	X	X	X	V	X
	A, sfr	A, CY ← A – sfr	3	X	X	X	V	X
	saddr, saddr	(saddr), CY ← (saddr) – (saddr)	3	X	X	X	V	X
	A, mem	A, CY ← A – (mem)	2-4	X	X	X	V	X
	mem, A	(mem), CY ← (mem) – A	2-4	X	X	X	V	X

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
8-Bit Arithmetic (cont)								
SUBC	A, #byte	A, CY ← A – byte – CY	2	X	X	X	V	X
	saddr, #byte	(saddr), CY ← (saddr) – byte – CY	3	X	X	X	V	X
	sfr, #byte	sfr, CY ← sfr – byte – CY	4	X	X	X	V	X
	r, r1	r, CY ← r – r1 – CY	2	X	X	X	V	X
	A, saddr	A, CY ← A – (saddr) – CY	2	X	X	X	V	X
	A, sfr	A, CY ← A – sfr – CY	3	X	X	X	V	X
	saddr, saddr	(saddr), CY ← (saddr) – (saddr) – CY	3	X	X	X	V	X
	A, mem	A, CY ← A – (mem) – CY	2-4	X	X	X	V	X
	mem, A	(mem), CY ← (mem) – A – CY	2-4	X	X	X	V	X
8-Bit Logic								
AND	A, #byte	A ← A ∧ byte	2	X	X		P	
	saddr, #byte	(saddr) ← (saddr) ∧ byte	3	X	X		P	
	sfr, #byte	sfr ← sfr ∧ byte	4	X	X		P	
	r, r1	r ← r ∧ r1	2	X	X		P	
	A, saddr	A ← A ∧ (saddr)	2	X	X		P	
	A, sfr	A ← A ∧ sfr	3	X	X		P	
	saddr, saddr	(saddr)← (saddr) ∧ (saddr)	3	X	X		P	
	A, mem	A ← A ∧ (mem)	2-4	X	X		P	
	mem, A	(mem) ← (mem) ∧ A	2-4	X	X		P	
OR	A, #byte	A ← A ∨ byte	2	X	X		P	
	saddr, #byte	(saddr) ← (saddr) ∨ byte	3	X	X		P	
	sfr, #byte	sfr ← sfr ∨ byte	4	X	X		P	
	r, r1	r ← r ∨ r1	2	X	X		P	
	A, saddr	A ← A ∨ (saddr)	2	X	X		P	
	A, sfr	A ← A ∨ sfr	3	X	X		P	
	saddr, saddr	(saddr)← (saddr) ∨ (saddr)	3	X	X		P	
	A, mem	A ← A ∨ (mem)	2-4	X	X		P	
	mem, A	(mem) ← (mem) ∨ A	2-4	X	X		P	
XOR	A, #byte	A ← A ⊻ byte	2	X	X		P	
	saddr, #byte	(saddr) ← (saddr) ⊻ byte	3	X	X		P	
	sfr, #byte	sfr ← sfr ⊻ byte	4	X	X		P	
	r, r1	r ← r ⊻ r1	2	X	X		P	
	A, saddr	A ← A ⊻ (saddr)	2	X	X		P	
	A, sfr	A ← A ⊻ sfr	3	X	X		P	
	saddr, saddr	(saddr)← (saddr) ⊻ (saddr)	3	X	X		P	
	A, mem	A ← A ⊻ (mem)	2-4	X	X		P	
	mem, A	(mem) ← (mem) ⊻ A	2-4	X	X		P	

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
8-Bit Logic (cont)								
CMP	A, #byte	A – byte	2	X	X	X	V	X
	saddr, #byte	(saddr) – byte	3	X	X	X	V	X
	sfr, #byte	sfr – byte	4	X	X	X	V	X
	r, r1	r – r1	2	X	X	X	V	X
	A, saddr	A – (saddr)	2	X	X	X	V	X
	A, sfr	A – sfr	3	X	X	X	V	X
	saddr, saddr	(saddr) – (saddr)	3	X	X	X	V	X
	A, mem	A – (mem)	2-4	X	X	X	V	X
	mem, A	(mem) – A	2-4	X	X	X	V	X
16-Bit Arithmetic								
ADDW	AX, #word	AX, CY ← AX + word	3	X	X	X	V	X
	saddrp, #word	(saddrp), CY ← (saddrp) + word	4	X	X	X	V	X
	sfrp, #word	sfrp, CY ← sfrp + word	5	X	X	X	V	X
	rp, rp1	rp, CY ← rp + rp1	2	X	X	X	V	X
	AX, saddrp	AX, CY ← AX + (saddrp)	2	X	X	X	V	X
	AX, sfrp	AX, CY ← AX + sfrp	3	X	X	X	V	X
	saddrp, saddrp	(saddrp), CY ← (saddrp) + (saddrp)	3	X	X	X	V	X
SUBW	AX, #word	AX, CY ← AX – word	3	X	X	X	V	X
	saddrp, #word	(saddrp), CY ← (saddrp) – word	4	X	X	X	V	X
	sfrp, #word	sfrp, CY ← sfrp – word	5	X	X	X	V	X
	rp, rp1	rp, CY ← rp – rp1	2	X	X	X	V	X
	AX, saddrp	AX, CY ← AX – (saddrp)	2	X	X	X	V	X
	AX, sfrp	AX, CY ← AX – sfrp	3	X	X	X	V	X
	saddrp, saddrp	(saddrp), CY ← (saddrp) – (saddrp)	3	X	X	X	V	X
CMPW	AX, #word	AX – word	3	X	X	X	V	X
	saddrp, #word	(saddrp) – word	4	X	X	X	V	X
	sfrp, #word	sfrp – word	5	X	X	X	V	X
	rp, rp1	rp – rp1	2	X	X	X	V	X
	AX, saddrp	AX – (saddrp)	2	X	X	X	V	X
	AX, sfrp	AX – sfrp	3	X	X	X	V	X
	saddrp, saddrp	(saddrp) – (saddrp)	3	X	X	X	V	X
Multiplication/Division								
MULU	r1	AX ← AX x r1	2					
DIVUW	r1	AX (quotient), r1 (remainder) ← AX ÷ r1	2					
MULUW	rp1	AX (high-order 16 bits), rp1 (low-order 16 bits) ← AX x rp1	2					
DIVUX	rp1	AXDE (quotient), rp1 (remainder) ← AXDE ÷ rp1	2					
MULW (Note 3)	rp1	AX (high-order 16 bits), rp1 (low-order 16 bits) ← AX x rp1	2					

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Sum-of-Products								
MACW	n	AXDE ← (B) x (C) + AXDE, B ← B + 2, C ← C + 2, n ← n–1. End if n = 0 or P/V = 1	3	X	X	X	V	X
Table Shift								
MOVTBLW	!addr 16, n (Note 4)	(addr16 + 2) ← (addr16), n ← n–1, addr16 ← addr16 – 2. End if n = 0	4					
Incrememt/Decrement								
INC	r1	r1 ← r1 + 1	1	X	X	X	V	
	saddr	(saddr) ← (saddr) + 1	2	X	X	X	V	
DEC	r1	r1 ← r1 – 1	1	X	X	X	V	
	saddr	(saddr) ← (saddr) – 1	2	X	X	X	V	
INCW	rp2	rp2 ← rp2 + 1	1					
	saddrp	(saddrp) ← (saddrp) + 1	3					
DECW	rp2	rp2 ← rp2 – 1	1					
	saddrp	(saddrp) ← (saddrp) – 1	3					
Shift/Rotate								
ROR	r1, n	(CY, $r1_7 \leftarrow r1_0$, $r1_{m-1} \leftarrow r1_m$) x n times	2				P	X
ROL	r1, n	(CY, $r1_0 \leftarrow r1_7$, $r1_{m+1} \leftarrow r1_m$) x n times	2				P	X
RORC	r1, n	(CY ← $r1_0$, $r1_7$ ← CY, $r1_{m-1} \leftarrow r1_m$) x n times	2				P	X
ROLC	r1, n	(CY ← $r1_7$, $r1_0$ ← CY, $r1_{m+1} \leftarrow r1_m$) x n times	2				P	X
SHR	r1, n	(CY ← $r1_0$, $r1_7$ ← 0, $r1_{m-1} \leftarrow r1_m$) x n times	2	X	X	0	P	X
SHL	r1, n	(CY ← $r1_7$, $r1_0$ ← 0, $r1_{m+1} \leftarrow r1_m$) x n times	2	X	X	0	P	X
SHRW	rp1, n	(CY ← $rp1_0$, $rp1_{15}$ ← 0, $rp1_{m-1} \leftarrow rp1_m$) x n times	2	X	X	0	P	X
SHLW	rp1, n	(CY ← $rp1_{15}$, $rp1_0$ ← 0, $rp1_{m+1} \leftarrow rp1_m$) x n times	2	X	X	0	P	X
ROR4	[rp1]	$A_{3-0} \leftarrow (rp1)_{3-0}$, $(rp1)_{7-4} \leftarrow A_{3-0}$, $(rp1)_{3-0} \leftarrow (rp1)_{7-4}$	2					
ROL4	[rp1]	$A_{3-0} \leftarrow (rp1)_{7-4}$, $(rp1)_{3-0} \leftarrow A_{3-0}$, $(rp1)_{7-4} \leftarrow (rp1)_{3-0}$	2					
BCD Adjustment								
ADJBA		Decimal adjust accumulator after add	2	X	X	X	P	X
ADJBS		Decimal adjust accumulator after subtract	2	X	X	X	P	X
Data Expansion								
CVTBW		X ← A, $A_{6-0} \leftarrow A_7$	1					
Bit Manipulation								
MOV1	CY, saddr.bit	CY ← (saddr.bit)	3					X
	CY, sfr.bit	CY ← sfr.bit	3					X
	CY, A.bit	CY ← A.bit	2					X
	CY, X.bit	CY ← X.bit	2					X
	CY, PSWH.bit	CY ← PSWH.bit	2					X

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Bit Manipulation (cont)								
MOV1 (cont)	CY, PSWL.bit	CY ← PSWL.bit	2					X
	saddr.bit, CY	(saddr.bit) ← CY	3					
	sfr.bit, CY	sfr.bit ← CY	3					
	A.bit, CY	A.bit ← CY	2					
	X.bit, CY	X.bit ← CY	2					
	PSWH.bit, CY	PSWH.bit ← CY	2					
	PSWL.bit, CY	PSWL.bit ← CY	2	X	X	X	X	
AND1	CY, saddr.bit	CY ← CY ∧ (saddr.bit)	3					X
	CY, /saddr.bit	CY ← CY ∧ $\overline{\text{(saddr.bit)}}$	3					X
	CY, sfr.bit	CY ← CY ∧ sfr.bit	3					X
	CY, /sfr.bit	CY ← CY ∧ $\overline{\text{sfr.bit}}$	3					X
	CY, A.bit	CY ← CY ∧ A.bit	2					X
	CY, /A.bit	CY ← CY ∧ $\overline{\text{A.bit}}$	2					X
	CY, X.bit	CY ← CY ∧ X.bit	2					X
	CY, /X.bit	CY ← CY ∧ $\overline{\text{X.bit}}$	2					X
	CY, PSWH.bit	CY ← CY ∧ PSWH.bit	2					X
	CY, /PSWH.bit	CY ← CY ∧ $\overline{\text{PSWH.bit}}$	2					X
	CY, PSWL.bit	CY ← CY ∧ PSWL.bit	2					X
	CY, /PSWL.bit	CY ← CY ∧ $\overline{\text{PSWL.bit}}$	2					X
OR1	CY, saddr.bit	CY ← CY ∨ (saddr.bit)	3					X
	CY, /saddr.bit	CY ← CY ∨ $\overline{\text{(saddr.bit)}}$	3					X
	CY, sfr.bit	CY ← CY ∨ sfr.bit	3					X
	CY, /sfr.bit	CY ← CY ∨ $\overline{\text{sfr.bit}}$	3					X
	CY, A.bit	CY ← CY ∨ A.bit	2					X
	CY, /A.bit	CY ← CY ∨ $\overline{\text{A.bit}}$	2					X
	CY, X.bit	CY ← CY ∨ X.bit	2					X
	CY, /X.bit	CY ← CY ∨ $\overline{\text{X.bit}}$	2					X
	CY, PSWH.bit	CY ← CY ∨ PSWH.bit	2					X
	CY, /PSWH.bit	CY ← CY ∨ $\overline{\text{PSWH.bit}}$	2					X
	CY, PSWL.bit	CY ← CY ∨ PSWL.bit	2					X
	CY, /PSWL.bit	CY ← CY ∨ $\overline{\text{PSWL.bit}}$	2					X
XOR1	CY, saddr.bit	CY ← CY ⊻ (saddr.bit)	3					X
	CY, sfr.bit	CY ← CY ⊻ sfr.bit	3					X
	CY, A.bit	CY ← CY ⊻ A.bit	2					X
	CY, X.bit	CY ← CY ⊻ X.bit	2					X
	CY, PSWH.bit	CY ← CY ⊻ PSWH.bit	2					X
	CY, PSWL.bit	CY ← CY ⊻ PSWL.bit	2					X

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Bit Manipulation (cont)								
SET1	saddr.bit	(saddr.bit) ← 1	2					
	sfr.bit	sfr.bit ← 1	3					
	A.bit	A.bit ← 1	2					
	X.bit	X.bit ← 1	2					
	PSWH.bit	PSWH.bit ← 1	2					
	PSWL.bit	PSWL.bit ← 1	2	X	X	X	X	X
CLR1	saddr.bit	(saddr.bit) ← 0	2					
	sfr.bit	sfr.bit ← 0	3					
	A.bit	A.bit ← 0	2					
	X.bit	X.bit ← 0	2					
	PSWH.bit	PSWH.bit ← 0	2					
	PSWL.bit	PSWL.bit ← 0	2	X	X	X	X	X
NOT1	saddr.bit	(saddr.bit) ← $\overline{\text{(saddr.bit)}}$	3					
	sfr.bit	sfr.bit ← $\overline{\text{sfr.bit}}$	3					
	A.bit	A.bit ← $\overline{\text{A.bit}}$	2					
	X.bit	X.bit ← $\overline{\text{X.bit}}$	2					
	PSWH.bit	PSWH.bit ← $\overline{\text{PSWH.bit}}$	2					
	PSWL.bit	PSWL.bit ← $\overline{\text{PSWL.bit}}$	2	X	X	X	X	X
SET1	CY	CY ← 1	1					1
CLR1	CY	CY ← 0	1					0
NOT1	CY	CY ← $\overline{\text{CY}}$	1					X
Subroutine Linkage								
CALL	!addr16	$(SP-1) \leftarrow (PC + 3)_H$, $(SP - 2) \leftarrow (PC + 3)_L$, PC ← addr16, SP ← SP – 2	3					
	rp1	$(SP-1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, $PC_H \leftarrow rp1_H$, $PC_L \leftarrow rp1_L$, SP ← SP – 2	2					
	[rp1]	$(SP-1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, $PC_H \leftarrow (rp1 + 1)$, $PC_L \leftarrow (rp1)$, SP ← SP – 2	2					
CALLF	!addr11	$(SP-1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, SP ← SP – 2	2					
CALLT	[addr5]	$(SP-1) \leftarrow (PC + 1)_H$, $(SP - 2) \leftarrow (PC + 1)_L$, $PC_H \leftarrow$ (TPFx8000H + 2 x addr5 + 41H), $PC_L \leftarrow$ (TPFx8000H + 2 x addr5 + 40H), SP ← SP – 2	1					
BRK		(SP–1) ← PSWH, (SP – 2) ← PSWL, $(SP - 3) \leftarrow (PC + 1)_H$, $(SP - 4) \leftarrow (PC + 1)_L$, $PC_L \leftarrow$ (003EH), $PC_H \leftarrow$ (003FH), SP ← SP – 4, IE ← 0	1					
RET		$PC_L \leftarrow$ (SP), $PC_H \leftarrow$ (SP + 1), SP ← SP + 2	1					
RETB		$PC_L \leftarrow$ (SP), $PC_H \leftarrow$ (SP + 1), PSWL ← (SP + 2), PSWH ← (SP + 3), SP ← SP + 4	1	R	R	R	R	R
RETI		$PC_L \leftarrow$ (SP), $PC_H \leftarrow$ (SP + 1), PSWL ← (SP + 2), PSWH ← (SP + 3), SP ← SP + 4	1	R	R	R	R	R

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Stack Manipulation								
PUSH	sfrp	(SP – 1) ← sfr_H, (SP – 2) ← sfr_L, SP ← SP – 2	3					
	post	{(SP – 1) ← rpp_H, (SP – 2) ← rpp_L, SP ← SP – 2} x n (Note 5)	2					
	PSW	(SP – 1) ← PSWH, (SP – 2) ← PSWL, SP ← SP – 2	1					
PUSHU	post	{(UP – 1) ← rpp_H, (UP – 2) ← rpp_L, UP ← UP – 2} x n (Note 5)	2					
POP	sfrp	sfr_L ← (SP), sfr_H ← (SP + 1), SP ← SP + 2	3					
	post	{rpp_L ← (SP), rpp_H ← (SP + 1), SP ← SP + 2} x n (Note 5)	2					
	PSW	PSWL ← (SP), PSWH ← (SP + 1), SP ← SP + 2	1	R	R	R	R	R
POPU	post	{rpp_L ← (UP), rpp_H ← (UP + 1), UP ← UP + 2} x n (Note 5)	2					
MOVW	SP, #word	SP ← word	4					
	SP, AX	SP ← AX	2					
	AX, SP	AX ← SP	2					
INCW	SP	SP ← SP + 1	2					
DECW	SP	SP ← SP – 1	2					
Pin Level Test								
CHKL	sfr	(Pin level) $\overline{\vee}$ (internal signal level)	3	X	X		P	
CHKLA	sfr	A ← (Pin level) $\overline{\vee}$ (internal signal level)	3	X	X		P	
Unconditional Branch								
BR	!addr16	PC ← addr16	3					
	rp1	PC_H ← $rp1_H$, PC_L ← $rp1_L$	2					
	[rp1]	PC_H ← (rp1 + 1), PC_L ← (rp1)	2					
	$addr16	PC ← PC + 2 + jdisp8	2					
Conditional Branch								
BC, BL	$addr16	PC ← PC + 2 + jdisp8 if CY = 1	2					
BNC, BNL	$addr16	PC ← PC + 2 + jdisp8 if CY = 0	2					
BZ, BE	$addr16	PC ← PC + 2 + jdisp8 if Z = 1	2					
BNZ, BNE	$addr16	PC ← PC + 2 + jdisp8 if Z = 0	2					
BV, BPE	$addr16	PC ← PC + 2 + jdisp8 if P/V = 1	2					
BNV, BPO	$addr16	PC ← PC + 2 + jdisp8 if P/V = 0	2					
BN	$addr16	PC ← PC + 2 + jdisp8 if S = 1	2					
BP	$addr16	PC ← PC + 2 + jdisp8 if S = 0	2					
BGT	$addr16	PC ← PC + 3 + jdisp8 if (P/V $\overline{\vee}$ S) $\vee$ Z = 0	3					
BGE	$addr16	PC ← PC + 3 + jdisp8 if P/V $\overline{\vee}$ S = 0	3					
BLT	$addr16	PC ← PC + 3 + jdisp8 if P/V $\overline{\vee}$ S = 1	3					
BLE	$addr16	PC ← PC + 3 + jdisp8 if (P/V $\overline{\vee}$ S) $\vee$ Z = 1	3					
BH	$addr16	PC ← PC + 3 + jdisp8 if Z $\vee$ CY = 0	3					
BNH	$addr16	PC ← PC + 3 + jdisp8 if Z $\vee$ CY = 1	3					

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Conditional Branch								
BT	saddr.bit, $addr16	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	3					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 1	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 1	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 1	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 1	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 1	3					
BF	saddr.bit, $addr16	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	4					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 0	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 0	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 0	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 0	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 0	3					
BTCLR	saddr.bit, $addr16	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	4					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 1 then reset X.bit	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 1 then reset PSWH.bit	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 1 then reset PSWL.bit	3	X	X	X	X	X
BFSET	saddr.bit, $addr16	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0 then set (saddr.bit)	4					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 0 then set sfr.bit	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 0 then set A.bit	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 0 then set X.bit	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 0 then set PSWH.bit	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 0 then set PSWL.bit	3	X	X	X	X	X
DBNZ	r2, $addr16	r2 ← r2 – 1, then PC ← PC + 2 + jdisp8 if r2 = 0	2					
	saddr, $addr16	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) = 0	3					
Context Switching								
BRKCS	RBn	$RBS_{2\text{-}0}$ ← n, PC_H ↔ R5, PC_L ↔ R4, R7 ← PSWH, R6 ← PSWL, RSS ← 0, IE ← 0	2					
RETCS	!addr16	PC_H ← R5, PC_L ← R4, R5 ← $addr16_H$, R4 ← $addr16_L$, PSWH ← R7, PSWL ← R6	3	R	R	R	R	R
RETCSB	!addr16	PC_H ← R5, PC_L ← R4, R5 ← $addr16_H$, R4 ← $addr16_L$, PSWH ← R7, PSWL ← R6	4	R	R	R	R	R

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
String Manipulation								
MOVM	[DE+], A	(DE+) ← A, C ← C–1 End if C = 0	2					
	[DE–], A	(DE–) ← A, C ← C–1 End if C = 0	2					
MOVBK	[DE+], [HL+]	(DE+) ← (HL+), C ← C–1 End if C = 0	2					
	[DE–], [HL–]	(DE–) ← (HL–), C ← C–1 End if C = 0	2					
XCHM	[DE+], A	(DE+) ↔ A, C ← C–1 End if C = 0	2					
	[DE–], A	(DE–) ↔ A, C ← C–1 End if C = 0	2					
XCHBK	[DE+], [HL+]	(DE+) ↔ (HL+), C ← C–1 End if C = 0	2					
	[DE–], [HL–]	(DE–) ↔ (HL–), C ← C–1 End if C = 0	2					
CMPME	[DE+], A	(DE+) – A, C ← C–1 End if C = 0 or Z = 0	2	X	X	X	V	X
	[DE–], A	(DE–) – A, C ← C–1 End if C = 0 or Z = 0	2	X	X	X	V	X
CMPBKE	[DE+], [HL+]	(DE+) – (HL+), C ← C–1 End if C = 0 or Z = 0	2	X	X	X	V	X
	[DE–], [HL–]	(DE–) – (HL–), C ← C–1 End if C = 0 or Z = 0	2	X	X	X	V	X
CMPMNE	[DE+], A	(DE+) – A, C ← C–1 End if C = 0 or Z = 1	2	X	X	X	V	X
	[DE–], A	(DE–) – A, C ← C–1 End if C = 0 or Z = 1	2	X	X	X	V	X
CMPBKNE	[DE+], [HL+]	(DE+) – (HL+), C ← C–1 End if C = 0 or Z = 1	2	X	X	X	V	X
	[DE–], [HL–]	(DE–) – (HL–), C ← C–1 End if C = 0 or Z = 1	2	X	X	X	V	X
CMPMC	[DE+], A	(DE+) – A, C ← C–1 End if C = 0 or CY = 0	2	X	X	X	V	X
	[DE–], A	(DE–) – A, C ← C–1 End if C = 0 or CY = 0	2	X	X	X	V	X
CMPBKC	[DE+], [HL+]	(DE+) – (HL+), C ← C–1 End if C = 0 or CY = 0	2	X	X	X	V	X
	[DE–], [HL–]	(DE–) – (HL–), C ← C–1 End if C = 0 or CY = 0	2	X	X	X	V	X
CMPMNC	[DE+], A	(DE+) – A, C ← C–1 End if C = 0 or CY = 1	2	X	X	X	V	X
	[DE–], A	(DE–) – A, C ← C–1 End if C = 0 or CY = 1	2	X	X	X	V	X
CMPBKNC	[DE+], [HL+]	(DE+) – (HL+), C ← C–1 End if C = 0 or CY = 1	2	X	X	X	V	X
	[DE–], [HL–]	(DE–) – (HL–), C ← C–1 End if C = 0 or CY = 1	2	X	X	X	V	X
CPU Control								
MOV	STBC, #byte	STBC ← byte (Note 6)	4					
	WDM, #byte	WDM ← byte (Note 6)	4					
SWRS		RSS ← $\overline{RSS}$	1					
SEL	RBn	RBS_{2-0} ← n, RSS ← 0	2					
	RBn, ALT	RBS_{2-0} ← n, RSS ← 1	2					
NOP		No operation	1					
EI		IE ← 1 (Enable interrupt)	1					
DI		IE ← 0 (Disable interrupt)	1					

Instruction Set (cont)

Notes:

(1) A special instruction is used to write to STBC and WDM.

(2) One byte move instruction when [DE], [HL], [DE+], [DE–], [HL+], or [HL–] is specified for mem.

(3) 16-bit signed multiply instruction

(4) Addressing range is 0FE00H to 0FEFFH.

(5) rpp refers to register pairs specified in post byte. "n" is the number of register pairs specified in post byte.

(6) Trap if data bytes in operation code are not one's complement. If trap, then:
(SP–1) ← PSWH, (SP–2) ← PSWL, (SP–3) ← $(PC–4)_H$,
(SP–4) ← $(PC–4)_L$, PC_L ← (003CH), PC_H ← (003DH).
SP ← SP–4, IE ← 0.

SOLDERING

Packaging and Soldering Information

Part Number	Package	Package Drawing	Recommended Soldering Code (Note 1)
μPD78350GC-3BE	64-pin plastic QFP (3.0-mm height)	P64GC-80-3BE	IR30-107-1 VP15-107-1 WS60-107-1
μPD78P352AG-xxx-22	64-pin plastic QFP (1.7-mm height)	P64G-80-22-1	Contact NEC
μPD78P352G-22			IR30-107-2 VP15-107-2
μPD78P352KK	64-pin ceramic LCC with window	X80KW-80B	Not intended for soldering

Soldering Conditions

Method (Note 1)	Code (Note 2)	Soldering Conditions	Exposure Limit (Note 3)
Infrared ray reflow	IR30-107-1 IR30-107-2	Package peak temp: 230°C Time: 30 sec (210°C min)	Max no. of days: 7 (thereafter, 10 hours baking at 125°C is required)
Vapor phase reflow	VP15-107-1 VP15-107-2	Package peak temp: 215°C Time: 40 sec (200°C min)	Max no. of days: 7 (thereafter, 10 hours baking at 125°C is required)
Wave soldering	WS60-107-1	Solder bath temp: 260°C max Time: 10 sec max	
Pin partial heating	—	Temperature: 300°C max Time: 3 sec max (per device side)	

Notes:

(1) Do not use different soldering methods together. However, on all devices, the pin partial heating method can be used alone or in combination with other soldering methods.

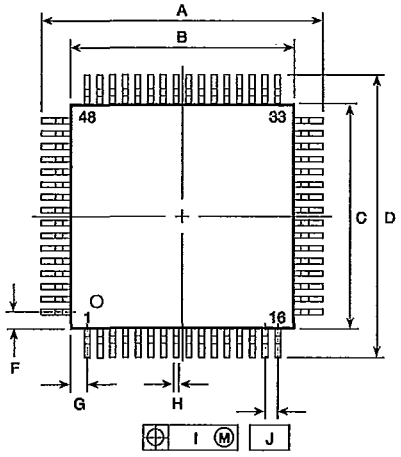
(2) The maximum number of soldering operations is one or two as indicated by the last digit of the soldering code: -1 or -2

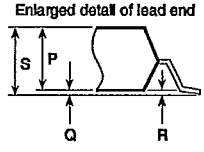
(3) Maximum no. of days refers to the number of days after unpacking the dry pack. Storage conditions are 25°C and 65% RH max.

PACKAGE DRAWINGS

64-Pin Plastic QFP (3.0-mm height) (Dwg No. P64GC-80-3BE)

Item	Millimeters	Inches
A	17.6 ± 0.4	.693 ± .016
B	14.0 ± 0.2	.551 $^{+.009}_{-.008}$
C	14.0 ± 0.2	.551 $^{+.009}_{-.008}$
D	17.6 ± 0.4	.693 ± .016
F	1.0	.039
G	1.0	.039
H	0.35 ± 0.10	.014 $^{+.004}_{-.005}$
I	0.15	.006
J	0.8 (TP)	.031 (TP)
K	1.8 ± 0.2	.071 ± .008
L	0.8 ± 0.2	.031 $^{+.009}_{-.008}$
M	0.15 $^{+0.10}_{-0.05}$	.006 $^{+.004}_{-.003}$
N	0.15	.006
P	2.7	.106
Q	0.1 ± 0.1	.004 ± .004
R	0.1 ± 0.1	.004 ± .004
S	3.0 max	.118 max



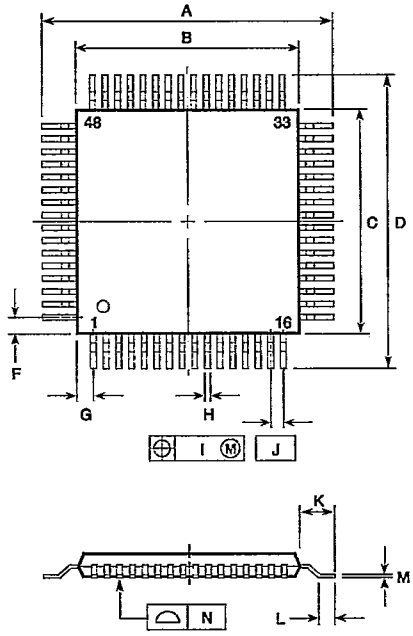


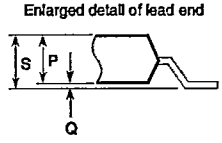
P64GC-80-3BE

83CL-91598 (12/92)

64-Pin Plastic QFP (1.7-mm height) (Dwg No. P64G-80-22-1)

Item	Millimeters	Inches
A	18.4 ± 0.4	.724 $^{+.017}_{-.016}$
B	14.0 ± 0.2	.551 $^{+.009}_{-.008}$
C	14.0 ± 0.2	.551 $^{+.009}_{-.008}$
D	18.4 ± 0.4	.724 $^{+.017}_{-.016}$
F	1.0	.039
G	1.0	.039
H	0.35 ± 0.10	.014 $^{+.004}_{-.005}$
I	0.15	.006
J	0.8 (TP)	.031 (TP)
K	2.2 ± 0.2	.087 $^{+.008}_{-.009}$
L	1.0 ± 0.2	.039 $^{+.009}_{-.008}$
M	0.15 $^{+0.10}_{-0.05}$	.006 $^{+.004}_{-.003}$
N	0.15	.006
P	1.5 ± 0.1	.059 ± .004
Q	0.0 ± 0.1	.000 ± .004
S	1.7 max	.067 max



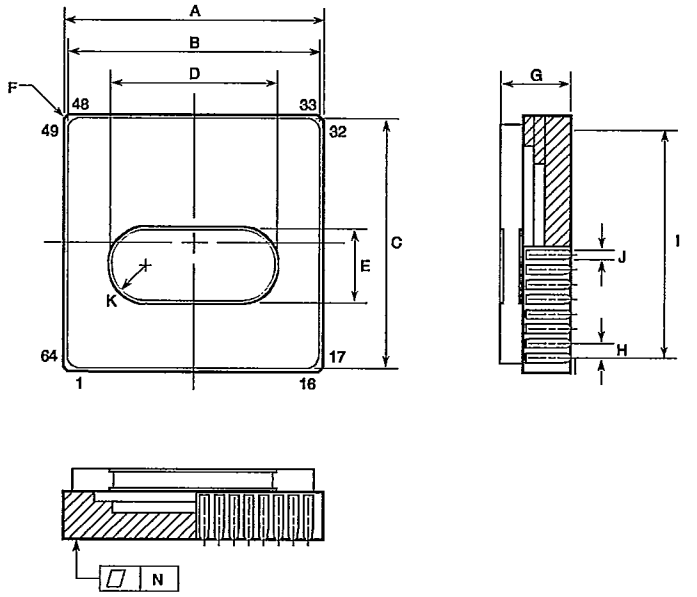


P64G-80-22-1

46NR-670B (1/90)

64-Pin Ceramic LCC With Window (Dwg No. X80KW-80B)

Item	Millimeters	Inches
A	14.00 ± 0.18	.551 ± .007
B	13.60 ± 0.15	.535 ± .006
C	13.60 ± 0.15	.535 ± .006
D	9.0	.354 Typ.
E	4.0	.157 Typ.
F	C 0.3	C .012
G	3.185 ± 0.371	.125 ± .015
H	0.8 ± 0.1	.031 ± .004
I	12.0 ± 0.15	.472 ± .006
J	0.51 ± 0.1	.020 ± .004
K	R 2.0	R .079
N	0.08	.003



X80KW-80B

83CL-9160B (9/93)