NEC
NEC Electronics Inc.

μPD78356 Family (μPD78355/356/P356) 16-/8-Bit, K-Series Microcontrollers With A/D Converter and Convolution Capability

Preliminary

September 1993

Description

The μPD78355, μPD78356, and μPD78P356 are K-Series® microcontrollers. These 16-/8-bit devices —with a minimum instruction time of 125 ns at 32 MHz—are designed for high-speed, real-time process control. They feature a 16-bit CPU, a 16-/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 with or without a saturation word provides hardware convolution capability; a 16-bit subtract and accumulate absolute value instruction provides correlation capability.

On-board memory includes 2048 bytes of RAM and 48K bytes of mask ROM, UV EPROM, or one-time programmable (OTP) ROM. A ROMless version is also available. The UV EPROM and OTP ROM versions feature a PROM error correction function capable of correcting one 1-bit error per four bytes of code. This achieves a significant improvement in reliability over devices without error correction and is suited for applications that require high reliability under rigorous conditions.

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 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, office automation, and industrial control/robotics markets.

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

Features

- Complete single-chip microcontroller
  - 16-bit ALU
  - 2048 bytes of RAM
  - 48K bytes of ROM (μPD78356) or PROM (μPD78P356)
- Powerful instruction set
  - 16-bit unsigned and signed multiply
  - 16-bit unsigned divide
  - 16-bit multiply and accumulate instruction with or without saturation word
  - 16-bit subtraction and accumulate absolute value instructions
  - 1-bit and 8-bit logic instructions
  - String instructions
- Minimum instruction time: 125 ns at 32 MHz
- 5-byte instruction prefetch queue
- Memory expansion
  - 8- or 16-bit external data bus
  - 64K-byte address space
- Large I/O capacity
  - Up to 57 I/O port lines (μPD78355)
  - Up to 76 I/O port lines (μPD78356/P356)
- Memory-mapped, on-chip peripherals (special function registers)
- Real-time pulse unit (RPU)
  - Two 16-bit interval timers
  - Two 16-bit timer/event counters
  - One 10-bit interval timer
  - One 16-bit up/down counter
  - Ten 16-bit capture/compare registers
  - Five external interrupt/capture lines
  - Three external event counter inputs
  - Three external timer clear inputs
  - Ten timer outputs
- Two pulse-width modulated (PWM) output lines with 8-, 10-, or 12-bit precision
- One 8-bit real-time output port
- Eight-channel, high-speed 10-bit A/D converter; conversion time: 2 μs at 32 MHz
- Two-channel, 8-bit D/A converter

50596

Features (cont)

- Three-channel serial communications interface
  - — Asynchronous serial interface (UART)
  - — Clock synchronous serial interface 0
    - Full-duplex, three-wire mode
    - NEC serial bus interface (SBI) mode
  - — Clock synchronous serial interface 1
    - Full-duplex three-wire mode
    - Pin switching function
- 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

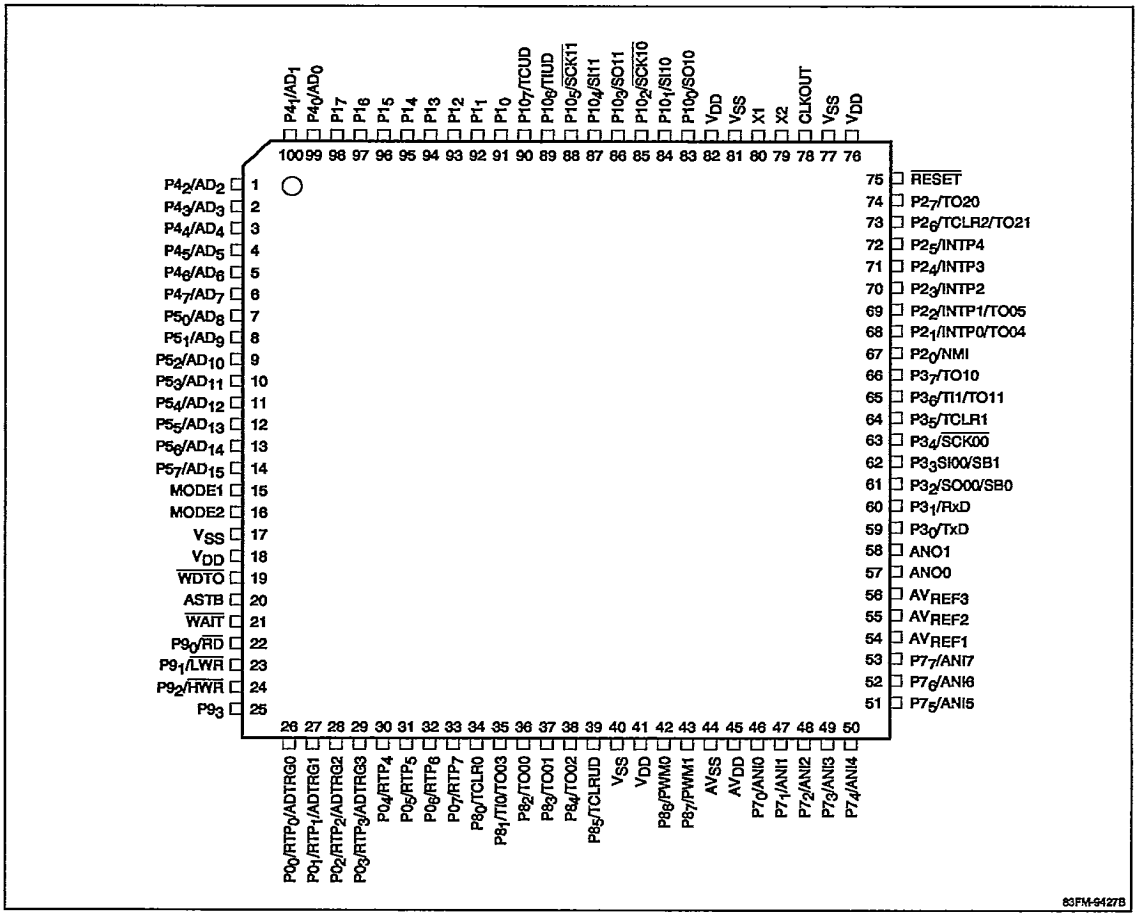
Ordering Information

Part Number	Package	ROM
μPD78355GC-7EA	100-pin plastic QFP (Dwg P100GC-50-7EA)	ROMless
μPD78356GC-xxx-7EA	100-pin plastic QFP (Dwg P100GC-50-7EA)	48K mask ROM
μPD78P356GC-7EA	100-pin plastic QFP (Dwg P100GC-50-7EA)	48K OTP ROM
μPD78P356KP-S	120-pin ceramic LCC with window (Dwg X120KW-80A)	48K UV EPROM

xxx indicates ROM code suffix.

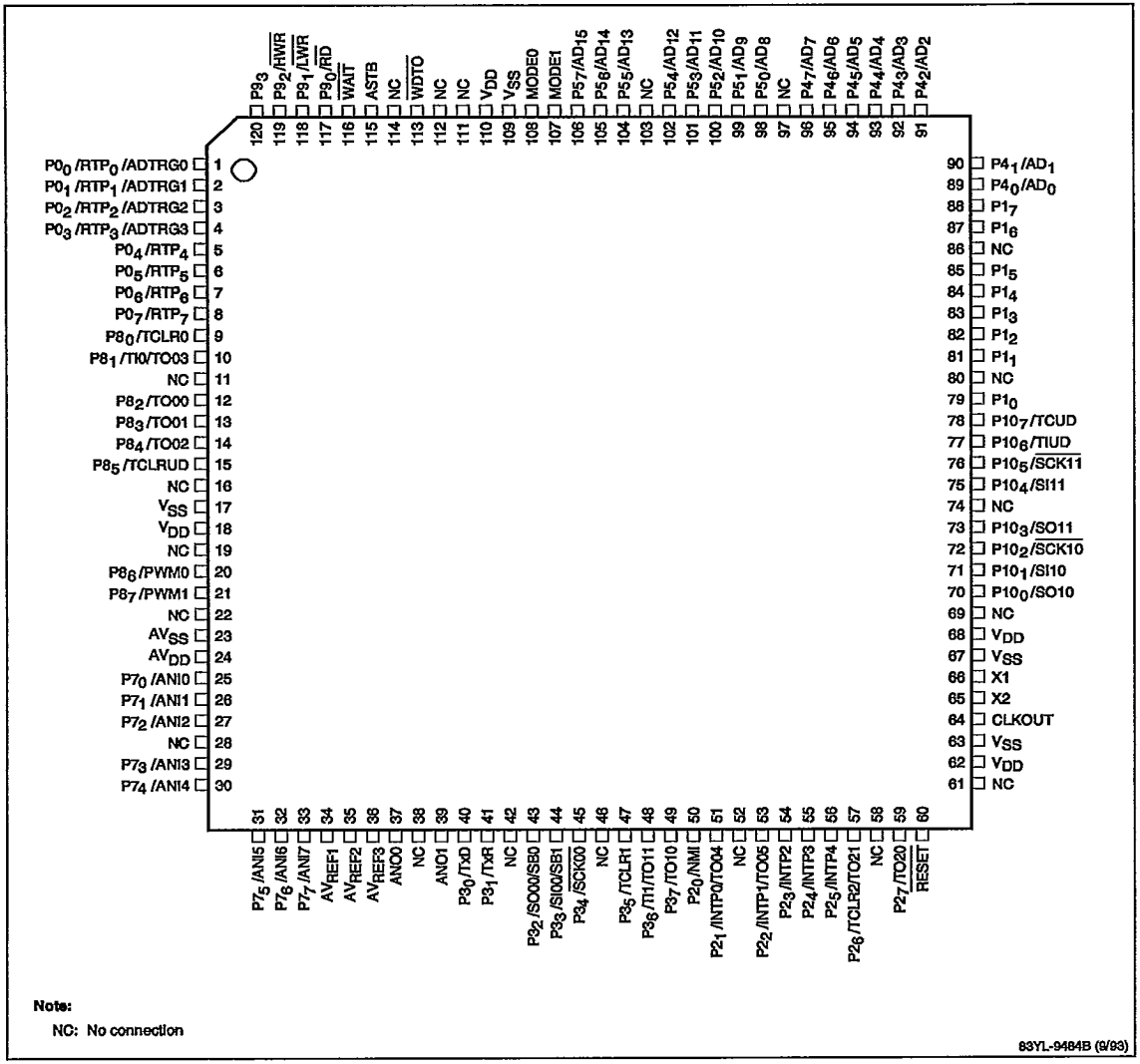
Pin Configurations

100-Pin Plastic QFP



Pin Configurations (cont)

120-Pin Ceramic LCC



Note:

NC: No connection

83YL-9484B (9/93)

Pin Functions; Normal Operating Mode

Pin Name	Function	Alternate Pin Name	Alternate Function
$P0_0$	Port 0; 8-bit, bit-selectable I/O port	RTP_0 ADTRG0	Real-time output port External trigger input for A/D converter
$P0_1$		RTP_1 ADTRG1	Real-time output port External trigger input for A/D converter
$P0_2$		RTP_2 ADTRG2	Real-time output port External trigger input for A/D converter
$P0_3$		RTP_3 ADTRG3	Real-time output port External trigger input for A/D converter
$P0_4$ - $P0_7$		RTP_4 - RTP_7	Real-time output port
$P1_0$ - $P1_7$	Port 1; 8-bit, bit selectable I/O port		
$P2_0$	Port 2; 8-bit, bit-selectable I/O port ($P2_0$ is input only)	NMI	External nonmaskable interrupt
$P2_1$		INTP0 TO04	External maskable interrupt Timer output from real-time pulse unit
$P2_2$		INTP1 TO05	External maskable interrupt Timer output from real-time pulse unit
$P2_3$		INTP2	External maskable interrupt
$P2_4$		INTP3	External maskable interrupt
$P2_5$		INTP4	External maskable interrupt
$P2_6$		TCLR2 TO21	Clear input to real-time pulse unit Timer output from real-time pulse unit
$P2_7$		TO20	Timer output from real-time pulse unit
$P3_0$	Port 3; 8-bit, bit-selectable I/O port	TxD	Asynchronous serial transmit data output
$P3_1$		RxD	Asynchronous serial receive data input
$P3_2$		SO00 SB0	Serial data output; three-wire serial I/O mode I/O bus for NEC serial bus interface mode
$P3_3$		SI00 SB1	Serial data input; three-wire serial I/O mode I/O bus for NEC serial bus interface mode
$P3_4$		$\overline{SCK00}$	Serial clock I/O for synchronous serial interface
$P3_5$		TCLR1	Clear input to real-time pulse unit
$P3_6$		TI1 TO11	External clock to timer 1 Timer output from real-time pulse unit
$P3_7$		TO10	Timer output from real-time pulse unit
$P4_0$ - $P4_7$	Port 4; 8-bit, byte-selectable I/O port (78356/P356)	AD_0 - AD_7	Low-order 8 bits of external multiplexed address/data bus
$P5_0$ - $P5_7$	Port 5; 8-bit, bit-selectable I/O port (78356/P356)	AD_8 - AD_{15}	High-order 8 bits of external multiplexed address/data bus
$P7_0$ - $P7_7$	Port 7; 8-bit input port	ANI0 - ANI7	Analog inputs to A/D converter

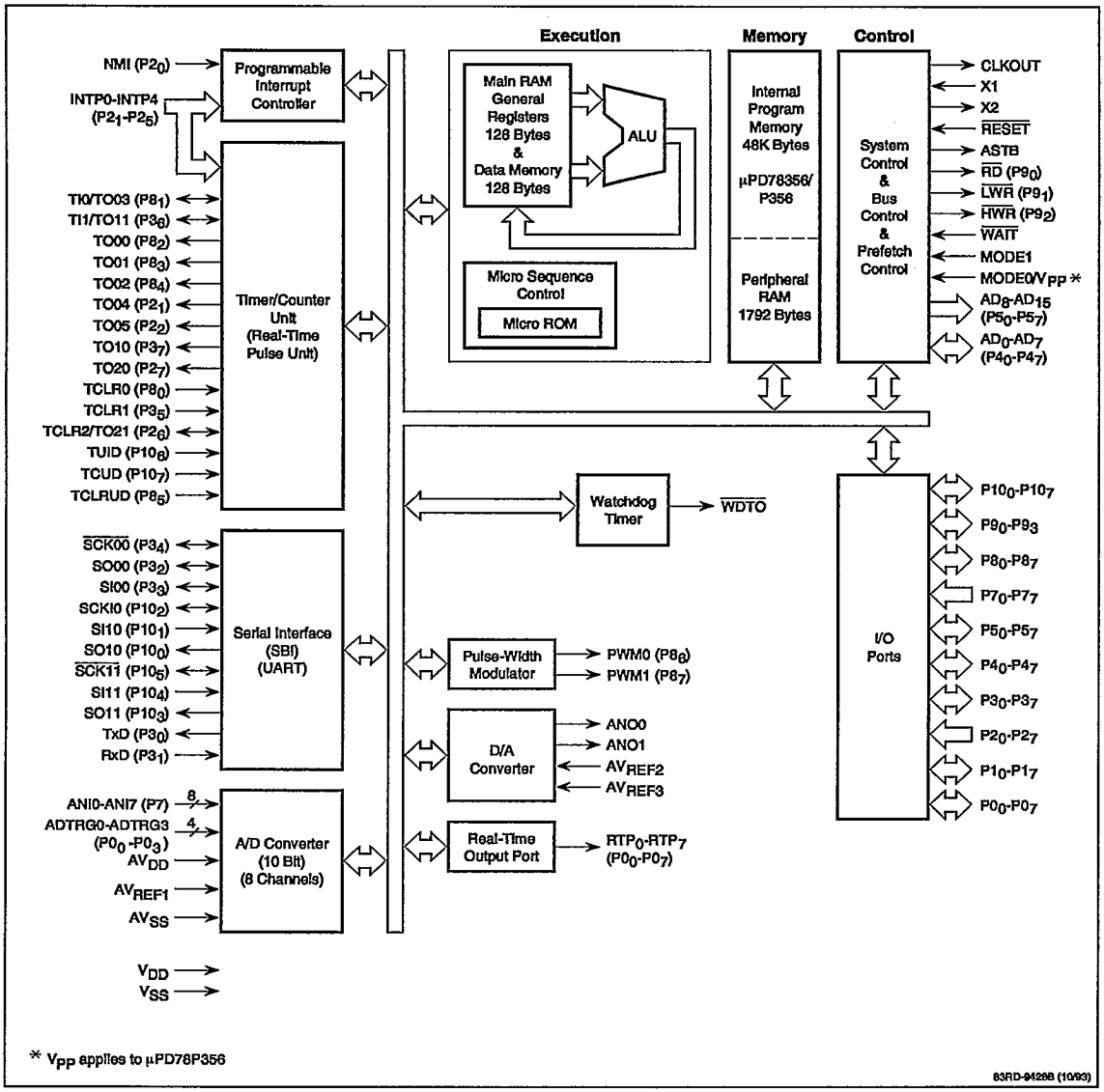
Pin Functions; Normal Operating Mode (cont)

Pin Name	Function	Alternate Pin Name	Alternate Function
$P8_0$	Port 8; 8-bit, bit-selectable I/O port	TCLR0	Clear input to real-time pulse unit
$P8_1$		TI0 TO03	External count clock to timer 0 Timer output from real-time pulse unit
$P8_2$		TO00	Timer output from real-time pulse unit
$P8_3$		TO01	Timer output from real-time pulse unit
$P8_4$		TO02	Timer output from real-time pulse unit
$P8_5$		TCLRUD	Clear input to real-time pulse unit
$P8_6$		PWM0	Pulse-width modulated output
$P8_7$		PWM1	Pulse-width modulated output
$P9_0$	Port 9; 4-bit, bit-selectable I/O port ($P9_0$ to $P9_2$ on 78356/P356)	$\overline{RD}$	External memory read strobe output
$P9_1$		$\overline{LWR}$	External memory write strobe output to low-order 8-bits in memory
$P9_2$		$\overline{HWR}$	External memory write strobe output to high-order 8-bits in memory
$P9_3$			
$P10_0$	Port 10; 8-bit, bit-selectable I/O port	SO10	Serial data output; three-wire serial I/O mode
$P10_1$		SI10	Serial data input; three-wire serial I/O mode
$P10_2$		$\overline{SCK10}$	Serial clock I/O for synchronous serial interface
$P10_3$		SO11	Serial data output; three-wire serial I/O mode
$P10_4$		SI11	Serial data input; three-wire serial I/O mode
$P10_5$		$\overline{SCK11}$	Serial clock I/O for synchronous serial interface
$P10_6$		TIUD	External clock for up/down counter
$P10_7$		TCUD	Up/down counter count direction control signal
ANO0, ANO1	Analog outputs from D/A converter		
ASTB	Address strobe output; used to latch address for external memory		
CLKOUT	Output of the system clock		
MODE0, MODE1	Set MODE0 and MODE1 to V_{SS} to access internal program memory on 78356/P356. If all program memory is external, set MODE0 to V_{DD} and MODE1 to to V_{SS} for an 8-bit data bus. Or set both to V_{DD} for a 16-bit data bus. To place 78P356 in programming mode, set MODE0 to V_{DD} and MODE1 to V_{SS}. The level of this pin cannot be changed during normal operation.		
NC	Pins labeled NC are not internally connected and may be connected to V_{SS}		
$\overline{RESET}$	External system reset input		
$\overline{WAIT}$	A low-level input adds wait states to the external bus cycle		
$\overline{WDTO}$	Open-drain output from the watchdog timer		

Pin Functions; Normal Operating Mode (cont)

Pin Name	Function	Alternate Pin Name	Alternate Function
X1	Crystal connection or external clock input		
X2	Crystal connection or open for external clock		
AV_{DD}	A/D converter power input		
AV_{REF1}	A/D converter reference voltage high		
AV_{REF2}	D/A converter reference voltage high		
AV_{REF3}	D/A converter reference voltage low		
AV_{SS}	A/D converter ground		
V_{DD}	+5 volt power input		
V_{SS}	Ground		

Block Diagram; μPD78356 Family



FUNCTIONAL DESCRIPTION

Central Processing Unit

The central processing unit (CPU) of the μPD78356 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 0.875 μs and the divide in 2.69 μs 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 13.44 μs at 32 MHz.

A subtract and accumulate absolute values instruction, SACW [DE+], [HL+], subtracts corresponding factors of two tables and calculates the sum of the absolute values of these subtractions. The total execution time for two tables of 10 terms each is 17.125 μs at 32 MHz.

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 32 MHz, the clock is 16 MHz with a 62.5 ns clock cycle. Since instructions execute in two or more cycles, the minimum instruction time is 125 ns.

Internal RAM

The μPD78356 family has a maximum of 2048 bytes of internal RAM. The upper 256-byte area (FE00H-FEFFH) features high-speed data access of one data word per two internal system clocks and is known as "main RAM." The remainder (F700H-FDFFH) is accessed at the same speed as external memory (one word per three internal system clocks) and is known as "peripheral RAM." The μPD78356 family can be programmed to have 1K or 2K bytes of internal 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.

Internal Program Memory

The 78356 contains up to 48K bytes of mask ROM and the 78P356 up to 48K bytes of UV EPROM or OTP ROM. The 78356/P356 can be programmed to have 16K, 24K, 32K, or 48K bytes of internal program memory by using the memory expansion mode register (MM). Instructions are fetched from this internal memory at a maximum rate of one word every two internal system clocks. The 78355 does not have internal program memory.

External Memory

The μPD78356 family has a 64K-byte address space. The 78356/P356 can access 0, 256, 4K, or 16K bytes of external memory in the area from C000H to F6FFH. External memory can be either ROM, RAM, or peripheral devices as required. The 78356/P356 can have an 8- or 16-bit wide external data bus with a 16-bit wide external address bus. The data bus size is specified on 16K-byte boundaries by the programmable wait control register (PWC). The upper 16K-block includes only external memory addresses C000H to F6FFH and the external SFR area, FFD0H to FFDFH.

For 8-bit data bus operation of the 78356/P356, data bits are multiplexed with low-order address bits at port 4. The high-order address bits are taken from port 5 as required. 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 address latch (ASTB), read ($\overline{RD}$), and write ($\overline{LWR}$) strobes are provided from port 9.

For 16-bit data bus operation of the 78356/P356, low-order data bits are multiplexed with low-order address bits at port 4. High-order data bits are multiplexed with high-order address bits at port 5. The address latch, read, and two write strobes ($\overline{LWR}$ and $\overline{HWR}$) are provided from port 9.

The 78355 does not have ports 4 and 5. The MODE0 and MODE1 pins specify whether the ROMless 78355 has an 8- or 16-bit data bus. When set for an 8-bit data bus, it has eight dedicated high-order address lines and eight multiplexed low-order address and data lines. When set for a 16-bit bus, it has 16 dedicated address/data lines. All memory below address F700H must be external.

Table 1 summarizes the operation of the μPD78356 family during word and byte accesses (on even or odd addresses) to external memory and the external SFR access area in 16-bit data bus mode.

Table 1. 16-Bit Bus Access Cycles

Addr	Operation	$\overline{RD}$ †	$\overline{LWR}$ †	$\overline{HWR}$ †	Data
Word Access					
Even	Write	1	0	0	AD_0 - AD_{15}
	Read	0	1	1	AD_0 - AD_{15}
Odd*	Write				
	1st byte	1	1	0	AD_8 - AD_{15}
	2nd byte	1	0	1	AD_0 - AD_7
	Read				
	1st byte	0	1	1	AD_8 - AD_{15}
	2nd byte	0	1	1	AD_0 - AD_7
Byte Access					
Even	Write	1	0	1	AD_0 - AD_7
	Read	0	1	1	AD_0 - AD_7
Odd	Write	1	1	0	AD_8 - AD_{15}
	Read	0	1	1	AD_8 - AD_{15}

* Word access to an odd address is accomplished by two byte accesses: 1st byte from odd address; 2nd byte from odd address plus 1.

† 1 = Inactive, 0 = Active

The programmable wait control register (PWC) also allows the programmer to specify one or two additional wait states if they are required for low-speed memory or external peripheral devices. These wait states are specified independently in 16K blocks and are applicable to internal ROM, external memory, and the external SFR access area. If additional wait states are required, an external WAIT pin is provided.

In addition, the width of the ASTB signal can be increased by one internal clock cycle to allow more precharge time for dynamic RAMs or more decoding time for addresses. This address wait can be enabled in 32K-byte blocks by using the PWC register and is also applicable to internal ROM, external memory, and the external SFR access area.

Program Fetch

The μPD78356 family allows opcode fetch in the area between 0000H and FDFFH; fetches from addresses F700H to FDFFH are always from the peripheral RAM. The μPD78356 family contains 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 internal memory, a minimum of two internal system clocks are required for each byte, and the queue can hold 5 bytes. If the instruction is fetched from external memory, a minimum of three internal system clocks are required from each byte, and the queue can hold 3 bytes.

Instructions can be fetched from internal memory in either high-speed or normal fetch cycle mode using the 16-bit bus. The PWC register is used to select the mode for each 16K block. In high-speed fetch cycle mode, two internal system clocks are required to fetch an instruction word from internal ROM/PROM or the peripheral RAM. In normal fetch cycle mode, each word fetched from internal ROM/PROM requires three, four, or five internal system clocks (address wait can also be included) depending on the setting of the PWC register.

Instructions can be fetched from external memory using either an 8- or 16-bit bus. Only normal fetch cycle mode is available and each byte or word fetched requires three, four, or five internal system clocks. One address wait state can also be included depending on the setting of the PWC register.

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 $\overline{RESET}$ 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)

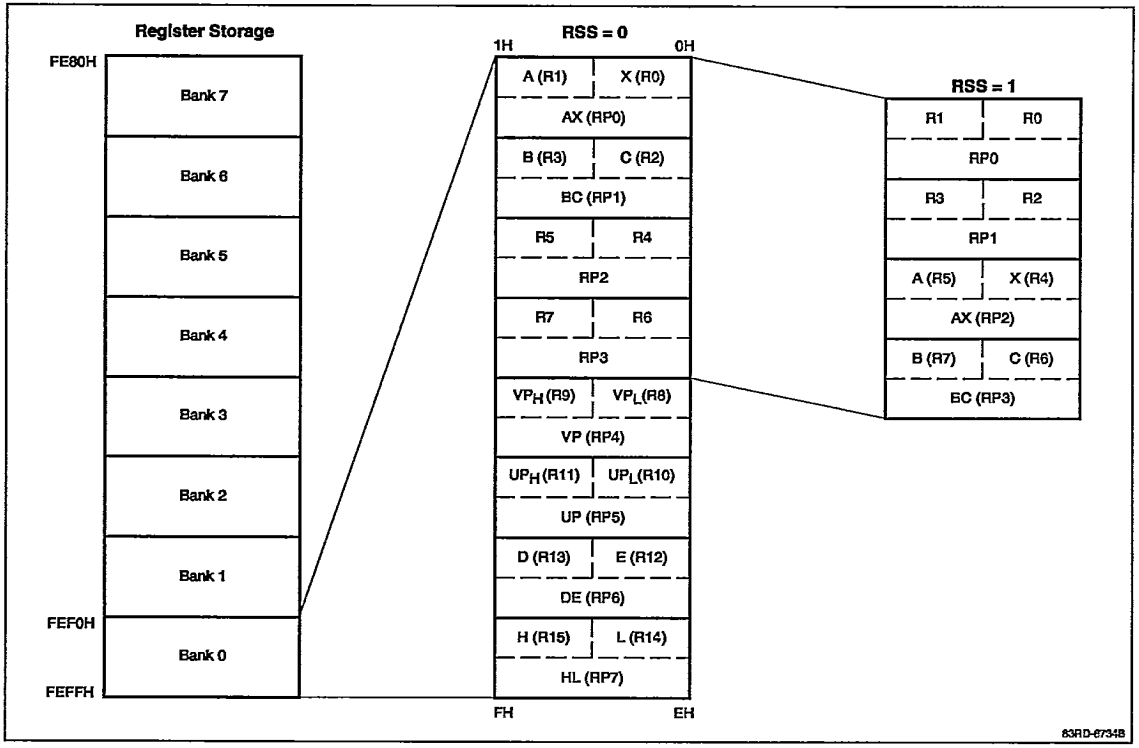
General Registers

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 1 illustrates the general register configuration.

Figure 1. General Registers



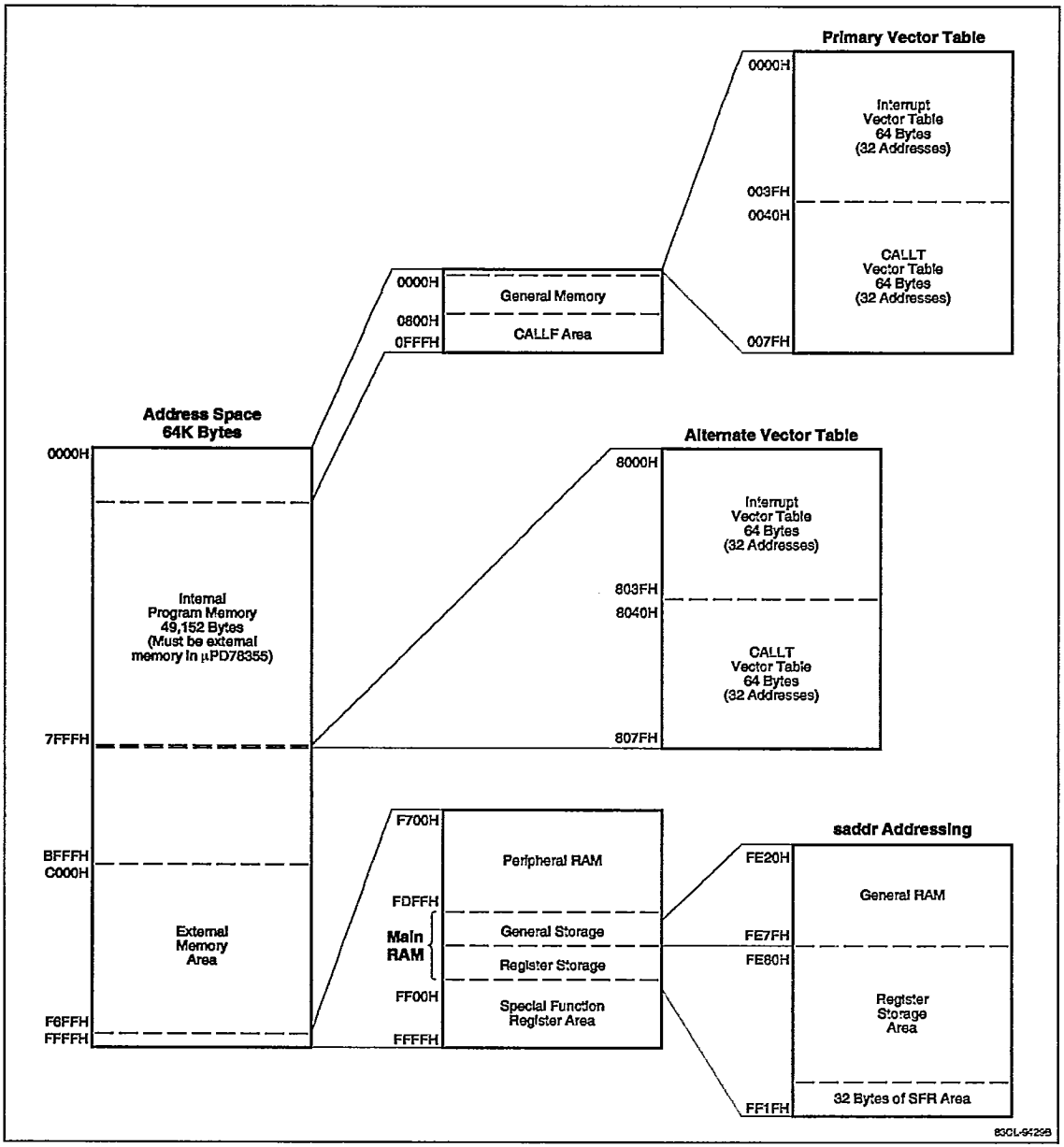
Addressing

The μPD78356 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 2 is the memory map of the μPD78356 family.

Figure 2. 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 2 lists the special function registers.

Table 2. 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 (Note 1)	P2	R/W	x	x	—	Undefined
FF03H	Port 3	P3	R/W	x	x	—	Undefined
FF04H	Port 4	P4	R/W	x	x	—	Undefined
FF05H	Port 5	P5	R/W	x	x	—	Undefined
FF07H	Port 7	P7	R	x	x	—	Undefined
FF08H	Port 8	P8	R/W	x	x	—	Undefined
FF09H	Port 9	P9	R/W	x	x	—	Undefined
FF0AH	Port 10	P10	R/W	x	x	—	Undefined
FF10H-FF11H	Capture/compare register 00	CC00	R/W	—	—	x	Undefined
FF12H-FF13H	Capture/compare register 01	CC01	R/W	—	—	x	Undefined
FF14H-FF15H	Capture/compare register 02	CC02	R/W	—	—	x	Undefined
FF16H-FF17H	Capture/compare register 30	CC30	R/W	—	—	x	Undefined
FF18H-FF19H	Capture/compare register 31	CC31	R/W	—	—	x	Undefined
FF1AH-FF1BH	Compare register 00	CM00	R/W	—	—	x	Undefined
FF1CH-FF1DH	Compare register 01	CM01	R/W	—	—	x	Undefined
FF1EH-FF1FH	Compare register 02	CM02	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
FF22H	Port 1 mode register (Note 2)	PM2	R/W	x	x	—	FFH
FF23H	Port 3 mode register	PM3	R/W	x	x	—	FFH
FF25H	Port 5 mode register	PM5	R/W	x	x	—	FFH
FF28H	Port 8 mode register	PM8	R/W	x	x	—	FFH
FF29H	Port 9 mode register	PM9	R/W	x	x	—	0FH
FF2AH	Port 10 mode register	PM10	R/W	x	x	—	FFH
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
FF36H-FF37H	Timer register 3	TM3	R	—	—	x	00H
FF38H-FF39H	Timer register 4	TM4	R	—	—	x	00H

Table 2. Special Function Registers (cont)

Address	Register (SFR)	Symbol	R/W	Access Units (Bits) 1	8	16	State After Reset
FF3AH-FF3BH	Presettable up/down counter register	UDC	R/W	—	—	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
FF40H	Port 0 mode control register	PMC0	R/W	x	x	—	00H
FF42H	Port 2 mode control register (Note 3)	PMC2	R/W	x	x	—	01H
FF43H	Port 3 mode control register	PMC3	R/W	x	x	—	00H
FF44H	Pullup resistor option register L	PUOL	R/W	x	x	—	00H
FF45H	Pullup resistor option register H	PUOH	R/W	x	x	—	00H
FF48H	Port 8 mode control register	PMC8	R/W	x	x	—	00H
FF4AH	Port 10 mode control register	PMC10	R/W	x	x	—	00H
FF50H-FF51H	Compare register 03	CM03	R/W	—	—	x	Undefined
FF52H-FF53H	Compare register 10	CM10	R/W	—	—	x	Undefined
FF54H-FF55H	Compare register 11	CM11	R/W	—	—	x	Undefined
FF56H-FF57H	Compare register 20	CM20	R/W	—	—	x	Undefined
FF58H-FF59H	Compare register 21	CM21	R/W	—	—	x	Undefined
FF5AH-FF5BH	Compare register 40	CM40	R/W	—	—	x	Undefined
FF5CH-FF5DH	Up/down counter compare register 0	CMUD0	R/W	—	—	x	Undefined
FF5EH-FF5FH	Up/down counter compare register 1	CMUD1	R/W	—	—	x	Undefined
FF60H	Real-time output port register L	RTPL	R/W	x	x	—	Undefined
FF61H	Real-time output port register H	RTPH	R/W	x	x	—	Undefined
FF62H	Port read control register	PRDC	R/W	x	x	—	00H
FF63H	Real-time output port mode register	RTPM	R/W	x	x	—	00H
FF68H	A/D converter mode register 0	ADM0	R/W	x	x	—	00H
FF69H	A/D converter mode register 1	ADM1	R/W	x	x	—	07H
FF6AH	D/A conversion setup register 0	DACS0	R/W	x	x	—	00H
FF6BH	D/A conversion setup register 1	DACS1	R/W	x	x	—	00H
FF6AH-FF6BH	D/A conversion setup register	DACS	R/W	—	—	x	0000H
FF70H	Timer unit mode register 0	TUM0	R/W	x	x	—	00H
FF71H	Timer unit mode register 1	TUM1	R/W	x	x	—	00H
FF72H	Timer unit mode register 2	TUM2	R/W	x	x	—	00H
FF73H	Timer unit mode register 3	TUM3	R/W	x	x	—	00H
FF74H	Timer control register 0	TMC0	R/W	x	x	—	00H
FF75H	Timer control register 1	TMC1	R/W	x	x	—	00H
FF76H	Timer control register 2	TMC2	R/W	x	x	—	04H
FF77H	Up/down counter control register	UDCC	R/W	x	x	—	00H
FF78H	Timer output control register 0	TOC0	R/W	x	x	—	00H
FF79H	Timer output control register 1	TOC1	R/W	x	x	—	00H
FF7AH	Timer output control register 2	TOC2	R/W	x	x	—	00H
FF7BH	Timer overflow status register (Note 4)	TOVS	R/W	x	x	—	00H
FF7CH	Noise protection control register	NPC	R/W	x	x	—	00H

Table 2. Special Function Registers (cont)

Address	Register (SFR)	Symbol	R/W	Access Units (Bits) 1	8	16	State After Reset
FF80H	Clock synchronous serial interface mode register 0	CSIM0	R/W	x	x	—	00H
FF82H	Serial bus interface control register (Note 5)	SBIC	R/W	x	x	—	00H
FF86H	Serial I/O shift register 0	SIO0	R/W	x	x	—	Undefined
FF88H	Asynchronous serial interface mode register	ASIM	R/W	x	x	—	80H
FF8AH	Asynchronous serial interface status register	ASIS	R	x	x	—	00H
FF8CH	Serial reception buffer: UART	RxB	R	—	x	—	Undefined
FF8EH	Serial transmission shift register: UART	TxS	W	—	x	—	Undefined
FF90H	Clock synchronous serial interface mode register 1	CSIM1	R/W	x	x	—	00H
FF96H	Serial I/O shift register 1	SIO1	R/W	x	x	—	Undefined
FFA0H	PWM control register	PWMC	R/W	x	x	—	00H
FFA2H	PWM register 0L	PWM0L	R/W	x	x	—	Undefined
FFA2H-FFA3H	PWM register 0	PWM0	R/W	—	—	x	Undefined
FFA4H	PWM register 1L	PWM1L	R/W	x	x	—	Undefined
FFA4H-FFA5H	PWM register 1	PWM1	R/W	—	—	x	Undefined
FFA8H	Inservice priority register	ISPR	R	x	x	—	00H
FFAAH	Interrupt mode control register	IMC	R/W	x	x	—	80H
FFACH	Interrupt mask register 0L	MK0L	R/W	x	x	—	FFH
FFACH-FFADH	Interrupt mask register 0	MK0	R/W	—	—	x	FFFFH
FFADH	Interrupt mask register 0H	MK0H	R/W	x	x	—	FFH
FFAEH	Interrupt mask register 1L	MK1L	R/W	x	x	—	FFH
FFAEH-FFAFH	Interrupt mask register 1	MK1	R/W	—	—	x	00FFH
FFB0H-FFB1H	A/D conversion result register 0	ADCR0	R	—	—	x	Undefined
FFB1H	A/D conversion result register 0H	ADCR0H	R	—	x	—	Undefined
FFB2H-FFB3H	A/D conversion result register 1	ADCR1	R	—	—	x	Undefined
FFB3H	A/D conversion result register 1H	ADCR1H	R	—	x	—	Undefined
FFB4H-FFB5H	A/D conversion result register 2	ADCR2	R	—	—	x	Undefined
FFB5H	A/D conversion result register 2H	ADCR2H	R	—	x	—	Undefined
FFB6H-FFB7H	A/D conversion result register 3	ADCR3	R	—	—	x	Undefined
FFB7H	A/D conversion result register 3H	ADCR3H	R	—	x	—	Undefined
FFB8H-FFB9H	A/D conversion result register 4	ADCR4	R	—	—	x	Undefined
FFB9H	A/D conversion result register 4H	ADCR4H	R	—	x	—	Undefined
FFBAH-FFBBH	A/D conversion result register 5	ADCR5	R	—	—	x	Undefined
FFBBH	A/D conversion result register 5H	ADCR5H	R	—	x	—	Undefined
FFBCH-FFBDH	A/D conversion result register 6	ADCR6	R	—	—	x	Undefined
FFBDH	A/D conversion result register 6H	ADCR6H	R	—	x	—	Undefined
FFBEH-FFBFH	A/D conversion result register 7	ADCR7	R	—	—	x	Undefined
FFBFH	A/D conversion result register 7H	ADCR7H	R	—	x	—	Undefined
FFC0H	Standby control register (Note 6)	STBC	R/W	—	x	—	0000 x000B
FFC1H	CPU control word	CCW	R/W	x	x	—	00H

Table 2. Special Function Registers (cont)

Address	Register (SFR)	Symbol	R/W	Access Units (Bits) 1	8	16	State After Reset
FFC2H	Watchdog timer mode register (Note 6)	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 SFR area	—	R/W	x	x	—	Undefined
FFE0H	Interrupt control register (INTOV0)	OVIC0	R/W	x	x	—	43H
FFE1H	Interrupt control register (INTOV3)	OVIC3	R/W	x	x	—	43H
FFE2H	Interrupt control register (INTP0/INTCC00)	PIC0	R/W	x	x	—	43H
FFE3H	Interrupt control register (INTP1/INTCC01)	PIC1	R/W	x	x	—	43H
FFE4H	Interrupt control register (INTP2/INTCC02)	PIC2	R/W	x	x	—	43H
FFE5H	Interrupt control register (INTP3/INTCC30)	PIC3	R/W	x	x	—	43H
FFE6H	Interrupt control register (INTP4/INTCC31)	PIC4	R/W	x	x	—	43H
FFE7H	Interrupt control register (INTCM00)	CMIC00	R/W	x	x	—	43H
FFE8H	Interrupt control register (INTCM01)	CMIC01	R/W	x	x	—	43H
FFE9H	Interrupt control register (INTCM02)	CMIC02	R/W	x	x	—	43H
FFEAH	Interrupt control register (INTCM03)	CMIC03	R/W	x	x	—	43H
FFEBH	Interrupt control register (INTCM10)	CMIC10	R/W	x	x	—	43H
FFECH	Interrupt control register (INTCM11)	CMIC11	R/W	x	x	—	43H
FFEDH	Interrupt control register (INTCM20)	CMIC20	R/W	x	x	—	43H
FFEEH	Interrupt control register (INTCM21)	CMIC21	R/W	x	x	—	43H
FFEFH	Interrupt control register (INTCM40)	CMIC40	R/W	x	x	—	43H
FFF0H	Interrupt control register (INTCMUD0)	CMICUD0	R/W	x	x	—	43H
FFF1H	Interrupt control register (INTCMUD1)	CMICUD1	R/W	x	x	—	43H
FFF2H	Interrupt control register (INTSER)	SERIC	R/W	x	x	—	43H
FFF3H	Interrupt control register (INTSR)	SRIC	R/W	x	x	—	43H
FFF4H	Interrupt control register (INTST)	STIC	R/W	x	x	—	43H
FFF5H	Interrupt control register (INTCSI0)	CSIIC0	R/W	x	x	—	43H
FFF6H	Interrupt control register (INTCSI1)	CSIIC1	R/W	x	x	—	43H
FFF7H	Interrupt control register (INTAD)	ADIC	R/W	x	x	—	43H

Notes:

(1) P2 bit 0 is read only

(2) PM2 bit 0 is always 1.

(3) PMC2 bit 0 is always 1.

(4) TOVS bits 6 and 7 are always 0; bits 1, 2, 4, and 5 are read/write; bits 0 and 3 are read only.

(5) SBIC bits 5 and 7 are read/write; bits 2, 3, and 6 are read only; bits 0, 1, and 4 are write only.

(6) Protected register that can be written by a special instruction only.

Input/Output Ports

The 78355 has a total of 57 I/O lines. The 78356/P356 have an additional 19 for a total of 76 I/O lines. Ports P0, P1, P3, P8, and P10 are tri-state, 8-bit input/output ports and P7 is an 8-bit input port. Port P2 is an 8-bit I/O port with pin $P2_0$ always an input. Bit 3 of port 9 is available as an I/O line at all times. All the I/O bits in P0 to P3 and P8 to P10 can be individually selected for either input or output.

Each pin of P2 can be programmed for rising or falling edge detection; pins 1 to 7 can also be programmed for both rising and falling edge detection.

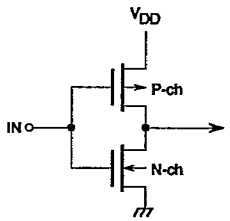
Software programmable internal pullup resistors are available at each pin of ports P0 to P3 and P8 to P10 except $P2_0$. These resistors are enabled on a port basis for all I/O pins set to input mode.

The output level of the P0 to P3 and P8 to P10 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 3 shows the structure of each port pin.

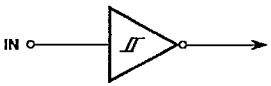
The 19 additional I/O lines in the 78356/P356 are P4, P5, and bits 0 to 2 of P9. These ports are available if external memory or memory-mapped external circuitry is not being used. Port 4 is shared with the low-order address/data bus (AD_0 to AD_7) and is byte-selectable for input or output. Port 5 is shared with the high-order address bus in 8-bit bus mode or the high-order address/data bus in 16-bit bus mode. Depending on the amount of external memory used, either 8, 4, 2, or 0 bits of port 5 are available for bit-selectable I/O. Port 9 is a 4-bit, bit-selectable I/O port. One of its pins is shared with the read strobe and two others with the high and low write strobes. Internal pullup resistors are available at each pin of port 4 and 5 when they are set to input mode.

Figure 3. I/O Circuits

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

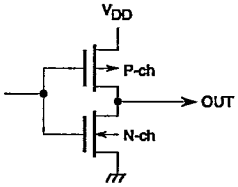


Type 2. $P2_0$, $\overline{\text{RESET}}$

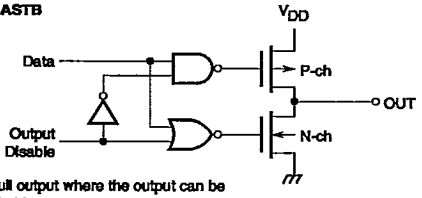


Schmitt trigger input with hysteresis characteristics.

Type 3. CLKOUT

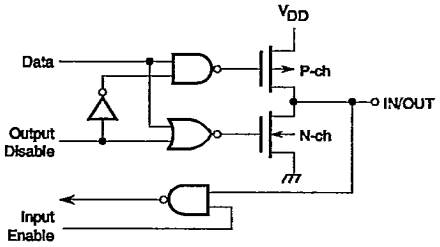


Type 4. ASTB

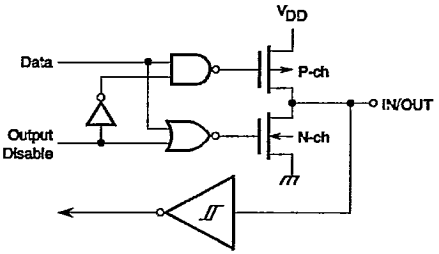


Push-pull output where the output can be placed in high impedance (Both the P and N channels are off.)

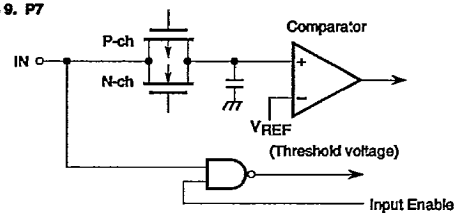
Type 5. $P0_4$-$P0_7$, P1, $P3_0$, $P3_1$, $P3_7$, P4, P5, $P8_2$-$P8_4$, $P8_6$, $P8_7$, P9, $P10_0$, $P10_3$



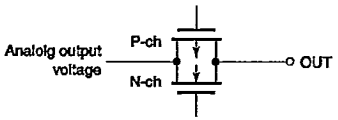
Type 8. $P0_0$-$P0_3$, $P2_1$-$P2_7$, $P3_2$-$P3_6$, $P8_0$, $P8_1$, $P8_5$, $P10_1$, $P10_2$, $P10_4$-$P10_7$



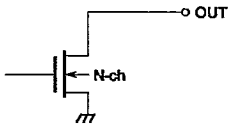
Type 9. P7



Type 12. ANO0, ANO1



Type 14. $\overline{\text{WDTO}}$



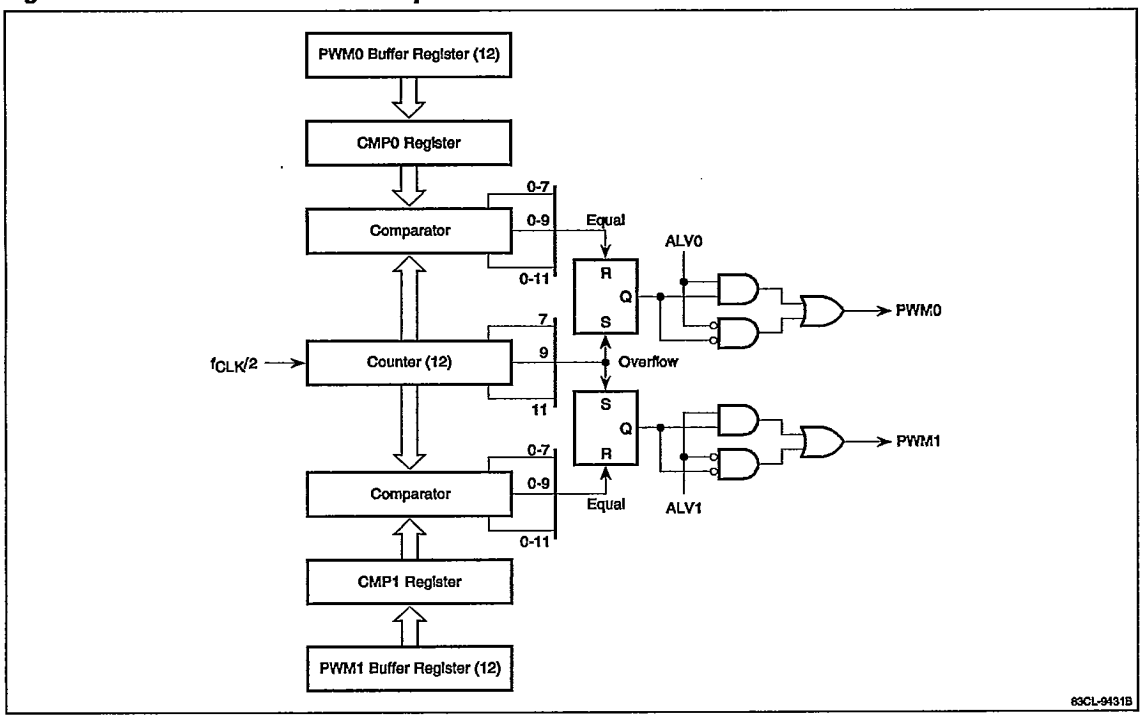
83CL-9430B (9/93)

Pulse-Width Modulated Outputs

The μPD78356 family has two high-speed, pulse-width modulated (PWM) outputs. A single 12-bit, free-running counter counts the internal system clock $f_{CLK}/2$ and serves both outputs. The resolution is 125 ns per bit at 32 MHz. By setting the counter and comparator to either 8, 10, or 12 bits, repetition rates of 31.2, 7.8, and 1.9 kHz, respectively, can be achieved.

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 8 bits 6 and 7, respectively.

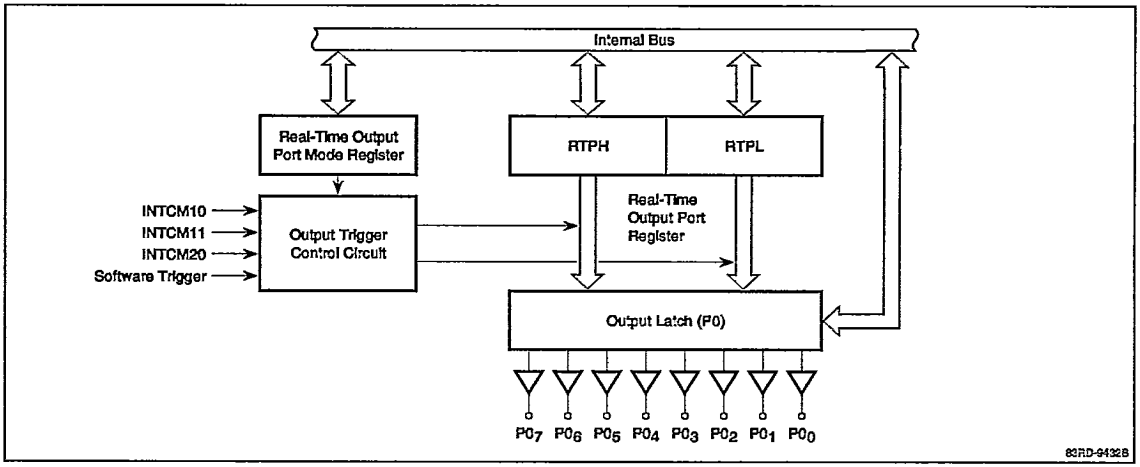
Figure 4. Pulse-Width Modulated Outputs



Real-Time Output Port

Port 0 can function on a bit-selectable basis as a real-time output port. See figure 5. Real-time port bits can be directly written under program control or they can be written under control of timing signals INTCM10, INTCM11, and INTCM20 generated by the real-time pulse unit. The latter method provides output timing that is independent of interrupt latency.

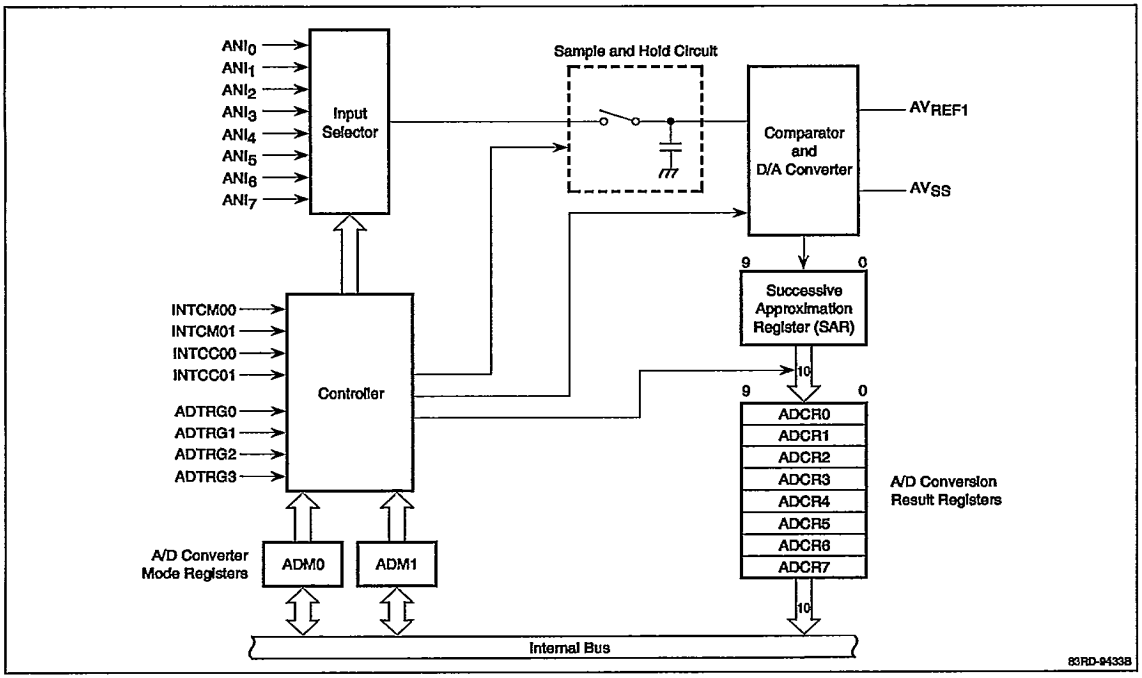
Figure 5. Real-Time Output Port



A/D Converter

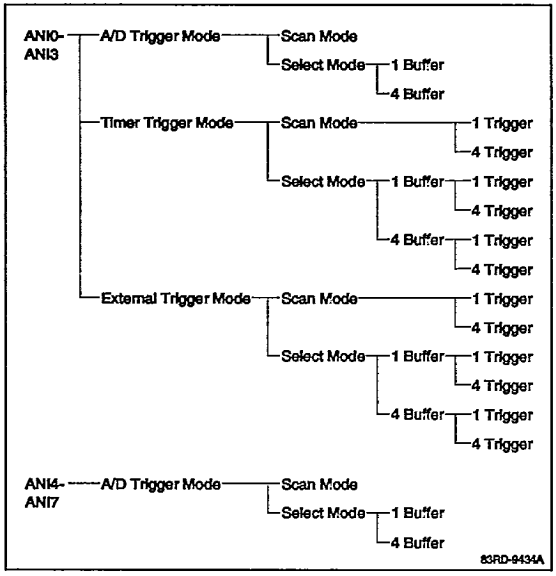
The analog-to-digital (A/D) converter (see figure 6) uses the successive-approximation method for converting up to eight multiplexed analog inputs into 10-bit digital data. The minimum conversion time per input is 2 μs at 32-MHz operation. There are eight 16-bit A/D conversion results registers (ADCR0 to ADCR7). During word access, the low-order 10-bits contain the result and the upper 6-bits are set to zero. During byte access, the high-order 8-bits of the 10-bit A/D conversion result are read. This converted data can be easily transferred to memory by the macro service function.

Figure 6. A/D Converter



The A/D converter of the μPD78356 family supports a wide variety of operating modes. See figure 7. A/D conversion can be started by one of three modes: A/D trigger mode, timer trigger mode, and external trigger mode. Analog inputs ANI0 to ANI3 can be started by all three modes; ANI4 to ANI7 can be started only by the A/D trigger mode. In A/D trigger mode, conversion is started by writing data into the A/D converter mode register 0 (ADM0) and proceeds automatically.

Figure 7. A/D Converter Operating Modes



In timer trigger mode, coincidence signals from compare registers CM00, CM01, CC00, and CC01 associated with timer 0 start the conversions. Either all four signals are used (four-trigger mode) or only the coincidence signal CM00 is used (one-trigger mode). The timing and sequence of the A/D conversions are controlled by these trigger signals. Conversions can be performed once or repeatedly.

In external trigger mode, four external signals ADTRG0 to ADTRG3 start the conversions. Either all four signals (four-trigger mode) or one signal ADTRG0 (one-trigger mode) starts the conversions. The timing and sequence of the A/D conversions are controlled by these trigger signals.

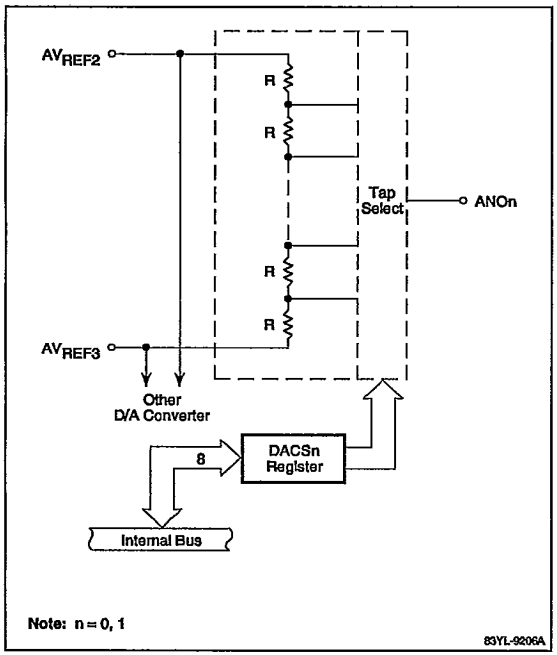
The A/D converter can operate in either scan mode or select mode. In scan mode, from one to eight analog inputs (ANI0 to ANI7) can be programmed for conversion. The A/D converter selects each of the inputs (order and timing sequence are based on the trigger mode), converts the data, and stores it in its associated A/D conversion result register (ADCR0 to ADCR7). An interrupt (INTAD) is generated when all inputs have been converted.

In select mode, only one of the eight A/D inputs can be selected for conversion. Data from this input can be stored in its associated ADCR register (one-buffer mode) or four values can be stored sequentially in registers ADCR0 to ADCR3 (four-buffer mode). In one-buffer mode, an interrupt (INTAD) is generated after each conversion. In four-buffer mode, an interrupt is generated after four conversions. The timing of the conversions depends on the trigger mode selected.

D/A Converter

The μPD78356 family has two digital-to-analog (D/A) converters as shown in figure 8. The 8-bit digital data, written to the D/A conversion setup registers (DASCn; n = 0, 1), selects one of the 256 taps on a resistor ladder between AV_{REF2} and AV_{REF3}. The selected voltage becomes the analog output at the ANOn pin. ANOn is a high-impedance output and requires an external buffer to drive a low-impedance load.

Figure 8. D/A Converter

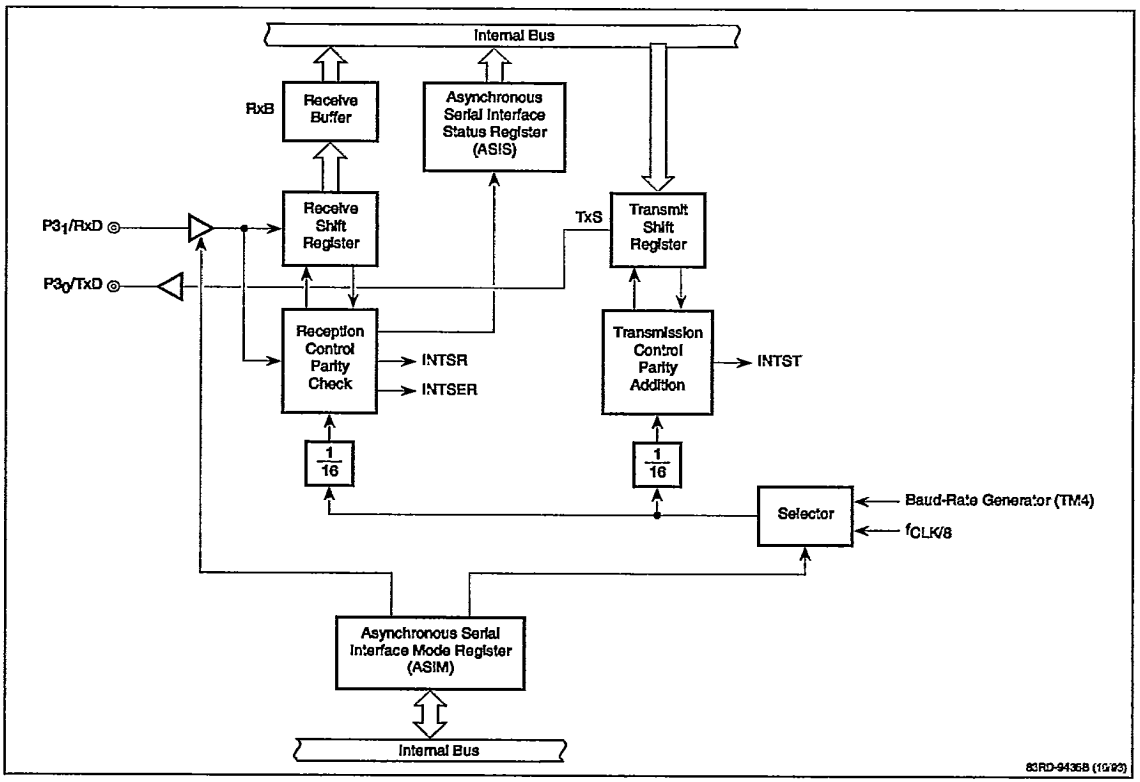


Serial Interfaces

The μPD78356 family has three independent serial interfaces: one asynchronous and two clock synchronous. All three share an internal baud-rate generator.

The asychrounous serial interface is a standard universal asynchronous receiver transmitter (UART). The UART (figure 9) permits full-duplex operation and can be programmed for 7 or 8 bits of data after the start bit, followed by one or two stop bits. Odd, even, zero, or no parity can also be selected. The serial clock for the UART is from the internal system clock divided by eight or from the internal baud-rate generator. The UART generates three interrupts: INTST (transmission complete), INTSR (reception complete), and INTSER (reception error).

Figure 9. Asynchronous Serial Interface

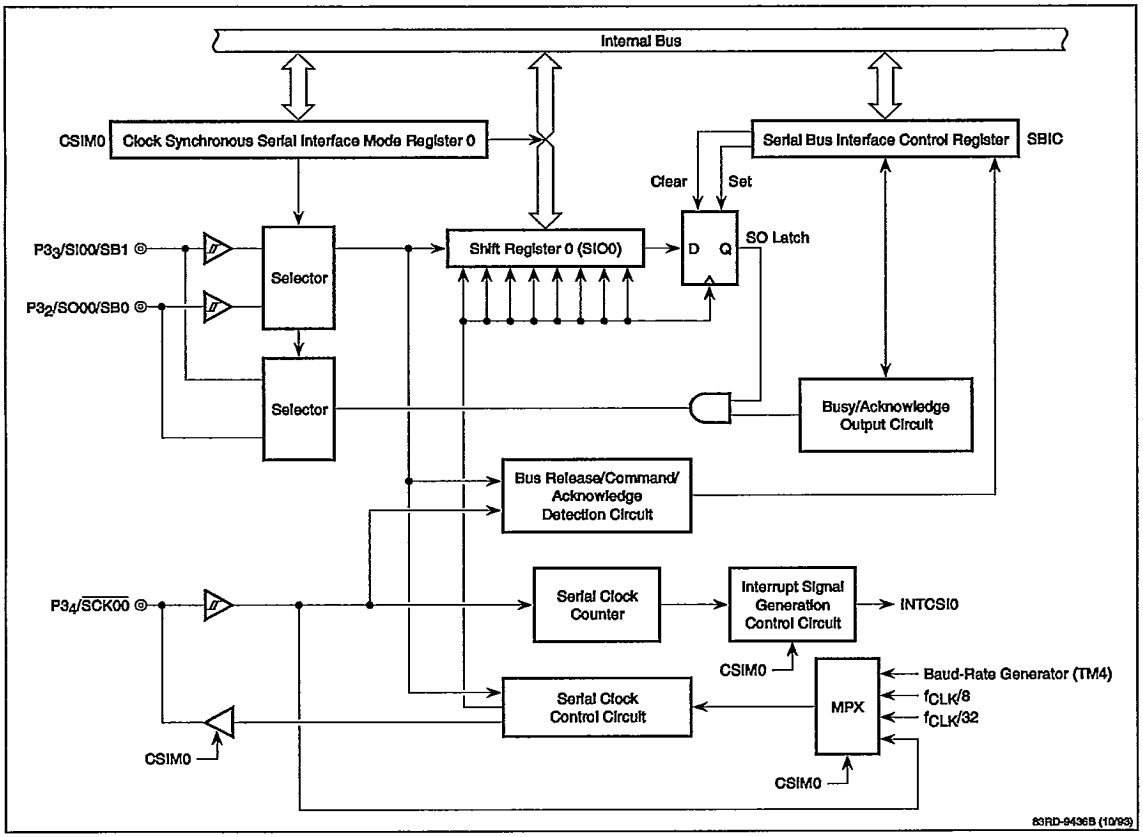


Clock synchronous serial interface 0 (figure 10) is an 8-bit interface that operates in either a three-wire serial I/O mode or NEC serial bus interface (SBI) mode.

In the three-wire serial I/O mode, the 8-bit shift register (SIO) is loaded with a byte of data and eight clock pulses are generated. These eight pulses shift the byte of data out the SO line (either MSB or LSB first) and in from the SI line, providing full-duplex operation. This interface can also be set to receive or transmit data only. The INTCSI0 interrupt is generated after each 8-bit transfer. One of two internal clocks, an external clock, or the internal baud-rate generator clocks the data.

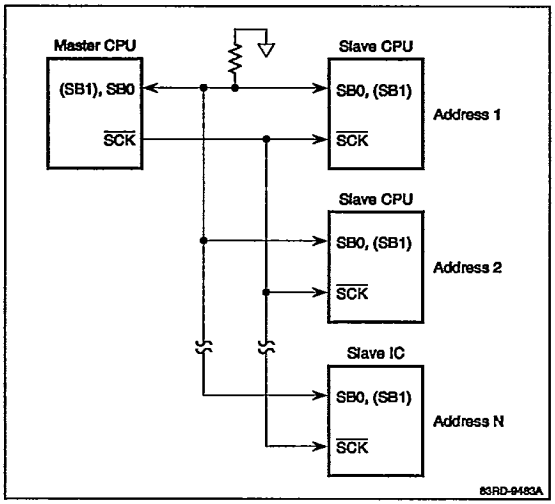
The NEC SBI mode is a two-wire, high-speed proprietary serial interface available on most devices in the NEC μPD75xxx and μPD78xxx product lines. Devices are connected in a master/slave configuration. See figure 11. There is only one master device at a time; all others are slaves. The master sends addresses, commands, and data over one of the serial bus lines (SB0 or SB1) using a fixed hardware protocol synchronized with the SCK line.

Figure 10. Clock Synchronous Serial Interface 0



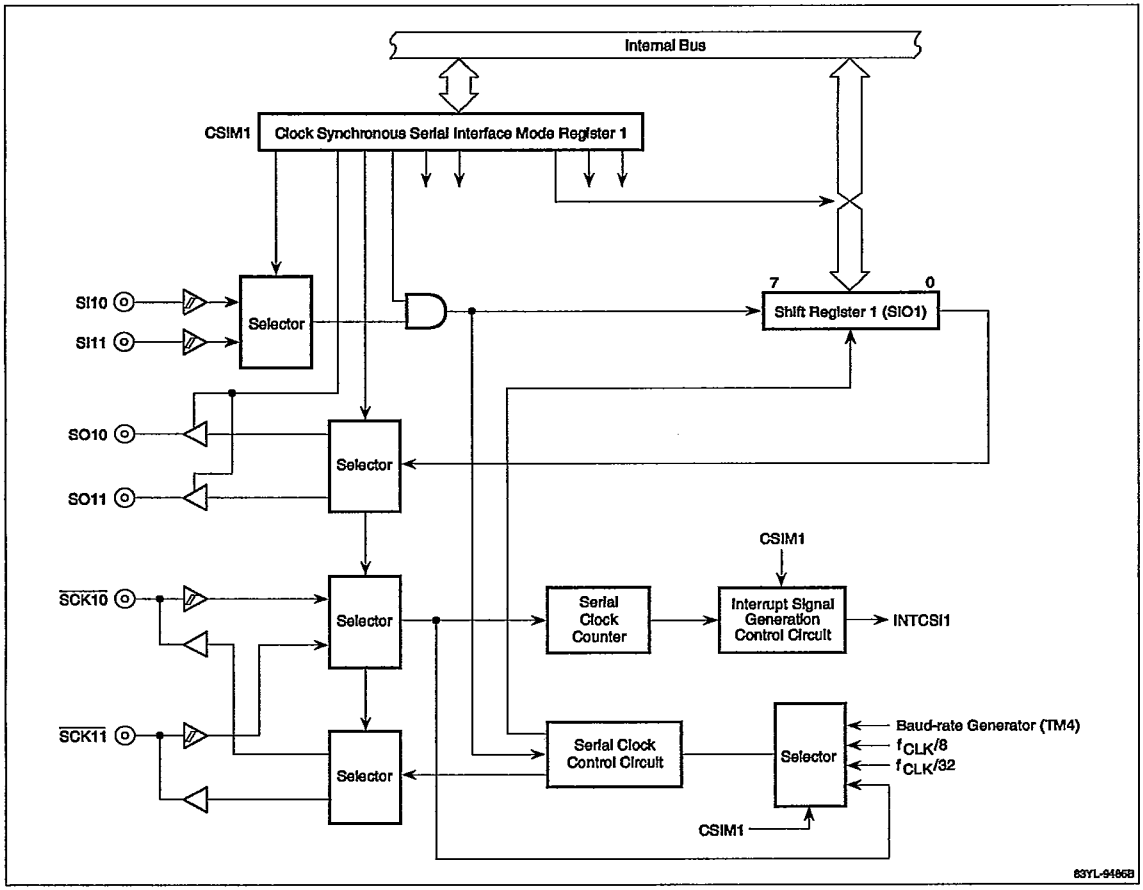
Each slave 78355/356/P356 can be programmed in software to respond to any one of 256 addresses. There are also 256 commands and 256 data types. Since all commands are user-definable, any software protocol, simple or complex, can be defined. It is even possible to develop commands to change a slave into a master and the previous master into a slave.

Figure 11. SBI Mode Master/Slave Configuration



Clock synchronous serial interface 1 (figure 12) with a pin switching function is also an 8-bit interface that operates only in the three-wire serial I/O mode. It can be switched under program control between two sets of I/O pins: $\overline{SCK10}$, SO10, and SI10 or $\overline{SCK11}$, SO11, and SI11. With the exception of this pin switching function, its operation in three-wire serial I/O mode is identical to clock synchronous interface 0.

Figure 12. Clock Synchronous Serial Interface 1 With Pin Switching



A dedicated baud-rate generator can be programmed to provide a common serial clock to the asynchronous and clock synchronous serial interfaces. The baud-rate generator uses timer 4 and compare register CM40 of the real-time pulse unit to generate a serial clock from one of four internal clocks. By choosing the correct oscillator frequency, the baud-rate generator is capable of generating all the commonly used baud rates from 2400 to 154K b/s for the UART and 150 to 2M b/s for the clock synchronous interfaces.

Real-Time Pulse Unit

The real-time pulse unit (RPU, figure 13) can be used as an interval timer, to measure pulse widths and frequencies, to generate puse-width modulated outputs, to count external events, to control the real-time output port and the A/D converter, and to generate the serial clock. It consists of two 16-bit timer/counters (TM0 and TM1), two 16-bit timers (TM2 and TM3), one 10-bit interval timer (TM4), one 16-bit up/down counter (UDC), ten 16-bit compare registers, five 16-bit registers that can be used for either capture or compare, and ten timed output latches.

All the timers count various clocks derived from the internal system clock. Timers TM0 and TM1 also count external events on the TI0 and TI1 pins, respectively. All timers are cleared by an external reset. Timers TM0 to TM2 can also be cleared by either an external clear input or a coincidence interrupt from one of its compare registers. Timers TM3 and TM4 are cleared only by a coincidence interrupt from one of its associated compare registers. When any of timers TM0 to TM4 overflow, an overflow bit is set; in the case of timers TM0 and TM3, an overflow interrupt is also generated. Timer TM4 can also be used as the baud-rate generator for the serial interfaces.

Capture events for TM0 can be triggered by external maskable interrupts INTP0 to INTP2; capture events for TM3 can be triggered by INTP3 and INTP4. Compare events associated with timers TM0 to TM4 can be used to generate interrupts, control timed output pins, or both. In addition, three of them (INTCM10, INTCM11, INTCM20) can control the real-time output port and four of them (INTCM00, INTCM01, INTCC00, INTCC01) can control the A/D converter. The timed output latches share pins in ports P2, P3, and P8. Five of them can be toggled or set and reset by compare events, and the remaining five can be toggled. These latches can generate pulse-width modulated outputs.

Figure 13. Real-Time Pulse Unit (Sheet 1 of 3)

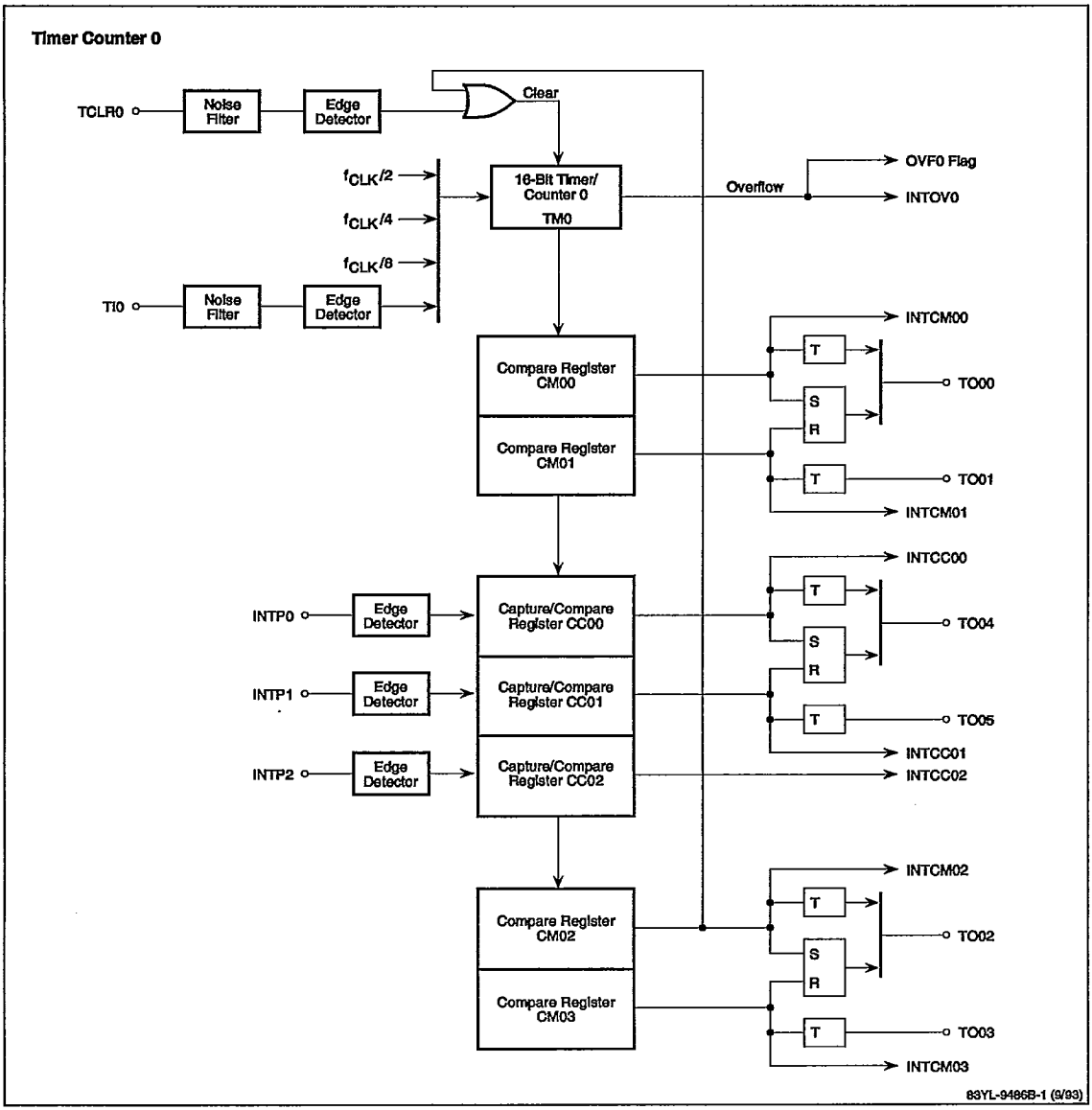


Figure 13. Real-Time Pulse Unit (Sheet 2 of 3)

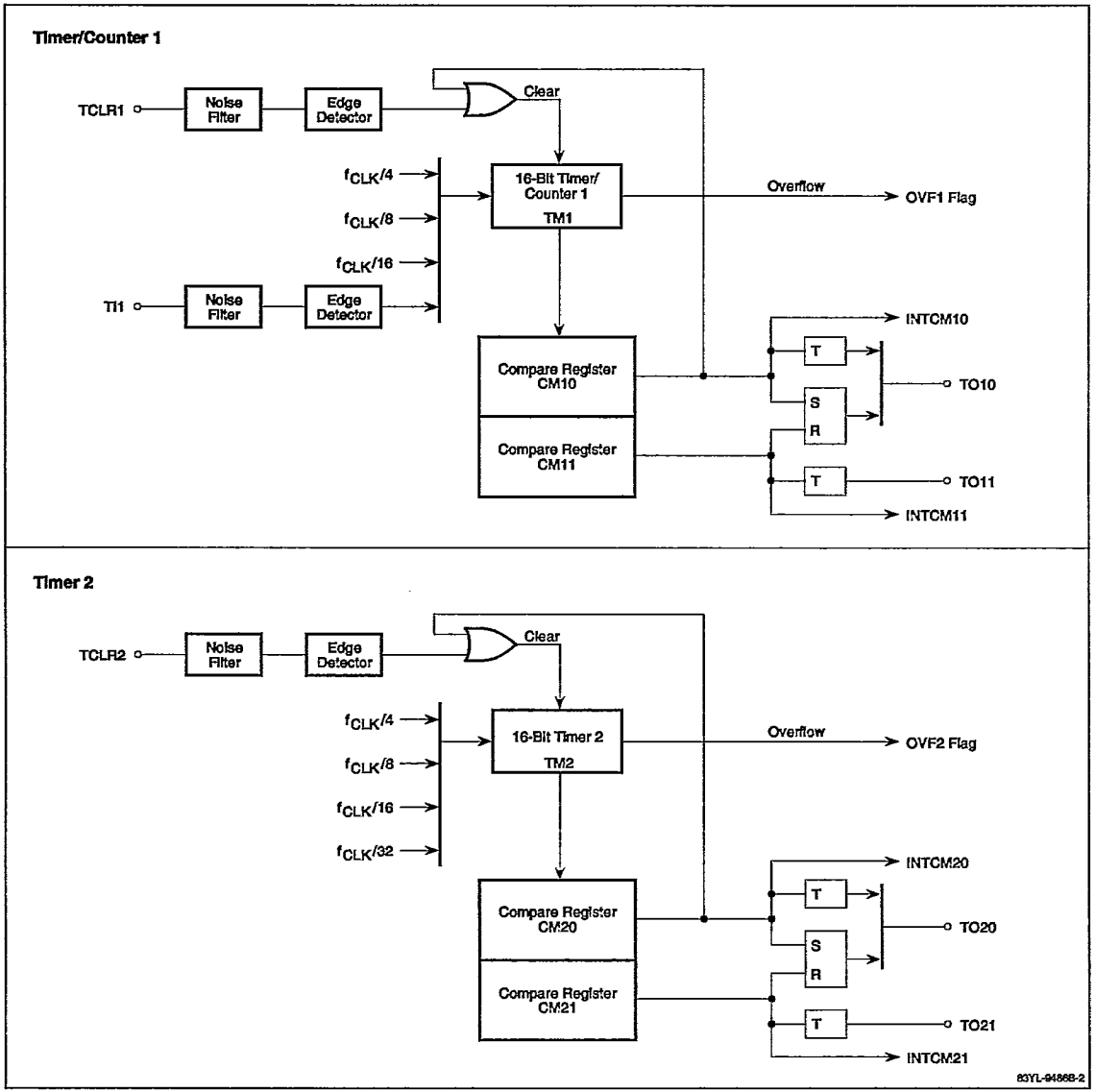
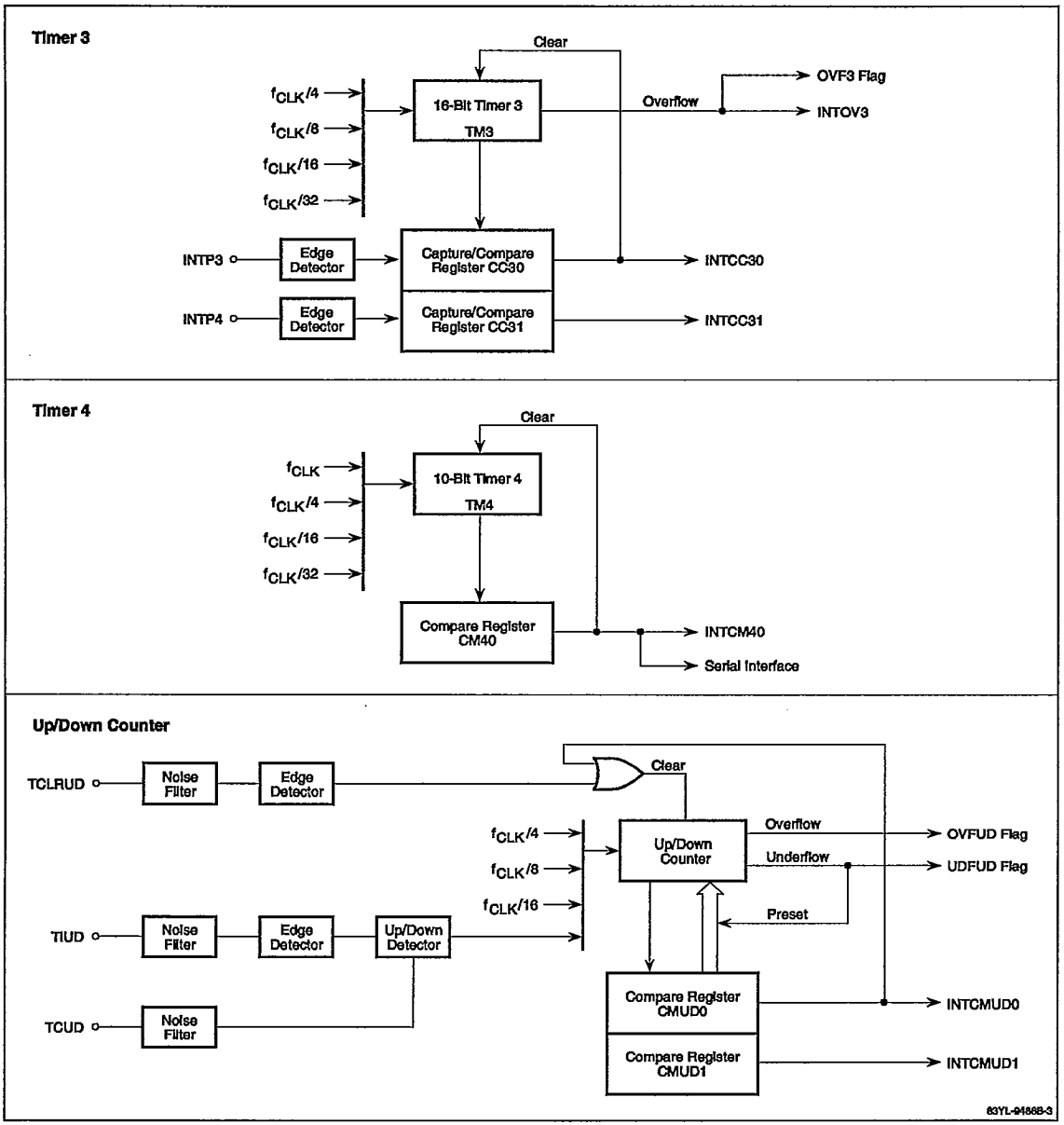


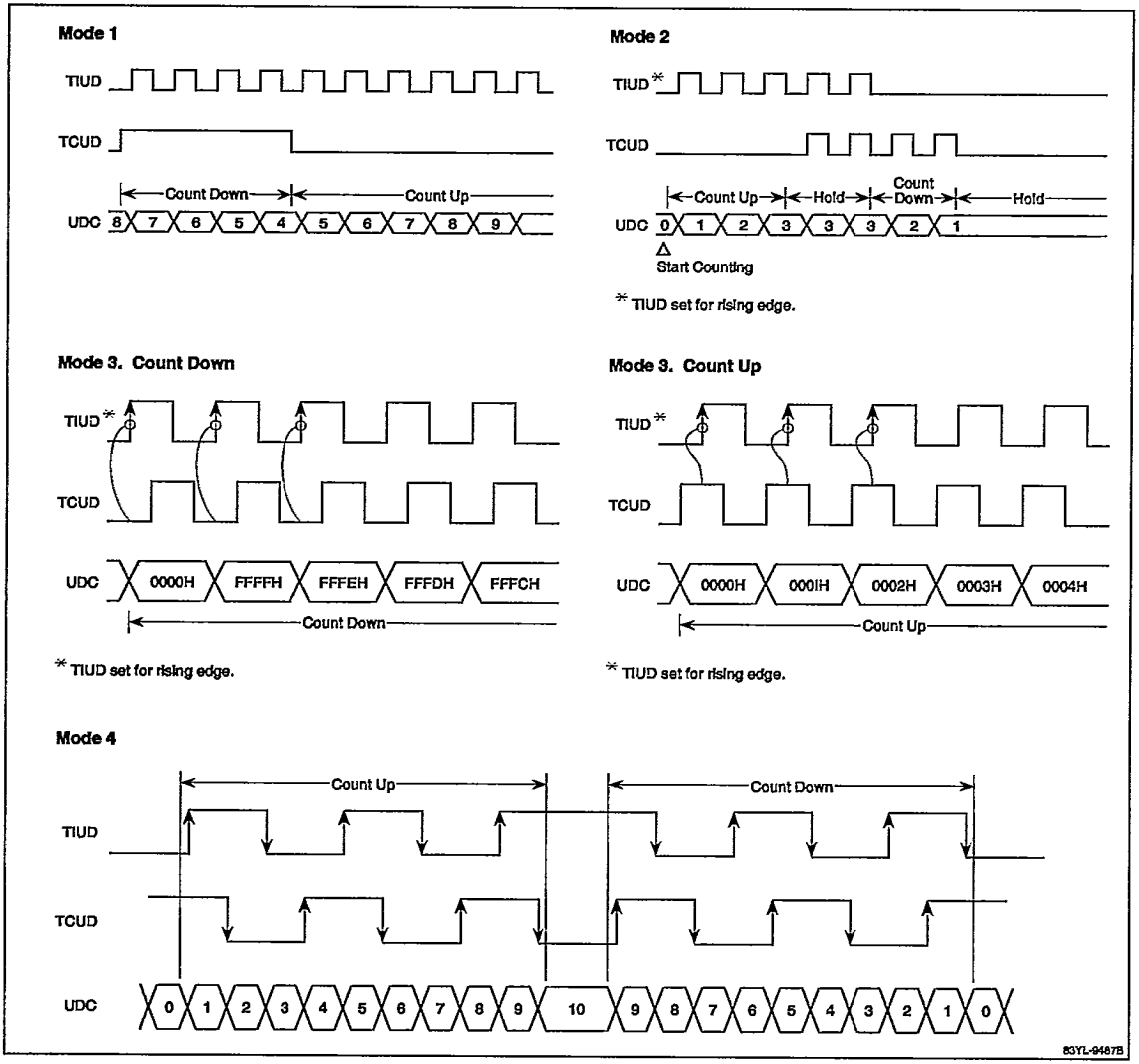
Figure 13. Real-Time Pulse Unit (Sheet 3 of 3)



The 16-bit up/down counter (UDC) can count the internal system clock (divided by 4, 8, or 16) or count external events on the TIUD pin. When the counter overflows, an overflow bit is set. If the counter underflows, an underflow bit is set. The UDC can be cleared by an external clear input (TCLRUD) or by a coincidence signal from its compare register.

When counting external events, the UDC can be programmed to operate in four modes. See figure 14. In mode 1, the UDC counts external events on the TIUD pin. When direction control pin TCUD is high, the UDC counts down. When TCUD is low, the UDC counts up.

Figure 14. Up/Down Counter; Modes 1, 2, 3, 4



In mode 2, when the preset edge (rising, falling, or both) is detected at the TIUD pin, the UDC counts up; when a rising edge is detected at the TCUD pin, the UDC counts down. If TIUD and TCUD are active simultaneously, they are not counted and the value in the UDC is held.

Modes 3 and 4 are designed to count the output of a two-phase shaft encoder on a servomotor. In mode 3, two signals having a 90-degree phase shift are entered into the TIUD and TCUD pins. When the preset edge (rising, falling, or both) is detected on the TIUD pin, the signal level on TCUD is sampled. If the level of TCUD is low, the UDC counts down; if the level is high, the UDC counts up.

When mode 4 is specified, quadrature counting is enabled. The UDC is incremented or decremented at positive and negative transitions of both input signals. Whether it is incremented or decremented is dependent upon the relative phase of the two signals as illustrated in figure 14.

The μPD78356 family has programmable noise detection on the external clock inputs and external clear inputs to the RPU. The noise detection time can be set for each input at 4 or 16 internal system clocks by the noise protection control register (NPC).

Interrupts

The μPD78356 family has 24 maskable hardware interrupt sources: 5 software selectable as external or internal and 19 internal. The four external maskable interrupts share pins with port 2. Any of them, INTP0 and INTP4, can also be used to trigger capture events in the real-time pulse unit. 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 3.

Table 3. 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	003CH
			BRK instruction	CPU		003EH	003EH
			BRKCS instruction (Initiates context switch)	CPU			
Nonmaskable		NMI	NMI input pin	External		0002H	8002H
		INTWDT	Watchdog timer overflow	Internal		0004H	8004H
Maskable	0	INTOV0	Timer 0 overflow	Internal	FE06H	0006H	8006H
	1	INTOV3	Timer 3 overflow	Internal	FE06H	0006H	8006H
	2	INTP0 INTCC00	INTP0 pin CC00 coincidence	External Internal	FE0AH	000AH	800AH
	3	INTP1 INTCC01	INTP1 pin CC01 coincidence	External Internal	FE0CH	000CH	800CH
	4	INTP2 INTCC02	INTP2 pin CC02 coincidence	External Internal	FE0EH	000EH	800EH
	5	INTP3 INTCC30	INTP3 pin CC30 coincidence	External Internal	FE10H	0010H	8010H
	6	INTP4 INTCC31	INTP4 pin CC31 coincidence	External Internal	FE12H	0012H	8012H
	7	INTCM00	CM00 coincidence	Internal	FE14H	0014H	8014H
	8	INTCM01	CM01 coincidence	Internal	FE16H	0016H	8016H
	9	INTCM02	CM02 coincidence	Internal	FE18H	0018H	8018H
	10	INTCM03	CM03 coincidence	Internal	FE1AH	001AH	801AH
	11	INTCM10	CM10 coincidence	Internal	FE1CH	001CH	801CH
	12	INTCM11	CM11 coincidence	Internal	FE1EH	001EH	801EH
	13	INTCM20	CM20 coincidence	Internal	FE20H	0020H	8020H
	14	INTCM21	CM21 coincidence	Internal	FE22H	0022H	8022H
	15	INTCM40	CM40 coincidence	Internal	FE24H	0024H	8024H
	16	INTCMUD0	CMUD0 coincidence	Internal	FE26H	0026H	8026H
	17	INTCMUD1	CMUD1 coincidence	Internal	FE28H	0028H	8028H
	18	INTSER	Asynchronous serial interface reception error	Internal	FE2AH	002AH	802AH
	19	INTSR	End of asynchronous serial interface reception	Internal	FE2CH	002CH	802CH
	20	INTST	End of asynchronous serial interface transmission	Internal	FE2EH	002EH	802EH
	21	INTCSI0	End of clocked serial interface CSI0 transmission/reception	Internal	FE30H	0030H	8030H
	22	INTCSI1	End of clocked serial interface CSI1 transmission/reception	Internal	FE32H	0032H	8032H
	23	INTAD	End of A/D conversion	Internal	FE34H	0034H	8034H
Reset		RESET	RESET pin	External		0000H	0000H

* 0 is the highest priority.

Interrupt Servicing

The μPD78356 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 μPD78356 family has 24 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 15.

There are also three mask flag register, MK0L, MK0H, and MK1L, 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 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 15. 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		**Interrrupt Mask Flag**
0		Interrupt request enabled
1		Interrupt will be pending
xxISMxx		**Macro Service Enable**
0		Software interrrupt
1		Macro service
xxCSExx		**Context Switch Enable**
0		Vector interrupt
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 than 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.

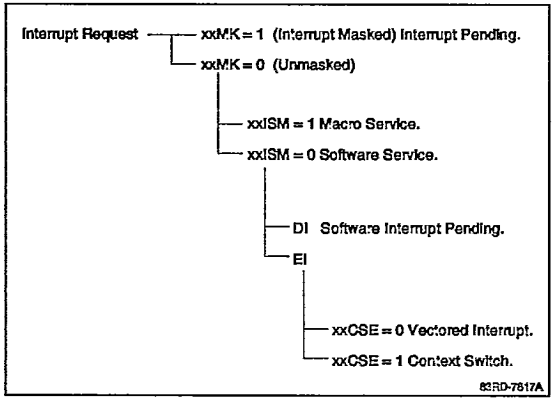
Interrupt requests programmed to be handled by macro service have priority over all software interrupt service regardless of the assigned priority level, and

macro service requests are accepted even when the interrupt enable bit in the PSW is set to the disable state. See figure 16.

The default priorities listed in table 3 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 16. Interrupt Service Sequence



Vectored Interrupt

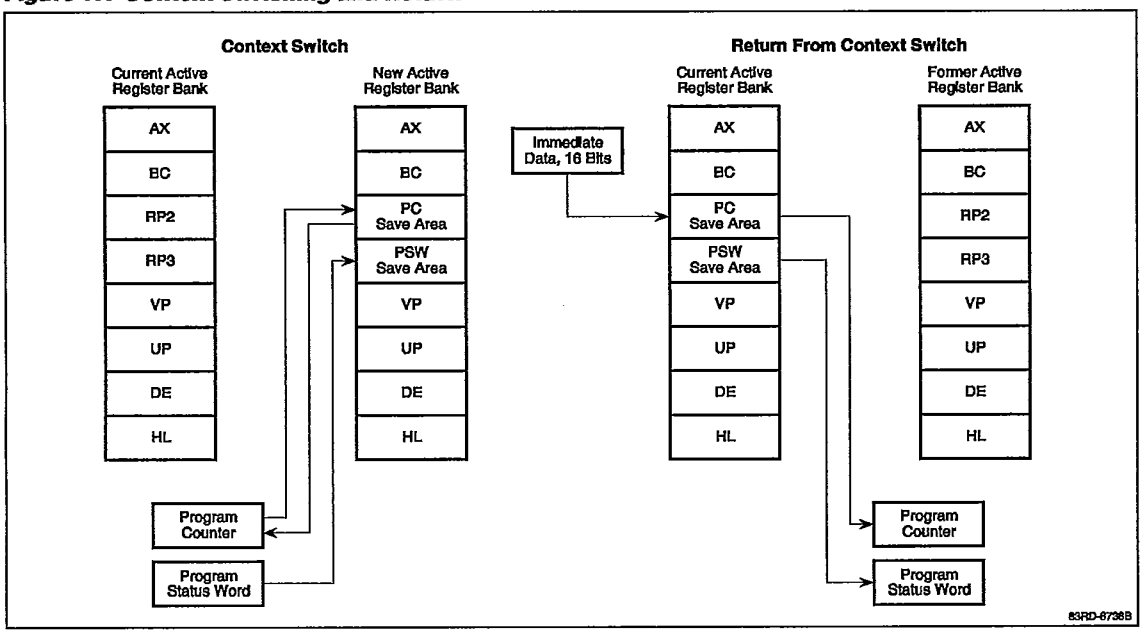
When vectored interrupt is specified for a given interrupt request, (1) the program status word and the program counter are saved on the stack, (2) the processor's priority is raised to that specified for the interrupt, (3) the IE bit in the PSW is set to zero, and (4) the routine whose address is in 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 μPD78356 family device resumes the interrupted routine. The corresponding interrupt request flag is cleared before executing the interrupt service routine.

Context Switch

When context switching (figure 17) 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 reverses 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 17. 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 on-chip main RAM. The function to be performed is specified in the control word.

The μPD78356 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, any nonmaskable interrupt, 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 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, watchdog timer output pin $\overline{WDT0}$ goes active low for a period of 32 system clocks. The $\overline{WDT0}$ pin can be connected to the $\overline{RESET}$ pin or used to control external circuitry. Three program-selectable intervals are available: 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 μPD78356 family is reset by taking the $\overline{RESET}$ pin low. The reset circuit contains a noise filter to protect against spurious system resets caused by noise. On power-up, the $\overline{RESET}$ 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 $\overline{RESET}$ pin going high. While $\overline{RESET}$ is low, all external lines except $\overline{WDTO}$, CLKOUT, V_{SS}, V_{DD}, AV_{SS}, AV_{DD}, AV_{REF1}, AV_{REF2}, AV_{REF3}, X1, and X2 are in the high-impedance state.

ELECTRICAL SPECIFICATIONS (Preliminary)

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 78P356	−0.5 to V_{DD} + 0.5 V
$P2_0$/NMI (A9) of 78P356	−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	140 mA
Output current, high; I_{OH}	
Each output pin	−1.0 mA
Total	30 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.

Operating Conditions

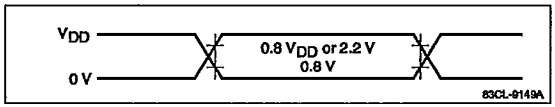
Oscillator Frequency, f_{XX}	T_A	V_{DD}
8 to 32 MHz	−10 to +70°C	+5.0 V ±10%

Capacitance

T_A = 25°C; V_{DD} = V_{SS} = 0 V

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

AC Timing Test Points



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}		75	107	mA	Operating mode; 78355/356
			86	125	mA	Operating mode; 78P356
	I_{DD2}		40	60	mA	HALT mode; 78355/356
			40	60	mA	HALT mode; 78P356
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 $\overline{RESET}$, X1, $P0_0$ - $P0_3$, P2, $P3_2$ - $P3_6$, $P8_0$ - $P8_1$, $P8_5$, $P10_1$, $P10_2$, $P10_4$ - $P10_7$.

AC Characteristics

T_A = −10 to +70°C; V_{DD} = +5.0 V ±10%; V_{SS} = 0 V; f_{XX} = 32 MHz

Parameter	Symbol	Calculation Formula	Min	Max	Unit	Conditions
External Memory Read/Write Operation						
System clock cycle time (Note 1)	t_{CYK}	—	62.5	250	ns	C_L = 50 pF
Address setup time to ASTB ↓	t_{SAST}	(0.5 + a)T − 24	7		ns	C_L = 100 pF (Note 2)
Address hold after ASTB ↓	t_{HSTA}	0.5T − 16	15		ns	C_L = 100 pF
$\overline{RD}$ ↓ to address floating	t_{FRA}	—		0	ns	C_L = 100 pF
Address to data input valid	t_{DAID}	(2.5 + a + n)T − 56		100	ns	C_L = 100 pF (Note 2, 3)
$\overline{RD}$ ↓ to data input valid	t_{DRID}	(1.5 + n)T − 44		49	ns	C_L = 100 pF (Note 3)
ASTB ↓ to $\overline{RD}$ ↓ delay time	t_{DSTR}	0.5T − 16	15		ns	C_L = 100 pF
Data hold time from $\overline{RD}$ ↑	t_{HRID}	—	0		ns	C_L = 100 pF
$\overline{RD}$ ↑ to next address active	t_{DRA}	0.5T − 14	17		ns	C_L = 100 pF
$\overline{RD}$ width low	t_{WRL}	(1.5 + n)T − 30	63		ns	C_L = 100 pF (Note 3)
ASTB width high	t_{WSTH}	(0.5 + a)T − 17	14		ns	C_L = 100 pF (Note 2)
$\overline{LWR}$ or $\overline{HWR}$ to data output	t_{DWOD}	0.5T − 10		21	ns	C_L = 100 pF
ASTB ↓ to $\overline{LWR}$ ↓ or $\overline{HWR}$ ↓ delay	t_{DSTW}	0.5T − 16	15		ns	C_L = 100 pF
Data setup time to $\overline{LWR}$ ↑ or $\overline{HWR}$ ↑	t_{SODW}	(1 + n)T − 5	57		ns	C_L = 100 pF (Note 3)
Data hold time after $\overline{LWR}$ ↑ or $\overline{HWR}$ ↑	t_{HWOD}	—	8		ns	C_L = 100 pF
$\overline{LWR}$ or $\overline{HWR}$ width, low	t_{WWL}	(1.5 + n)T − 30	63		ns	C_L = 100 pF (Note 2, 3)
$\overline{WAIT}$ setup time from address	t_{SAWT}	(a + n)T − 15		110	ns	C_L = 100 pF (Notes 2, 4)
$\overline{WAIT}$ hold time from address	t_{HAWT}	(0.5 + a + n)T	156		ns	C_L = 100 pF (Notes 2, 4)

AC Characteristics (cont)

Parameter	Symbol	Calculation Formula	Min	Max	Unit	Conditions
$\overline{WAIT}$ setup time from $\overline{RD}$ ↓ or $\overline{WR}$ ↓	t_{SRWRY}	(n – 1)T – 25		37	ns	C_L = 100 pF (Note 4)
$\overline{WAIT}$ hold time from $\overline{RD}$ ↓ or $\overline{WR}$ ↓	t_{HRWRY}	(n – 0.5)T – 14	79		ns	C_L = 100 pF (Note 4)
ASTB ↑ delay time from $\overline{WR}$ ↑	t_{DWST}	1.5T – 15	TBD		ns	C_L = 100 pF
Serial Port Operation						
$\overline{SCK}$ cycle time	t_{CYSK}	—	8T		ns	$\overline{SCK}$ output
			500		ns	$\overline{SCK}$ input
$\overline{SCK}$ width, low	t_{WSKL}	—	4T – 40		ns	$\overline{SCK}$ output
			210		ns	$\overline{SCK}$ input
$\overline{SCK}$ width, high	t_{WSKH}	—	4T – 40		ns	$\overline{SCK}$ output
			210		ns	$\overline{SCK}$ input
SI setup time to $\overline{SCK}$ ↑	t_{SRXSK}	—	80		ns	
SI hold time after $\overline{SCK}$	t_{HSKRX}	—	80		ns	
$\overline{SCK}$ ↓ to SO delay time	t_{DSKTX}	—		110	ns	

Definitions:

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

(2) a = address wait state: a = 0 or 1

(3) n = number of wait states specified by the external wait pin $\overline{WAIT}$ and the PWC register

Notes:

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

(2) No address wait state

(3) No wait states

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

A/D Converter Characteristics

T_A = –10 to +70°C; AV_{DD} = V_{DD} = +5 V ±10% or +3 V ±10%; V_{SS} = AV_{SS} = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Resolution				10	Bit	
Total error*				1.2	%	
Quantization error				±1/2	LSB	
Conversion time	t_{CONV}	32			t_{CYK}	AD trigger mode
		37			t_{CYK}	Timer/external trigger mode
Sampling time	t_{SAMP}	7.5			t_{CYK}	
Analog input voltage	A_{IAN}	–0.3		AV_{REF1} + 0.3	V	
Analog input impedance	R_{AN}		TBD		MΩ	
Reference voltage	AV_{REF1}	TBD		V_{DD}	V	
AV_{REF1} current	AI_{REF1}		3.0	9	mA	f_{XX} = 32 MHz
			TBD	TBD	mA	ADM0.7(CS) = 0
AV_{DD} supply current	AI_{DD}		3.3	13	mA	Operational mode

* Does not include quantization error. Percentage of full-scale value is shown.

D/A Converter Characteristics

T_A = −10 to +70°C; AV_{REF2} = V_{DD} = +5 V ±10%; AV_{REF3} = V_{SS} = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Resolution				8	Bit	
Overall error				TBD	%	
Settling time				2	μs	Load condition 2 MΩ, 30 pF
Output resistance	R_O		20		kΩ	When DACS0 and DACS1 are set to 7FH
Analog reference voltage	AV_{REF2}	0.75 V_{DD}		V_{DD}	V	
	AV_{REF3}	V_{SS}		0.2 V_{DD}	V	
Reference supply input current	AI_{REF2}	0		5	mA	
	AI_{REF3}	−5		0	mA	

Up/Down Counter Operation

T_A = −10 to +70°C; V_{DD} = +5 V ±10% V_{SS} = 0 V; T = t_{CYK} (ns)

Parameter	Symbol	Min	Max	Unit	Conditions
TIUD high-, low-level width	t_{WTIUH}, t_{WTIUL}	aT		ns	Except mode 4
		bT		ns	Mode 4
TCUD high-, low-level width	t_{WTCUH}, t_{WTCUL}	aT		ns	Except mode 4
		bT		ns	Mode 4
TCLRUD high-, low-level width	t_{WCLUH}, t_{WCLUL}	aT		ns	
TCUD setup time to TIUD	t_{STCU}	0		ns	Mode 3, rise
TCUD hold time from TIUD ↑	t_{HTCU}	2T		ns	Mode 3, rise
TIUD setup time to TCUD	t_{S4TIU}	4T		ns	Mode 4
TIUD hold time from TCUD	t_{H4TIU}	4T		ns	Mode 4
TIUD, TCUD cycle time	t_{CYC4}		1	MHz	

Note: a, b are defined by NPC register.
NPC.bit = 0, then a = 4, b = 8; NPC.bit = 1, then a = b = 16.

Other Operations

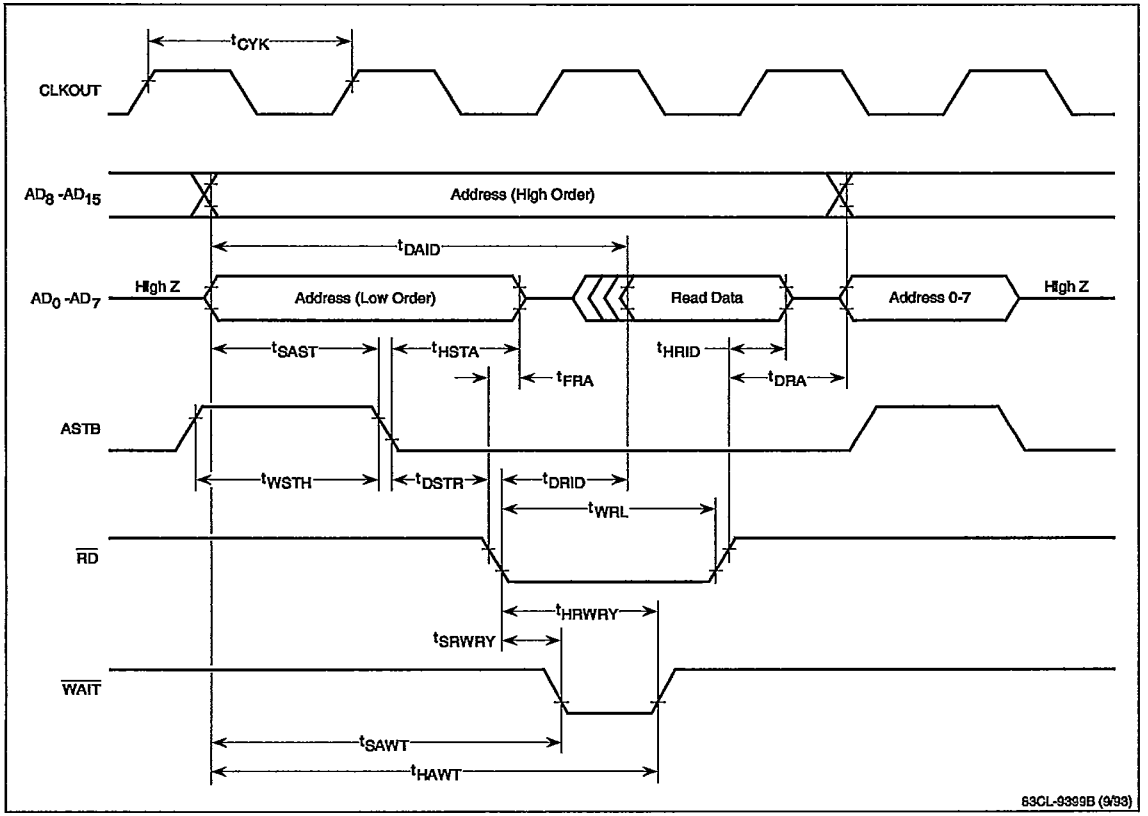
T_A = −10 to +70°C; V_{DD} = +5 V ±10%; V_{SS} = 0 V; T = t_{CYK} (ns)

Parameter	Symbol	Min	Max	Unit	Conditions
NMI high-, low-level width	t_{WNIH}, t_{WNIL}	2		μs	
INTP0 to INTP4 high-, low-level width	t_{WINH}, t_{WINL}	4T		ns	
$\overline{\text{RESET}}$ high-, low-level width	t_{WRSH}, t_{WRSL}	2		μs	
TIn high-, low-level width	t_{WTINH}, t_{WTINL}	aT		ns	
TCLRn high-, low-level width	t_{WCLNH}, t_{WCLNL}	aT		ns	
ADTRGn high-, low-level width	t_{WADNH}, t_{WADNL}	4T		ns	
ADC external trigger inputs ADTRGn to ADTRGm (valid edge to valid edge)	$t_{ADCININ}$	37T		ns	

Note: a is defined by NPC register (a = 4 or 16).

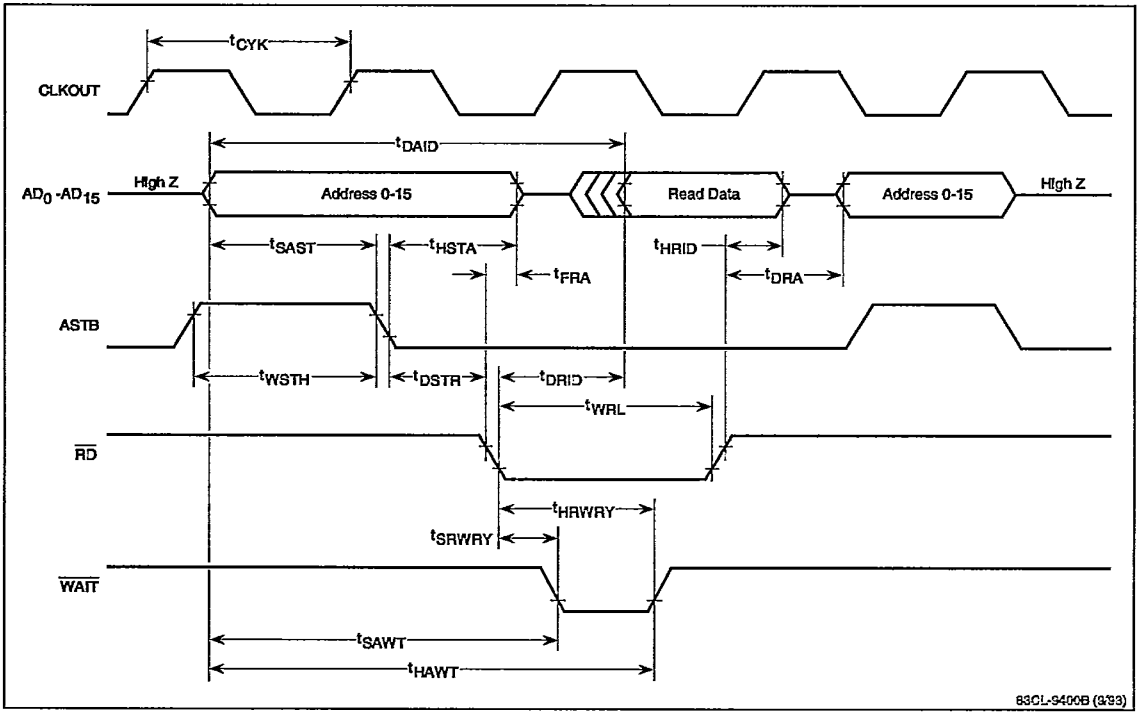
Timing Waveforms

Read Operation (8-Bit Bus)



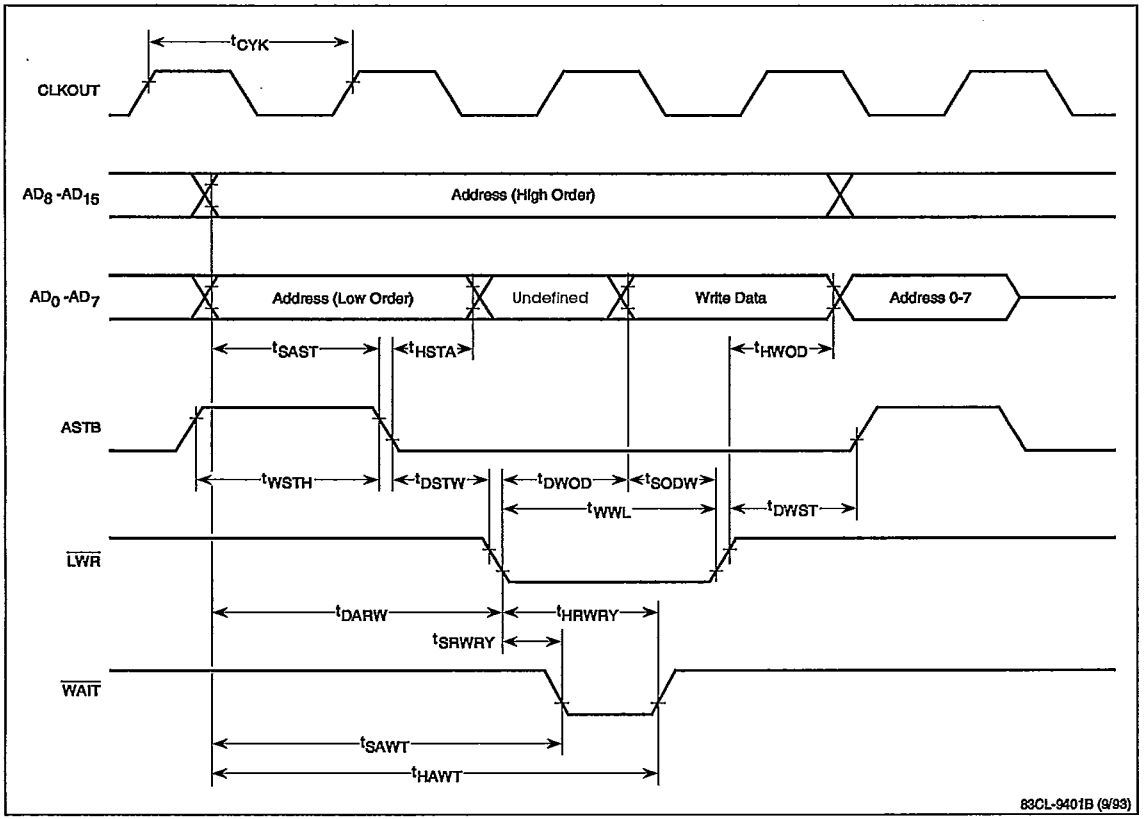
Timing Waveforms (cont)

Read Operation (16-Bit Bus)



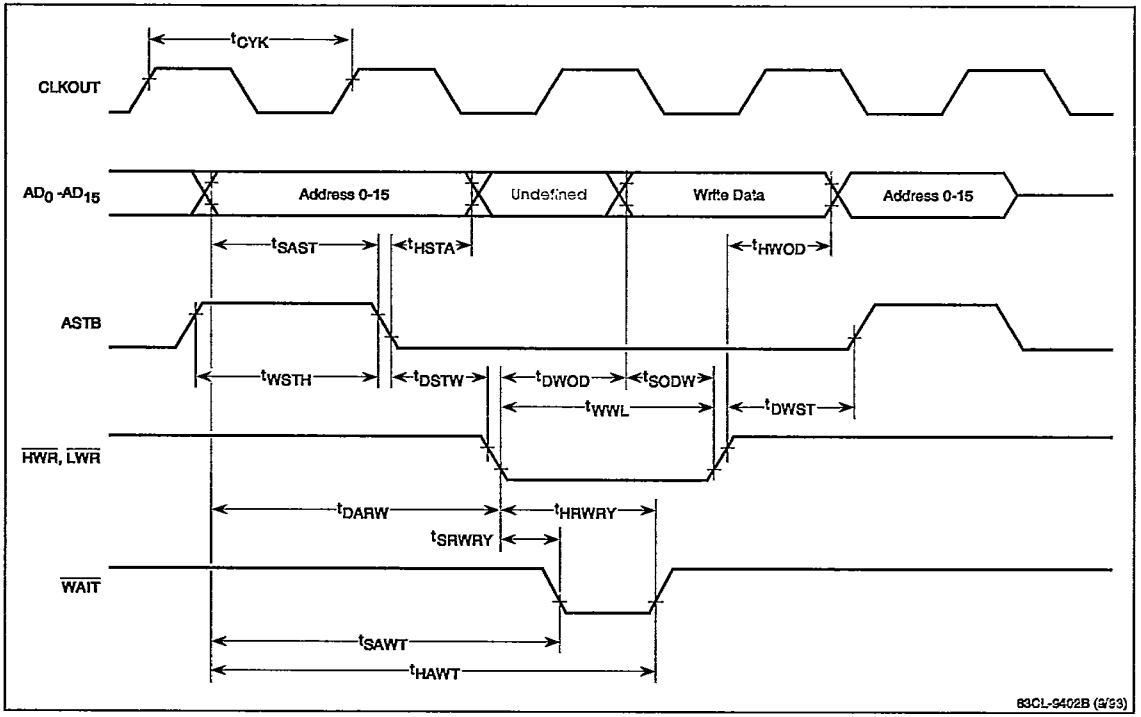
Timing Waveforms (cont)

Write Operation (8-Bit Bus)



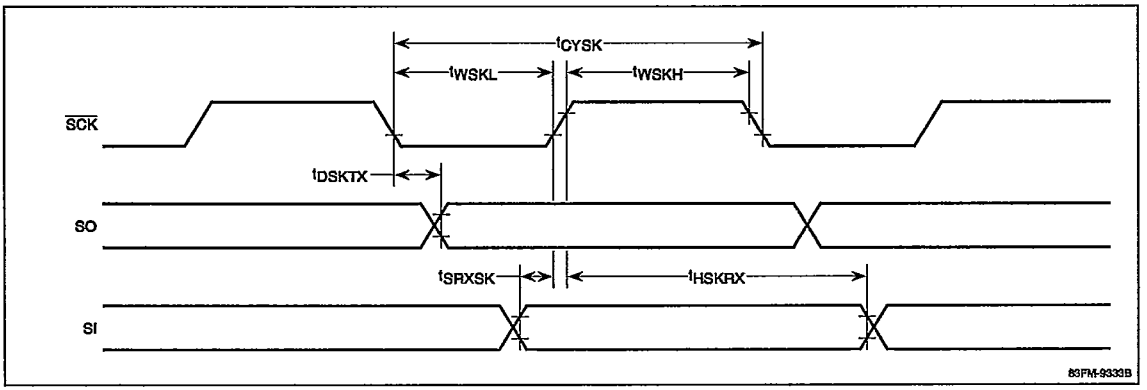
Timing Waveforms (cont)

Write Operation (16-Bit Bus)

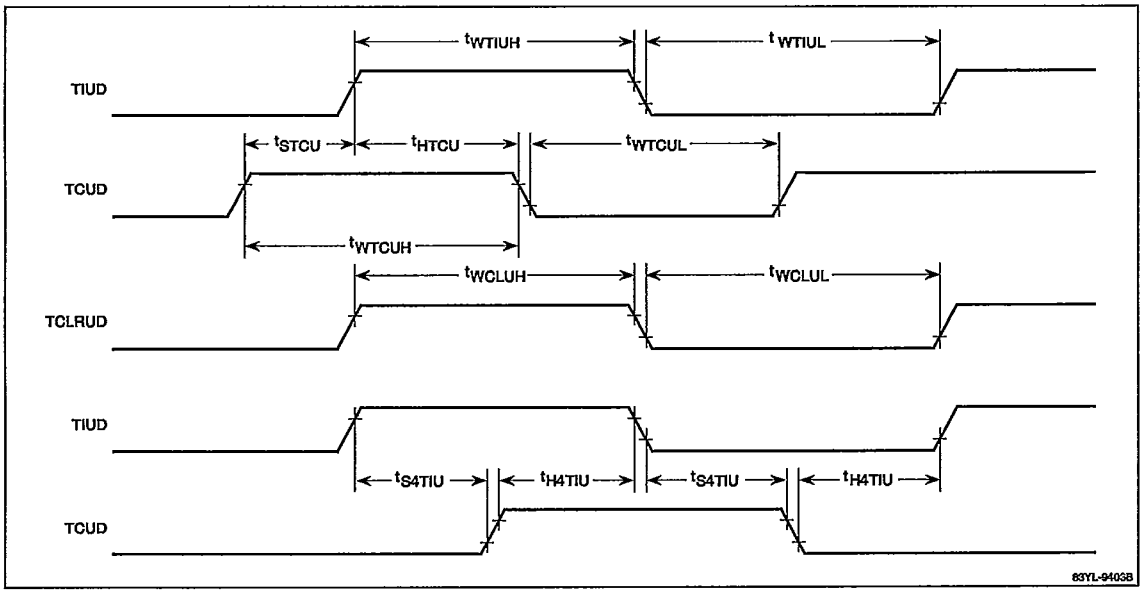


Timing Waveforms (cont)

Serial Port Operation, Clock Synchronous Mode

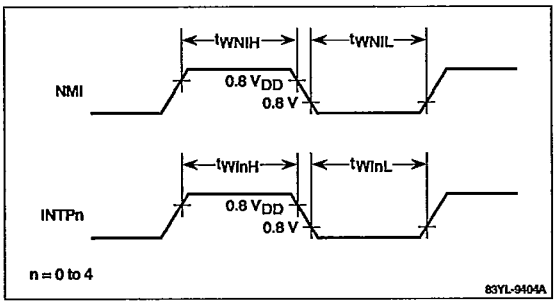


Up/Down Counter Input

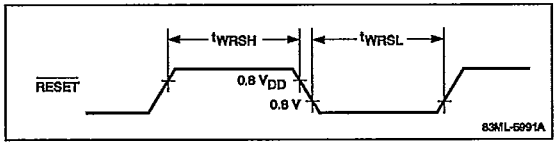


Timing Waveforms (cont)

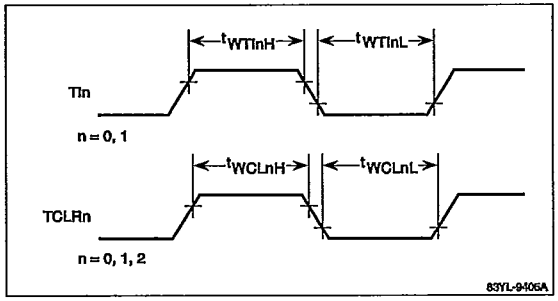
Interrupt Input



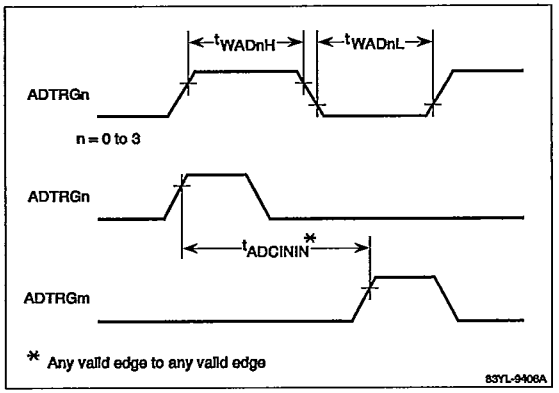
Reset Input



Timer Input



A/D Converter Trigger Input



PROM PROGRAMMING

The PROM in the μPD78P356 is one-time programmable (OTP) or ultraviolet erasable (UV EPROM). The 49,152 x 8-bit PROM has the programming characteristics of an NEC μPD27C1001A, including both page and byte programming modes. Table 4 shows the functions of the μPD78P356 pins in normal operating mode and PROM programming mode.

Table 4. 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	$P1_0$	$\overline{CE}$
Output enable	$P1_1$	$\overline{OE}$
Program voltage	MODE0/V_{PP}	MODE0/V_{PP}
Mode voltage	MODE1, $P2_1$, $\overline{RESET}$	MODE1, $P2_1$, $\overline{RESET}$

The μPD78P356 also includes a PROM error correction function capable of correcting one 1-bit error per four bytes of code. Each device contains 49,152 bytes of PROM for program storage (0000H to BFFFH) and 12,288 bytes of PROM for error correction code (C000H to EFFFH). A four-byte ECC Control Word (ECW) is located at addresses F000H to F003H and controls the ECC circuitry. For additional protection, four bytes of ECC data for the ECW are located at addresses F004H to F007H. The error correction code and information to be stored in the ECW are automatically generated and added to your HEX file by the ECCGEN program supplied with the RA78K3 Relocatable Assembler Package.

PROM Programming Mode

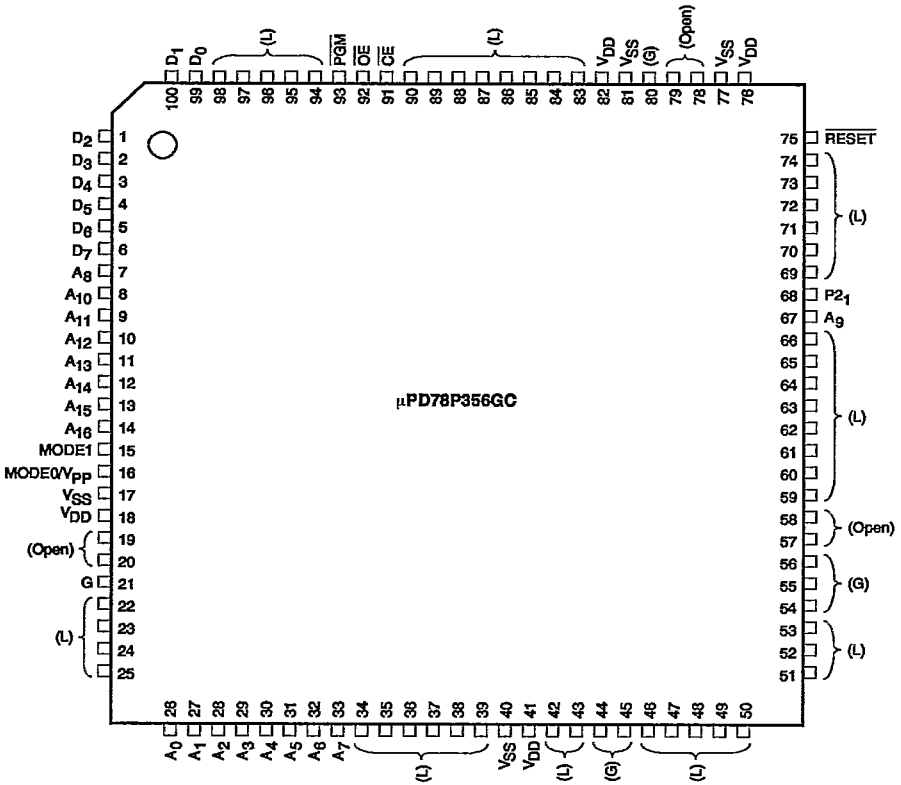
When MODE1, $P2_1$, and $\overline{RESET}$ pins are set low, the μPD78P356 enters the PROM programming mode. Operation in this mode is determined by the setting of $\overline{CE}$, $\overline{OE}$, $\overline{PGM}$, MODE0/V_{PP}, and V_{DD} pins as indicated in table 5.

Table 5. 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 18. Pin Functions in μPD78P356 PROM Programming Mode; 100-Pin Plastic QFP



Notes:

Recommended connections for pins not used during PROM programming:

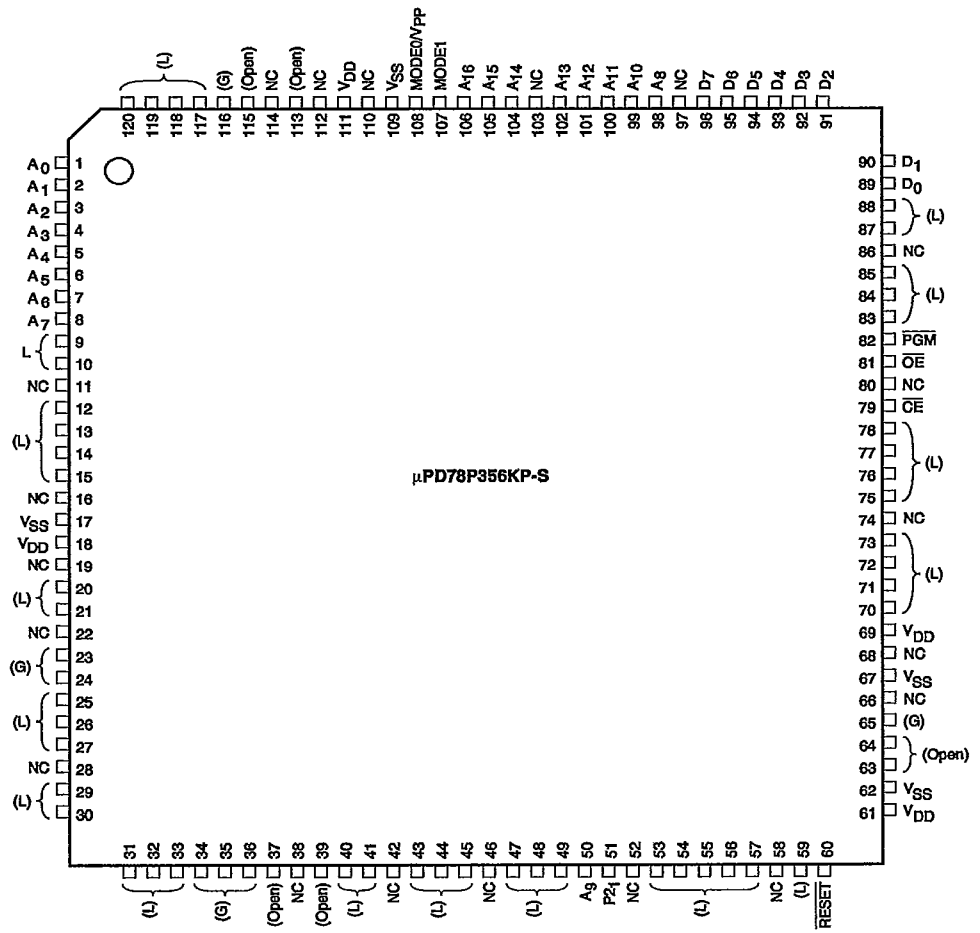
(L) Connect each pin through a resistor to V_{SS}.

(G) Connect to V_{SS}.

(Open) No connection.

83YL-9407B

Figure 19. Pin Functions in μPD78P356 PROM Programming Mode; 120-Pin Ceramic LCC



Notes:

Recommended connections for pins not used during PROM programming:

- (L) Connect each pin through a resistor to V_{SS}.
- (G) Connect to V_{SS}.
- (Open) No connection.
- NC Connect to V_{SS} to prevent noise.

83YL-9408B (10/93)

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 figures 18 and 19. 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 figures 18 and 19. 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 figures 18 and 19. 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 fluorescent 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 $\overline{\text{RESET}}$, X1, $P0_0$ - $P0_3$, P2, $P3_2$ - $P3_6$, $P8_0$ - $P8_1$, $P8_5$, $P10_1$ - $P10_2$, $P10_4$ - $P10_7$.

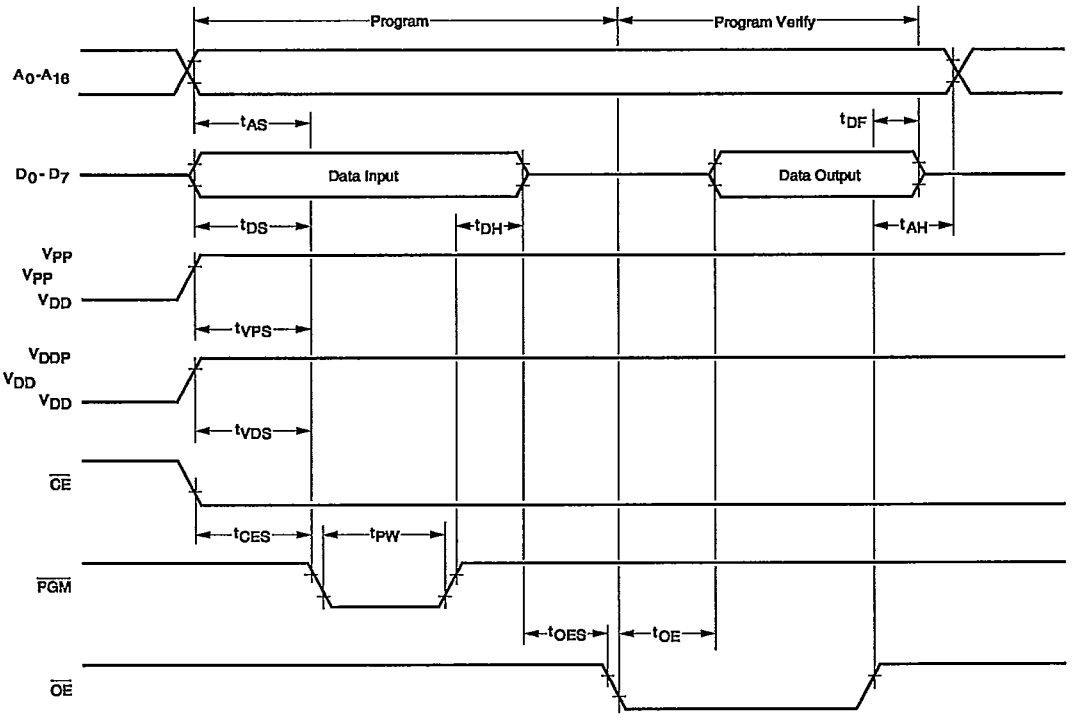
AC Programming Characteristics

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}.

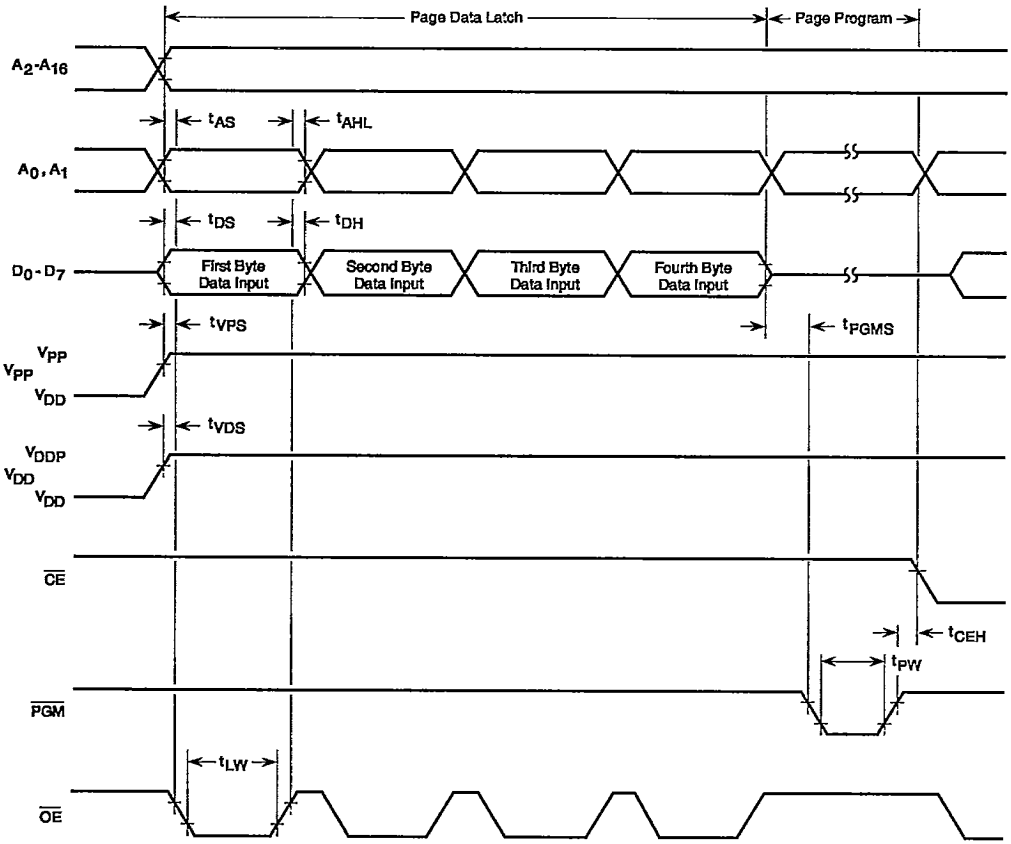
[2] V_{PP} must not be greater than +13.5 V, including overshoot.

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

83CL-9156B

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}.

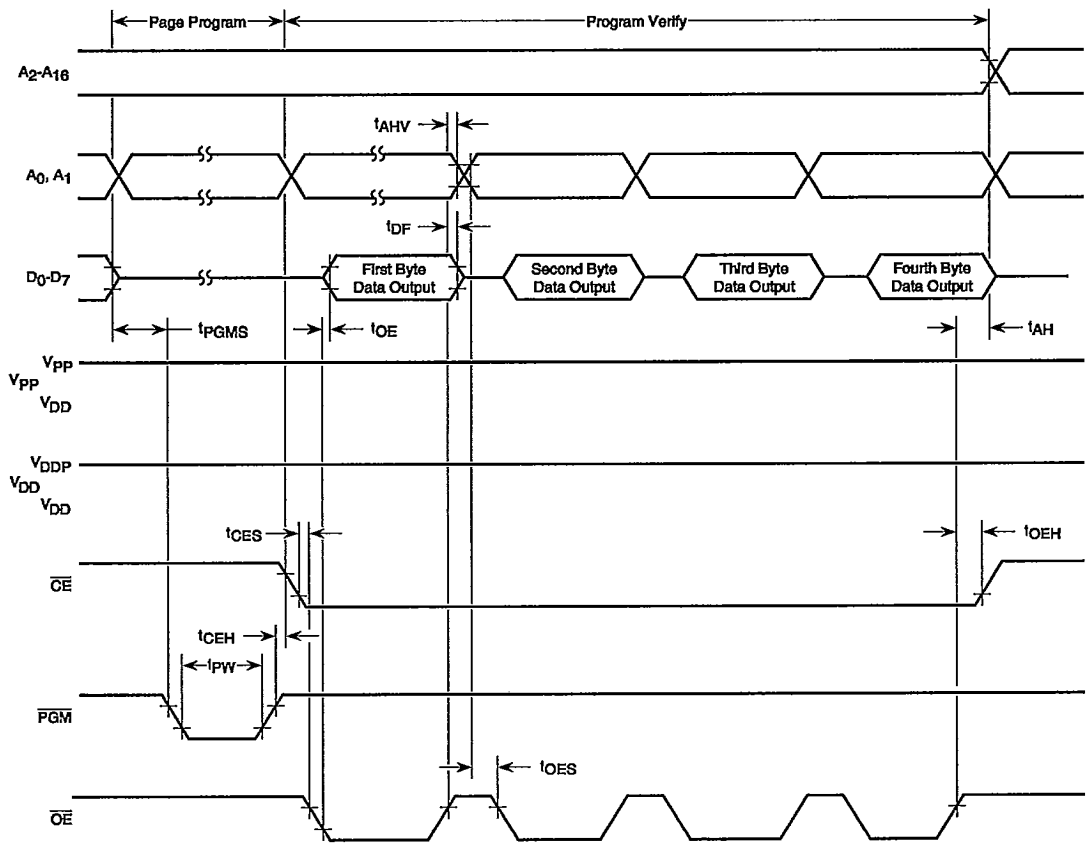
[2] V_{PP} must not be greater than +13.5 V, including overshoot.

[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}.

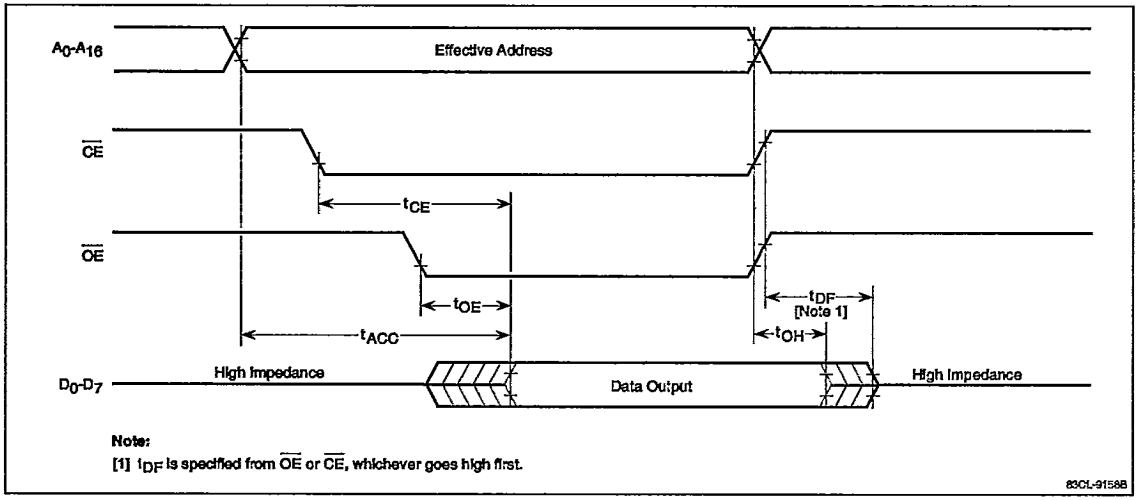
[2] V_{PP} must not be greater than +13.5 V, including overshoot.

[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 μPD78356 family is upward compatible with the μPD78322, and μPD78352 families. Four new instructions (MACW, MACSW, SACW, and MOVTBL) have been added to the μPD78322 and two (MACSW and SACW) to the μPD78352 . These additional instructions facilitate digital signal processing.

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 operation of the convolution instruction with saturation word MACSW is identical to instruction MACW except when the instuction terminates with the P/V flag set. In this case, the AXDE register will be set to 7FFFFFFFH by an overflow or 80000000H by an underflow.

Correlation instruction SACW subtracts corresponding factors of two tables and calculates the sum of the absolute values of these subtractions. Instruction MOVTBL 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.
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
laddr16	16-bit absolute address specified by an immediate address or label
$addr16	Relative branch address or label
addr16	16-bit address
laddr11	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

Instruction Set Symbols (cont)

Symbol	Definition
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
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 ← A 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
Sum-of-Products With Saturation								
MACSW	n	AXDE ← (B) x (C) + AXDE, B ← B+2, C ← C + 2, n ← n–1, if overflow (P/V = 1) then AXDE ← 7FFFFFFFH, if underflow (P/V = 1) then AXDE ← 80000000H. End if n = 0 or P/V = 1	3	x	x	x	x	x
Correlation Operation								
SACW	[DE+], [HL+]	AX ← AX + |(DE)–(HL)|, DE ← DE+2, HL ← HL + 2, C ← C–1. End if C = 0 or CY = 1	4	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					
Increment/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, $r1_7$ ← $r1_0$, $r1_{m-1}$ ← $r1_m$) x n times	2				P	X
ROL	r1, n	(CY, $r1_0$ ← $r1_7$, $r1_{m+1}$ ← $r1_m$) x n times	2				P	X
RORC	r1, n	(CY ← $r1_0$, $r1_7$ ← CY, $r1_{m-1}$ ← $r1_m$) x n times	2				P	X
ROLC	r1, n	(CY ← $r1_7$, $r1_0$ ← CY, $r1_{m+1}$ ← $r1_m$) x n times	2				P	X
SHR	r1, n	(CY ← $r1_0$, $r1_7$ ← 0, $r1_{m-1}$ ← $r1_m$) x n times	2	X	X	0	P	X
SHL	r1, n	(CY ← $r1_7$, $r1_0$ ← 0, $r1_{m+1}$ ← $r1_m$) x n times	2	X	X	0	P	X
SHRW	rp1, n	(CY ← $rp1_0$, $rp1_{15}$ ← 0, $rp1_{m-1}$ ← $rp1_m$) x n times	2	X	X	0	P	X
SHLW	rp1, n	(CY ← $rp1_{15}$, $rp1_0$ ← 0, $rp1_{m+1}$ ← $rp1_m$) x n times	2	X	X	0	P	X
ROR4	[rp1]	A_{3-0} ← $(rp1)_{3-0}$, $(rp1)_{7-4}$ ← A_{3-0}, $(rp1)_{3-0}$ ← $(rp1)_{7-4}$	2					
ROL4	[rp1]	A_{3-0} ← $(rp1)_{7-4}$, $(rp1)_{3-0}$ ← A_{3-0}, $(rp1)_{7-4}$ ← $(rp1)_{3-0}$	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} ← A_7	1					

Instruction Set (cont)

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

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Bit Manipulation (cont)								
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
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					

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Subroutine Linkage (cont)								
BRK		(SP–1) ← PSWH, (SP – 2) ← PSWL, (SP – 3) ← $(PC + 1)_H$, (SP – 4) ← $(PC + 1)_L$, PC_L ← (003EH), PC_H ← (003FH), SP ← SP – 4, IE ← 0	1					
RET		PC_L ← (SP), PC_H ← (SP + 1), SP ← SP + 2	1					
RETB		PC_L ← (SP), PC_H ← (SP + 1), PSWL ← (SP + 2), PSWH ← (SP + 3), SP ← SP + 4	1	R	R	R	R	R
RETI		PC_L ← (SP), PC_H ← (SP + 1), PSWL ← (SP + 2), PSWH ← (SP + 3), SP ← SP + 4	1	R	R	R	R	R
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) $\forall$ (internal signal level)	3	X	X		P	
CHKLA	sfr	A ← (Pin level) $\forall$ (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					

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Conditional Branch (cont)								
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 ∀ S) ∨ Z = 0	3					
BGE	$addr16	PC ← PC + 3 + jdisp8 if P/V ∀ S = 0	3					
BLT	$addr16	PC ← PC + 3 + jdisp8 if P/V ∀ S = 1	3					
BLE	$addr16	PC ← PC + 3 + jdisp8 if (P/V ∀ S) ∨ Z = 1	3					
BH	$addr16	PC ← PC + 3 + jdisp8 if Z ∨ CY = 0	3					
BNH	$addr16	PC ← PC + 3 + jdisp8 if Z ∨ CY = 1	3					
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

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Conditional Branch (cont)								
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_{2-0} ← 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
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

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
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					

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.

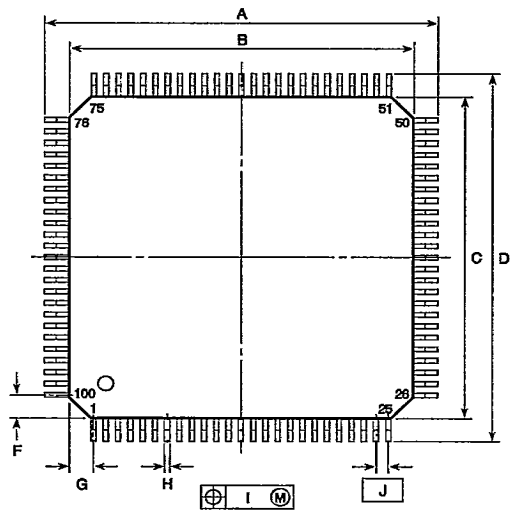
(5) rpp refers to register pairs specified in post byte. "n" is the number of register pairs specified in post byte.

(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.

PACKAGE DRAWINGS

100-Pin Plastic QFP (Dwg No. P100GC-50-7EA)

Item	Millimeters	Inches
A	16.0 ± 0.4	.630 ± .016
B	14.0 ± 0.2	.551 +.009 −.008
C	14.0 ± 0.2	.551 +.009 −.008
D	16.0 ± 0.4	.630 ± .016
F	1.0	.039
G	1.0	.039
H	0.20 ± 0.10	.008 ± .004
I	0.08	.003
J	0.5 (T.P.)	.020 (T.P.)
K	1.0 ± 0.2	.039 +.009 −.008
L	0.5 ± 0.2	.020 +.008 −.009
M	0.15 ± 0.05	.006 ± .002
N	0.10	.004
P	1.45	.057
Q	0.1 ±0.1	.004 ± .004
S	1.7 max	.067 max



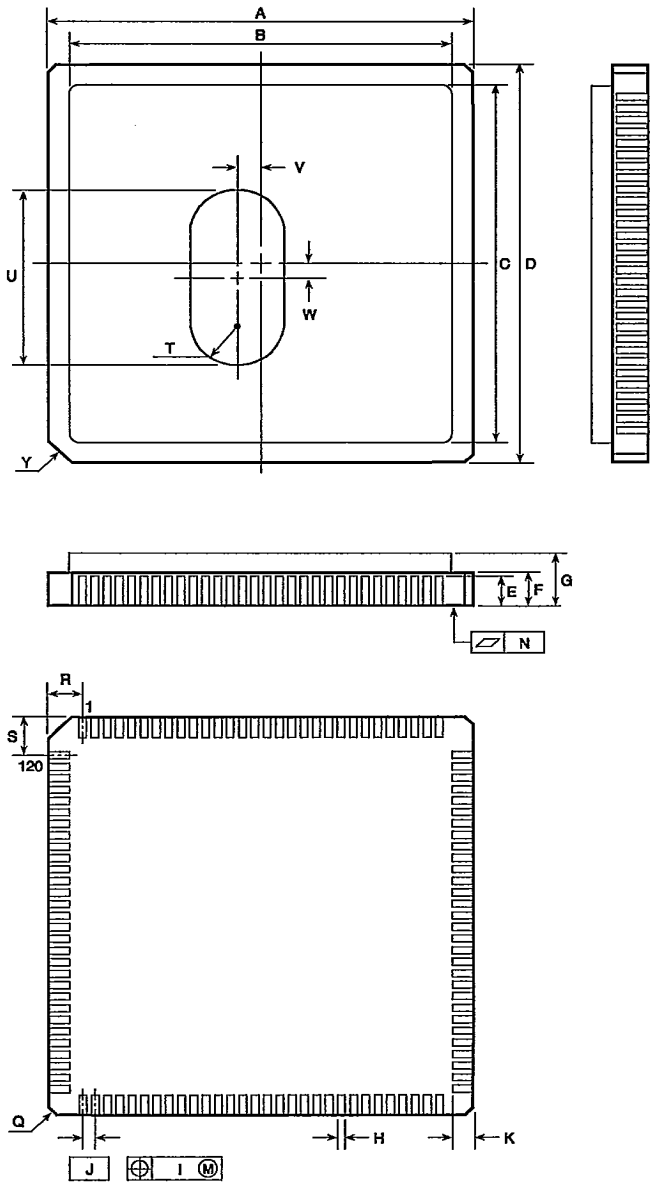
P100GC-50-7EA

83RD-7596B (6/93)

PACKAGE DRAWINGS (cont)

120-Pin Ceramic LCC With Window (Dwg No. X120KW-80A)

Item	Millimeters	Inches
A	27.3 ± 0.27	1.075 ± .011
B	24.5	.965
C	24.5	.965
D	27.3 ± 0.27	1.075 ± .011
E	1.94	.076
F	2.14	.084
G	3.57 max	.141 max
H	0.51 ± 0.10	.020 ± .004
I	0.08	.003
J	0.8 (TP)	.031 (TP)
K	1.0 ± 0.15	.039 ± .006
N	0.10	.004
Q	0.3 cor	.012 cor
R	2.05	.081
S	2.05	.081
T	3.0 rad	.118 rad
U	12.0	.472
V	1.5	.060
W	1.0	.039
Y	1.0 cor	.039 cor



X120KW-80A

83YL-9409B (5/93)