NEC
NEC Electronics Inc.

μPD78322 Family (μPD78320/322/P322) 16/8-Bit, K-Series Microcomputers With A/D Converter, Real-Time Output Ports

July 1993

Description

The μPD78320, μPD78322, and μPD78P322 are members of the K-Series® of microcomputers. These 16/8-bit microcomputers—with a minimum instruction time of 250 ns at 25 MHz—are designed for high-speed, real-time process control. They feature a 16-bit CPU, an 8-bit external data bus, eight banks of main registers, an advanced interrupt handling facility, and a powerful set of memory-mapped on-chip peripherals.

On-board memory includes 640 bytes of RAM and 16K bytes of mask ROM, UV EPROM, or one-time programmable (OTP) ROM. A ROMless version is also available.

The advanced interrupt handling facility has three levels of programmable hardware priority control and three methods of servicing interrupt requests, including vectoring, hardware context switching, and macro service.

The macro service facility reduces the CPU overhead involved in servicing peripheral interrupts by transferring data between the memory-mapped special function registers (SFRs) and memory without the use of time consuming interrupt service routines. In addition, the macro service facility can be used to perform certain CPU functions, such as event counting, math-oriented data alterations, data comparisons, or A/D converter buffering .

The combination of 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.

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

Features

- Complete single-chip microcomputer
  - 16-bit ALU
  - 640 bytes of RAM
  - 16K bytes of ROM (μPD78322/P322)
- Powerful instruction set
  - 16-bit multiply and divide
  - 1-bit and 8-bit logic instructions
  - String instructions
- Minimum instruction time: 250 ns at 16 MHz
- 3-byte instruction prefetch queue
- Memory expansion
  - 8085 bus-compatible
  - 64K-byte address space
- Large I/O capacity
  - Up to 55 I/O port lines (μPD78322/P322)
  - Up to 37 I/O port lines (μPD78320)
- Memory-mapped on-chip peripherals (special function registers)
- Real-time pulse unit
  - 16/18-bit free-running timer
  - 16-bit timer/event counter
  - Six 16-bit compare registers
  - Four 18-bit capture registers
  - Six external interrupt/capture lines
  - One external event counter/interrupt line
  - Six timer-controlled output lines
- 10-bit, eight-channel A/D converter; on-chip sample and hold amplifier
- Two-channel serial communications interface
  - Asynchronous serial interface (UART)
  - Clock-synchronized interface
  - Full-duplex, three-wire mode
  - NEC serial bus interface (SBI) mode
  - Dedicated baud-rate generator
- Programmable priority interrupt controller (three levels)
- Three methods of interrupt service
  - Vectored interrupts
  - Context switching with hardware save of all general registers
  - Macro service mode with choice of nine different functions
- Watchdog timer with dedicated output
- STOP and HALT standby functions
- 5-volt CMOS technology

50269-1

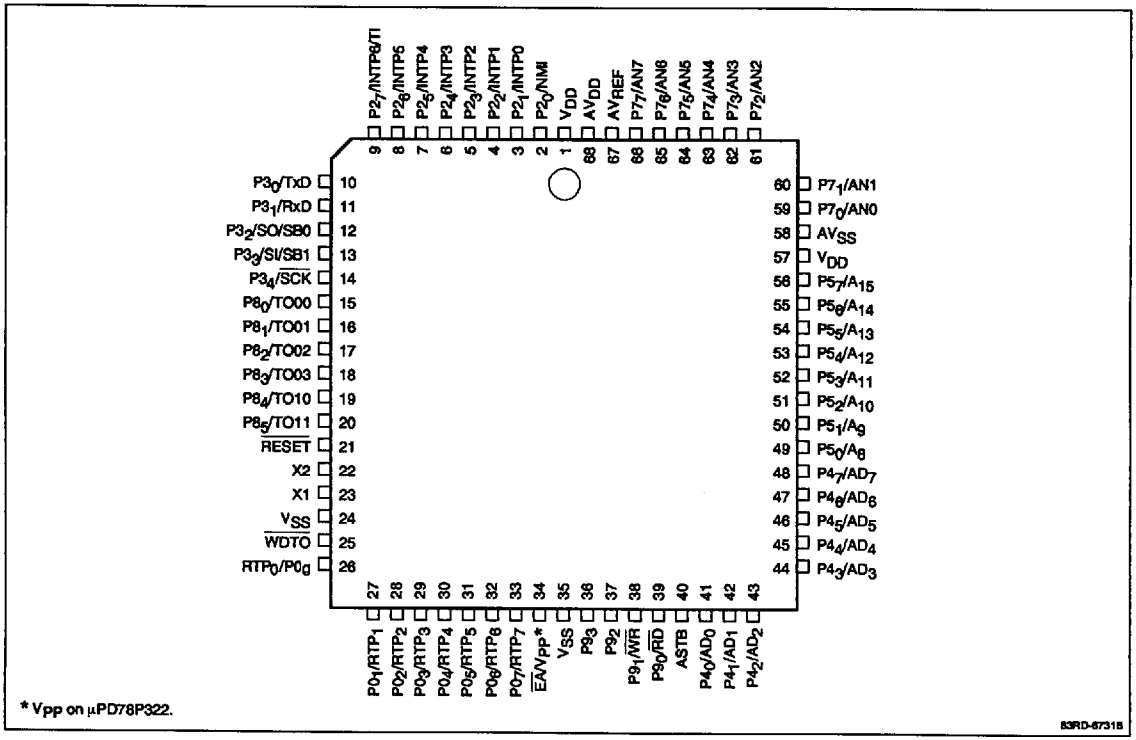
Ordering Information

Part Number	Operating Temperature Range	External Clock	Package	Package Drawing	ROM
μPD78320GF	−10 to +70°C	8 to 16 MHz	80-pin plastic QFP	P80GF-80-3B9-1	ROMless
GF(A)	−40 to +85°C				
GF(A1)	−40 to +110°C	8 to 12 MHz			
GF(A2)	−40 to +125°C				
μPD78320L	−10 to +70°C	8 to 16 MHz	68-pin PLCC	P68L-50A1-1	
L(A)	−40 to +85°C				
L(A1)	−40 to +110°C	8 to 12 MHz			
L(A2)	−40 to +125°C				
μPD78322GF-xxx	−10 to +70°C	8 to 16 MHz	80-pin plastic QFP	P80GF-80-3B9-1	16K mask ROM
GF(A)-xxx	−40 to +85°C				
GF(A1)-xxx	−40 to +110°C	8 to 12 MHz			
GF(A2)-xxx	−40 to +125°C				
μPD78322L-xxx	−10 to +70°C	8 to 16 MHz	68-pin PLCC	P68L-50A1-1	
L(A)-xxx	−40 to +85°C				
L(A1)-xxx	−40 to +110°C	8 to 12 MHz			
L(A2)-xxx	−40 to +125°C				
μPD78P322GF	−10 to +70°C	8 to 16 MHz	80-pin plastic QFP	P80GF-80-3B9-1	16K OTP ROM
L	−10 to +70°C	8 to 16 MHz	68-pin PLCC	P68L-50A1-1	
μPD78P322KE	−10 to +70°C	8 to 16 MHz	80-pin ceramic LCC w/ window	x80KW-80A	16K UV EPROM
KC	−10 to +70°C	8 to 16 MHz	68-pin ceramic LCC w/ window	x68KW-50A	

xxx indicates ROM code suffix

Pin Configurations

68-Pin PLCC or Ceramic LCC

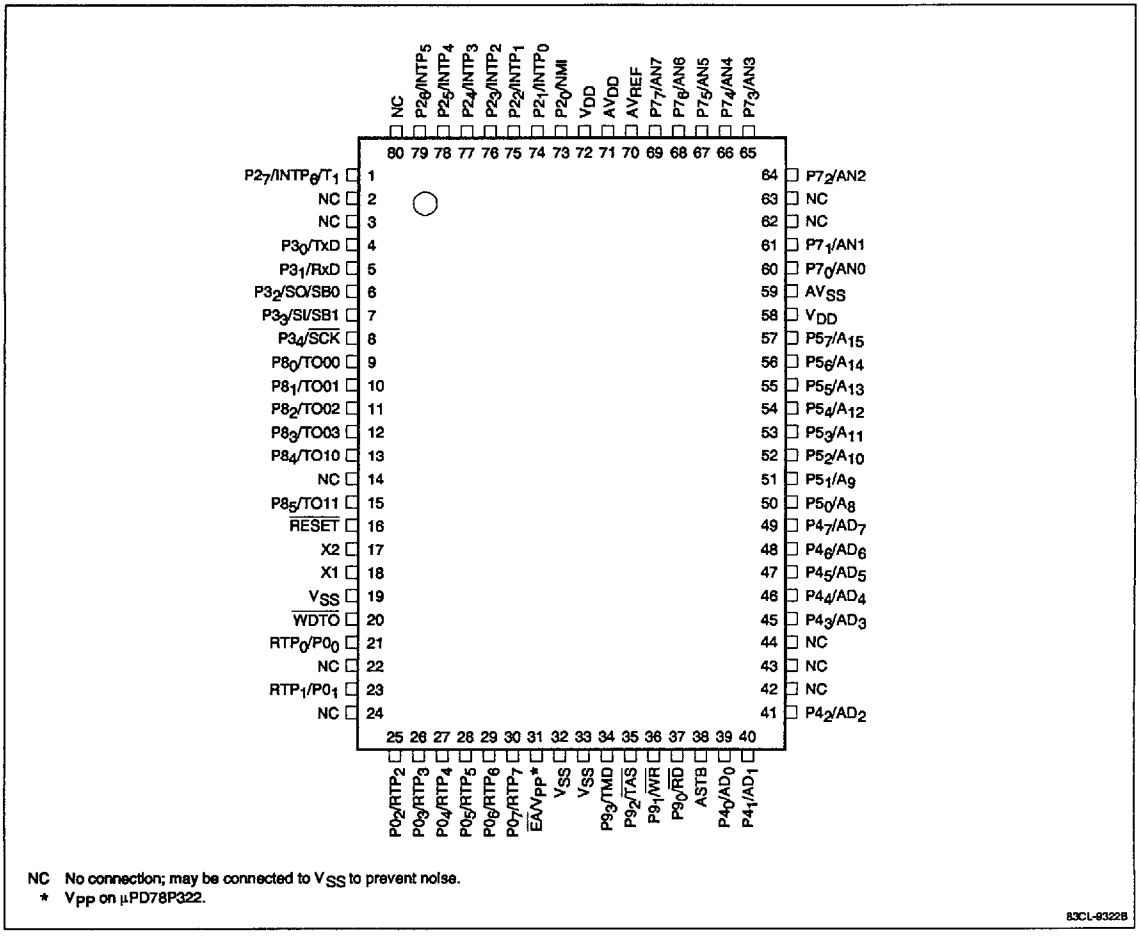


* Vpp on μPD78P322.

83RD-67315

Pin Configurations (cont)

80-Pin Plastic QFP or Ceramic LCC



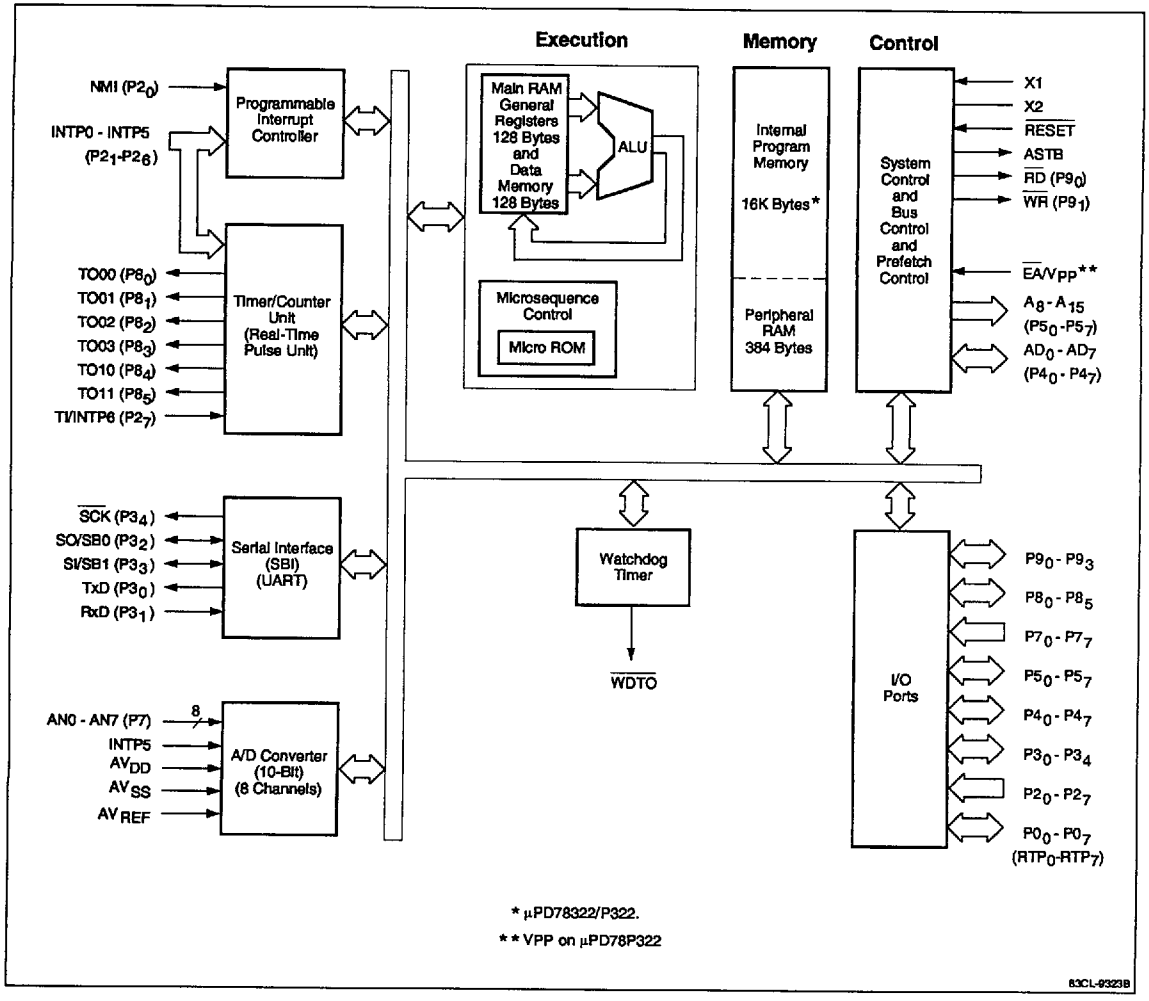
NC No connection; may be connected to V_{SS} to prevent noise.
* V_{PP} on μPD78P322.

83CL-9322B

Pin Functions

Symbol	First Function	Symbol	Second Function
$P0_0$ - $P0_7$	Port 0; 8-bit, bit-selectable I/O port	RTP_0 - RTP_7	Bit-selectable, timer-controlled, real-time output port
$P2_0$	Port 2; 8-bit input port	NMI	External nonmaskable interrupt
$P2_1$ - $P2_6$		INTP0 - INTP5	Maskable external interrupts; edge-selectable
$P2_7$		INTP6/TI	Maskable external interrupt or timer input
$P3_0$	Port 3; 5-bit, bit selectable I/O port	TxD	Asynchronous serial transmit
$P3_1$		RxD	Asysnchronous serial receive
$P3_2$		SO/SB0	SO: serial data output for 3-wire serial I/O mode. SB0: I/O bus for NEC serial bus interface (SBI).
$P3_3$		SI/SBI	SI: serial data input for 3-wire serial I/O mode. SB1: I/O bus for NEC serial bus interface (SBI).
$P3_4$		SCK	Serial clock input or output
$P4_0$ - $P4_7$	Port 4; 8-bit, byte-selectable I/O port	AD_0 - AD_7	Low-order byte of external address/data bus
$P5_0$ - $P5_7$	Port 5; 8-bit, bit-selectable I/O port	A_8 - A_{15}	High-order byte of external address bus
$P7_0$ - $P7_7$	Port 7; 8-bit input port	AN0 - AN7	Inputs for A/D converter
$P8_0$	Port 8; 6-bit, bit-selectable I/O port	TO00	Timer (RPU) output lines
$P8_1$		TO01	
$P8_2$		TO02	
$P8_3$		TO03	
$P8_4$		TO10	
$P8_5$		TO11	
$P9_0$	Port 9; 4-bit, bit-selectable I/O port	$\overline{RD}$	External read strobe
$P9_1$		$\overline{WR}$	External write strobe
$P9_2$ - $P9_3$			
ASTB	External address latch strobe		
$\overline{EA}$	External access control on μPD78320/322; a high level enables access to on-chip ROM; a low level is applied if all program memory is external. Must be tied low for the μPD78320.		
$\overline{RESET}$	External system reset input		
$\overline{WDTO}$	Watchdog timer output		
X1	Crystal oscillator connection or external clock input.		
X2	Crystal oscillator connection; not necessary to connect with external clock input.		
AV_{REF}	A/D converter reference voltage input		
AV_{DD}	A/D converter +5-volt power input		
AV_{SS}	A/D converter ground		
V_{DD}	+5-volt power input		
V_{SS}	Ground		
V_{PP}	PROM write-verify power input on μPD78P322 only. Must be tied to V_{DD} for normal operation		
NC	Not connected internally. May be connected to V_{SS}		

μPD78322 Family Block Diagram



* μPD78322/P322.

** VPP on μPD78P322

83CL-9323B

FUNCTIONAL DESCRIPTION

Central Processing Unit

The central processing unit (CPU) of the μPD78322 family features 16-bit arithmetic including 16-by-16-bit multiply, both unsigned and signed, and 32-by-16-bit unsigned divide (producing a 32-bit quotient and a 16-bit remainder). The signed multiply executes in 3.5 μs and the divide in 5.38 μs at 16 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 16 MHz, the clock is 8 MHz. Since some instructions execute in two cycles, the minimum instruction time is 250 ns.

On-Chip RAM

The μPD78322 family has a total of 640 bytes of on-chip RAM. The upper 256-byte area (FE00H-FEFFH) features high-speed access of one word of data per internal system clock and is known as "main RAM." The remainder (FC80H-FDFFH) is accessed at the same speed as external memory (1 byte per three internal system clocks) and is known as "peripheral RAM." The general register banks and the macro service control words are stored in main RAM. The remainder of main RAM and any unused register bank locations are available for general storage.

On-Chip PROM

The μPD78322 contains 16K bytes of internal ROM; the μPD78P322 contains 16K bytes of UV EPROM or one-time programmable ROM. Instructions are fetched from this on-chip memory at a maximum rate of 1 byte every internal system clock through the high-speed fetch mode. The μPD78320 does not have on-chip PROM.

External Memory

The μPD78322 family has a 64K-byte address space. The μPD78322/P322 can access 0, 256, 4K, 16K, or 48K bytes of external memory in the area from 4000H to FFFFH. External memory can be either ROM or RAM (or both) as required. The μPD78322/P322 have an 8-bit wide external data bus and a 16-bit wide external address bus. The low-order 8 bits of the address bus are multiplexed to provide the 8-bit data bus and are supplied by I/O port 4.

High-order address bits are taken from port 5 as required. Address latch, read, and write strobes are also provided. The memory mode register (MM) controls the size of the external memory. It can be programmed for 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 μPD78320 does not have ports 4 and 5. It has eight dedicated high-order address lines and eight dedicated address/data lines. All memory below address FC80H must be external.

The programmable wait control register (PWC) 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 for internal and external memory are specified independently.

Program Fetch

The μPD78322 family allows opcode fetch in the area between 0000H and FFFFH under the following constraints: from FC80H to FDFFH, opcodes will be fetched from the peripheral RAM; from FFE0H to FFFFH, opcodes will be fetched from external memory only. The μPD78322 family contains a 3-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.

Instruction bytes can be fetched from on-chip memory in either high-speed or ordinary fetch cycle mode. The fetch cycle control register (FCC) is used to select the mode. In high-speed fetch cycle mode, one internal system clock is required to fetch each instruction byte from on-chip memory. In ordinary fetch cycle mode, each byte to be fetched requires three, four, or five internal system clocks depending on the setting of the PWC register.

Each instruction byte fetched from external memory requires three, four, or five internal system clocks 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. During reset, the program counter is loaded with the address stored in locations 0000H and 0001H.

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

CPU Control Word. The CPU control word (CCW) selects the origin of the interrupt vector and CALLT tables. If the TPF bit (bit 1) is zero, the origin is 0000H; if the TPF bit is one, the origin is 8000H. The CCW is a special function register located at address FFC1H. The addresses of the vectors for the RESET input, operation-code trap, and BRK instruction are fixed at 0000H, 0003CH, 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	LT	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
LT	Interrupt priority level transition flag
CY	Carry bit (or 1-bit accumulator for logic)

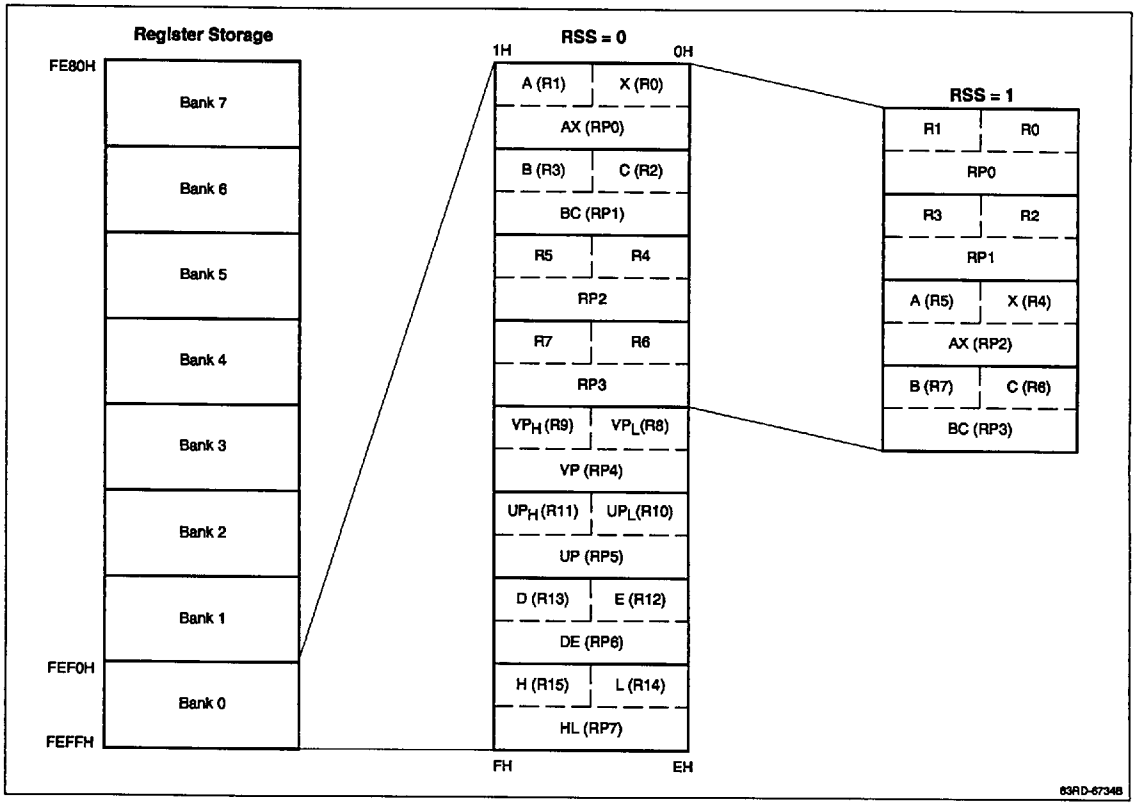
General Registers

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 identify active register banks.

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



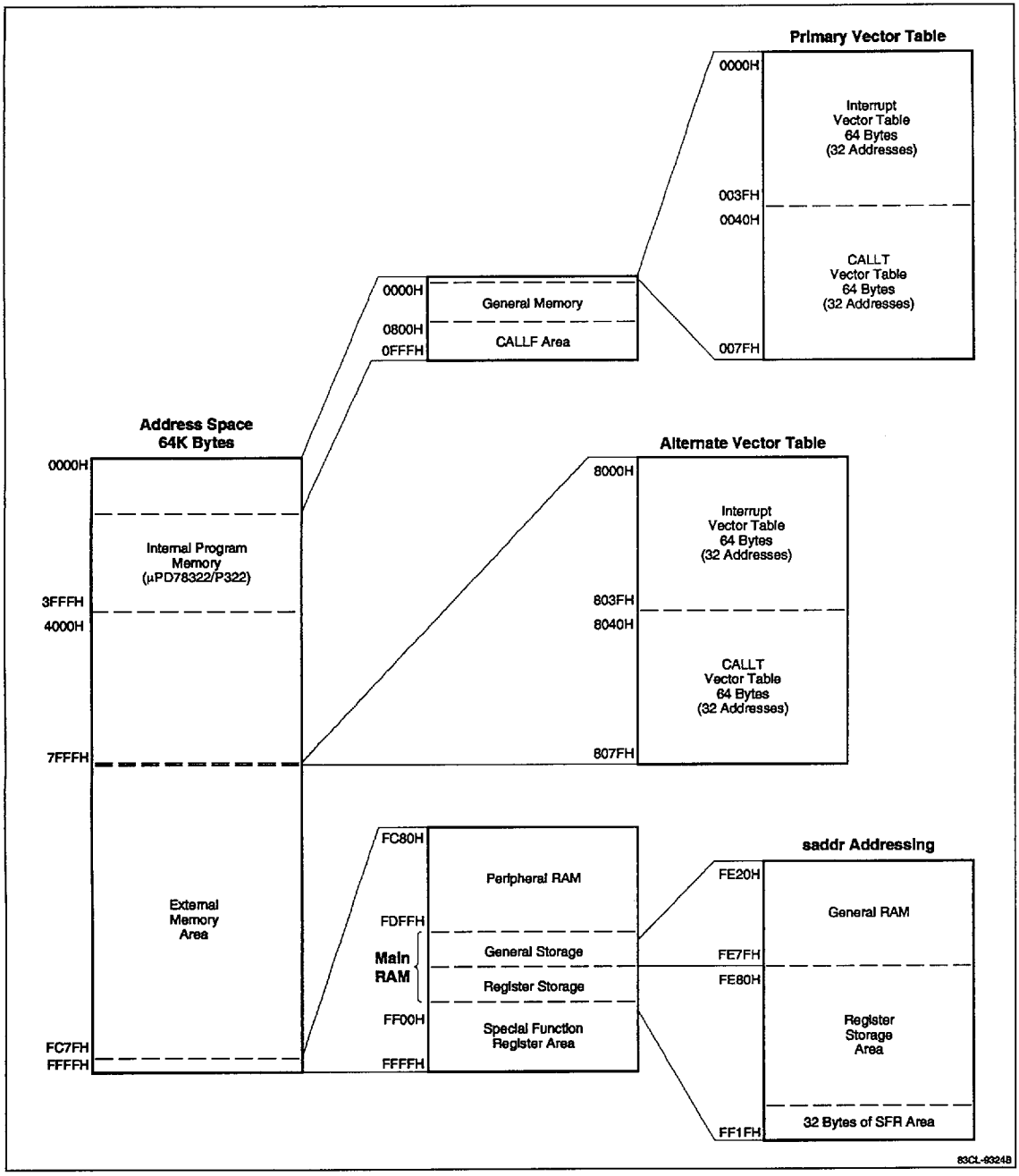
Addressing

The μPD78322 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, whereas 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 decrement, saddr, saddr indirect, SFR, based, indexed, and based indexed. There are also 8-bit and 16-bit immediate operands. Figure 3 is the memory map of the μPD78322 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 except the port mode registers and the asynchronous serial transmission shift register can be read under program control, and most can also be written. They are either 8 or 16 bits, as required, and many of the 8-bit registers are capable of single-bit access as well.

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

Table 1. Special Function Registers

Address	Register	Symbol	R/W	Access Unit (Bits)			State After Reset
				1	8	16	
FF00H	Port 0	P0	R/W	X	X	—	Undefined
FF02H	Port 2	P2	R	—	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	—	Undefined
FF08H	Port 8	P8	R/W	X	X	—	Undefined
FF09H	Port 9	P9	R/W	X	X	—	Undefined
FF0AH-FF0BH	Free-running counter (lower 16 bits)*	TM0LW	R	—	—	X	0000H
FF10H-FF11H	Capture register X0 (lower 16 bits)*	CTX0LW	R	—	—	X	Undefined
FF12H-FF13H	Capture register 01 (lower 16 bits)*	CT01LW	R	—	—	X	Undefined
FF14H-FF15H	Capture register 02 (lower 16 bits)*	CT02LW	R	—	—	X	Undefined
FF16H-FF17H	Capture register 03 (lower 16 bits)*	CT03LW	R	—	—	X	Undefined
FF18H-FF19H	Capture/compare register X0 (lower 16 bits)*	CCX0LW	R/W	—	—	X	Undefined
FF1AH-FF1BH	Capture/compare register 01 (lower 16 bits)*	CC01LW	R/W	—	—	X	Undefined
FF20H	Port 0 mode register	PM0	W	—	X	—	FFH
FF23H	Port 3 mode register	PM3	W	—	X	—	xxx1 1111B
FF25H	Port 5 mode register	PM5	W	—	X	—	FFH
FF28H	Port 8 mode register	PM8	W	—	X	—	xx11 1111B
FF29H	Port 9 mode register	PM9	W	—	X	—	xxxx 1111B
FF2AH-FF2BH	Free running counter (high 16 bits)*	TM0UW	R	—	—	X	0000H
FF2CH-FF2DH	Timer register 1	TM1	R	—	—	X	0000H
FF30H-FF31H	Capture register X0 (high 16 bits)*	CTX0UW	R	—	—	X	Undefined
FF32H-FF33H	Capture register 01 (high 16 bits)*	CT01UW	R	—	—	X	Undefined
FF34H-FF35H	Capture register 02 (high 16 bits)*	CT02UW	R	—	—	X	Undefined
FF36H-FF37H	Capture register 03 (high 16 bits)*	CT03UW	R	—	—	X	Undefined
FF38H-FF39H	Capture/compare register X0 (high 16 bits)*	CCX0UW	R/W	—	—	X	Undefined
FF3AH-FF3BH	Capture/compare register 01 (high 16 bits)*	CC01UW	R/W	—	—	X	Undefined
FF40H	Port 0 mode control register	PMC0	W	—	X	—	00H
FF41H	Real-time output port set register	RTPS	R/W	X	X	—	00H

Table 1. Special Function Registers (cont)

Address	Register	Symbol	R/W	Access Unit (Bits) 1	8	16	State After Reset
FF43H	Port 3 mode control register	PMC3	W	—	X	—	xxx0 0000B
FF48H	Port 8 mode control register	PMC8	W	—	X	—	xx00 0000B
FF4CH-FF4DH	Baud rate generator	BRG	R/W	—	—	X	Undefined
FF60H	Real-time output port register	RTP	R/W	X	X	—	Undefined
FF61H	Real-time output port reset register	RTPR	R/W	X	X	—	00H
FF62H	Port read control register	PRDC	R/W	X	X	—	00H
FF68H	A/D converter mode register	ADM	R/W	X	X	—	00H
FF6AH	A/D converter result register (16-bit access)	ADCR	R	—	—	X	Undefined
FF6BH	A/D converter result register (high 8 bits)	ADCRH	R	—	X	—	Undefined
FF70H-FF71H	Compare register 00	CM00	R/W	—	—	X	Undefined
FF72H-FF73H	Compare register 01	CM01	R/W	—	—	X	Undefined
FF74H-FF75H	Compare register 02	CM02	R/W	—	—	X	Undefined
FF76H-FF77H	Compare register 03	CM03	R/W	—	—	X	Undefined
FF7CH-FF7DH	Compare register 10	CM10	R/W	—	—	X	Undefined
FF7EH-FF7FH	Compare register 11	CM11	R/W	—	—	X	Undefined
FF80H	Clock synchronized serial interface mode register	CSIM	R/W	X	X	—	00H
FF82H	Serial bus interface control register	SBIC	R/W	X	X	—	00H
FF86H	Serial I/O shift register	SIO	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	—	00H
FF8CH	Serial receive buffer: UART	RXB	R	—	X	—	Undefined
FF8EH	Serial transmit shift register: UART	TXS	W	—	X	—	Undefined
FFB0H	Timer control register	TMC	R/W	X	X	—	00H
FFB1H	Baud rate generator mode register	BRGM	R/W	X	X	—	00H
FFB2H	Prescalar mode register	PRM	R/W	X	X	—	00H
FFB8H	Timer output control register 0	TOC0	R/W	X	X	—	00H
FFB9H	Timer output control register 1	TOC1	R/W	X	X	—	00H
FFBFH	Real-time pulse unit mode register	RPUM	R/W	X	X	—	00H
FFC0H	Standby control register	STBC	R/W**	X	X	—	0000 X000B
FFC1H	CPU control word	CCW	R/W	X	X	—	00H
FFC2H	Watchdog timer mode register	WDM	R/W**	X	X	—	00H
FFC4H	Memory extension mode register	MM	R/W	X	X	—	00H
FFC6H	Programmable wait control register	PWC	R/W	X	X	—	22H
FFC9H	Fetch cycle control register	FCC	R/W	X	X	—	00H
FFD0H-FFDFH	External access area		R/W	X	X	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0L/IF0	R/W	X	X	X	00H
FFE1H	Interrupt request flag register 0H	IF0H	R/W	X	X	—	00H
FFE2H	Interrupt request flag register 1L	IF1L/IF1	R/W	X	X	X	00H

Table 1. Special Function Registers (cont)

Address	Register	Symbol	R/W	Access Unit (Bits) 1	8	16	State After Reset
FFE4H	Interrupt mask flag register 0L	MK0L/ MK0	R/W	X	X	X	FFH
FFE5H	Interrupt mask flag register 0H	MK0H	R/W	X	X	—	FFH
FFE6H	Interrupt mask flag register 1L	MK1L/ MK1	R/W	X	X	X	xxxx x111B
FFE8H	Priority selection buffer register 0L	PB0L/ PB0	R/W	X	X	X	00H
FFE9H	Priority selection buffer register 0H	PB0H	R/W	X	X	—	00H
FFEAH	Priority selection buffer register 1L	PB1L/ PB1	R/W	X	X	X	00H
FFECH	Interrupt service mode selection register 0L	ISM0L/ ISM0	R/W	X	X	X	00H
FFEDH	Interrupt service mode selection register 0H	ISM0H	R/W	X	X	—	00H
FFEEH	Interrupt service mode selection register 1L	ISM1L/ ISM1	R/W	X	X	X	00H
FFF0H	Context switch enable register 0L	CSE0L/ CSE0	R/W	X	X	X	00H
FFF1H	Context switch enable register 0H	CSE0H	R/W	X	X	—	00H
FFF2H	Context switch enable register 1L	CSE1L/ CSE1	R/W	X	X	X	00H
FFF4H	External interrupt mode register 0	INTM0	R/W	X	X	—	00H
FFF5H	External interrupt mode register 1	INTM1	R/W	X	X	—	00H
FFF8H	In-service priority register	ISPR	R	—	X	—	00H
FFF9H	Priority selection register	PRSL	R/W	X	X	—	00H

* Lower or upper 16 bits of an 18-bit register.
** Protected location: special instruction required for write.

Input/Output Ports

The μPD78322 family has six ports providing a total of 37 I/O lines. P0, P3, P8, and P9 are tri-state input/output ports of 8, 5, 6, and 2 bits, respectively; each bit can be individually selected for input or output. P2 and P7 are 8-bit input ports.

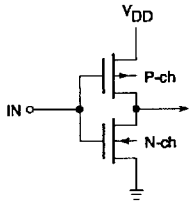
P2 functions only in the control mode as input pins for the NMI signal, the INTP0 to INTP5 interrupt signals, and the INTP6/TI interrupt signal or external count clock for timer 1 (TI). However, any masked interrupt automatically becomes an input line and the state of all the pins can be read by the program using a read instruction to port 2. Each pin of P2 can be programmed for rising, falling, or both rising and falling edge detection.

The output level of the P0, P3, P8, and P9 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.

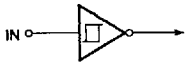
The μPD78322/P322 have two additional input/output ports, P4 and P5, and two additional I/O pins in P9. All these I/O lines are available if external memory or memory-mapped external circuitry is not being used. Port 4 is shared with the low-order address/data bus (AD_0 to AD_7) and is byte-selectable for input or output. Port 5 is shared with the high-order address bus (A_8 to A_{15}). Depending on the amount of external memory used, either 8, 6, 4, or 0 bits are available for bit-selectable I/O. Port 9 is a 4-bit, bit-selectable I/O port; two of its pins are shared with the read and write strobes.

Figure 3. I/O Circuits

Type 1. $\overline{EA}$

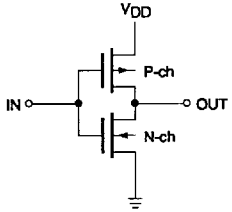


Type 2. P2, $\overline{RESET}$

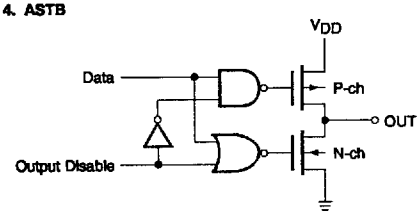


Schmitt trigger input with hysteresis characteristics.

Type 3. $\overline{WDTO}$

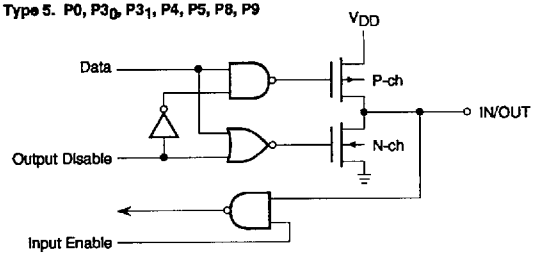


Type 4. ASTB

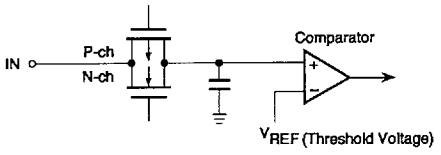


Push-pull output that can be placed in high impedance (both P-ch and N-ch off).

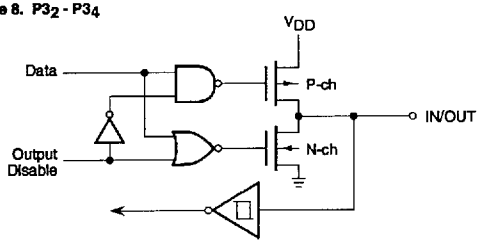
Type 5. P0, $P3_0$, $P3_1$, P4, P5, P8, P9



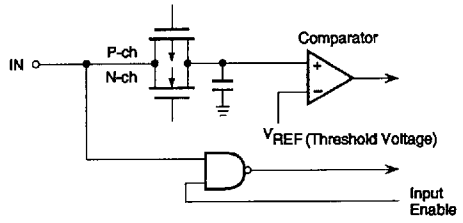
Type 7.



Type 8. $P3_2$ - $P3_4$



Type 9. P7



83CL-9325B

Real-Time Output Port

Port 0 can function on a bit-selectable basis as a real-time output port. Selected bits can be directly written under program control, or they can be set or cleared under control of timing signals generated by the real-time pulse unit. Timing by the latter method is independent of interrupt latency.

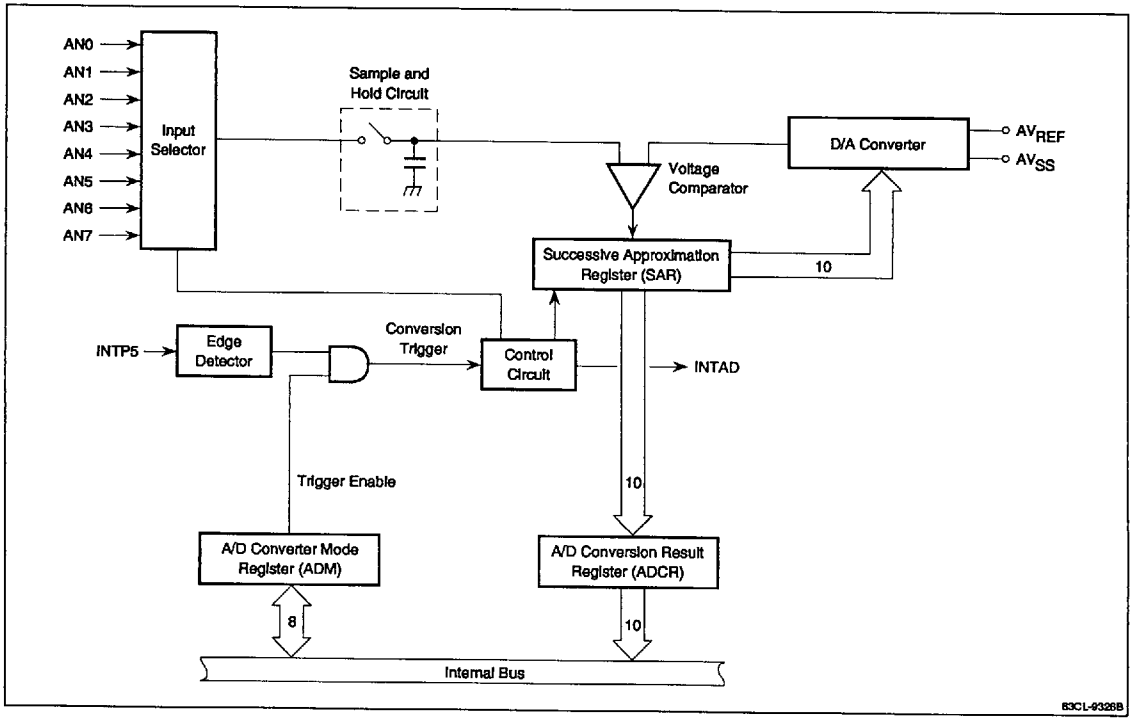
A/D Converter

The analog-to-digital (A/D) converter (figure 4) uses the successive-approximation method for converting up to eight multiplexed analog inputs into 10-bit digital data. The conversion time per input is 18 μs at 16-MHz operation. A/D conversion can be started by an external interrupt, INTP5, or under software control.

The A/D converter can operate in scan mode or select mode. In scan mode, inputs AN0 - AN3 or AN4 - AN7 can be programmed for conversion. The A/D converter selects each of the four inputs in order, converts the data, stores it in the A/D conversion result (ADCR) register, and generates an interrupt (INTAD). This converted data can be easily transferred to memory by the macro service function. In select mode, any one of the eight A/D inputs can be selected for conversion.

Once the A/D converter is started by INTP5 or software, conversion continues until it is disabled by software. The ADCR register is continually updated and either the full 10 bits or only the upper 8 bits of the conversion can be read at any time.

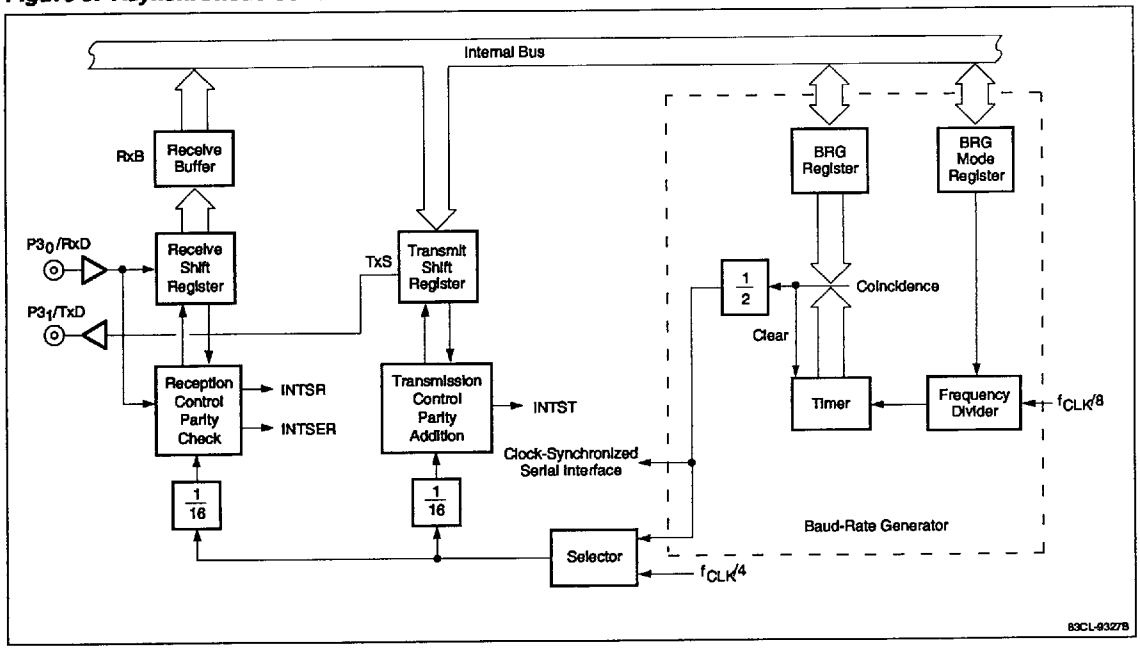
Figure 4. A/D Converter



Serial Interface

The μPD78322 family has two independent serial interfaces with a dedicated baud-rate generator. The first is a standard universal asynchronous receiver transmitter (UART). The UART (figure 5) 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.

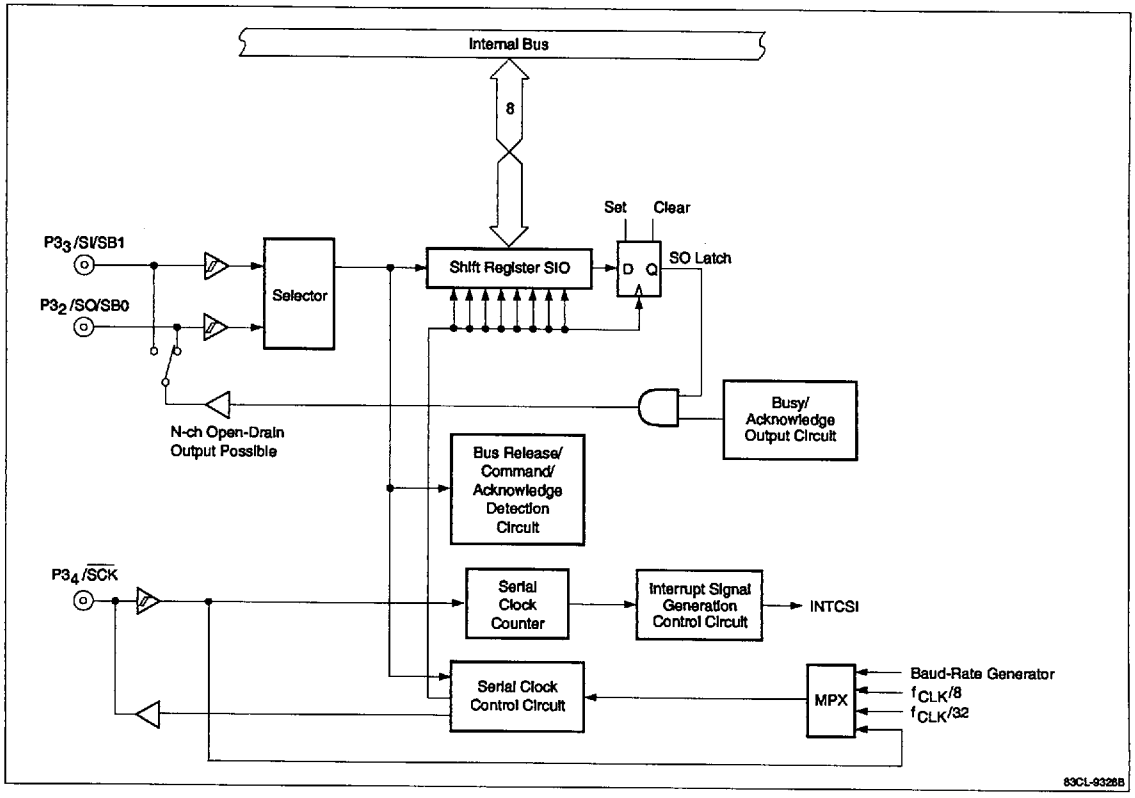
Figure 5. Asynchronous Serial Interface



The source of the serial clock for the UART is the internal system clock (divided by 4) or the on-chip baud-rate generator. The UART generates three interrupts: INTST (transmission complete), INTSR (reception complete), and INTSER (reception error).

The second interface is an 8-bit clock-synchronized serial interface (figure 6). It can be operated in either three-wire serial I/O mode or NEC serial bus interface (SBI) mode.

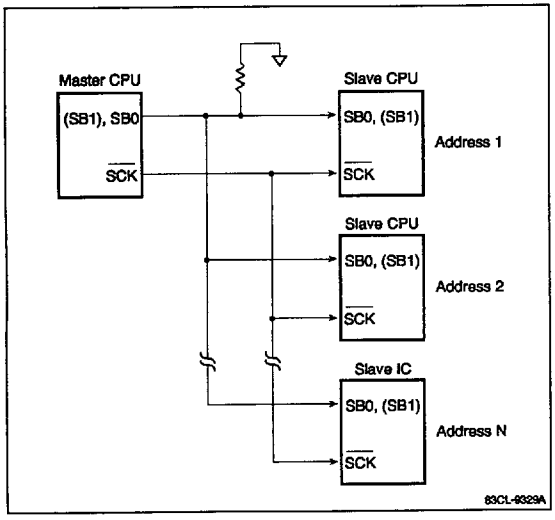
Figure 6. Clock-Synchronized Serial Interface



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 on the SO line (either MSB or LSB first) and in on the SI line, providing full-duplex operation. This interface can also be set to receive only or to transmit only. The INTCSI 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 (figure 7). 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 sysnchronized with the $\overline{SCK}$ line.

Figure 7. SBI Mode Master/Slave Configuration



Each slave μPD78322 family 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.

A dedicated baud-rate generator can be programmed to provide the serial clock to both asynchronous and clock-synchronized serial interfaces. By choosing the correct oscillator frequency, the baud-rate generator is capable of generating all of the commonly used baud rates from 75 to 19,200 b/s.

Real-Time Pulse Unit

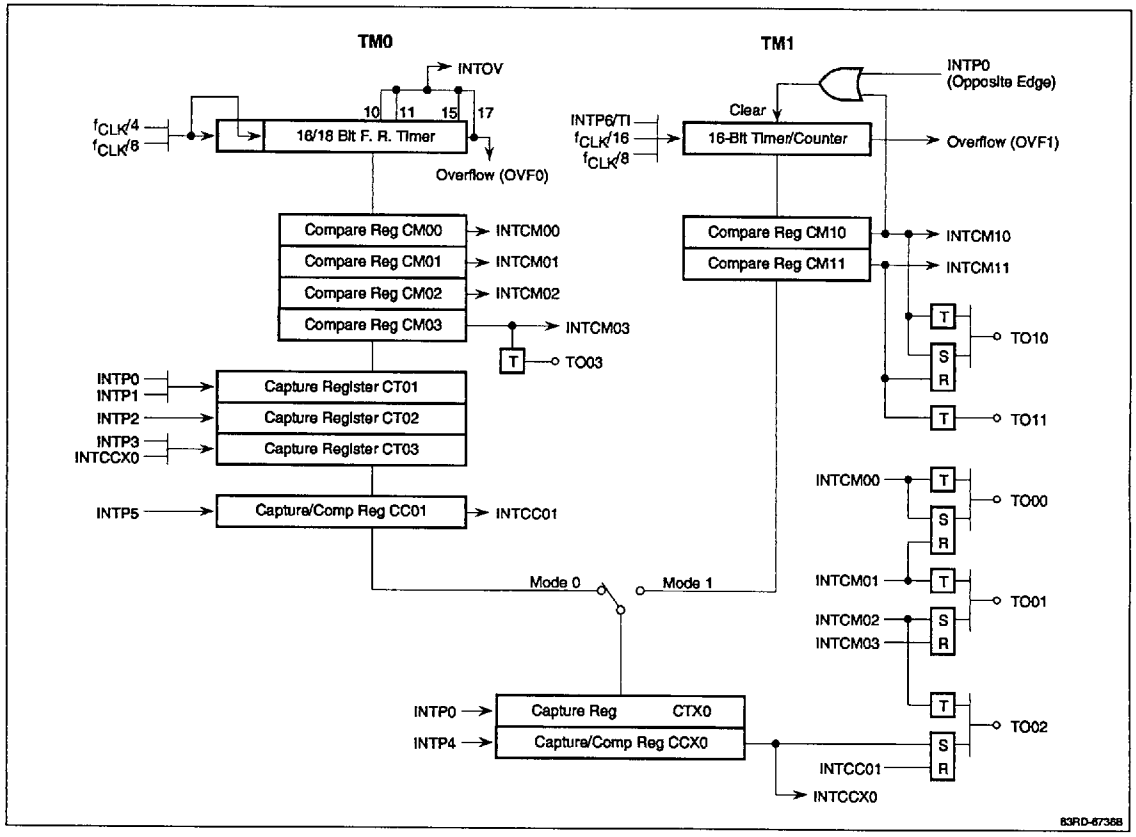
The real-time pulse unit (RPU, figure 8) can function as an interval timer to measure pulse widths and frequencies, generate pulse-width modulated outputs, count external events, and control the real-time output port. It consists of 18-bit free-running timer TM0, 16-bit timer/counter TM1, six 16-bit compare registers, four 18-bit capture registers, two 18-bit registers (capture or compare), and six timed output latches.

TM0 always counts the internal system clock (divided by 4 or 8) and can be reset by an external reset pulse only. TM1 can count either the internal system clock (divided by 8 or 16) or external events. TM1 can be reset by a compare event (a match between a timer and an associated compare register) or by an external signal in INTP0.

Capture events can be triggered by external maskable interrupts INTP0 - INTP5, and compare events can generate interrupts, control timed output pins, or both. In addition, interrupts INTCM03 and INTCCX0 can control the real-time output port

The timed output latches share the six pins of port P8. Four latches can be toggled or set and reset by compare events, and the remaining two can be toggled. These latches, with the macro service facility, can generate up to four pulse-width modulated outputs.

Figure 8. Real-Time Pulse Unit



Interrupts

The μPD78322 family has 19 maskable hardware interrupt sources: 7 external and 12 internal. The external maskable interrupts share pins with port P2. Six of them, INTP0 to INTP5, can also be used to trigger capture events in the real-time pulse unit. In addition, there are two nonmaskable interrupts, two software interrupts, and RESET. The software interrupts, generated by the BRK or BRKCS instruction and the operation code trap, are not maskable. See table 2.

Table 2. Interrupt Sources

Type of Request	Default Priority	Signal Name	Source	Location	Macro Service Control Word	Vector Address TPF = 0	Vector Address TPF = 1
Software	—	—	Operation code trap	CPU	—	003CH	
	—	—	Break instruction	CPU	—	003EH	
Nonmaskable	—	NMI	NMI input pin	External	—	0002H	8002H
	—	INTWDT	Watchdog timer overflow	Internal	—	0004H	8004H
Maskable	0	INTOV	Timer 0 overflow	Internal	FE06H	0006H	8006H
	1	INTP0	INTP0 pin	External	FE08H	0008H	8008H
	2	INTP1	INTP1 pin	External	FE0AH	000AH	800AH
	3	INTP2	INTP2 pin	External	FE0CH	000CH	800CH
	4	INTP3	INTP3 pin	External	FE0EH	000EH	800EH
	5	INTP4 INTCCX0	INTP4 pin CCX0 coincidence	External Internal	FE10H	0010H	8010H
	6	INTP5 INTCC01	INTP5 pin CC01 coincidence	External Internal	FE12H	0012H	8012H
	7	INTP6	INTP6 pin	External	FE14H	0014H	8014H
	8	INTCM00	CM00 coincidence	Internal	FE16H	0016H	8016H
	9	INTCM01	CM01 coincidence	Internal	FE18H	0018H	8018H
	10	INTCM02	CM02 coincidence	Internal	FE1AH	001AH	801AH
	11	INTCM03	CM03 coincidence	Internal	FE1CH	001CH	801CH
	12	INTCM10	CM10 coincidence	Internal	FE1EH	001EH	801EH
	13	INTCM11	CM11 coincidence	Internal	FE20H	0020H	8020H
	14	INTSER	Asynchronous serial Interface reception error	Internal	—	0022H	8022H
	15	INTSR	End of asynchronous serial Interface reception	Internal	FE24H	0024H	8024H
	16	INTST	End of asynchronous serial Interface transmission	Internal	FE26H	0026H	8026H
	17	INTCSI	End of clocked serial Interface transmission/reception	Internal	FE28H	0028H	8028H
	18	INTAD	End of A/D conversion	Internal	FE2AH	002AH	802AH
Reset	—	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$ pin	External	—	0000H	

Interrupt Servicing

The μPD78322 family provides three 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 μPD78322 family has ten 16-bit interrupt control registers. Each bit in each register is dedicated to one of the 19 maskable interrupt sources. The interrupt request flag registers (IF0, IF1) contain an interrupt request flag for each interrupt. The interrupt mask registers (MK0, MK1) are used to enable or disable any interrupt. The interrupt service mode registers (ISM0,

ISM1) specify whether an interrupt is processed by vectoring or macro service.

The priority specification buffer registers (PB0, PB1), in conjunction with the 8-bit priority specification register (PRSL), can be used to specify one of three priority levels for each interrupt. The context switching enable flag registers (CSE0, CSE1) specify whether an interrupt is processed by vectoring or context switching.

Two other 8-bit registers are associated with interrupt processing. The in-service 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.

The IE and LT bits of the program status word (PSW) are also used to control interrupts. If the IE bit is zero, all maskable interrupts, but not macro service, are disabled. The IE bit can be set or cleared using the EI and DI instructions, respectively, or by directly writing to the PSW. The IE bit is cleared each time an interrupt is accepted. The LT bit is set by hardware when a newly accepted maskable interrupt request is assigned a priority higher than the interrupt currently being serviced. The LT flag is used to control resetting the ISPR register when a return instruction from an interrupt service routine is executed.

Interrupt Priority

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

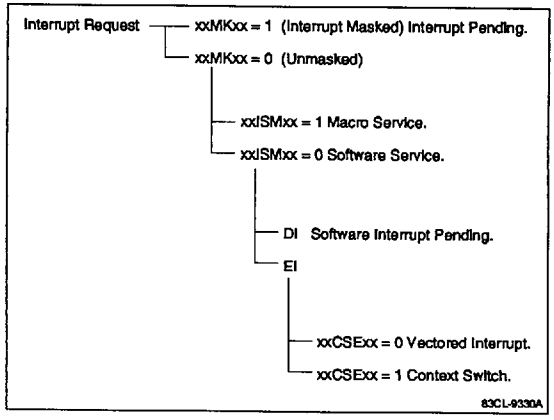
Three hardware-controlled priority levels are available for the maskable interrupts. Any one of the three levels can be assigned by software to each of the maskable interrupt lines. Interrupt requests of a priority equal to or higher than the processor's current priority level are accepted; lower priority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

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 9.

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

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

Figure 9. Interrupt Service Sequence



Vectored Interrupt

When vectored interrupt is specified for a given interrupt request, the program status word and the program counter are saved on the stack. The processor's priority is raised to that specified for the interrupt, the IE bit in the PSW is set to zero, and 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 μPD78322 family resumes the interrupted routine.

Context Switch

When context switching (figure 10) is specified for a given interrupt, the active register bank is changed to the register bank specified by the three low-order bits of the word in the interrupt vector table. The program counter is loaded from RP2 of the new register bank, the old program counter and program status word are saved in RP2 and RP3 of the new register bank, and the IE bit in the PSW is set to zero.

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

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 (or when some other completion condition is met), a software service routine is entered. Either vectored interrupt or context switch can be specified for entry to this routine, which is known as the macro service completion routine, and the routine is entered according to the specified priority.

Macro service is provided for all of the maskable interrupt requests except INTSER, the asynchronous serial interface receive error interrupt request. Each request has a dedicated macro service control word stored in on-chip main RAM. The function to be performed is specified in the control word. See table 3.

Figure 10. Context Switching and Return

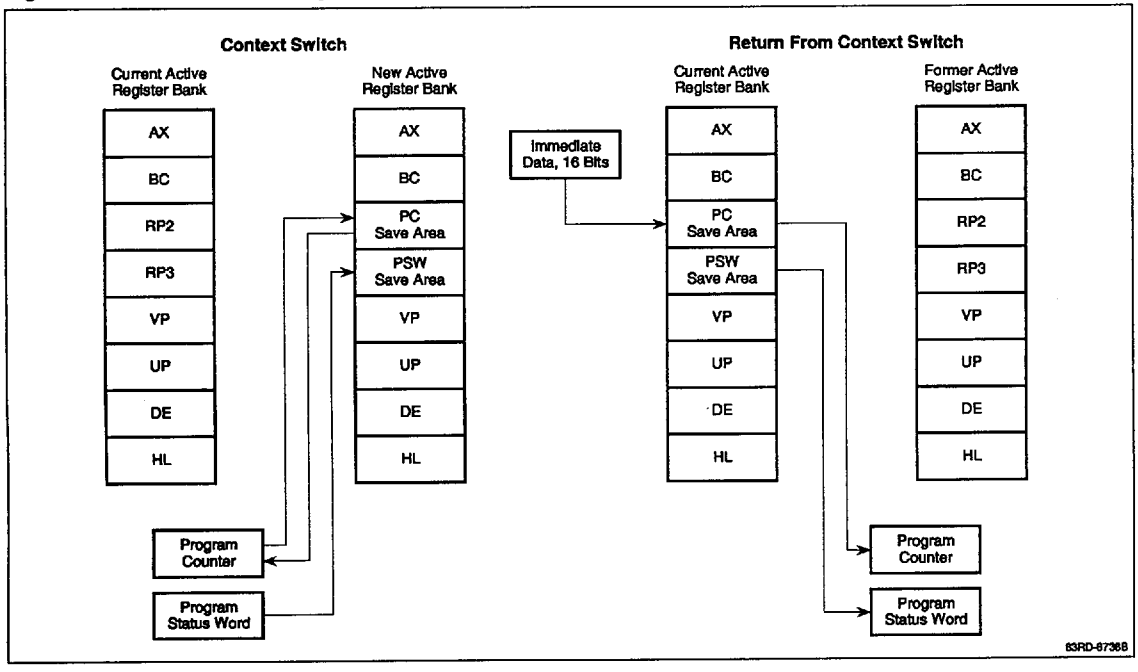


Table 3. Macro Service Functions

Control Word	Function
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.
DTACMP	Data compare. If the value of the special-function register requesting macro service matches a byte of data, the software service routine is entered. This mode can be used to detect an address match in SBI mode.
BITSHT	Data shift. Shifts the contents of a specified special function register one bit to the right (toward the LSB). The software service routine is entered when a data bit of 1 has been shifted out from the LSB. This mode can be used to control the real-time output port by shifting the values of the real-time output port set or reset registers.
BITLOG	Bit logic. Performs the logical AND or OR of a data byte and the contents of the specified special function register, stores the result back in the SFR, and enters the software service routine.
ADCBUF	A/D converter buffering. Stores the contents of the A/D conversion result register in a byte or word buffer in main RAM (FExxH). When the macro service counter reaches 0, A/D conversion is stopped but *no 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 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 byte or word buffer in main RAM (FExxH).
DTADIF-P	Data difference with memory pointer. Stores the difference between the current value of a specified special function register and its previous value in a byte or word buffer anywhere in the 64K-byte address space.
DTAADD	Data addition. Stores the sum of a byte or word of data and a specified special function register in the same or another specified special-function register.

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 nonmaskable interrupt request, unmasked maskable interrupt request, 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 $\overline{WDT0}$ can be connected to the $\overline{RESET}$ pin or used to control external circuitry. Three program-selectable intervals are available: 8.19, 32.7, and 131.0 ms at 16 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 μPD78322 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 $\overline{WDTO}$, AV_{REF}, AV_{DD}, AV_{SS}, V_{SS}, V_{DD}, X1, and X2 are in the high-impedance state.

ELECTRICAL SPECIFICATIONS

Note: Unless otherwise noted, these specifications apply to the entire μPD78322 family. Exceptions for extended temperature range devices may be singled out in the tables by the symbols below. The applicable symbol is also included in the device part number

Symbol	Operating Temperature Range
( )*	−10 to + 70°C (standard)
(A)	−40 to + 85°C (extended)
(A1)	−40 to + 110°C (extended)
(A2)	−40 to + 125°C (extended)

* ( ) = no suffix

Absolute Maximum Ratings

T_A = 25°C

Supply voltage, V_{DD}	−0.5 to +7.0 V
Supply voltage, AV_{DD}	−0.5 to V_{DD} + 0.5 V
Supply voltage, AV_{SS}	−0.5 to + 0.5 V
Supply voltage, V_{PP}	−0.5 to +13.5 V
Input voltage, V_I	
Except $P2_0$/NMI of μPD78P322	−0.5 to V_{DD} + 0.5 V
$P2_0$/NMI of μPD78P322	−0.5 to +13.5 V
Output voltage, V_O	−0.5 to V_{DD} + 0.5 V
Reference intput voltage, AV_{REF} ($f_{XX} \leq$ 16 MHz)	−0.5 to AV_{DD} + 0.3 V
Output current, low; I_{OL}	
Each output pin	4.0 mA
Total	90 mA
Output current, high; I_{OH}	
Each output pin	−1.0 mA
Total	−20 mA (A) devices
Operating temperature, T_{OPT}	
( ) devices	−10 to +70°C
(A) devices	−40 to +85°C
(A1) devices	−40 to +110°C
(A2) devices	−40 to +125°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.

Recommended Operating Conditions

Device	T_A	V_{DD}	Oscillator Freq, f_{XX}
μPD78320/322	−10 to +70°C	+5.0 V ±10%	8 to 16 MHz
μPD78P322	−10 to +70°C	+5.0 V ±5%	
(A) devices	−40 to +85°C	+5.0 V ±10%	
(A1) devices	−40 to +110°C	+5.0 V±10%	8 to 12 MHz
(A2) devices	−40 to +125°C		

Capacitance

T_A = 25°C; V_{DD} = V_{SS} = 0 V

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

† C_I = 10 pF on (A), (A1), and (A2) devices.

Oscillator Characteristics

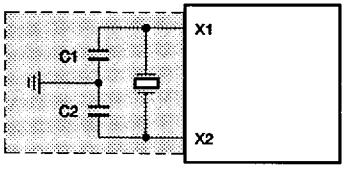
V_{DD} = +5 V ±10% (±5% for μPD78P322); V_{SS} = 0 V
Devices and their T_A ratings are defined earlier by symbols ( ), (A), (A1), and (A2)

Oscillator	Parameter	Symbol	Min	Max	Unit	Conditions
Ceramic or cystal resonator	Oscillation frequency	f_{XX}	8	16 12†	MHz	
External clock input at X1	Frequency	f_X	8	16 12†	MHz	
	Rise time, fall time	t_{XR}, t_{XF}	0	20	ns	
	Low-level, high-level width	t_{WXL}, t_{WXH}	25 46†	80 100†	ns	

† Applicable to (A1) and (A2) devices.

Recommended Oscillator Circuits

A. Resonator Circuit

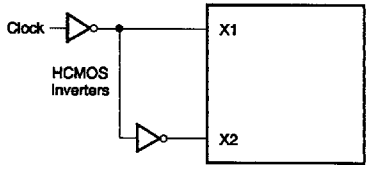


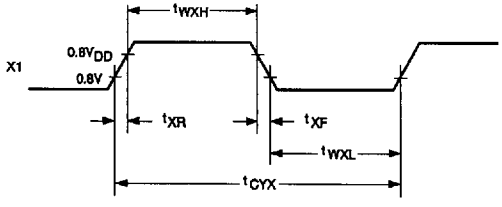
Mount resonator as close as possible to pins X1 and X2.
Do not place other signal lines in shaded area.

Crystal Resonator

Manufacturer	Part No.	C1, C2
Kinseki Co. Ltd.	HC49/U-S or HC49/U	10 pF

B. External Clock Circuit





83CL-9331B

DC Characteristics

V_{DD} = +5.0 V ±10% (±5% for μPD78P322); V_{SS} = 0 V
Devices and their T_A ratings are defined earlier by symbols ( ), (A), (A1), and (A2)

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}		40	65	mA	Operating mode
	I_{DD2}		20	35 45 (Note 3)	mA	HALT mode
Data retention voltage	V_{DDDR}	2.5			V	STOP mode
Data retention current	I_{DDDR}		2	10 100 (Note 3)	μA	STOP mode; V_{DDDR} = 2.5 V
			10	50 1000 (Note 3)	μA	STOP mode; V_{DDDR} = 5.0 V ±10% (±5% for μPD78P322)

Notes:

(1) All except pins in Note 2.

(2) Pins $\overline{RESET}$, X1, X2, $P2_n$, INTPn, NMI, TI, $P3_2$/SB0/SO, $P3_3$/SB1/SI, $P3_4$/SCK.

(3) Applicable to (A1) and (A2) devices.

AC Characteristics

V_{DD} = +5.0 V ±10% (±5% for μPD78P322); V_{SS} = 0 V
Devices and their T_A ratings are defined earlier by symbols ( ), (A), (A1), and (A2)

		( ) and (A)		(A1) and (A2)			
Parameter	Symbol	Min	Max	Min	Max	Unit	Conditions
External Memory Read/Write Operation							
System clock cycle time	t_{CYK}	125	250	166	250	ns	t_{CYK} equals twice the period of the crystal or external clock input.
Address setup time to ASTB ↓	t_{SAST}	32		43		ns	t_{CYK} = 125 ns
Address hold time after ASTB ↓	t_{HSTA}	32		53		ns	
Address to $\overline{RD}$ ↓ delay time	t_{DAR}	85		126		ns	
$\overline{RD}$ ↓ to address floating	t_{FRA}		0		0	ns	
Address to data input	t_{DAID}		222		326	ns	
$\overline{RD}$ ↓ to data input	t_{DRID1}		112		174	ns	
ASTB ↓ to $\overline{RD}$ ↓ delay time	t_{DSTR}	42		63		ns	
Data hold time from $\overline{RD}$ ↑	t_{HRID}	0		0		ns	
$\overline{RD}$ ↑ to address active	t_{DRA}	50 ( ) 37 (A)		58		ns	
$\overline{RD}$ width low	t_{WRL}	157		219 (A1) 209 (A2)		ns	
ASTB width, high	t_{WSTH}	37		58		ns	
Address to $\overline{WR}$ ↓ delay time	t_{DAW}	85		126		ns	
ASTB ↓ to data output	t_{DSTOD}		102		123	ns	

AC Characteristics (cont)

Parameter	Symbol	( ) and (A) Min	( ) and (A) Max	(A1) and (A2) Min	(A1) and (A2) Max	Unit	Conditions
External Memory Read/Write Operation (cont)							
$\overline{WR}$ ↓ to data output	t_{DWOD}		40		40	ns	t_{CYK} = 125 ns
ASTB ↓ to $\overline{WR}$ ↓ delay	t_{DSTW}	42		63		ns	
Data setup time to $\overline{WR}$ ↑	t_{SODW}	147		204		ns	
Data hold time after $\overline{WR}$ ↑	t_{HWOD}	32		53		ns	
$\overline{WR}$ width, low	t_{WWL}	157		209		ns	
$\overline{WR}$ ↑ to $\overline{ASTB}$ delay time	t_{DWST}	42		63		ns	
Serial Port Operation							
$\overline{SCK}$ cycle time	t_{CYSK}	1000		1328		ns	$\overline{SCK}$ output
		1000		1328		ns	$\overline{SCK}$ input
$\overline{SCK}$ width low	t_{WSKL}	420		584		ns	$\overline{SCK}$ output
		420		584		ns	$\overline{SCK}$ input
$\overline{SCK}$ width high	t_{WSKH}	420		584		ns	$\overline{SCK}$ output
		420		584		ns	$\overline{SCK}$ input
SI setup time to $\overline{SCK}$	t_{SRXSK}	80		80		ns	
SI hold time after $\overline{SCK}$ ↑	t_{HSKRX}	80		80		ns	
$\overline{SCK}$ ↓ to SO delay time	t_{DSKTX}		210		210	ns	R = 1 kΩ; C = 100 pF
Other Operations							
NMI high/low-level width	t_{WNIH}, t_{WNIL}	5				μs	
INTP0 high/low-level width	t_{WI0H}, t_{WI0L}	8				t_{CYK}	
INTP1 high/low-level width	t_{WI1H}, t_{WI1L}	8				t_{CYK}	
INTP2 high/low-level width	t_{WI2H}, t_{WI2L}	8				t_{CYK}	
INTP3 high/low-level width	t_{WI3H}, t_{WI3L}	8				t_{CYK}	
INTP4 high/low-level width	t_{WI4H}, t_{WI4L}	8				t_{CYK}	
INTP5 high/low-level width	t_{WI5H}, t_{WI5L}	8				t_{CYK}	
INTP6 high/low-level width	t_{WI6H}, t_{WI6L}	8				t_{CYK}	
$\overline{RESET}$ high/low-level width	t_{WRSH}, t_{WRSL}	5				μs	
TI high/low-level width	t_{WTIH}, t_{WTIL}	8				t_{CYK}	During TM1 even-counter mode

Timing Dependent on t_{CYK}

Symbol	Calculation Formula	Min/Max	Unit
t_{DAID}	(2.5 + n)T – 90	Max	ns
t_{DAR}	T – 40	Min	ns
t_{DAW}	T – 40	Min	ns
t_{DRA}	0.5T – 12 ( ) 0.5T – 25 (A) (A1) (A2)	Min	ns
t_{DRID1}	(1.5 + n)T – 75	Max	ns
t_{DSTOD}	0.5T + 40	Max	ns
t_{DSTR}	0.5T – 20	Min	ns
t_{DSTW}	0.5T – 20	Min	ns
t_{DWST}	0.5T – 20	Min	ns
t_{DWOD}	0.5T – 11	Min	ns
t_{HSTA}	0.5T – 30	Min	ns
t_{HWOD}	0.5T – 30	Min	ns
t_{SAST}	0.5T – 30 ( ) (A) 0.5T – 40 (A1) (A2)	Min	ns
t_{SODW}	1.5T – 40 ( ) (A) 0.5T – 45 (A1) (A2)	Max	ns
t_{WRL}	(1.5 + n)T – 30	Min	ns

Timing Dependent on t_{CYK} (cont)

Symbol	Calculation Formula	Min/Max	Unit
t_{WSTH}	0.5T – 25	Min	ns
t_{WWL}	(1.5 + n)T – 30 ( ) (A) (1.5 + n)T – 40 (A1) (A2)	Min	ns
t_{WI0H}, t_{WI0L}	8T	Min	ns
t_{WI1H}, t_{WI1L}	8T	Min	ns
t_{WI2H}, t_{WI2L}	8T	Min	ns
t_{WI3H}, t_{WI3L}	8T	Min	ns
t_{WI4H}, t_{WI4L}	8T	Min	ns
t_{WI5H}, t_{WI5L}	8T	Min	ns
t_{WI6H}, t_{WI6L}	8T	Min	ns
t_{WTIH}, t_{WTIL}	8T	Min	ns

Notes:

(1) n is the number of wait cycles specified by the PWC register.

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

(3) Devices and their T_A ratings are defined earlier by symbols ( ), (A), (A1), and (A2)

A/D Converter Characteristics

V_{DD} = +5 V ±10% (±5% for μPD78P322); AV_{SS} = V_{SS} = 0 V; AV_{DD} = (V_{DD} – 0.5 V) to V_{DD}
Devices and their T_A ratings are defined earlier by symbols ( ), (A), (A1), and (A2)

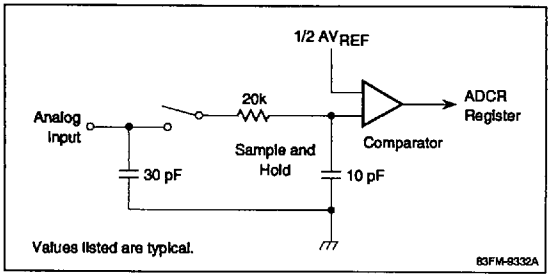
Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Resolution		10			Bit	
Relative accuracy*				±0.4 ( ) ±0.3 (A) (A1) (A2)	% †	AV_{REF} = 4.5 V to AV_{DD}
				±0.7 ( ) ±0.3 (A) (A1) (A2)	% †	AV_{REF} = 3.4 V to AV_{DD}
Quantization error				±1/2	LSB	
Conversion time	t_{CONV}	144			t_{CYK}	
Sampling time	t_{SAMP}	24			t_{CYK}	
Zero-scale error*			±1.5	±2.5	LSB	AV_{REF} = 4.5 V to AV_{DD}
			±1.5	±4.5	LSB	AV_{REF} = 3.4 V to AV_{DD}
Full-scale error*			±1.5	±2.5	LSB	AV_{REF} = 4.5 V to AV_{DD}
			±1.5	±4.5	LSB	AV_{REF} = 3.4 V to AV_{DD}
Nonlinearity error*			±1.5	±2.5	LSB	AV_{REF} = 4.5 V to AV_{DD}
			±1.5	±4.5	LSB	AV_{REF} = 3.4 V to AV_{DD}
Analog input voltage	V_{IAN}	–0.3		AV_{REF} + 0.3	V	
Reference voltage	AV_{REF}	3.4 or 4.5		AV_{DD}	V	
AV_{REF} current	AI_{REF}		1.0	3.0	mA	
AV_{DD} power current	AI_{DD}		2.0	6.0	mA	

A/D Converter Characteristics (cont)

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
A/D converter data retention current	AI_{DDR}		2	10 ( ) (A) 100 (A1) (A2)	μA	STOP mode; AV_{DDR} = 2.5 V
			10	50 ( ) (A) 1000 (A1) (A2)	μA	STOP mode; AV_{DDR} = 5 V ±10% (±5% for μPD78P322)
Analog input impedance	R_{AN}		10		MΩ	Nonsampling
						Sampling

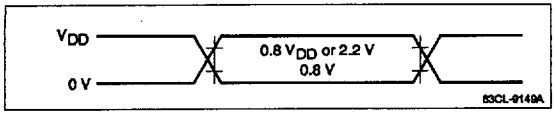
* Does not include quantization error
† Unit is percent of full-scale range (FSR)

A/D Converter Input Circuit

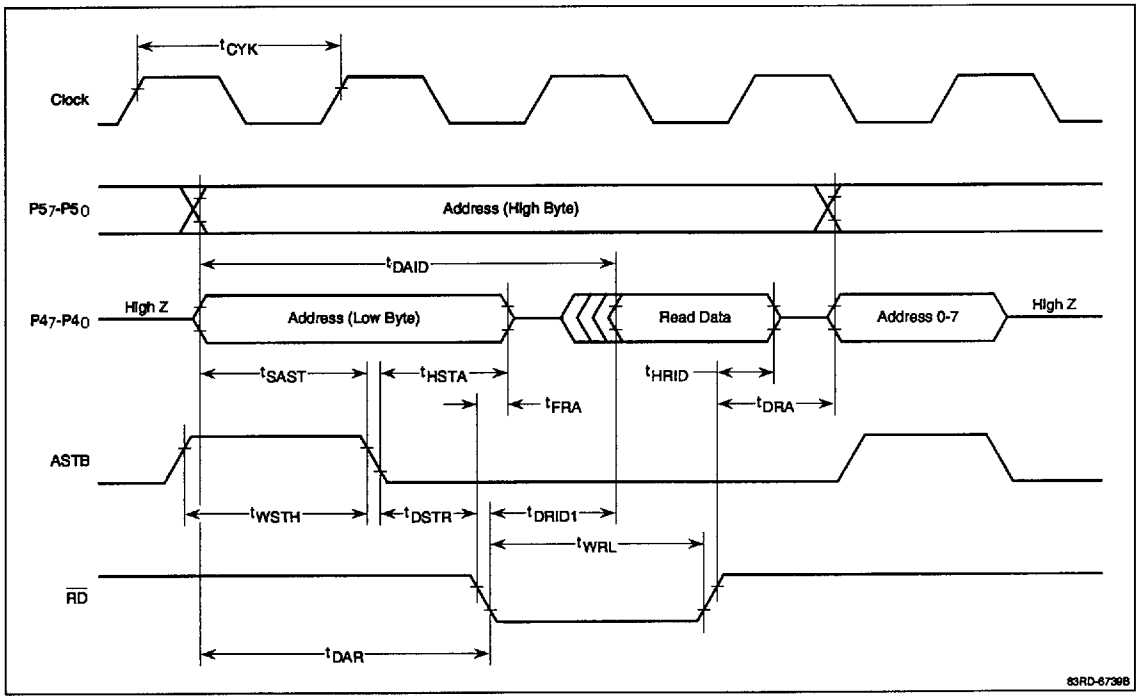


Timing Waveforms

AC Timing Test Points

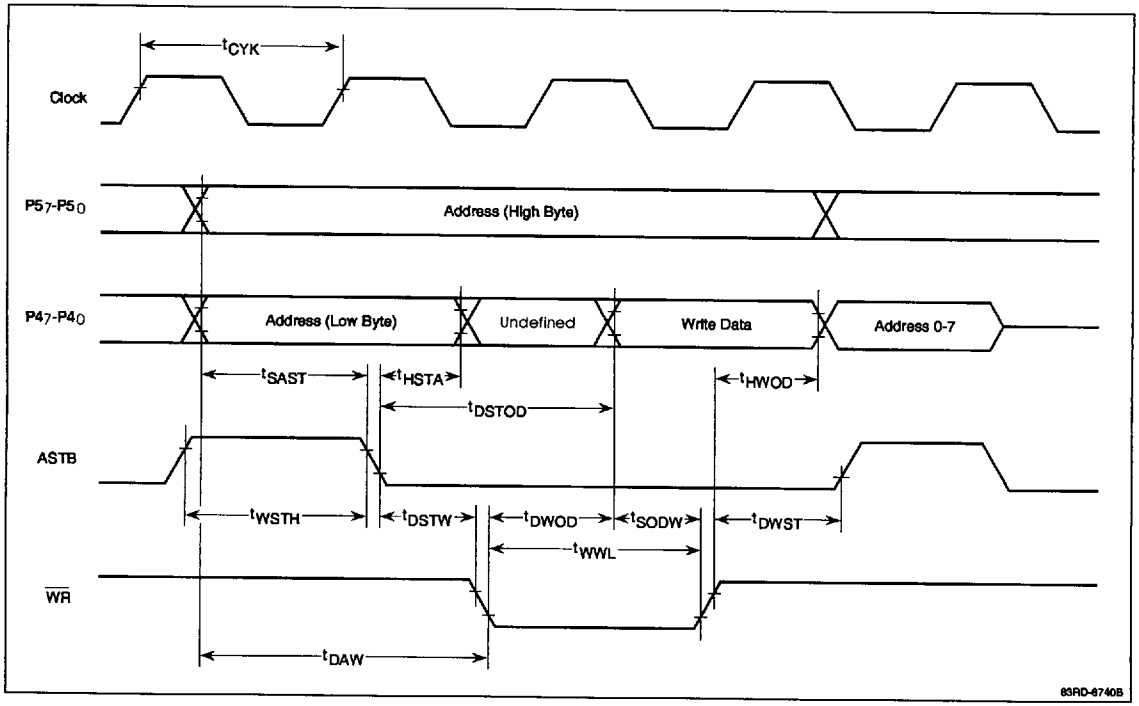


Read Operation

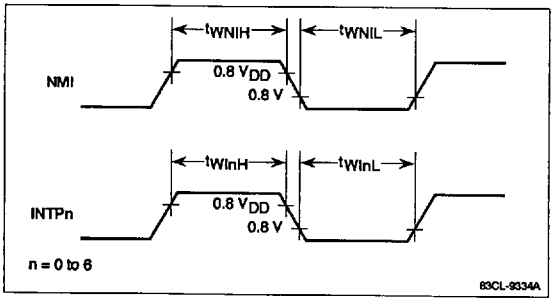


Timing Waveforms (cont)

Write Operation

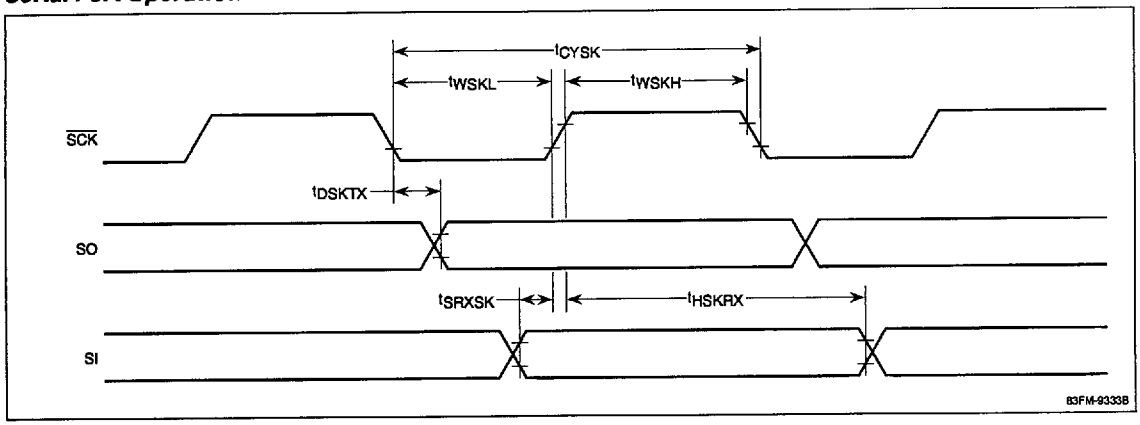


Interrupt Input

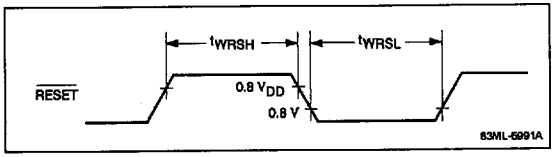


Timing Waveforms (cont)

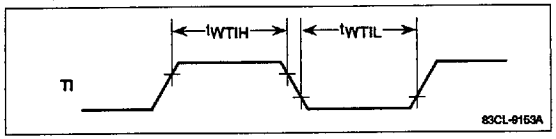
Serial Port Operation



Reset Input



TI Input



PROM PROGRAMMING

The PROM in the μPD78P322 is one-time programmable (OTP) or ultraviolet erasable (UV EPROM). The 16,384 x 8-bit PROM has the programming characteristics of an NEC μPD27C256A. Table 3 shows the functions of the μPD78P322 pins in normal operating mode and PROM programming mode.

PROM Programming Mode

When the $\overline{RESET}$ pin is set high and AV_{DD} is set low, the μPD78P322 enters the PROM programming mode. Operation in this mode is determined by the setting of the $\overline{CE}$, $\overline{OE}$, V_{PP}, and V_{DD} pins as indicated in table 4.

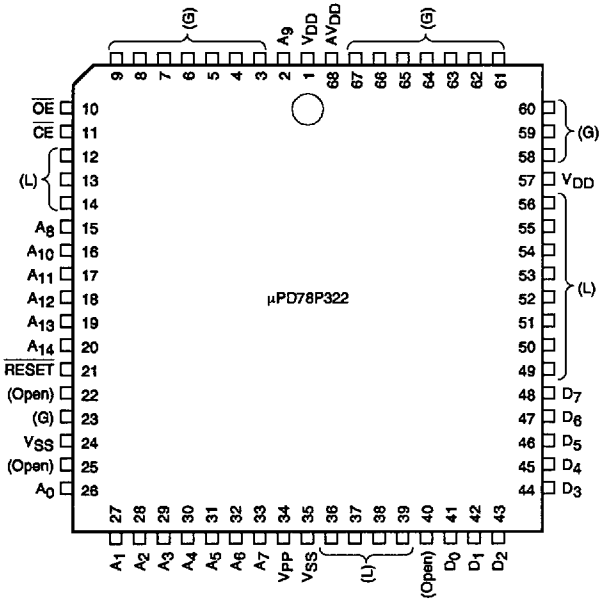
Table 3. Pin Functions During PROM Programming

Function	Normal Operating Mode	Programming Mode
Address input	$P0_0$ - $P0_7$, $P8_0$, $P2_0$, $P8_1$ - $P8_7$	A_0 - A_{14}
Data input	$P4_0$ - $P4_7$	D_0 - D_7
Chip enable/ program pulse	$P3_1$	$\overline{CE}$
Output enable	$P3_0$	$\overline{OE}$
Program voltage	V_{PP}	V_{PP}
Mode voltage	$\overline{RESET}$, AV_{DD}	$\overline{RESET}$, AV_{DD}

Table 4. Operation Modes For Programming

Mode	$\overline{RESET}$	AV_{DD}	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{DD}	D_0 - D_7
Program write	H	L	L	H	+12.5 V	+6.0 V	Data input
Program verify	H	L	H	L	+12.5 V	+6.0 V	Data output
Program inhibit	H	L	H	H	+12.5 V	+6.0 V	High impedance
Read	H	L	L	L	+5.0 V	+5.0 V	Data output
Ouput disable	H	L	L	H	+5.0 V	+5.0 V	High impedance
Standby	H	L	H	L/H	+5.0 V	+5.0 V	High impedance

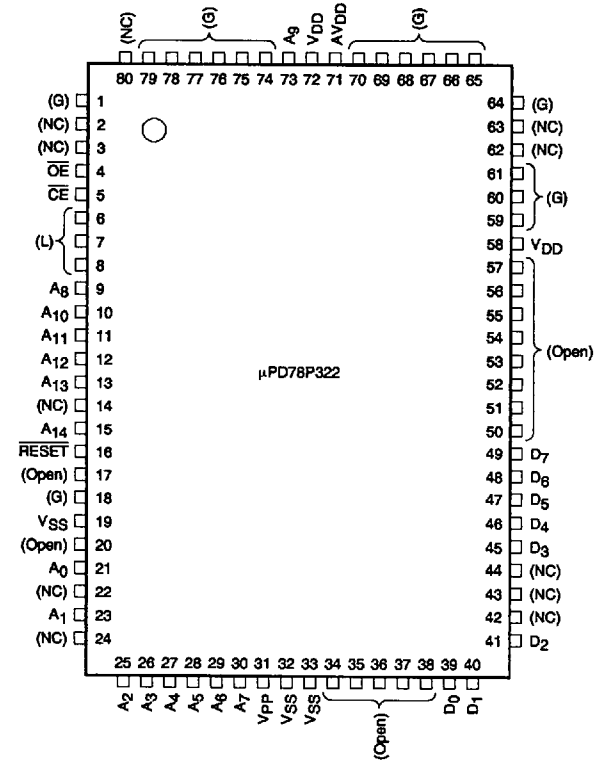
Figure 11. Pin Functions in μPD78P322 PROM Programming Mode; 68-Pin PLCC or LCC



Note:
Recommended connections for pins not used during PROM programming:
(L) Connect to V_{SS} through a resistor.
(G) Connect to V_{SS}.
(Open) No connection.

83CL-93358

Figure 12. Pin Functions in μPD78P322 PROM Programming Mode; 80-Pin QFP or LCC



Note:
Recommended connections for pins not used during PROM programming:
(L) Connect to V_{SS} through a resistor.
(G) Connect to V_{SS}.
(Open) No connection
(NC) Connect to V_{SS} to prevent noise.

83CL-9336B

PROM Write Procedure

Data can be written to the PROM by the following procedure.

(1) Set the pins not used for programming as indicated in figure 11 or 12. Set RESET high and AV_{DD} low. $\overline{CE}$ and $\overline{OE}$ should be high.

(2) Supply +6.0 V to V_{DD} pin and +12.5 V to V_{PP} pin.

(3) Provide initial address to pins A_0 - A_{14}.

(4) Provide write data.

(5) Input a 1-ms program pulse (active low) to $\overline{CE}$ 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 25 attempts, go to step 7.

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

(8) Perform one additional write with a program pulse width (in ms) equal to 3 times the number of writes performed in step 5.

(9) Increment address.

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

PROM Read Procedure

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

(1) Set the pins not used for programming as indicated in figure 11 or 12. Set RESET high and AV_{DD} low. $\overline{CE}$ and $\overline{OE}$ should be active high.

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

(3) Input address of data to be read to pins A_0 - A_{14}.

(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/cm^2 (ultraviolet ray intensity x exposure time) is required to completely erase the written data. Erasure by an ultraviolet lamp rated at 12,000 μW/cm^2 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 ±5°C; V_{SS} = 0 V

Parameter	Symbol	Symbol (Note 1)	Min	Typ	Max	Unit	Condition
High-level input voltage	V_{IH}	V_{IH}	2.2		V_{DDP} + 0.3	V	
Low-level input voltage	V_{IL}	V_{IL}	−0.3		0.8	V	
Input leakage current	I_{LIP}	I_{LI}			±10	μA	0 ≤ V_I ≤ V_{DDP} (Note 2)
High-level output voltage	V_{OH}	V_{OH}	2.4			V	I_{OH} = −400 μA
Low-level output voltage	V_{OL}	V_{OL}			0.45	V	I_{OL} = 2.0 mA
A9 pin input current	I_{A9}	—			±10	μA	A9 ($P2_0$/NMI)
Output leakage current	I_{LO}	—			10	μA	0 ≤ V_O ≤ V_{DDP}, $\overline{OE}$ = V_{IH}
PROG pin high-voltage input current	I_{IP}	—			±10	μA	
V_{DDP} power supply voltage	V_{DDP}	V_{DD}	5.75	6.0	6.25	V	Program memory write mode
			4.5	5.0	5.5	V	
V_{PP} power supply voltage	V_{PP}	V_{PP}	12.2	12.5	12.8	V	
				V_{PP} = V_{DDP}		V	
V_{DDP} power supply current	I_{DD}	I_{DD}		10	30	mA	
				10	30	mA	Program memory read mode $\overline{CE}$ = V_{IL}, V_I = V_{IH}
V_{PP} power supply current	I_{PP}	I_{PP}		10	30	mA	Program memory write mode $\overline{CE}$ = V_{IL}, $\overline{OE}$ = V_{IH}
				1	100	μA	Program memory read mode

Notes:

(1) Corresponding symbols for the μPD27C256A

(2) V_{DDP} is the V_{DD} pin during programming

AC Programming Characteristics

T_A = 25 ±5°C; V_{SS} = 0 V

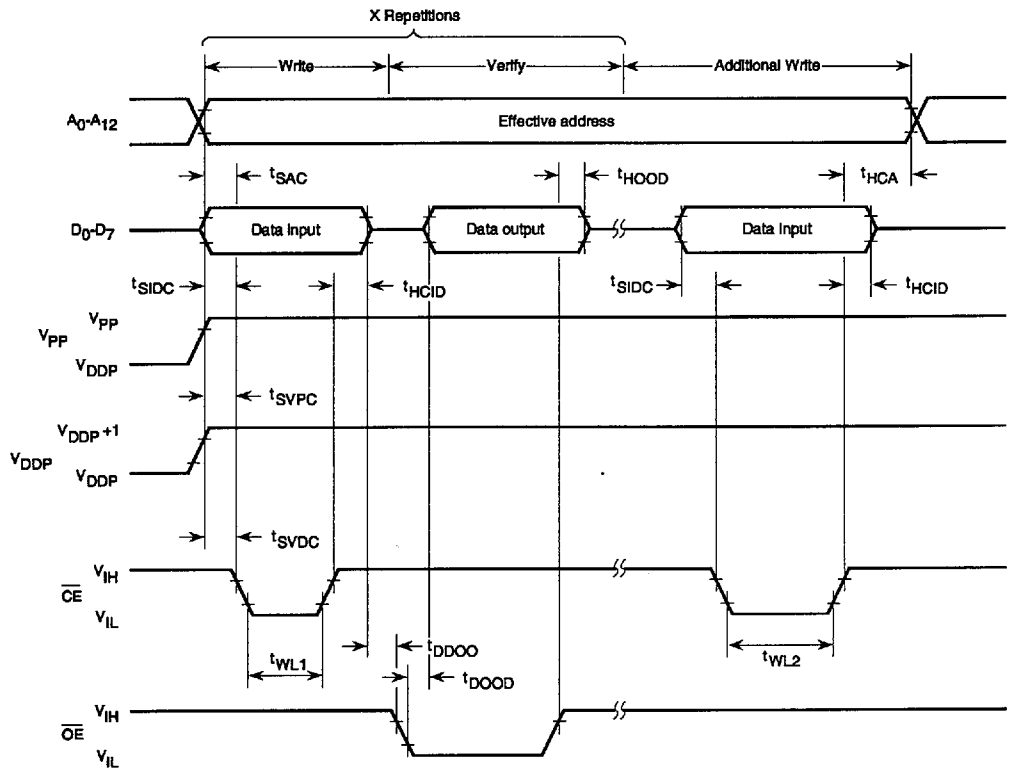
Parameter	Symbol	Symbol (Note 1)	Min	Typ	Max	Unit	Condition
Address setup time to $\overline{CE}$ ↓	t_{SAC}	t_{AS}	2			μs	
Data to to $\overline{OE}$ ↓ delay time	t_{DDOO}	t_{OES}	2			μs	
Input data setup time to $\overline{CE}$ ↓	t_{SIDC}	t_{DS}	2			μs	
Address hold time after $\overline{CE}$ ↑	t_{HCA}	t_{AH}	2			μs	
Input data hold time after $\overline{CE}$ ↑	t_{HCID}	t_{DH}	2			μs	
Output data hold time after $\overline{OE}$ ↑	t_{HOOD}	t_{DF}	0		130	ns	
V_{PP} setup time before $\overline{CE}$ ↓	t_{SVPC}	t_{VPS}	2			μs	
V_{DDP} setup time before $\overline{CE}$ ↓	t_{SVDC}	t_{VDS}	2			μs	
Initial program pulse width	t_{WL1}	t_{PW}	0.95	1.0	1.05	ms	
Additional program pulse width	t_{WL2}	t_{OPW}	2.85		78.75	ms	
Address to data output time	t_{DAOD}	t_{ACC}			2	μs	$\overline{OE}$ = V_{IL}
$\overline{OE}$ ↓ to data output time	t_{DOOD}	t_{OE}			1	μs	
Data hold time after $\overline{OE}$ ↓	t_{HCOD}	t_{DF}	0		130	ns	
Data hold time after address not valid	t_{HAOD}	t_{OH}	0			ns	$\overline{OE}$ = V_{IL}

Notes:

(1) Corresponding symbols for the μPD27C256A.

PROM Timing Diagrams

Write Mode



Notes:

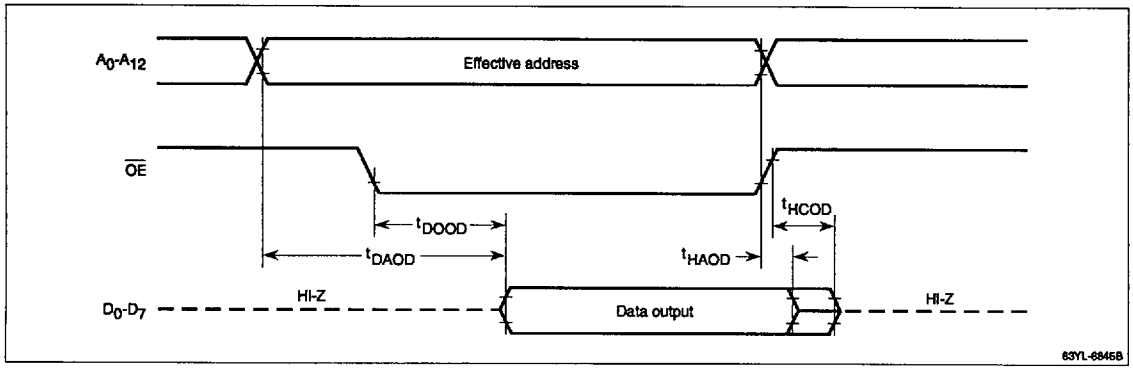
[1] V_{DDP} must be applied before V_{PP} is applied and must be removed after V_{PP} is removed.

[2] V_{PP} must not exceed +13 V including overshoot voltage.

83CL-9337B

PROM Timing Diagrams (cont)

Read Mode



INSTRUCTION SET

The μPD78322 family instruction set features 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

Instruction Set Symbols (cont)

Symbol	Definition
word	16 bits of immediate data or label
byte	8 bits of immediate data or label
jdisp8	8-bit two's complement displacement (immediate data displacement value –128 to +127)
bit	3 bits of immediate data (bit position in byte), or label
n	3 bits of immediate data
!addr16	16-bit absolute address specified by an immediate address or label
$addr16	Relative branch address or label
addr16	16-bit address
!addr11	11-bit immediate address or label
addr11	0800H-0FFFH: 0800H + (11-bit immediate address), or label
addr5	0040H-007EH: 0040H + 2 X (5-bit immediate address), or label
A	A register (8-bit accumulator)
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
R0-R15	Register 0 to register 15
AX	Register pair AX (16-bit accumulator)
BC	Register pair BC
DE	Register pair DE
HL	Register pair HL
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

Instruction Set Symbols (cont)

Symbol	Definition
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

* 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
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					
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	

Instruction Set (cont)

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

Instruction Set (cont)

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

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Stack Manipulation (cont)								
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) ∀ (internal signal level)	3	X	X		P	
CHKLA	sfr	A ← (Pin level) ∀ (internal signal level)	3	X	X		P	
Unconditional Branch								
BR	!addr16	PC ← addr16	3					
	rp1	PC_H ← $rp1_H$, PC_L ← $rp1_L$	2					
	[rp1]	PC_H ← (rp1 + 1), PC_L ← (rp1)	2					
	$addr16	PC ← PC + 2 + jdisp8	2					
Conditional Branch								
BC, BL	$addr16	PC ← PC + 2 + jdisp8 if CY = 1	2					
BNC, BNL	$addr16	PC ← PC + 2 + jdisp8 if CY = 0	2					
BZ, BE	$addr16	PC ← PC + 2 + jdisp8 if Z = 1	2					
BNZ, BNE	$addr16	PC ← PC + 2 + jdisp8 if Z = 0	2					
BV, BPE	$addr16	PC ← PC + 2 + jdisp8 if P/V = 1	2					
BNV, BPO	$addr16	PC ← PC + 2 + jdisp8 if P/V = 0	2					
BN	$addr16	PC ← PC + 2 + jdisp8 if S = 1	2					
BP	$addr16	PC ← PC + 2 + jdisp8 if S = 0	2					
BGT	$addr16	PC ← PC + 3 + jdisp8 if (P/V ∀ 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					

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Flags S	Z	AC	P/V	CY
Conditional Branch (cont)								
BT	saddr.bit, $addr16	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	3					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 1	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 1	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 1	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 1	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 1	3					
BF	saddr.bit, $addr16	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	4					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 0	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 0	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 0	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 0	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 0	3					
BTCLR	saddr.bit, $addr16	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	4					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 1 then reset X.bit	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 1 then reset PSWH.bit	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 1 then reset PSWL.bit	3	X	X	X	X	X
BFSET	saddr.bit, $addr16	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0 then set (saddr.bit)	4					
	sfr.bit, $addr16	PC ← PC + 4 + jdisp8 if sfr.bit = 0 then set sfr.bit	4					
	A.bit, $addr16	PC ← PC + 3 + jdisp8 if A.bit = 0 then set A.bit	3					
	X.bit, $addr16	PC ← PC + 3 + jdisp8 if X.bit = 0 then set X.bit	3					
	PSWH.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWH.bit = 0 then set PSWH.bit	3					
	PSWL.bit, $addr16	PC ← PC + 3 + jdisp8 if PSWL.bit = 0 then set PSWL.bit	3	X	X	X	X	X
DBNZ	r2, $addr16	r2 ← r2 − 1, then PC ← PC + 2 + jdisp8 if (r2) ≠ 0	2					
	saddr, $addr16	(saddr) ← (saddr) − 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0	3					
Context Switching								
BRKCS	RBn	RBS_{2-0} ← 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 (priority change)	3	R	R	R	R	R
RETCSB	!addr16	PC_H ← R5, PC_L ← R4, R5 ← $addr16_H$, R4 ← $addr16_L$, PSWH ← R7, PSWL ← R6 (no priority change)	4	R	R	R	R	R

Instruction Set (cont)

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

Instruction Set (cont)

Notes:

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

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

(3) 16-bit signed multiply instruction

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

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

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

SOLDERING

Packaging and Soldering Information

Part Number	Package	Package Drawing	Recommended Soldering Code
μPD78320GF	80-pin plastic QFP	P80GF-80-3B9-1	IR30-162
GF(A)			
GF(A1)			
GF(A2)			
μPD78320L	68-pin PLCC	P68L-50A1-1	IR30-00, VP15-00
L(A)			
L(A1)			
L(A2)			
μPD78322GF-xxx	80-pin plastic QFP	P80GF-80-3B9-1	IR30-162
GF(A)-xxx			
GF(A1)-xxx			
GF(A2)-xxx			
μPD78322L-xxx	68-pin PLCC	P68L-50A1-1	IR30-00, VP15-00
L(A)-xxx			VP15-00
L(A1)-xxx			VP15-00
L(A2)-xxx			VP15-00
μPD78P322GF	80-pin plastic QFP	P80GF-80-3B9-1	
μPD78P322L	68-pin PLCC	P68L-50A1-1	
μPD78P322KE	80-pin ceramic LCC with window	x80KW-80A	
μPD78P322KC	68-pin ceramic LCC with window	x68KW-50A	

Soldering Conditions

Method (Note 1)	Code	Soldering Conditions (Note 2)	Exposure Limit (Note 3)
Infrared reflow	IR30-00	Package peak temp: 230°C Time: 30 sec max (210°C min)	No limit
	IR30-162		Max no. of days: 2 (thereafter, 16 hours baking at 125°C is required)
Vapor phase	VP15-00	Package peak temp: 215°C Time: 40 sec max (200°C min)	No limit
Pin partial heating		Pin partial temp: 300°C max Time: 3 sec max (per device side)	—

Notes:

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

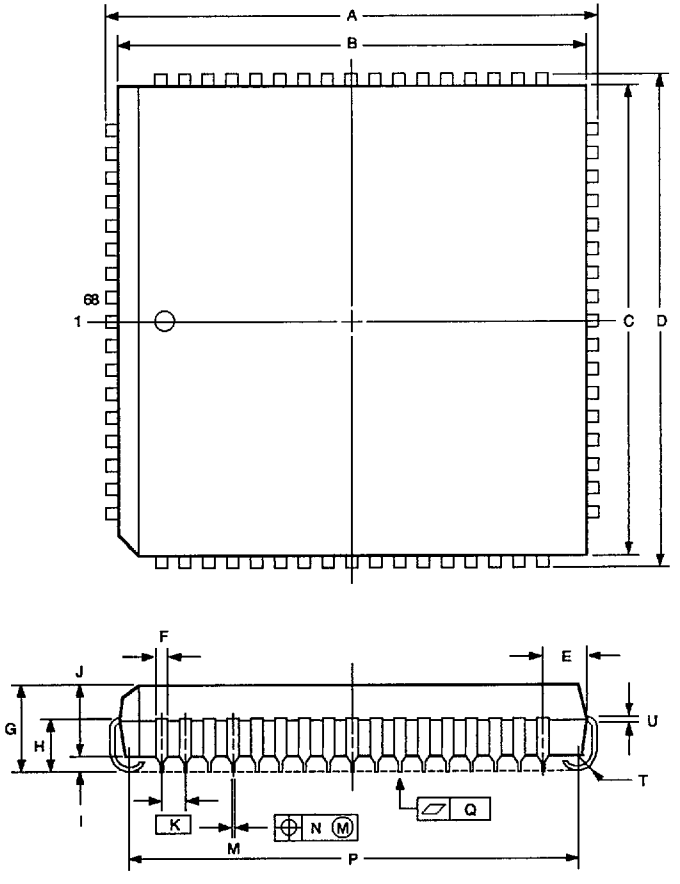
(2) The maximum number of soldering operations is one.

(3) Exposure Limit means maximum days storage at 25°C and 65% RH after the dry pack is opened. Thereafter, baking the devices is required before soldering.

PACKAGE DRAWINGS

68-Pin PLCC (Dwg No. P68L-50A1-1)

Item	Millimeters	Inches
A	25.2 ±0.2	.992 ±.008
B	24.20	.953
C	24.20	.953
D	25.2 ±0.2	.992 ±.008
E	1.94 ±0.15	.076 $^{+.007}_{-.006}$
F	0.6	.024
G	4.4 ±0.2	.173 $^{+.009}_{-.008}$
H	2.8 ±0.2	.110 $^{+.009}_{-.008}$
I	0.9 min	.035 min
J	3.4	.134
K	1.27 (TP)	.050 (TP)
M	0.40 ±0.10	.016 $^{+.004}_{-.005}$
N	0.12	.005
P	23.12 ±0.20	.910 $^{+.009}_{-.008}$
Q	0.15	.006
T	0.8 radius	.031 radius
U	0.20 $^{+0.10}_{-0.05}$	.008 $^{+.004}_{-.002}$



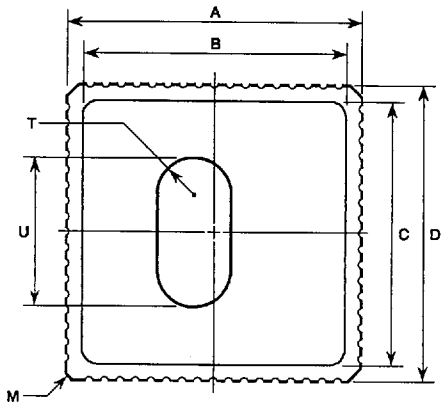
P68L-50A1-1

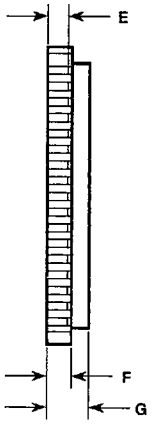
83YL-5561B (5/93)

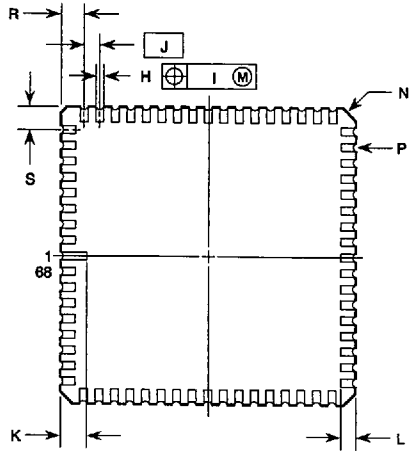
PACKAGE DRAWINGS (cont)

68-Pin Ceramic LCC With Window (Dwg No. x68KW-50A)

Item	Millimeters	Inches
A	24.13 ± 0.40	.95 ± .016
B	21.5	.846
C	21.5	.846
D	24.13 ± 0.40	.95 ± .016
E	1.65	.065
F	2.03	.08
G	3.50 max	.138 max
H	0.64 ± 0.10	.025 ± .004
I	0.12	.005
J	1.27	.05
K	2.16 ± 0.20	.085 ± .008
L	1.27 ± 0.20	.05 ± .008
M	C0.50	C.02
N	C1.02	C.04
P	R0.20	R.008
R	1.905	.075
S	1.905	.075
T	R3.0	R.118
U	12.0	.472







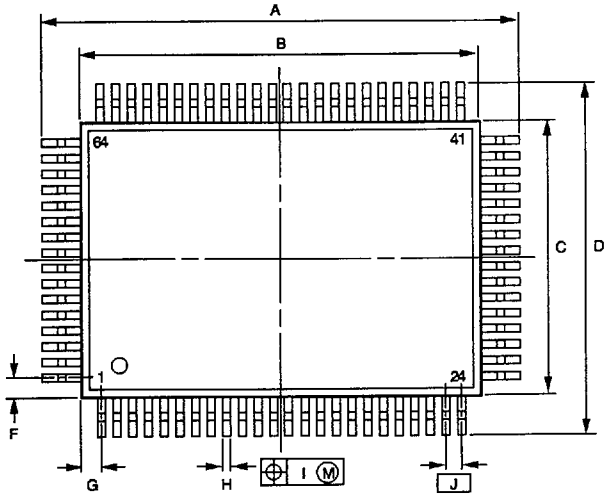
X68KW-50A

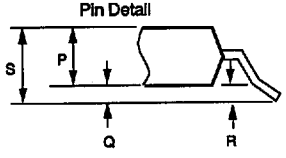
83FM-8390B

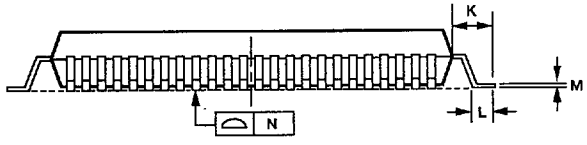
PACKAGE DRAWINGS (cont)

80-Pin Plastic QFP (Dwg No. P80GF-80-3B9-1)

Item	Millimeters	Inches
A	23.6 ±0.4	.929 ±.016
B	20.0 ±0.2	.787 $^{+.009}_{-.008}$
C	14.0 ±0.2	.551 $^{+.009}_{-.008}$
D	17.6 ±0.4	.693 ±.016
F	1.0	.039
G	0.8	.031
H	0.35 ±0.10	.014 $^{+.004}_{-.005}$
I	0.15	.006
J	0.8 (TP)	.031 (TP)
K	1.8 ±0.2	.071 $^{+.009}_{-.008}$
L	0.8 ±0.2	.031 $^{+.009}_{-.008}$
M	0.15 $^{+0.10}_{-0.05}$	.006 $^{+.004}_{-.002}$
N	0.15	.006
P	2.7	.106
Q	0.1 ±0.1	.004 ±.004
R	0.1 ±0.1	.004 ±.004
S	3.0 max	.118 max







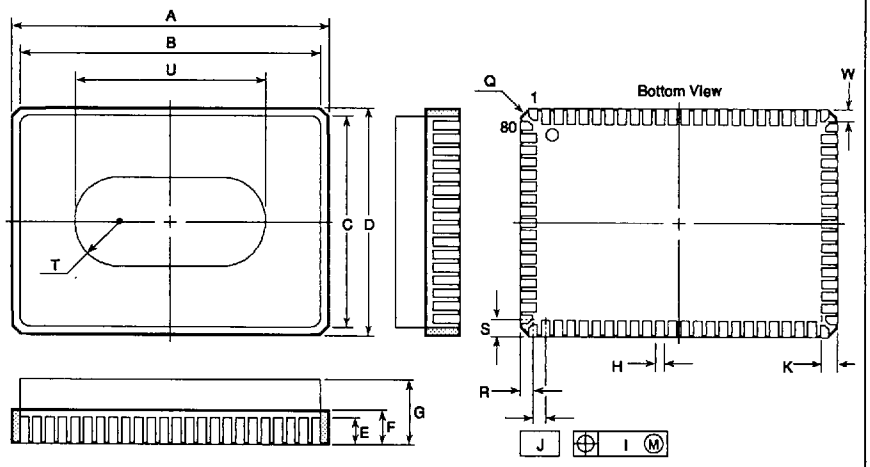
P80GF-80-3B9-1

83IH-5543B (10/91)

PACKAGE DRAWINGS (cont)

80-Pin Ceramic LCC With Window (Dwg No. x80KW-80A)

Item	Millimeters	Inches
A	20.0 ± 0.4	.787 $^{+.017}_{-.016}$
B	19.0	.748
C	13.2	.520
D	14.2 ± 0.4	.559 ± .016
E	1.64	.065
F	2.14	.084
G	4.064 max	.160 max
H	0.51 ± 0.10	.020 ± .004
I	0.08	.003
J	0.8 (TP)	.031 (TP)
K	1.0 ± 0.2	.039 $^{+.009}_{-.008}$
Q	0.5 cor	.020 cor
R	0.8	.031
S	1.1	.043
T	3.0 rad	.118 rad
U	12.0	.472
W	0.75 ± 0.2	.030 $^{+.008}_{-.009}$



X80KW-80A

49NR-617B (11/89)