

**NEC**

NEC Electronics Inc.

**μPD7822x****Advanced, 8-Bit****Real-Time Control Microcomputers  
With Analog Comparators**

T-49-19-07

T-49-19-59

**Description**

The μPD78220, μPD78224, and μPD78P224 are high-performance, 8-bit, single-chip microcomputers. They contain extended addressing capabilities for up to 1M byte of external memory. The devices also integrate sophisticated analog and digital peripherals as well as two low-power standby modes that make them ideal for low-power/battery backup applications.

The μPD7822x family focuses on embedded control with features such as hardware multiply and divide, two levels of interrupt response, four banks of main registers for multi-tasking, and macroservice for processor-independent peripheral and memory DMA. Augmenting this high-performance core are advanced components; for example, eight analog voltage comparators, two independent serial interfaces, several counter/timers for PWM outputs, and a real-time output port. On board memory includes 640 bytes of RAM and 16K bytes of mask ROM or OTP ROM.

The macroservice routine allows data to be transferred between any combination of memory and peripherals independent of the current program execution. The four banks of processor registers allow simplified context switching to be performed. Both features combined with powerful on-chip peripherals make this part ideal for a wide variety of embedded control applications.

- Memory expansion
  - 8085 bus-compatible
  - 64K program address space
  - 1M data address space
- Large I/O capacity: up to 71 I/O port lines
- Extensive timer/counter functions
  - One 16-bit timer/counter/event counter
  - Two 8-bit timer/counter/event counter
- Four timer-controlled PWM channels
- Two 4-bit real-time output ports
- Extensive interrupt handler
  - Vectored interrupt handling
  - Programmable priority
  - Macroservice mode
- Two independent serial ports
- Refresh output for pseudostatic RAM
- On-chip clock generator
  - 12-MHz maximum CPU clock frequency
  - 0.33-μs instruction cycle
- CMOS silicon gate technology
- 5-volt power supply

**Features**

- Complete single-chip microcomputer
  - 8-bit ALU
  - 16K ROM
  - 640 bytes RAM
  - Both 1-bit and 8-bit logic
- Instruction prefetch queue
- Hardware multiply and divide

**Ordering Information**

Part Number	ROM	Package
μPD78220L	ROMless	84-pin PLCC
μPD78220GJ		94-pin plastic QFP
μPD78224L	16K Mask ROM	84-pin PLCC
μPD78224GJ		94-pin plastic QFP
μPD78P224L	16K OTP ROM	84-pin PLCC
μPD78P224GJ		94-pin plastic QFP

**μPD7822x**

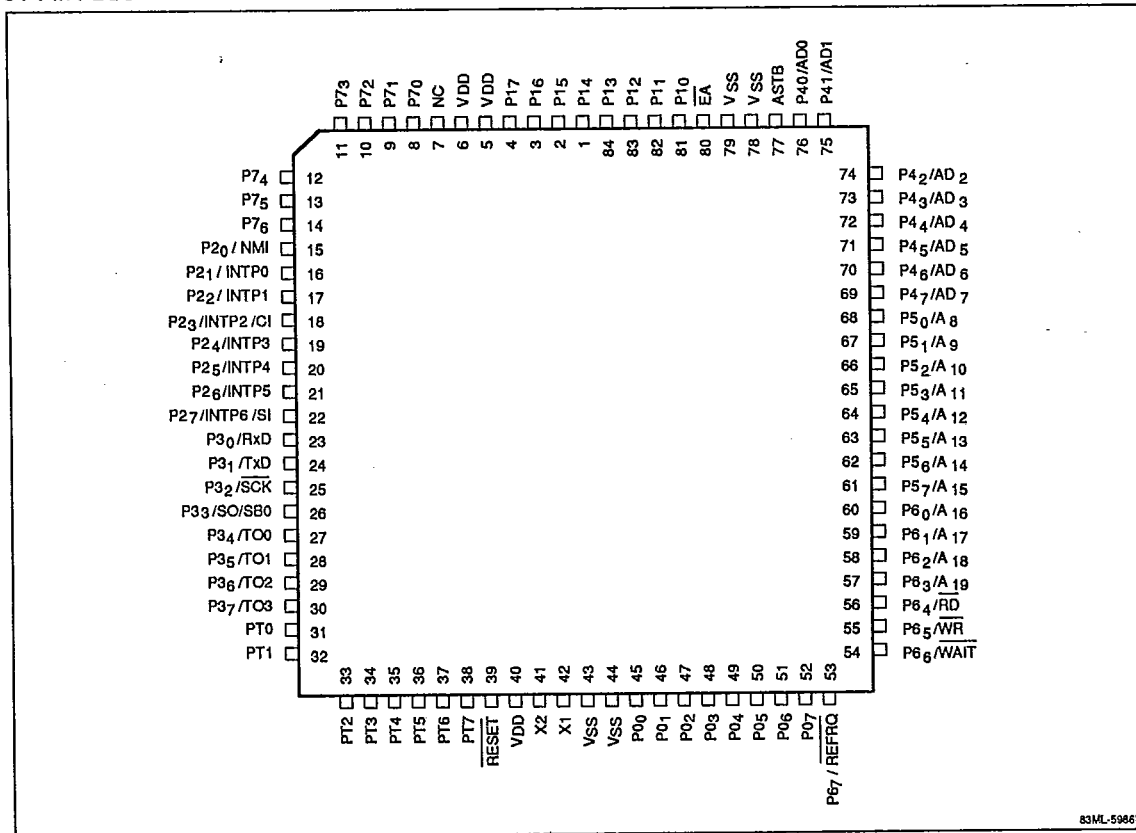
**Pin Identification**

Symbol	Function
P0 <sub>0</sub> -P0 <sub>7</sub>	Output port 0
P1 <sub>0</sub> -P1 <sub>7</sub>	I/O port 1
P2 <sub>0</sub> /NMI	Input port 2/Non-maskable interrupt input
P2 <sub>1</sub> -P2 <sub>2</sub> /INTP0-INTP1	Input port 2/Ext interrupt input/timer trigger
P2 <sub>3</sub> /INTP2/CI	Input port 2/Ext interrupt input/Clock Input
P2 <sub>4</sub> /INTP3	Input port 2/Ext interrupt input/timer trigger
P2 <sub>5</sub> /INTP4	Input port 2/External interrupt input
P2 <sub>6</sub> /INTP5	Input port 2/External interrupt input
P2 <sub>7</sub> /INTP6/SI	Input port 2/Ext interrupt input/Serial input
P3 <sub>0</sub> /RxD	I/O port 3/Serial receive input
P3 <sub>1</sub> /TxD	I/O port 3/Serial transmit output
P3 <sub>2</sub> /SCK	I/O port 3/Serial clock input/output
P3 <sub>3</sub> /SO/SB0	I/O port 3/Serial output/Serial bus I/O
P3 <sub>4</sub> -P3 <sub>7</sub> /TO0-TO3	I/O port 3/Timer output
P4 <sub>0</sub> -P4 <sub>7</sub> /AD <sub>0</sub> -AD <sub>7</sub>	I/O port 4/Lower address byte/data bus

Symbol	Function
P5 <sub>0</sub> -P5 <sub>7</sub> /A <sub>8</sub> -A <sub>15</sub>	I/O port 5/Upper address byte
P6 <sub>0</sub> -P6 <sub>3</sub> /A <sub>16</sub> -A <sub>19</sub>	Output port 6/Extended address nibble
P6 <sub>4</sub> /RD	I/O port 6/Read strobe output
P6 <sub>5</sub> /WR	I/O port 6/Write strobe output
P6 <sub>6</sub> /WAIT	I/O port 6/Wait input
P6 <sub>7</sub> /REFRQ	I/O port 6/Refresh output
P7 <sub>0</sub> -P7 <sub>6</sub>	I/O port 7
PT0-PT7	Port T analog inputs to voltage comparators
ASTB	Address strobe output
RESET	External reset input
EA	External memory access control input
X1, X2	External crystal or external clock input
VDD	Positive power supply input
VSS	Power return; normally ground
NC	No connection
IC	Internal connection; connect to VSS

**Pin Configurations**

**84-Pin PLCC**



83ML-5966B

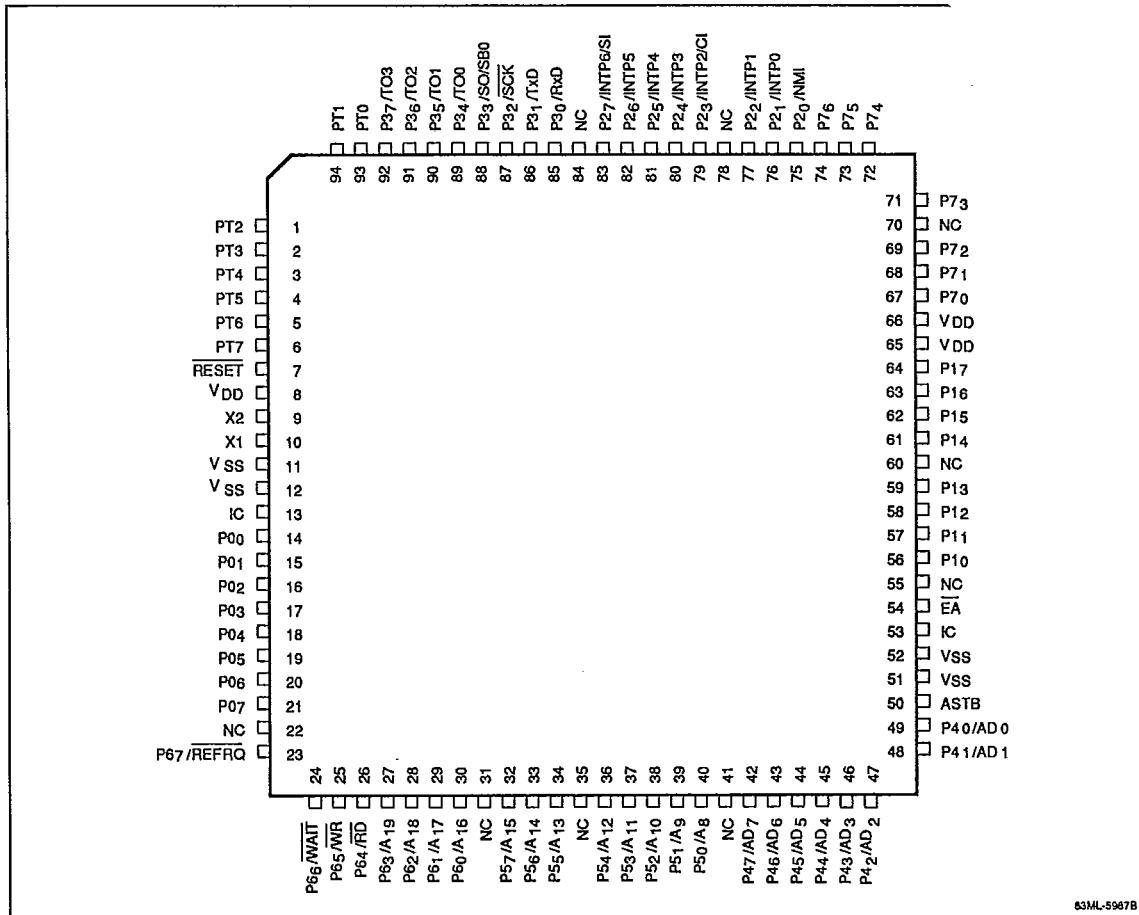


μPD7822x

T-49-19-07

94-Pin Plastic QFP

T-49-19-59



63ML-5967B

**NEC**

T-49-19-07

T-49-19-59

**μPD7822x****Pin Functions**

**P0<sub>0</sub>-P0<sub>7</sub>.** Port 0 is an 8-bit, tristate output port. Port 0 can also be configured as two 4-bit, real-time (timer-controlled) output ports.

**P1<sub>0</sub>-P1<sub>7</sub>.** Port 1 is an 8-bit bidirectional tristate port. Bits are individually programmable as input/output. Each pin is capable of driving an LED directly (8 mA).

**P2<sub>0</sub>-P2<sub>7</sub>.** Port 2 is an 8-bit input port.

**NMI.** Non-maskable interrupt input.

**INTP0-INTP6.** External interrupt inputs. INTP0, INTP1, and INTP3 are timer capture trigger inputs.

**Cl.** External clock input to the timer.

**SI.** Serial data input for three-line serial I/O mode.

**P3<sub>0</sub>-P3<sub>7</sub>.** Port 3 is an 8-bit tristate I/O port, each bit programmable as input/output.

**RxD.** Receive serial data input.

**TxD.** Transmit serial data output.

**SCK.** Serial shift clock output/input.

**SO.** Serial data output for three-line serial I/O mode.

**SB0.** I/O bus for the clocked serial interface.

**TO0-TO3.** Timer flip-flop outputs.

**P4<sub>0</sub>-P4<sub>7</sub>.** Port 4 is an 8-bit, bidirectional tristate port.

**AD<sub>0</sub>-AD<sub>7</sub>.** Multiplexed address/data bus used with external memory or expanded I/O.

**P5<sub>0</sub>-P5<sub>7</sub>.** Port 5 is an 8-bit, tristate output port.

**A<sub>8</sub>-A<sub>15</sub>.** Upper-order address bus used with external memory or expanded I/O.

**P6<sub>0</sub>-P6<sub>3</sub>.** Pins P6<sub>0</sub>-P6<sub>3</sub> of port 6 are outputs.

**A<sub>16</sub>-A<sub>19</sub>.** Extended-order address bus used with external memory.

**P6<sub>4</sub>-P6<sub>7</sub>.** Pins P6<sub>4</sub>-P6<sub>7</sub> of port 6 are individually programmable tristate input/output pins.

**RD.** Read strobe output used by external memory (or data registers) to place data on the I/O bus during a read operation.

**WR.** Write strobe output used by external memory (or data registers) to latch data from the I/O bus during a write operation.

**WAIT.** Wait signal input.

**REFRQ.** Refresh pulse output used by external pseudo-static memory.

**P7<sub>0</sub>-P7<sub>6</sub>.** Port 7 has seven individually programmable tristate I/O pins.

**PT0-PT7.** Port T is an eight-line input port. The analog voltage on each line is compared continuously with a programmable threshold voltage.

**ASTB.** Address strobe output used by external circuitry to latch the low-order 8 address bits during the first part of a read or write cycle.

**RESET.** A low level on this external reset input sets all registers to their specified reset values. This pin, together with P2<sub>0</sub>/NMI, sets the μPD78P224 in the PROM programming mode.

**EA.** Control signal input that selects external memory or internal ROM as the program memory. When EA is low, ROMless mode is initiated and external memory is accessed.

**X1, X2.** For frequency control of the internal clock oscillator, a crystal is connected to X1 and X2. If the clock is supplied by an external source, the clock signal is connected to X1 and the inverted clock signal is connected to X2.

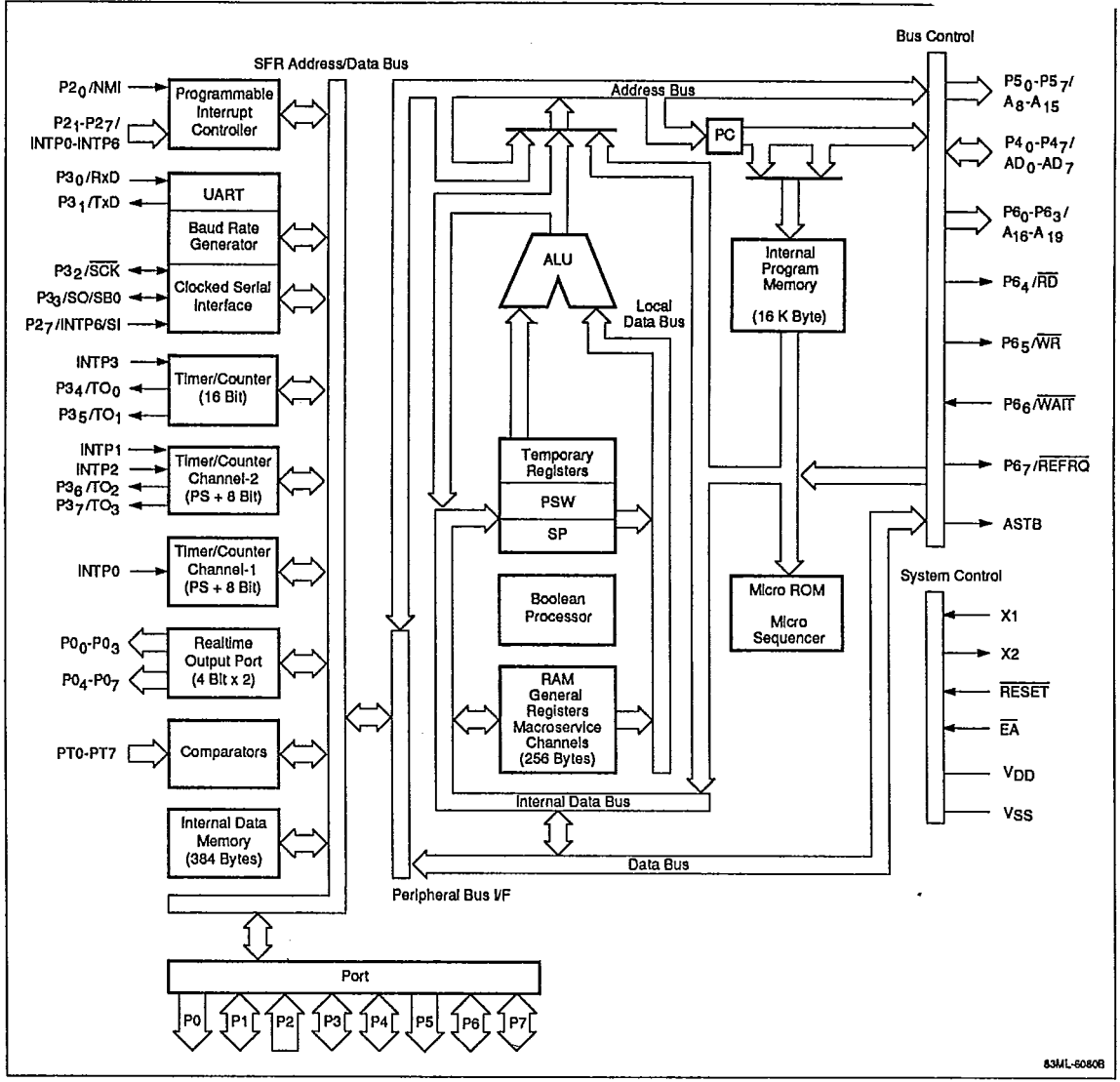


μPD7822x

T-49-19-07

μPD7822x Block Diagram

T-49-19-59



63ML-50808

**μPD7822x**

**FUNCTIONAL DESCRIPTION**

**Timing**

The maximum clock frequency is 12 MHz. The clock is derived from an external crystal or an external oscillator. The internal processor clock is two-phase and the machine states are executed at a rate of 6 MHz. The shortest instructions require two states (333 ns). The CPU contains a one-byte instruction prefetch. This allows a subsequent instruction to be fetched during the execution of an instruction that does not reference memory.

**Memory Map**

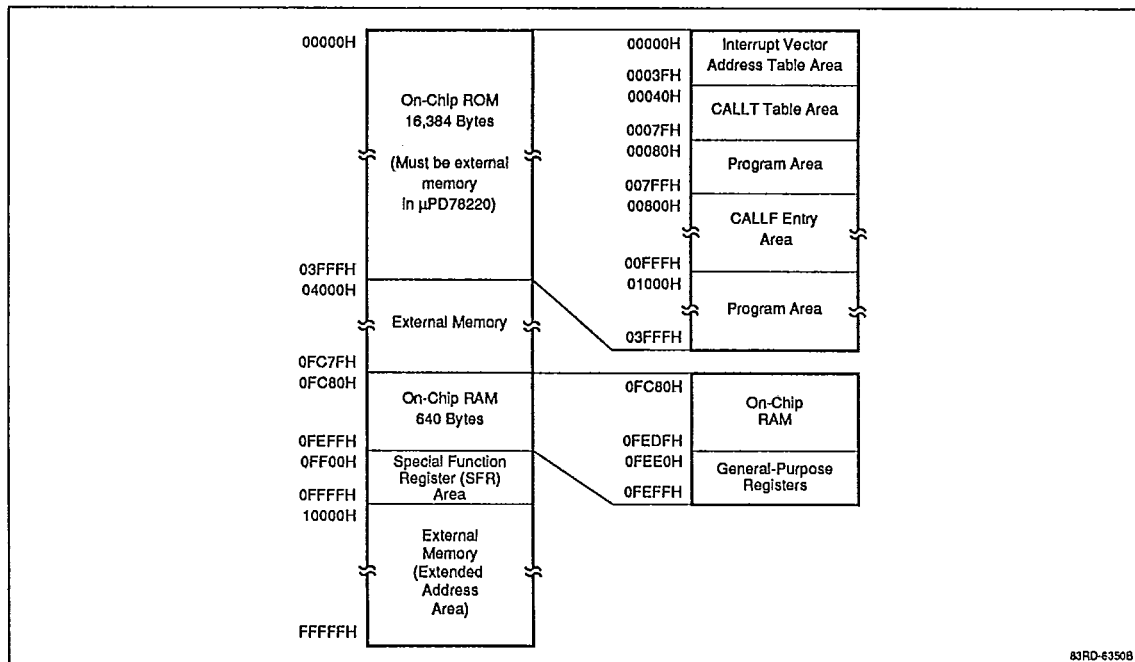
The μPD7822x has 1M bytes of address space. This address space is partitioned into 64K bytes of program

memory starting at address 00000H. (See figure 1.) The remainder of the 1M bytes can be accessed as data memory space.

External memory is supported by I/O port 4, an 8-bit multiplexed address/data bus. The memory mapping register controls the size of external memory as well as the number of added wait states. The upper address byte is derived from port 5, and the extended address nibble is derived from port 6.

The μPD78224 has on-chip mask ROM occupying the space from 00000H to 03FFFH. When the ROM is used and no other program or data space is required, ports 4, 5, and 6 are available as additional I/O ports.

**Figure 1. Memory Map**



63RD-63508



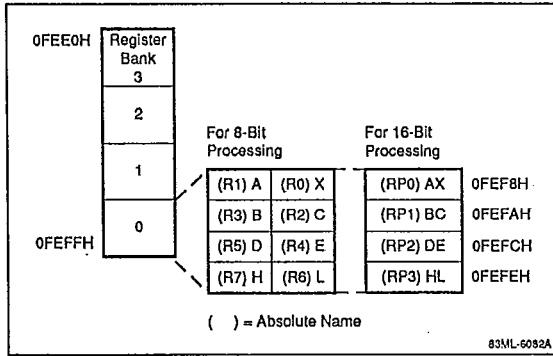
μPD7822x

T-49-19-07

**General-Purpose Registers**

The general-purpose registers are mapped into specific addresses in data memory. They are made up of four banks, each bank consisting of eight 8-bit or four 16-bit registers. The register bank used is specified by a CPU instruction. This can be checked by reading RBS0 and RBS1 in the program status word (PSW). The general-purpose register configuration is shown in figure 2.

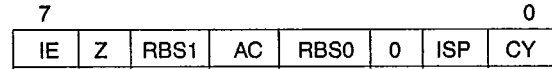
**Figure 2. Register Mapping**



**Special Registers**

T-49-19-59

There are three different special registers. The first is a 16-bit binary counter that holds the next program address to be executed and is named the program counter. The stack pointer is the second special 16-bit register. The stack pointer holds the address of the stack area (a last in, first out system). The third special register is an 8-bit program status word. This register contains various flags that are set or reset depending on the results of instruction execution. The program status word format is as follows:



- CY                    Carry flag
- ISP                    Interrupt priority status flag
- RBS0, RBS1        Register bank selection flags
- AC                    Auxiliary carry flag
- Z                     Zero flag
- IE                    Interrupt request enable flag

**Special Function Registers**

These registers are assigned to special functions such as the mode and control registers for on-chip peripheral hardware. They are mapped into the 256-byte memory space from 0FF00H to 0FFFFH. Table 1 is a list of special function registers.



μPD7822x

**NEC**

T-49-19-07

Table 1. Special Function Registers

T-49-19-59

Address	Special Function Register (SFR) Name	Symbol	R/W	Handleable Bit Unit			On Reset
				1 Bit	8 Bit	16 Bit	
0FF00H	Port 0	P0	R/W	0	0	–	Indeterminate
0FF01H	Port 1	P1	R/W	0	0	–	Indeterminate
0FF02H	Port 2	P2	R	0	0	–	Indeterminate
0FF03H	Port 3	P3	R/W	0	0	–	Indeterminate
0FF04H	Port 4	P4	R/W	0	0	–	Indeterminate
0FF05H	Port 5	P5	R/W	0	0	–	Indeterminate
0FF06H	Port 6	P6	R/W	0	0	–	x0H
0FF07H	Port 7	P7	R/W	0	0	–	Indeterminate
0FF0AH	Port 0 buffer register (low)	P0L	R/W	0	0	–	Indeterminate
0FF0BH	Port 0 buffer register (high)	P0H	R/W	0	0	–	Indeterminate
0FF0CH	Real-time output port control register	RTPC	R/W	0	0	–	00H
0FF10H, 0FF11H	16-bit compare register 0 (16-bit timer/counter)	CR00	R/W	–	–	0	Indeterminate
0FF12H, 0FF13H	16-bit compare register 1 (16-bit timer/counter)	CR01	R/W	–	–	0	Indeterminate
0FF14H	8-bit compare register (8-bit timer/counter 1)	CR10	R/W	–	0	–	Indeterminate
0FF15H	8-bit compare register (8-bit timer/counter 2)	CR20	R/W	–	0	–	Indeterminate
0FF16H	8-bit compare register (8-bit timer/counter 2)	CR21	R/W	–	0	–	Indeterminate
0FF17H	BRG 8-bit compare register	CR30	R/W	–	0	–	Indeterminate
0FF18H, 0FF19H	16-bit capture register (16-bit timer/counter)	CR02	R	–	–	0	Indeterminate
0FF1AH	8-bit capture register (8-bit timer/counter 2)	CR22	R	–	0	–	Indeterminate
0FF1CH	8-bit capture/compare register (8-bit timer/counter 1)	CR11	R/W	–	0	–	Indeterminate
0FF20H	Port 0 mode register	PM0	W	–	0	–	FFH
0FF21H	Port 1 mode register	PM1	W	–	0	–	FFH
0FF23H	Port 3 mode register	PM3	W	–	0	–	FFH
0FF25H	Port 5 mode register	PM5	W	–	0	–	FFH
0FF26H	Port 6 mode register	PM6	R/W	–	0	–	FFH
0FF27H	Port 7 mode register	PM7	W	–	0	–	7FH
0FF30H	Capture/compare control register 0	CRC0	W	–	0	–	10H
0FF31H	Timer output control register	TOC	W	–	0	–	00H
0FF32H	Capture/compare control register 1	CRC1	W	–	0	–	00H
0FF34H	Capture/compare control register 2	CRC2	W	–	0	–	00H
0FF43H	Port 3 mode control register	PMC3	R/W	0	0	–	00H
0FF50H, 0FF51H	16-bit timer register 0	TM0	R	–	–	0	0000H
0FF52H	8-bit timer register: CH-1	TM1	R	–	0	–	00H



**NEC****μPD7822x**

T-49-19-07

T-49-19-59

**Table 1. Special Function Registers (cont)**

Address	Special Function Register (SFR) Name	Symbol	R/W	Handleable Bit Unit			On Reset
				1 Bit	8 Bit	16 Bit	
0FF54H	8-bit timer register: CH-2	TM2	R	-	o	-	00H
0FF56H	BRG 8-bit timer register	TM3	R	-	o	-	00H
0FF5CH	Prescaler mode register 0	PRM0	W	-	o	-	00H
0FF5DH	Timer control register 0	TMC0	R/W	-	o	-	00H
0FF5EH	Prescaler mode register 1	PRM1	W	-	o	-	00H
0FF5FH	Timer control register 1	TMC1	R/W	-	o	-	00H
0FF6EH	Port T mode register	PMT	R/W	o	o	-	00H
0FF6FH	Port T	PT	R	o	o	-	Indeterminate
0FF80H	Clocked serial interface mode register	CSIM	R/W	o	o	-	00H
0FF82H	Serial bus interface control register	SBIC	R/W	o	o	-	00H
0FF86H	Serial shift register	SIO	R/W	-	o	-	Indeterminate
0FF88H	Asynchronous serial interface mode register	ASIM	R/W	o	o	-	80H
0FF8AH	Asynchronous serial interface status register	ASIS	R	o	o	-	00H
0FF8CH	Serial receive buffer: UART	RxB	R	-	o	-	Indeterminate
0FF8EH	Serial send shift register: UART	TxS	W	-	o	-	Indeterminate
0FFC0H	Standby control register	STBC	R/W	-	o	-	0000x000B
0FFC4H	Memory expansion mode register	MM	R/W	o	o	-	20H
0FFC5H	Programmable wait control register	PW	R/W	o	o	-	80H
0FFC6H	Refresh mode register	RFM	R/W	o	o	-	00H
0FFE0H	Interrupt request flag register L	IF0L IF0	R/W	o	o	o	Indeterminate
0FFE1H	Interrupt request flag register H	IF0H	R/W	o	o		Indeterminate
0FFE4H	Interrupt mask flag register L	MK0L MK0	R/W	o	o	o	FFFFH
0FFE5H	Interrupt mask flag register H	MK0H	R/W	o	o		FFFFH
0FFE8H	Priority specification flag register L	PR0L PR0	R/W	o	o	o	FFFFH
0FFE9H	Priority specification flag register H	PR0H	R/W	o	o		FFFFH
0FFECH	Interrupt service mode specification flag register L	ISM0L ISM0	R/W	o	o	o	0000H
0FFEDH	Interrupt service mode specification flag register H	ISM0H	R/W	o	o		0000H
0FFF4H	External interrupt mode register 0	INTM0	R/W	o	o	-	00H
0FFF5H	External interrupt mode register 1	INTM1	R/W	o	o	-	00H
0FFF8H	Interrupt status register	IST	R/W	o	o	-	00H

**6**

**Input/Output Ports**

Functions of ports P0-P7 and PT are explained below. All ports are 8 bits wide except P7, which is 7 bits wide.

Port	Function
P0	8-bit output port or two 4-bit real time output ports
P1	Bit programmable for input or output; large current capacity
P2	Input
P3	Bit programmable for input or output
P4	Input or output
P5	Output
P6 <sub>0</sub> -P6 <sub>3</sub>	Output
P6 <sub>4</sub> -P6 <sub>7</sub>	Bit programmable for input or output
P7	Bit programmable for input or output
PT	Inputs to eight voltage comparators

**Real-Time Output Port**

The real-time output port (figure 3) shares pins with port 0. The high and low nibbles may be treated separately or together. In the real-time output function, data stored beforehand in the buffer register is transferred to the output

latch simultaneously with the generation of either a timer interrupt or external interrupt. Using the real-time output function in conjunction with the macroservice function enables port 0 to output preprogrammed patterns at pre-programmed variable time intervals.

**Port T**

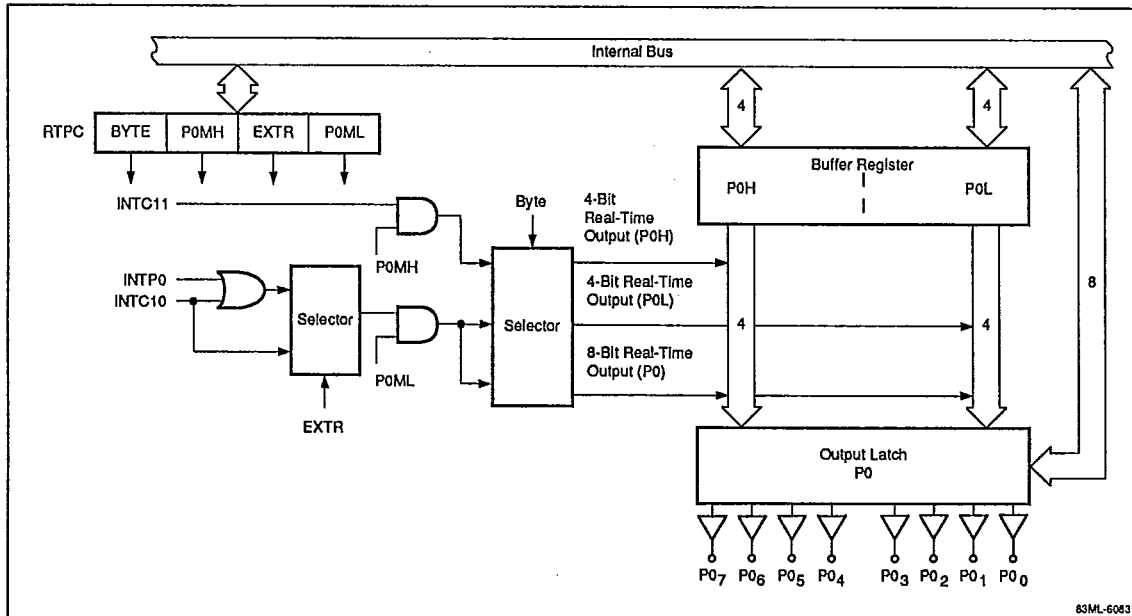
As shown in figure 4, the analog input voltage on each line of port T is compared with a programmable threshold voltage. The comparator output is 1 if the input voltage is higher than the threshold or 0 if it is lower.

Four bits from the PTM register are decoded to set the threshold voltage at one of 15 steps:  $V_{DD} \times 1/16$  through  $V_{DD} \times 15/16$ . Each comparator operates continuously as follows.

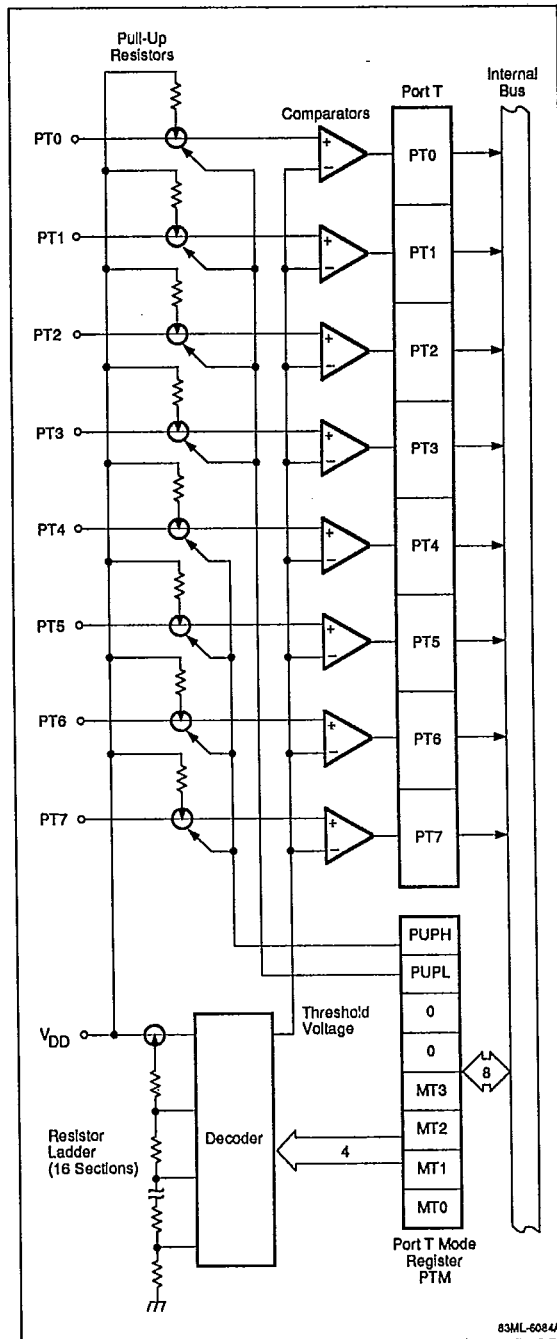
- (1) Threshold voltage is set by writing the PTM register.
- (2) As each comparison is completed, the result is latched in port T and the next comparison begins.
- (3) Unless the PTM register is rewritten, the threshold voltage is not changed.

Two bits from the PTM register specify the connection of pull-up resistors in 4-bit units. When PTM is set to 00H, the resistor ladder is released and threshold voltage is not supplied to the comparators. This can be done in the standby mode to eliminate unnecessary current drain.

**Figure 3. Real-Time Output Port**



**Figure 4. Comparator Port T**



**Serial Interface**

The μPD7822x has two independent serial interfaces.

- Asynchronous serial interface (UART) (figure 5)
- Clock-synchronized serial interface (figure 6)

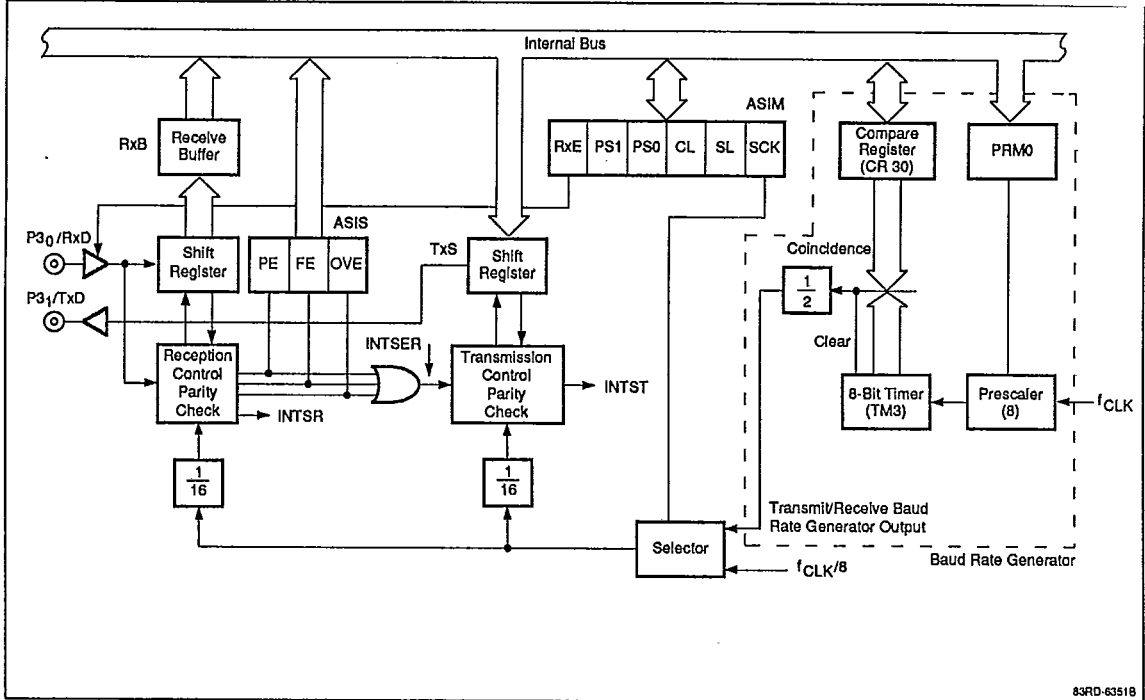
A universal asynchronous receiver transmitter (UART) is used as an asynchronous serial interface. This interface transfers one byte of data following a start bit. The μPD7822x contains a baud rate generator. This allows data to be transferred over a wide range of transfer rates.

The clock-synchronized serial interface has two different modes of operation:

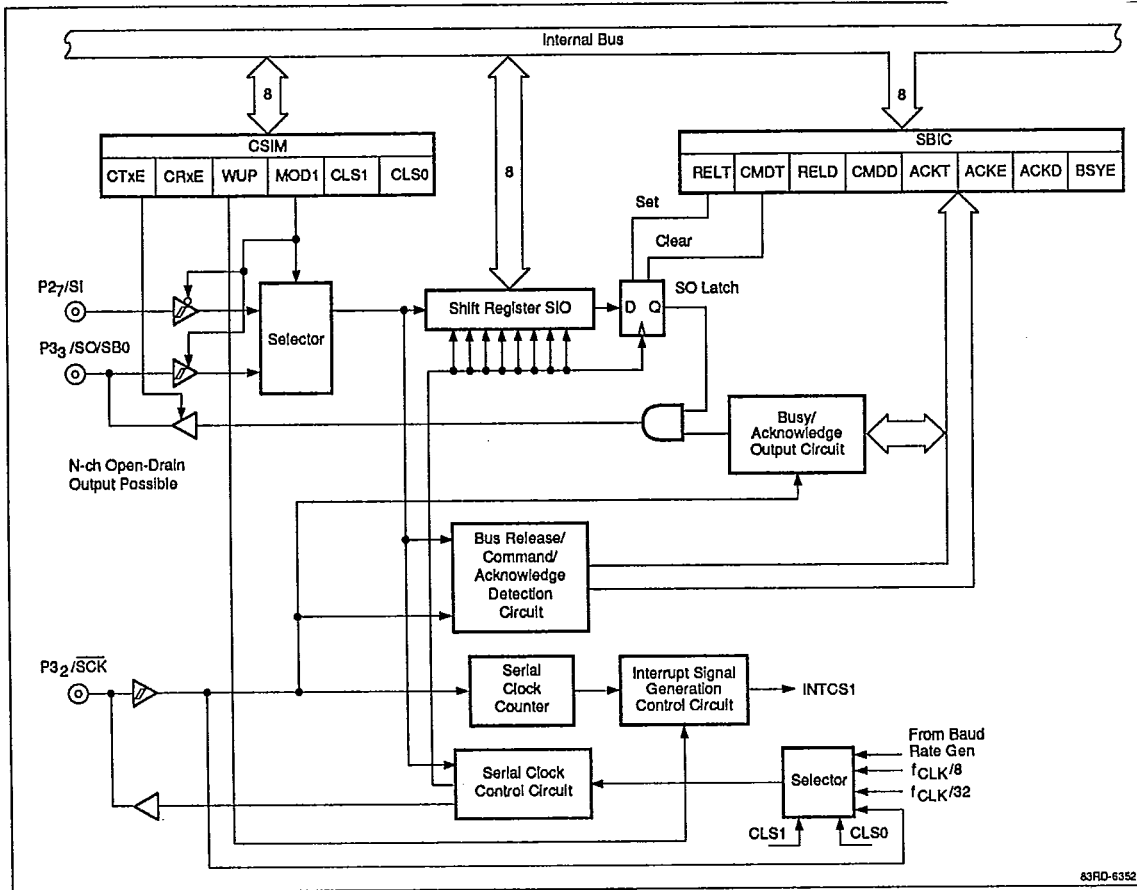
- Three-line serial I/O mode.  
In this mode, data 8 bits long is transferred along three lines: a serial clock (SCK) line and two serial bus lines (SO and SI). This mode is convenient when the μPD7822x is connected to peripheral I/Os and display controllers that have the conventional clock-synchronized serial interface.
- Serial bus interface mode (SBI).  
In this mode the μPD7822x can communicate data with several devices using the serial clock (SCK) and the serial data bus (SB0) lines. This mode conforms to NEC's serial bus format. In SBI mode, addresses that select a device to communicate with, commands that direct the device, and actual data are output to the serial data bus. A handshake line, which was required for connecting several devices in the conventional clock-synchronized serial interface, is not needed.



Figure 5. Asynchronous Serial Interface



83RD-63518

**Figure 6. Clock-Synchronized Serial Interface**

**Timer/Counters**

The μPD7822x has three timer/counters: one 16-bit and two 8-bit. The 16-bit timer/counter (figure 7) has the basic functionality of an interval timer, a programmable square-wave output, and a pulse width measurer. These functions can provide a digital delayed one-shot output, a pulse width modulated output, and a cycle measurer.

The two 8-bit timer/counters can provide the basic functions of an interval timer and a pulse width measurer. Timer/counter 1 can also be used as a timer for output trigger generation for the real-time output port. Timer/counter 2 can also provide an external event counter, a one-shot timer, a programmable square-wave output, a pulse-width modulated output, and a cycle measurer. See figures 8 and 9.

**Figure 7. 16-Bit Timer/Counter**

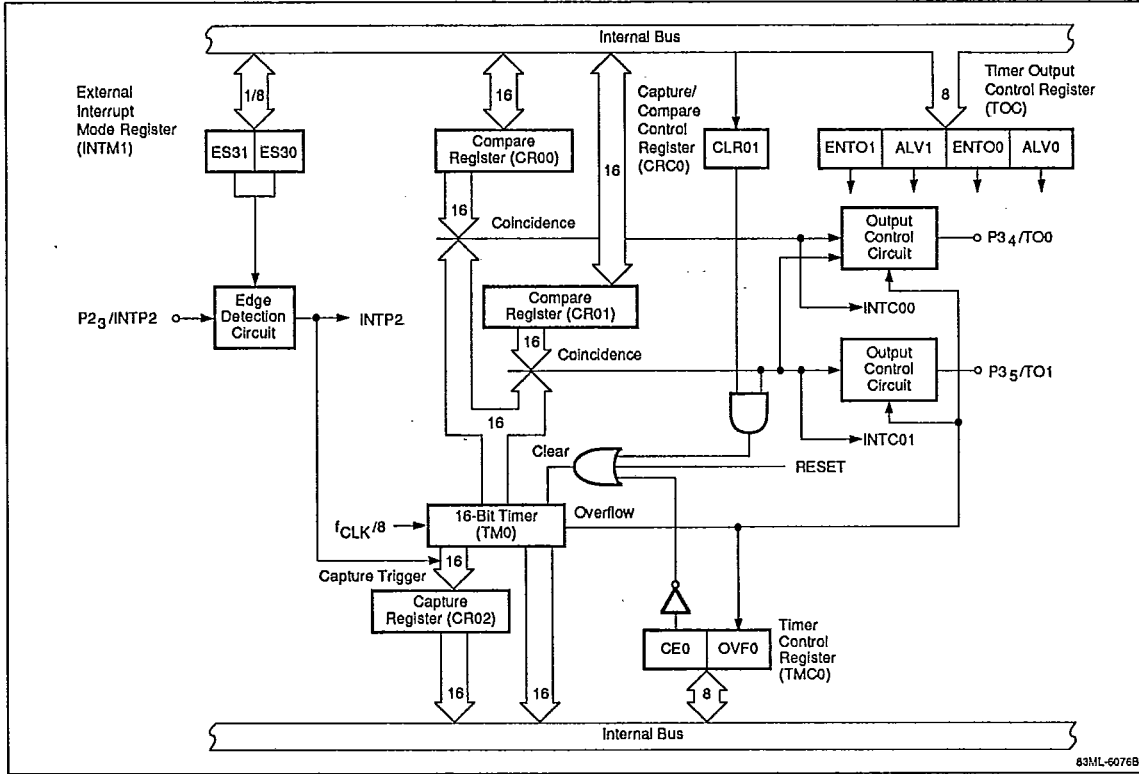
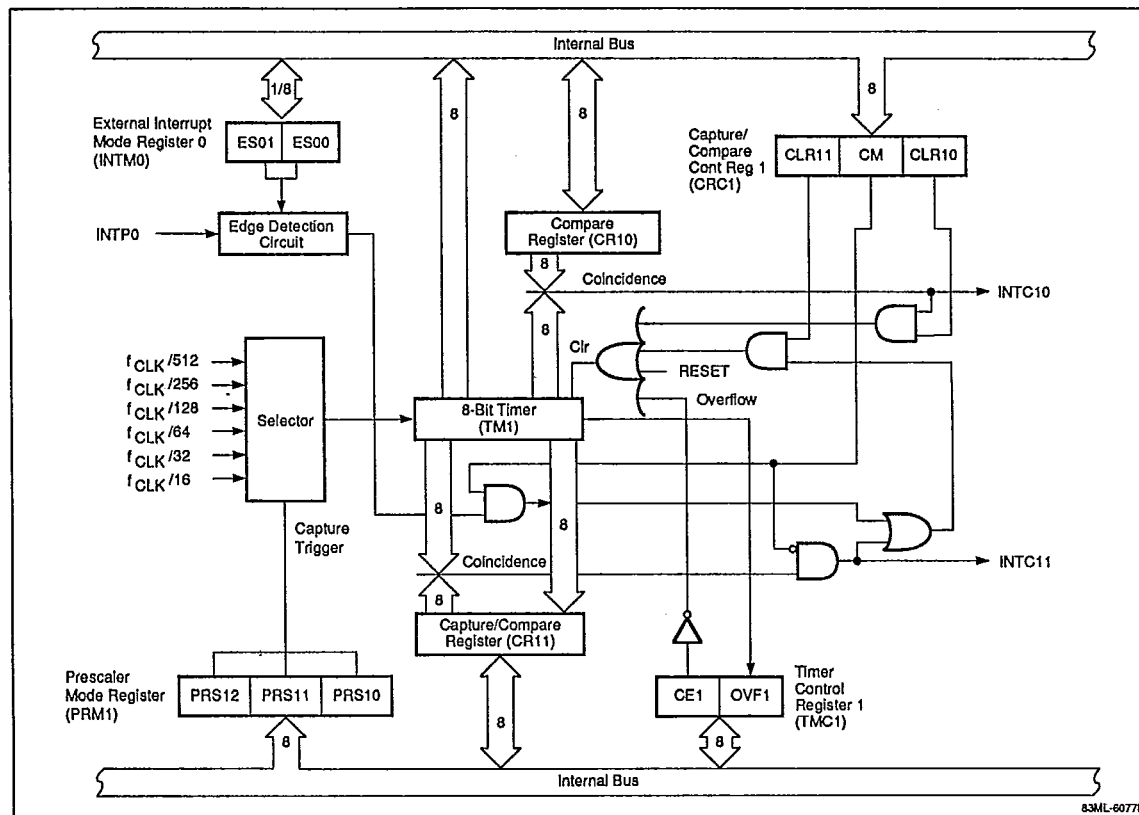
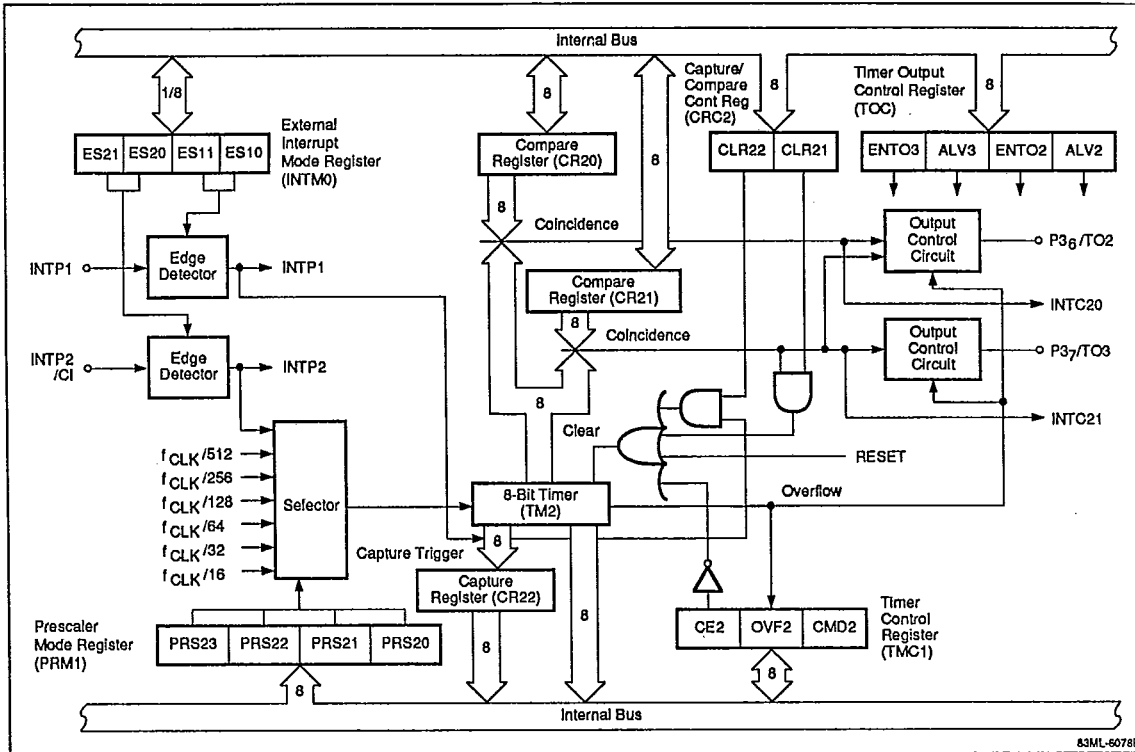




Figure 8. 8-Bit Timer/Counter 1



**Figure 9. 8-Bit Timer/Counter 2**

### Interrupts

There are 18 interrupt request sources; each source is allocated a location in the vector table. (See table 2.) There is one software interrupt request and one of the remaining 17 interrupts is non-maskable. The software interrupt and the non-maskable interrupt are unconditionally received even in the DI state. These two interrupts possess the maximum priority. The maskable interrupt requests are subject to mask control by the setting of the interrupt mask flag.

There are default priorities associated with each maskable interrupt and these can be assigned to either of two programmable priority levels. Interrupts may be serviced by the vectored interrupt method where a branch to a desired service program is executed. Interrupts may also be handled by the macroservice function where a preassigned process is performed without program intervention.





Table 2. Interrupt Sources and Vector Addresses

Interrupt Request Type	Default Priority	Interrupt Request Source	Macroservice Handling	Vector Table Address
Software	None	BRK instruction execution	-	003EH
Non-maskable	None	NMI (pin input edge detection)	-	0002H
Maskable	0	INTP0 (pin input edge detection)	-	0006H
	1	INTP1 (pin input edge detection)	-	0008H
	2	INTP2 (pin input edge detection)	-	000AH
	3	INTP3 (pin input edge detection)	-	000CH
	4	INTC00 (TM0-CR00 coincidence signal generation)	-	0014H
	5	INTC01 (TM0-CR01 coincidence signal generation)	-	0016H
	6	INTC10 (TM1-CR10 coincidence signal generation)	Yes	0018H
	7	INTC11 (TM1-CR11 coincidence signal generation)	Yes	001AH
	8	INTC21 (TM2-CR21 coincidence signal generation)	-	001CH
	9	INTP4 (pin input edge detection)	Yes	000EH
	10	INTP5 (pin input edge detection)	-	0010H
	11	INTP6 (pin input edge detection)	-	0012H
	12	INTSER (generation of asynchronous serial interface receive error)	-	0020H
	13	INTSR (end of asynchronous serial interface reception)	Yes	0022H
	14	INTST (end of asynchronous serial interface transmission)	Yes	0024H
15	INTCSI (end of clocked serial interface transfer)	Yes	0026H	



**Macroservice**

The macroservice function can be programmed to transfer data from a special function register to memory or from memory to a special function register. Transfer events are triggered by interrupt requests and take place without software intervention. There are six interrupt requests where macro servicing can be executed. The macroservice function is controlled by the macroservice mode register and the macroservice channel pointer. The macroservice mode register assigns the macro servicing mode and the macroservice channel pointer indicates the address of the memory location pointers. The location of each register and its corresponding interrupt is shown in figure 10.

**Refresh**

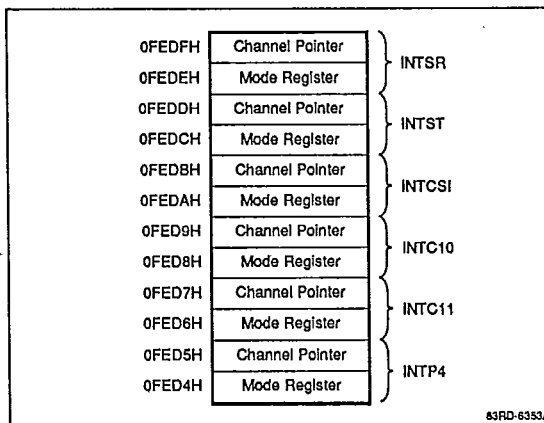
The refresh signal is used with a pseudostatic RAM. The refresh cycle can be set to one of four intervals ranging from 2.6 to 21.3 μs. The refresh is timed to follow a read or write operation so there is no interference.

**Standby Modes**

Halt and stop functions reduce system power consumption. In the halt mode, the CPU stops and the system clock continues to run. A release of the halt mode is initiated by an unmasked interrupt request, an NMI, or a RESET input. In the stop mode, the CPU and system clock are

both stopped, reducing the power consumption even further. The stop mode is released by an NMI input or a RESET input.

Figure 10. Macroservice Control Word Map



$\mu$ PD7822x**NEC**

T-49-19-07

T-49-19-59

**ELECTRICAL SPECIFICATIONS****Absolute Maximum Ratings** $T_A = +25^\circ\text{C}$ .

Item	Symbol	Conditions	Rating	Unit
Operating voltage	$V_{DD}$		-0.5 to +7.0	V
Input voltage	$V_I$		-0.5 to $V_{DD} + 0.5$	V
Output voltage	$V_O$		-0.5 to $V_{DD} + 0.5$	V
Low-level output current	$I_{OL}$	One output pin	30 (peak)	mA
			15 (mean value)	mA
		All output pins total	150 (peak)	mA
			100 (mean value)	mA
High-level output current	$I_{OH}$	One output pin	-2	mA
		All output pins total	-50	mA
Operating temperature	$T_{OPT}$		-40 to +85	$^\circ\text{C}$
Storage temperature	$T_{STG}$		-65 to +150	$^\circ\text{C}$

**Operating Frequency**

Oscillation Frequency	$T_A$	$V_{DD}$
$f_{XX} = 4$ to 12 MHz	-40 to +85 $^\circ\text{C}$	+5 V $\pm 5\%$
	-10 to +70 $^\circ\text{C}$	+5 V $\pm 10\%$

**Capacitance** $T_A = +25^\circ\text{C}$ ;  $V_{DD} = V_{SS} = 0$  V.

Item	Symbol	Typ	Max	Unit	Conditions
Input capacitance	$C_I$	20		pF	$f = 1$ MHz; pins not used for measurement are at 0 V
Output capacitance	$C_O$	20		pF	
Input/output capacitance	$C_{IO}$	20		pF	

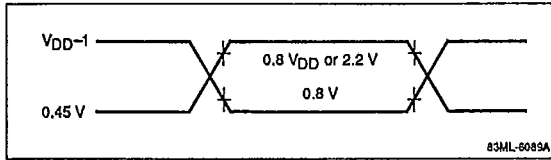
**DC Characteristics** $T_A = -40$  to +85 $^\circ\text{C}$ ;  $V_{DD} = +5$  V  $\pm 10\%$ ;  $V_{SS} = 0$  V.

Item	Symbol	Conditions	Min	Typ	Max	Unit
Low-level input voltage	$V_{IL}$	Except PT pins	0		0.8	V
High-level input voltage	$V_{IH1}$	Except PT pins and pins in Note 1	2.2		$V_{DD}$	V
	$V_{IH2}$	Pins in Note 1	0.8 $V_{DD}$		$V_{DD}$	V
Low-level output voltage	$V_{OL1}$	$I_{OL} = 2.0$ mA			0.45	V
	$V_{OL2}$	$I_{OL} = 8.0$ mA (Port PI pins)			1.0	V
High-level output voltage	$V_{OH1}$	$I_{OH} = -1.0$ mA	$V_{DD} - 1.0$			V
	$V_{OH2}$	$I_{OH} = -100$ $\mu\text{A}$	$V_{DD} - 0.5$			V
Input leakage current	$I_{LI}$	$V_I = 0$ to $V_{DD}$			$\pm 10$	$\mu\text{A}$
Output leakage current	$I_{LO}$	$V_O = 0$ to $V_{DD}$			$\pm 10$	$\mu\text{A}$
Pull-up current	$I_{PT}$	$V_I = 0$ V; PT pins		-150	-400	$\mu\text{A}$
$V_{DD}$ power supply current	$I_{DD1}$	Operating mode, $f_{XX} = 12$ MHz		16	40	mA
	$I_{DD2}$	HALT mode, $f_{XX} = 12$ MHz		7	20	mA
Data retention voltage	$V_{DDDR}$	STOP mode	2.5		5.5	V
Data retention current	$I_{DDDR}$	STOP mode $V_{DDDR} = 2.5$ V		2	20	$\mu\text{A}$
		$V_{DDDR} = 5$ V $\pm 10\%$		5	50	$\mu\text{A}$

**Notes:**

- (1) X1, X2, RESET, P2<sub>0</sub>/NMI, P2<sub>1</sub>/INTP0, P2<sub>2</sub>/INTP1, P2<sub>3</sub>/INTP2/CI, P2<sub>4</sub>/INTP3, P2<sub>5</sub>/INTP4, P2<sub>6</sub>/INTP5, P2<sub>7</sub>/INTP6/SI, P3<sub>2</sub>/SCK, P3<sub>3</sub>/SO/SB0, and EA pins.

Figure 11. Voltage Thresholds for Timing Measurements



**Read/Write Operation**

$T_A = -40$  to  $+85^\circ\text{C}$ ;  $V_{DD} = +5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$ ;  $f_{\text{CX}} = 12\text{ MHz}$ ;  $C_L = 100\text{ pF}$ .

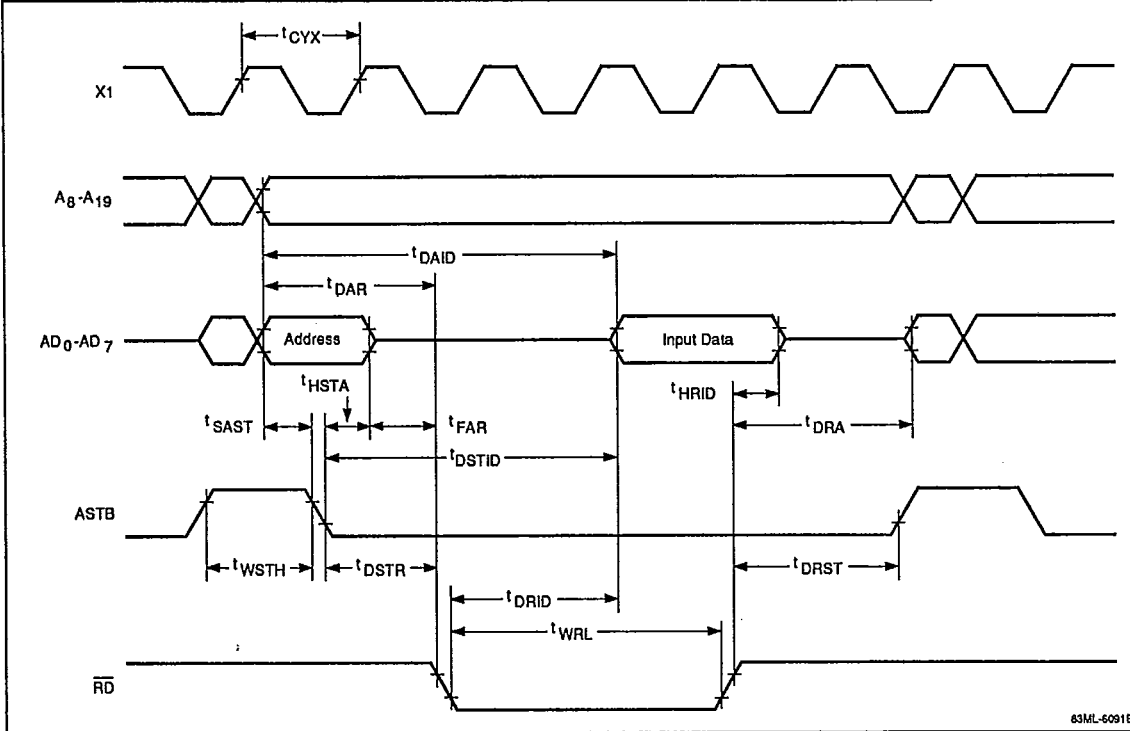
Item	Symbol	Conditions	Min	Max	Unit
X1 input clock cycle time	$t_{\text{CYX}}$		82	250	ns
Address setup time to $\text{ASTB} \downarrow$	$t_{\text{SAST}}$		52		ns
Address hold time from $\text{ASTB} \downarrow$ (Note 2)	$t_{\text{HSTA}}$	$R_L = 5\text{ k}\Omega$ , $C_L = 50\text{ pF}$	25		ns
Address to $\overline{\text{RD}} \downarrow$ delay time	$t_{\text{DAR}}$		129		ns
Address float time from $\overline{\text{RD}} \downarrow$	$t_{\text{FAR}}$		11		ns
Address to data input time	$t_{\text{DAID}}$			228	ns
$\text{ASTB} \downarrow$ to data input time	$t_{\text{DSTID}}$			181	ns
$\overline{\text{RD}} \downarrow$ to data input time	$t_{\text{DRID}}$			99	ns
$\text{ASTB} \downarrow$ to $\overline{\text{RD}} \downarrow$ delay time	$t_{\text{DSTR}}$		52		ns
Data hold time from $\overline{\text{RD}} \uparrow$	$t_{\text{HRID}}$		0		ns
$\overline{\text{RD}} \uparrow$ to address active time	$t_{\text{DRA}}$		124		ns
$\overline{\text{RD}} \uparrow$ to $\text{ASTB} \uparrow$ delay time	$t_{\text{DRST}}$		124		ns
$\overline{\text{RD}}$ low-level width	$t_{\text{WRL}}$		124		ns
$\text{ASTB}$ high-level width	$t_{\text{WSTH}}$		52		ns
Address to $\overline{\text{WR}} \downarrow$ delay time	$t_{\text{DAW}}$		129		ns
$\text{ASTB} \downarrow$ to data output time	$t_{\text{DSTOD}}$			142	ns
$\overline{\text{WR}} \downarrow$ to data output time	$t_{\text{DWOD}}$			60	ns
$\text{ASTB} \downarrow$ to $\overline{\text{WR}} \downarrow$ delay time	$t_{\text{DSTW1}}$		52		ns
	$t_{\text{DSTW2}}$	Refresh mode	129		ns
Data setup time to $\overline{\text{WR}} \uparrow$	$t_{\text{SODWR}}$		146		ns
Data setup time to $\overline{\text{WR}} \uparrow$ (Note 1)	$t_{\text{SODWF}}$	Refresh mode	22		ns
Data hold time from $\overline{\text{WR}} \uparrow$ (Note 2)	$t_{\text{HWOD}}$		20		ns
$\overline{\text{WR}} \uparrow$ to $\text{ASTB} \uparrow$ delay time	$t_{\text{DWST}}$		42		ns
$\overline{\text{WR}}$ low-level width	$t_{\text{WWL1}}$		196		ns
	$t_{\text{WWL2}}$	Refresh mode	114		ns
Address to $\overline{\text{WAIT}} \downarrow$ input time	$t_{\text{DAWT}}$			146	ns
$\text{ASTB} \downarrow$ to $\overline{\text{WAIT}} \downarrow$ input time	$t_{\text{DSTWT}}$			84	ns
$\overline{\text{WAIT}}$ hold time from X1 $\downarrow$	$t_{\text{HWTX}}$		0		ns
$\overline{\text{WAIT}}$ setup time to X1 $\uparrow$	$t_{\text{SWTX}}$		0		ns

**Notes:**

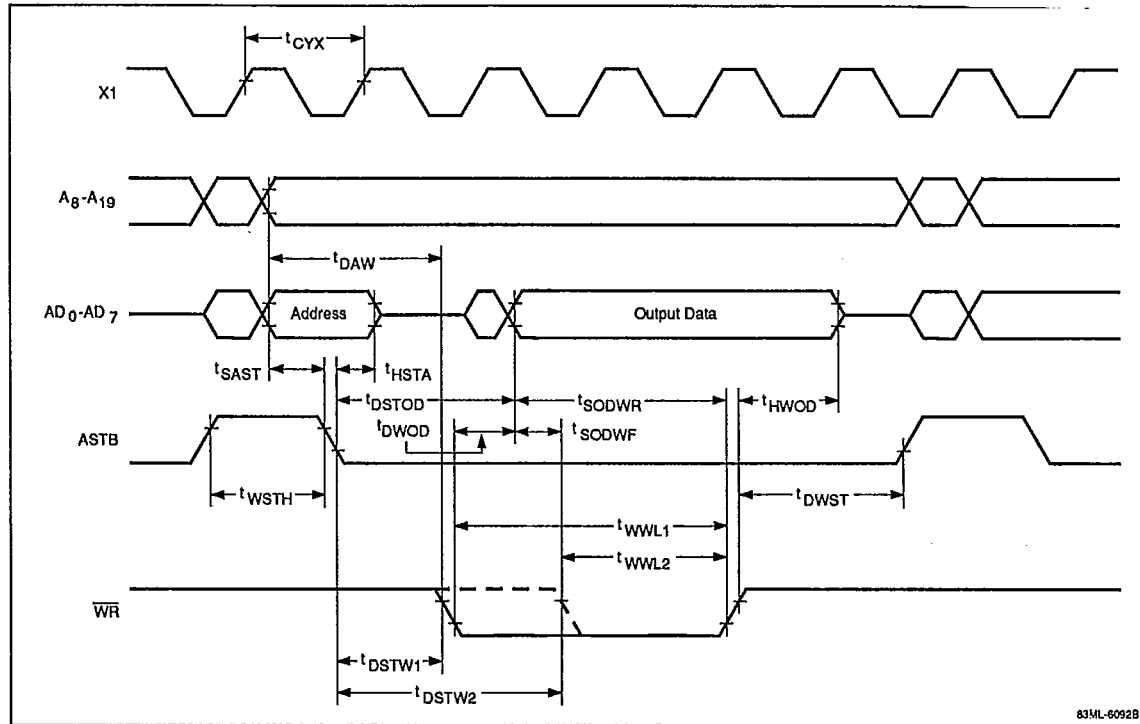
- (1) When accessing a pseudostatic RAM ( $\mu\text{PD4168}$ , etc.) that clocks in data at the falling edge of  $\overline{\text{WR}}$ , use  $t_{\text{SODWF}}$  instead of  $t_{\text{SODWR}}$  as the data setup time.
- (2) The hold time includes the time during which  $V_{\text{OH}}$  and  $V_{\text{OL}}$  are retained under the following load conditions:  $C_L = 100\text{ pF}$  and  $R_L = 2\text{ k}\Omega$ .



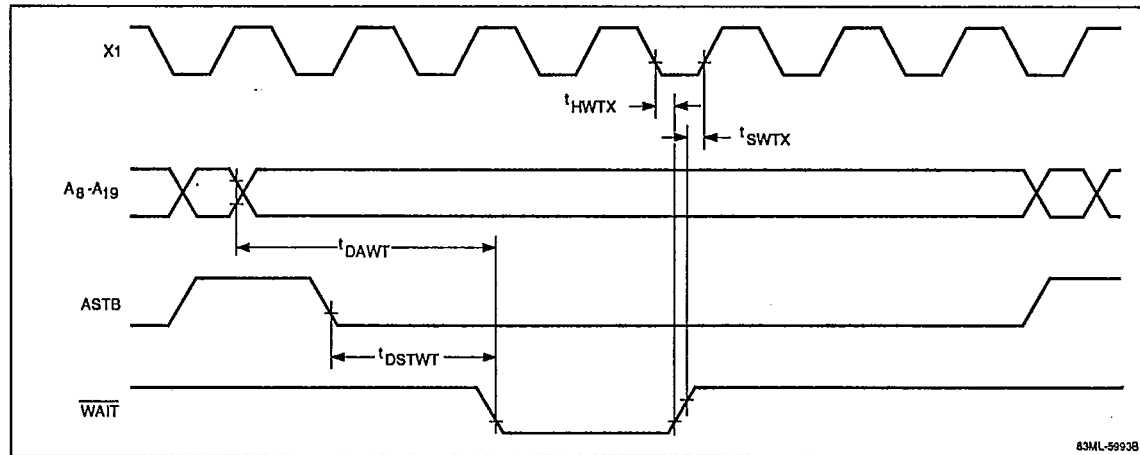
**Figure 12. Read Operation Timing**



**Figure 13. Write Operation Timing**



**Figure 14. External WAIT Input Timing**

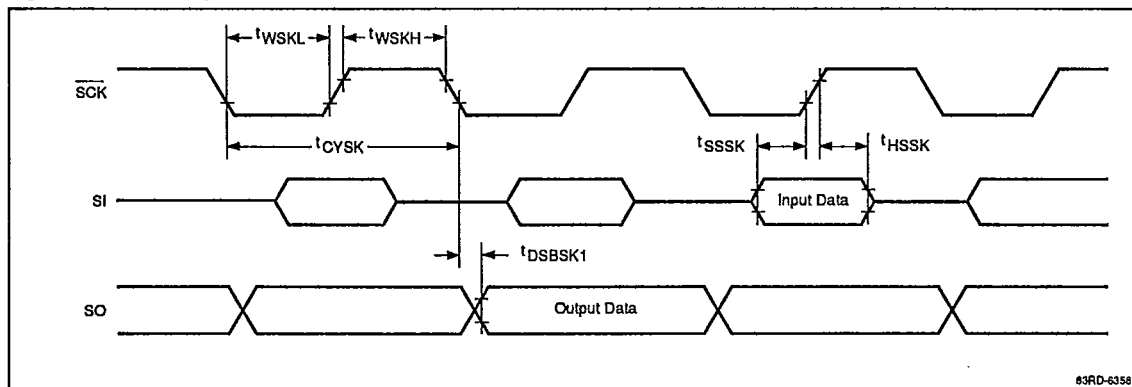


**Serial Port Operation**

T<sub>A</sub> = -40 to +85°C; V<sub>DD</sub> = +5 V ± 10%; V<sub>SS</sub> = 0 V; f<sub>XX</sub> = 12 MHz; C<sub>L</sub> = 100 pF.

Item	Symbol	Conditions	Min	Max	Unit
Serial clock cycle time	t <sub>CYSK</sub>	Input External clock	1.0		μs
		Output Internal clock/16	1.3		μs
		Internal clock/64	5.3		μs
Serial clock low-level width	t <sub>WSKL</sub>	Input External clock	420		ns
		Output Internal clock/16	556		ns
		Internal clock/64	2.5		μs
Serial clock high-level width	t <sub>WSKH</sub>	Input External clock	420		ns
		Output Internal clock/16	556		ns
		Internal clock/64	2.5		μs
SI, SB0 setup time to SCK ↑	t <sub>SSSK</sub>		150		ns
SI, SB0 hold time from SCK ↓	t <sub>HSSK</sub>		400		ns
SO/SB0 output delay time from SCK ↓	t <sub>DSBSK1</sub>	CMOS push-pull output (3-line serial I/O mode)	0	300	ns
	t <sub>DSBSK2</sub>	Open-drain output (SBI mode), R <sub>L</sub> = 1 kΩ	0	800	ns
SB0 high, hold time from SCK ↑	t <sub>HSBSK</sub>	SBI mode	4		t <sub>CYX</sub>
SB0 low, setup time to SCK ↓	t <sub>SSBSK</sub>	SBI mode	4		t <sub>CYX</sub>
SB0 low-level width	t <sub>WSBL</sub>		4		t <sub>CYX</sub>
SB0 high-level width	t <sub>WSBH</sub>		4		t <sub>CYX</sub>
RxD setup time to SCK ↑	t <sub>SRXSK</sub>		80		ns
RxD hold time after SCK ↓	t <sub>HSKRX</sub>		80		ns
SCK ↓ to TxD delay time	t <sub>DSKTX</sub>			210	ns

Figure 15. Clock-Synchronized Serial Interface Timing; Three-Line I/O Mode





μPD7822x

T-49-19-07

Figure 16. Clock-Synchronized Serial Interface Timing; SBI Mode

T-49-19-59

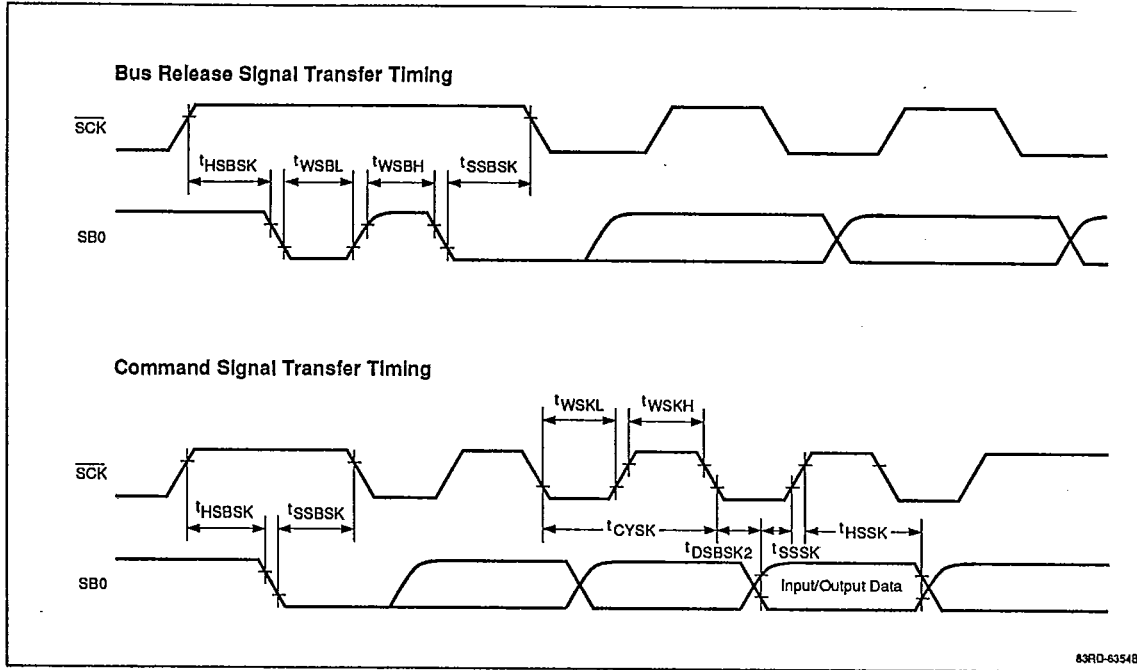
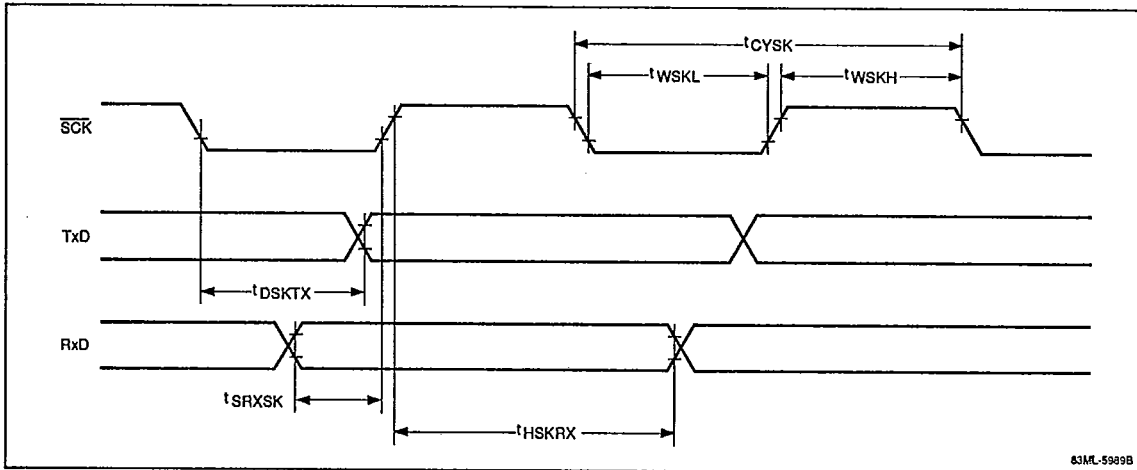


Figure 17. Asynchronous Mode Timing



**NEC**

T-49-19-07

**μPD7822x****Comparator Port Operation**

Item	Symbol	Conditions	Min	Max	Unit
Comparison accuracy	$V_{ACOMP}$	—		100	mV
		μPD78P224		100	mV
Comparison time	$t_{COMP}$		128	256	$t_{CYX}$
Sampling time	$t_{SAMP}$		62		$t_{CYX}$
PT Input voltage	$V_{IPT}$		0	$V_{DD}$	V

**Interrupt Timing Operation**

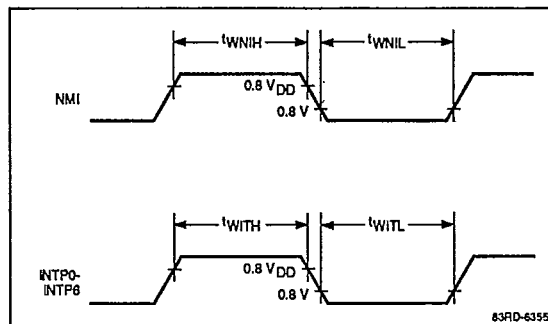
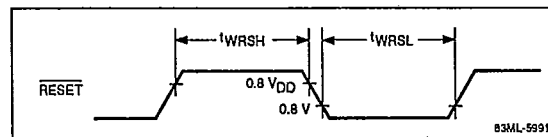
Item	Symbol	Conditions	Min	Max	Unit
NMI low-level width	$t_{WNIL}$		10		μs
NMI high-level width	$t_{WNIH}$		10		μs
INTP0-INTP6 low-level width	$t_{WITL}$		24		$t_{CYX}$
INTP0-INTP6 high-level width	$t_{WITH}$		24		$t_{CYX}$
RESET low-level width	$t_{WRSL}$		10		μs
RESET high-level width	$t_{WRSH}$		10		μs

**Data Retention Characteristics**

Item	Symbol	Conditions	Min	Typ	Max	Unit
Data retention voltage	$V_{DDDR}$	STOP mode	2.5		5.5	V
Data retention current	$I_{DDDR}$	$V_{DDDR} = 2.5\text{ V}$		2	20	μA
		$V_{DDDR} = 5\text{ V} \pm 10\%$		5	50	μA
$V_{DD}$ rise time	$t_{RVD}$		200			μs
$V_{DD}$ fall time	$t_{FVD}$		200			μs
$V_{DD}$ retention time (for STOP mode setting)	$t_{HVD}$		0			ms
STOP release signal input time	$t_{DREL}$		0			ms
Oscillation stabilization wait time	$t_{WAIT}$	Crystal resonator	30			ms
		Ceramic resonator	5			ms
Low-level input voltage	$V_{IL}$	Note 1	0		$0.1 V_{DDDR}$	V
High-level input voltage	$V_{IH}$	Note 1	$0.9 V_{DDDR}$		$V_{DDDR}$	V

**Notes:**

- (1) RESET, P<sub>20</sub>/NMI, P<sub>21</sub>/INTP0, P<sub>22</sub>/INTP1, P<sub>23</sub>/INTP2/CI, P<sub>24</sub>/INTP3, P<sub>25</sub>/INTP4, P<sub>26</sub>/INTP5, P<sub>27</sub>/INTP6/SI, P<sub>32</sub>/SCK, P<sub>33</sub>/SO/SB0, and EA pins.

**Figure 18. Interrupt Input Timing** T-49-19-59**Figure 19. Reset Input Timing**



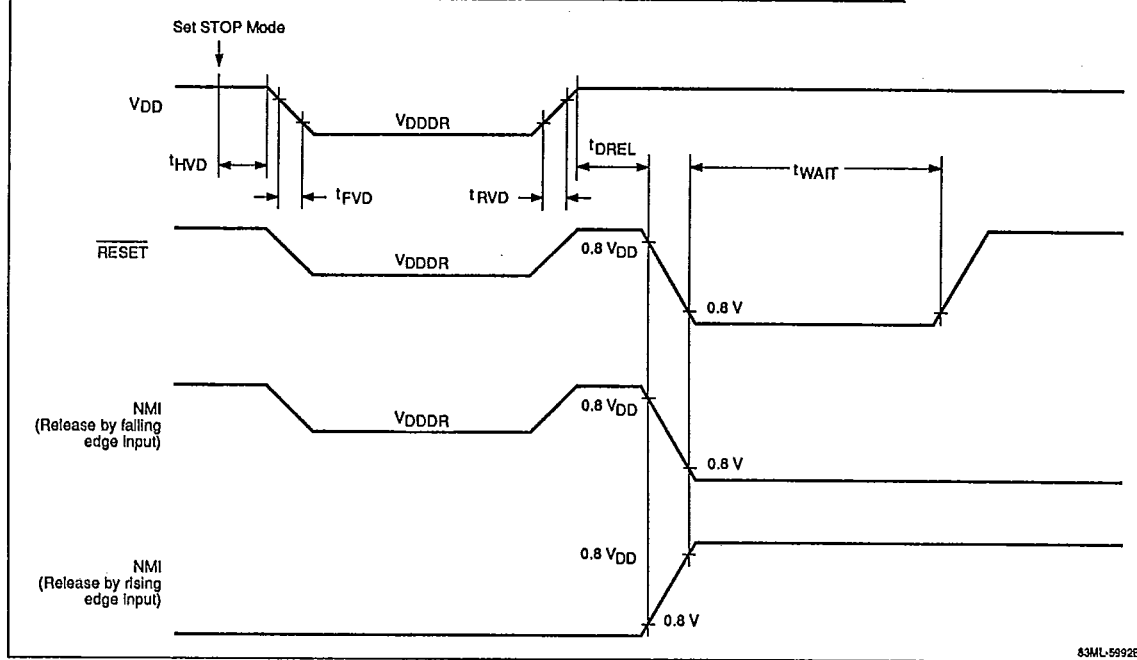


μPD7822x

T-49-19-07

Figure 20. Data Retention Characteristics

T-49-19-59



**NEC****μPD7822x**

T-49-19-07

T-49-19-59

**Timing Dependent on  $t_{CYX}$** 

Item	Symbol	Calculation Formula	Min/Max	12 MHz	Unit
X1 input clock cycle time	$t_{CYX}$		Min	82	ns
Address setup time to $\overline{ASTB} \downarrow$	$t_{SAST}$	$t_{CYX} - 30$	Min	52	ns
Address to $\overline{RD} \downarrow$ delay time	$t_{DAR}$	$2t_{CYX} - 35$	Min	129	ns
Address float time from $\overline{RD} \downarrow$	$t_{FAR}$	$t_{CYX}/2 - 30$	Min	11	ns
Address to data input time	$t_{DAID}$	$(4+2n)t_{CYX} - 100$	Max	228	ns
$\overline{ASTB} \downarrow$ to data input time	$t_{DSTID}$	$(3+2n)t_{CYX} - 65$	Max	181	ns
$\overline{RD} \downarrow$ to data input time	$t_{DRID}$	$(2+2n)t_{CYX} - 65$	Max	99	ns
$\overline{ASTB} \downarrow$ to $\overline{RD} \downarrow$ delay time	$t_{DSTR}$	$t_{CYX} - 30$	Min	52	ns
$\overline{RD} \uparrow$ to address active time	$t_{DRA}$	$2t_{CYX} - 40$	Min	124	ns
$\overline{RD} \uparrow$ to $\overline{ASTB} \uparrow$ delay time	$t_{DRST}$	$2t_{CYX} - 40$	Min	124	ns
$\overline{RD}$ low-level width	$t_{WRL}$	$(2+2n)t_{CYX} - 40$	Min	124	ns
$\overline{ASTB}$ high-level width	$t_{WSTH}$	$t_{CYX} - 30$	Min	52	ns
Address to $\overline{WR} \downarrow$ delay time	$t_{DAW}$	$2t_{CYX} - 35$	Min	129	ns
$\overline{ASTB} \downarrow$ to data output time	$t_{DSTOD}$	$t_{CYX} + 60$	Max	142	ns
$\overline{ASTB} \downarrow$ to $\overline{WR} \downarrow$ delay time	$t_{DSTW1}$	$t_{CYX} - 30$	Min	52	ns
	$t_{DSTW2}$	$2t_{CYX} - 35$ (refresh mode)	Min	129	ns
Data setup time to $\overline{WR} \uparrow$	$t_{SODWR}$	$(3+2n)t_{CYX} - 100$	Min	146	ns
Data setup time to $\overline{WR} \downarrow$	$t_{SODWF}$	$t_{CYX} - 60$ (refresh mode)	Min	22	ns
$\overline{WR} \uparrow$ to $\overline{ASTB} \uparrow$ delay time	$t_{DWST}$	$t_{CYX} - 40$	Min	42	ns
$\overline{WR}$ low-level width	$t_{WWL1}$	$(3+2n)t_{CYX} - 50$	Min	196	ns
	$t_{WWL2}$	$(2+2n)t_{CYX} - 50$ (refresh mode)	Min	114	ns
Address to $\overline{WAIT} \downarrow$ input time	$t_{DAWT}$	$3t_{CYX} - 100$	Max	146	ns
$\overline{ASTB} \downarrow$ to $\overline{WAIT} \downarrow$ input time	$t_{DSTWT}$	$2t_{CYX} - 80$	Max	84	ns

**Notes:**

(1) n indicates the number of wait states.



μPD7822x

T-49-19-07

Figure 21. Recommended Oscillator Circuit

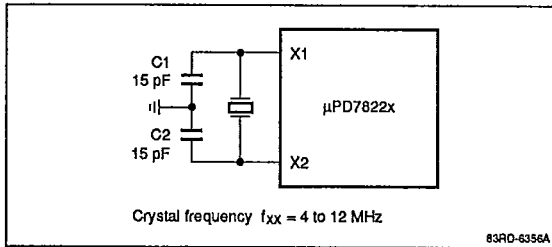
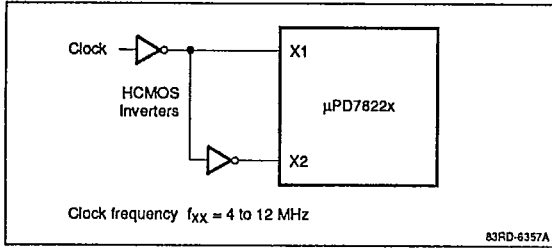


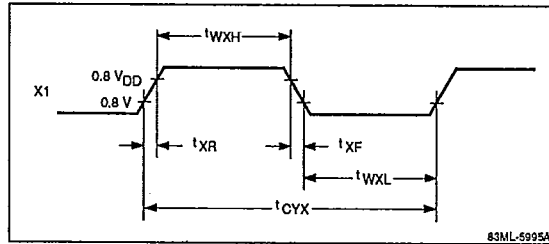
Figure 22. Recommended External Clock Circuit



External Clock Operation T-49-19-59

Item	Symbol	Conditions	Min	Max	Unit
X1 input low-level width	$t_{WXL}$		30	130	ns
X1 input high-level width	$t_{WXH}$		30	130	ns
X1 input rise time	$t_{XR}$		0	30	ns
X1 input fall time	$t_{XF}$		0	30	ns
X1 input clock cycle time	$t_{CYX}$		82	250	ns

Figure 23. External Clock Timing



**μPD7822x**

T-49-19-07 T-49-19-59

**μPD78P224 PROGRAMMING**

In the 78P224, the mask ROM of 78224 is replaced by a one-time programmable ROM (OTP ROM). The ROM is  $16,384 \times 8$  bits and can be programmed using a general-purpose PROM writer with a μPD27C256A programming mode.

The PA-78P224GJ/L are the socket adaptors used for configuring the μPD78P224 to fit a standard PROM socket.

Refer to tables 3 through 6 and figures 24 and 25 for special information applicable to PROM programming.

**Table 3. Pin Functions During PROM Programming**

Pin		Function
P0 <sub>0</sub> -P0 <sub>7</sub>	A <sub>0</sub> -A <sub>7</sub>	Input pins for PROM write/verify operations
P5 <sub>0</sub> /A <sub>8</sub>	A <sub>8</sub>	Input pin for PROM write/verify operation
P2 <sub>1</sub> /INTP <sub>0</sub>	A <sub>9</sub>	Input pin for PROM write/verify operation
P5 <sub>2</sub> -P5 <sub>6</sub> /A <sub>10</sub> -A <sub>14</sub>	A <sub>10</sub> -A <sub>14</sub>	Input pins for PROM write/verify operations
P4 <sub>0</sub> -P4 <sub>7</sub> /AD <sub>0</sub> -AD <sub>7</sub>	D <sub>0</sub> -D <sub>7</sub>	Data pins for PROM write/verify operations
P6 <sub>5</sub> /WR	$\overline{CE}$	Strobe data into the PROM
P6 <sub>4</sub> /RD	$\overline{OE}$	Enable a data read from the PROM
P2 <sub>0</sub> /NMI	NMI	PROM programming mode is entered by applying a high voltage to this pin
RESET	$\overline{RESET}$	PROM programming mode requires applying a low voltage to this pin
EA	V <sub>pp</sub>	High voltage applied to this pin for program write/verify
V <sub>DD</sub>	V <sub>DD</sub>	Positive power supply pin
V <sub>SS</sub>	V <sub>SS</sub>	Ground

**Table 4. Summary of Operation Modes for PROM Programming**

Mode	NMI	$\overline{RESET}$	$\overline{CE}$	$\overline{OE}$	V <sub>pp</sub>	V <sub>DD</sub>	D <sub>0</sub> -D <sub>7</sub>
Program write	+12.5 V	L	L	H	+12.5 V	+6 V	Data input
Program verify	+12.5 V	L	H	L	+12.5 V	+6 V	Data output
Program Inhibit	+12.5 V	L	H	H	+12.5 V	+6 V	High Z
Read out	+12.5 V	L	L	L	+5 V	+5 V	Data output
Output disable	+12.5 V	L	L	H	+5 V	+5 V	High Z
Standby	+12.5 V	L	H	L/H	+5 V	+5 V	High Z

**Notes:**

When +12.5 V is applied to V<sub>pp</sub> and +6 V to V<sub>DD</sub>, both  $\overline{CE}$  and  $\overline{OE}$  cannot be set to low level (L) simultaneously.



μPD7822x

T-49-19-07

T-49-19-59

**Table 5. DC Programming Characteristics**

T<sub>A</sub> = 25 ± 5°C, V<sub>IP</sub> = 12.5 ± 0.5 V applied to NMI pin, V<sub>SS</sub> = 0 V.

Parameter	Symbol	Symbol*	Condition	Min	Typ	Max	Unit
High-level input voltage	V <sub>IH</sub>	V <sub>IH</sub>		2.4		V <sub>DDP</sub> + 0.3	V
Low-level input voltage	V <sub>IL</sub>	V <sub>IL</sub>		-0.3		0.8	V
Input leakage current	I <sub>LIP</sub>	I <sub>LI</sub>	V <sub>I</sub> = 0 to V <sub>DDP</sub>			10	μA
High-level output voltage	V <sub>OH1</sub>	V <sub>OH</sub>	I <sub>OH</sub> = -400 μA	2.4			V
	V <sub>OH2</sub>	V <sub>OH2</sub>	I <sub>OH</sub> = -100 μA	V <sub>DD</sub> - 0.7			V
Low-level output voltage	V <sub>OL</sub>	V <sub>OL</sub>	I <sub>OH</sub> = 2.1 mA			0.45	V
Output leakage current	I <sub>LO</sub>		V <sub>O</sub> = 0 to V <sub>DDP</sub> ; $\overline{OE} = V_{IH}$			10	μA
NMI pin high-voltage input current	I <sub>IP</sub>					±10	μA
V <sub>DDP</sub> power voltage	V <sub>DDP</sub>	V <sub>CC</sub>	Program memory write mode	5.75	6.0	6.25	V
			Program memory read mode	4.5	5.0	5.5	V
V <sub>PP</sub> power voltage	V <sub>PP</sub>	V <sub>PP</sub>	Program memory write mode	12.2	12.5	12.8	V
			Program memory read mode		V <sub>PP</sub> = V <sub>DDP</sub>		V
V <sub>DDP</sub> power current	I <sub>DD</sub>	I <sub>CC</sub>	Program memory write mode		5	30	mA
			Program memory read mode $\overline{OE} = V_{IL}, V_I = V_{IH}$		5	30	mA
V <sub>PP</sub> power current	I <sub>PP</sub>	I <sub>PP</sub>	Program memory write mode $\overline{OE} = V_{IL}, \overline{OE} = V_{IH}$		5	30	mA
			Program memory read mode		1	100	μA

\* Corresponding symbols of the μPD27C256A.



**Table 6. AC Programming Characteristics**

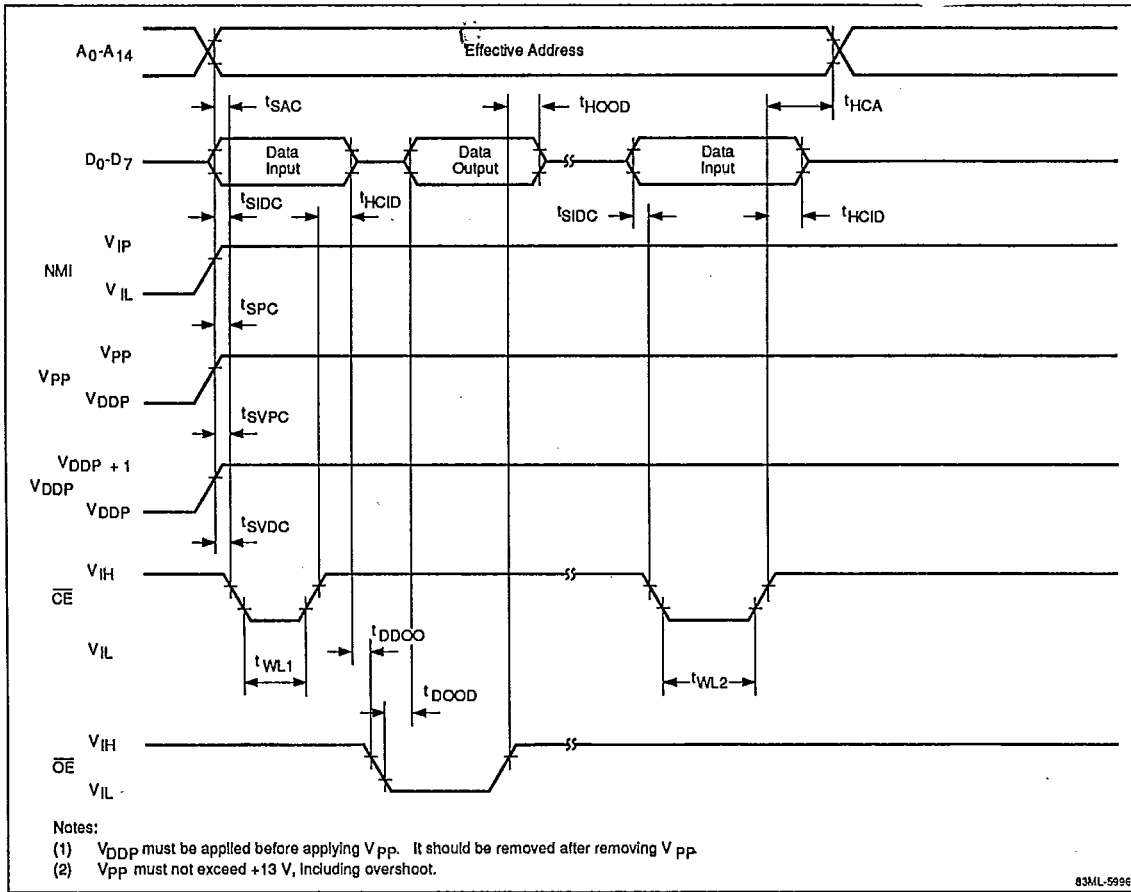
T<sub>A</sub> = 25 ± 5°C, V<sub>IP</sub> = 12.5 ± 0.5 V applied to NMI pin, V<sub>SS</sub> = 0 V, V<sub>DD</sub> = 6 ± 0.25 V, V<sub>PP</sub> = 12.5 ± 0.3 V.

Parameter	Symbol	Symbol*	Condition	Min	Typ	Max	Unit
Address setup time to $\overline{CE} \downarrow$	t <sub>SAC</sub>	t <sub>AS</sub>		2			μs
Data to $\overline{OE} \downarrow$ delay time	t <sub>DDO</sub>	t <sub>OES</sub>		2			μs
Input data setup time to $\overline{CE} \downarrow$	t <sub>SIDC</sub>	t <sub>DS</sub>		2			μs
Address hold time from $\overline{CE} \uparrow$	t <sub>HCA</sub>	t <sub>AH</sub>		2			μs
Input data hold time from $\overline{CE} \uparrow$	t <sub>HCID</sub>	t <sub>DH</sub>		2			μs
Output data hold time from $\overline{OE} \uparrow$	t <sub>HOD</sub>	t <sub>DF</sub>		0		130	ns
V <sub>PP</sub> setup time to $\overline{CE} \downarrow$	t <sub>SVPC</sub>	t <sub>VPS</sub>		1			ms
V <sub>DDP</sub> setup time to $\overline{CE} \downarrow$	t <sub>SVDC</sub>	t <sub>VDS</sub>		1			ms
Initial program pulse width	t <sub>WL1</sub>	t <sub>PW</sub>		0.95	1.0	1.05	ms
Additional program pulse width	t <sub>WL2</sub>	t <sub>OPW</sub>		2.85		78.75	ms
NMI high-voltage input setup time to $\overline{CE} \downarrow$	t <sub>SPC</sub>			2			μs
Address to data output time	t <sub>DAOD</sub>	t <sub>ACC</sub>	$\overline{CE} = \overline{OE} = V_{IL}$			200	ns
$\overline{CE} \downarrow$ to data output time	t <sub>DCOD</sub>	t <sub>CE</sub>	$\overline{OE} = V_{IL}$			200	ns
$\overline{OE} \downarrow$ to data output time	t <sub>DOOD</sub>	t <sub>OE</sub>	$\overline{CE} = V_{IL}$			75	ns
Data hold time from $\overline{OE} \uparrow$	t <sub>HOD</sub>	t <sub>DF</sub>	$\overline{CE} = V_{IL}$	0		60	ns
Data hold time from address	t <sub>HAOD</sub>	t <sub>OH</sub>	$\overline{CE} = \overline{OE} = V_{IL}$	0			ns

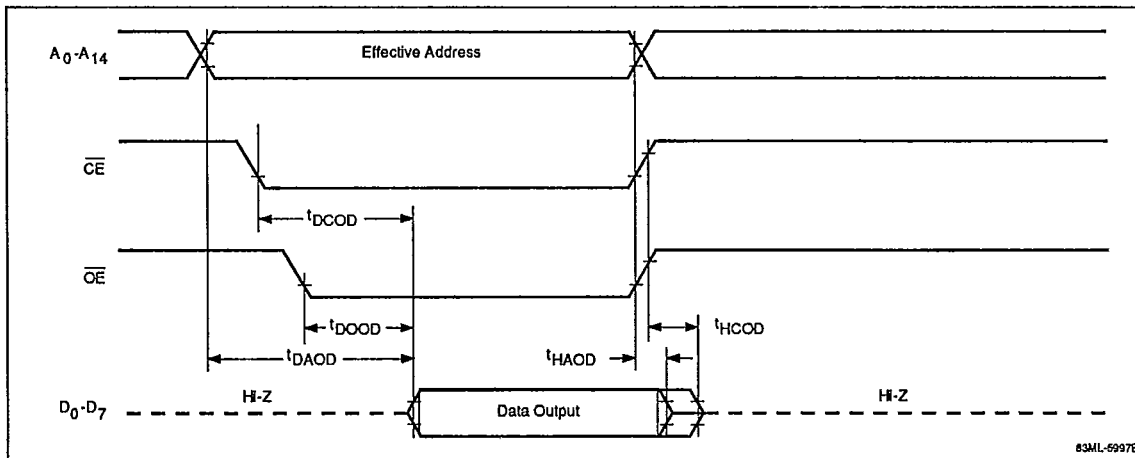
\* Corresponding symbols of the μPD27C256A.

**μPD7822x**

**Figure 24. PROM Write Mode Timing**



**Figure 25. PROM Read Mode Timing**





**PROM Write Procedure**

- (1) Connect the  $\overline{\text{RESET}}$  pin to a low level and apply +12.5 V to the NMI pin.
- (2) Apply +6 V to the  $V_{DD}$  pin and +12.5 V to the  $V_{pp}$  pin.
- (3) Provide the initial address.
- (4) Provide write data.
- (5) Provide 1-ms program pulse (active low) to the  $\overline{\text{CE}}$  pin.
- (6) This bit is now verified with a pulse (active low) to the  $\overline{\text{OE}}$  pin. If the data has been written, proceed to step 8; if not, repeat steps 4 to 6. If the data cannot be correctly written after 25 attempts, go to step 7.
- (7) Classify as defective and stop write operation.
- (8) Provide write data and supply program pulse (for additional writing) for 3 ms times the number of repeats performed between steps 4 to 6.
- (9) Increment the address.
- (10) Repeat steps 4 to 9 until the end address.

**PROM Read Procedure**

- (1) Fix the  $\overline{\text{RESET}}$  pin to a low level and apply +12.5 V to the NMI pin.
- (2) Apply +5 V to the  $V_{DD}$  and  $V_{pp}$  pins.
- (3) Input the address of the data to be read to pins  $A_0$ - $A_{14}$ .
- (4) Read mode is entered with a pulse (active low) on both the  $\overline{\text{CE}}$  and  $\overline{\text{OE}}$  pins.
- (5) Data is output to the  $D_0$  to  $D_7$  pins.

**INSTRUCTION SET**

All microcomputers in the μPD7822x family have a 1-byte instruction lookahead buffer. This allows the first byte of the next opcode in program memory to be fetched while the current opcode is being executed. This pipeline architecture allows instruction fetch and execute cycles to overlap. An instruction can be fetched from program memory while data is being read from or written to RAM or an I/O port.

The advantage of the pipeline is that one instruction can be executed while another is being fetched, virtually halving the time required for these two operations and thereby reducing overall program execution time.

**Operands and Operations**

Refer to tables 7 and 8 for the meanings of symbols in the operand and operations columns of the Instruction Set table.

Specify operands in accordance with the rules of operand representation; for details, refer to the assembler specifications. If two or more description methods are available, select one. The symbols +, -, #, !, \$, /, [ ], and & are keywords and must be used in conjunction with each instruction.

When describing immediate data as a label, use one of the following modifiers: +, -, #, !, \$, /, [ ], and &. Symbols r and rp can be described in both the function name and absolute name.

**Table 7. Operands**

Symbol	Meaning
+	Autoincrement
-	Autodecrement
#	Immediate data
!	Absolute address
\$	Relative address
/	Bit inversion
[ ]	Indirect addressing
&	Subbank; 1M-byte expansion space
r	Register Function name: X, A, C, B, E, D, L, H Absolute name: R0 to R7
r1	Register group 1: C, B
rp	Register pair Function name: AX, BC, DE, HL Absolute name: RP0 to RP3
sfr	Special function register: P0, P2-P7, P0H, P0L, RTPC, CR10, CR11, CR20, CR21, CR22, CR30, PM0, PM3, PM5, PM6, PMC3, PU0, CRC0-CRC2, TOC, TM1-TM3, TMC0, TMC1, PRM0, PRM1, ADM, ADCR, CSIM, SBIC, SIO, ASIM, ASIS, RxB, TxS, BRGC, STBC (dedicated instruction only), MM, PW, RFM, IF0L, IF0H, MK0L, MK0H, PR0L, PR0H, ISM0L, ISM0H, INTM0, INTM1, IST



**NEC**

T-49-19-07

**μPD7822x**

T-49-19-59

**Table 7. Operands (cont)**

Symbol	Meaning
sfrp	Special function register pair: CR00-CR02, TM0, IF0, MK0, PR0, ISM0
mem	Memory address indirectly addressed Register indirect mode: [DE], [HL], [DE+], [HL+], [DE-], [HL-] Base mode: [DE+ byte], [HL+ byte], [SP+ byte] Indexed mode: word[A], word[B], word[DE], word[HL]
mem1	Memory addressed by means of Indirect addressing group 1: [DE], [HL]
saddr	Memory address indirectly addressed: FE20H-FF1FH immediate data or label
saddrp	Memory address addressed by means of direct addressing pair: FE20H-FF1EH immediate data (LSB=0; odd address) or label
addr16	16-bit address: 0000H-FFFFH immediate data or label
addr11	11-bit address: 800H-FFFH immediate data or label
addr5	5-bit address: 40H-7EH immediate data or label
word	16-bit data: 16-bit immediate data or label
byte	8-bit data: 8-bit immediate data or label
bit	3-bit data: 3-bit immediate data or label
n	Number of shift bits: 3-bit immediate data (0-7)
RBn	Register bank: RB0-RB3

**Table 8. Registers and Flags**

Symbol	Meaning
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-R7	Registers 0 to 7 (absolute names)
AX	Register pair (AX); 16-bit accumulator
BC	Register pair (BC)
DE	Register pair (DE)
HL	Register pair (HL)
RP0-RP3	Register pairs 0 to 3 (absolute names)
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary flag
Z	Zero flag
RBS1-RBS0	Register bank select flags
IE	Interrupt enable flag
STBC	Standby control register
( )	Memory contents indicated by address or register contents in ( )
xxH	Hexadecimal number
xH, xL	Higher 8 bits and lower 8 bits of 16-bit register pair





**Clocks**

The clock field specifies the number of clocks required under the conditions defined by the four column headings as follows:

- IROM Program in internal ROM is executed.
- IRAM Program in external ROM is executed and internal RAM is accessed.
- SFR Program in external ROM is executed and special function register is accessed.
- EMEM Program in external ROM is executed and external memory is accessed.

In a shift-rotate instruction, n in the clock field indicates the number of bits by which data is shifted.

The hyphen (-) indicates a range of values; for example 10-13 means 10, 11, 12, or 13.

The virgule symbol (/) means either/or; for example, a/b means either a or b.

The number of clocks when execution is branched by a conditional branch instruction is shown after the symbol (/).

The number of clocks for instruction having the saddr or saddrp operand and when an SFR is accessed with FF00H to FFFFH described as saddr or saddrp is shown after the symbol (/).

**Bytes and Clocks**

The number of bytes and clocks for instructions with a mem or &mem operand depends on the particular instruction and the memory addressing mode (register indirect, base, or indexed). Table 9 is applicable when the program in internal ROM is executed (ROM clock column of the Instruction Set table). Table 10 is applicable when the program in external ROM is executed (IRAM, SFR, and EMEM clock columns).

**Flags**

The symbols in the flag field have the following meanings.

- Blank No change
- 0 Cleared to 0
- 1 Set to 1
- x Set or cleared depending on the result
- R Value previously saved is restored

**Operation Codes**

T-49-19-59

Table 11 defines the symbols used in the operation code field.

**Registers and Register Pairs.** The r, r1, and rp operands are specified in the opcode by one or more bits as shown in figure 26. For example, 001 as bits R<sub>2</sub>R<sub>1</sub>R<sub>0</sub> (or R<sub>6</sub>R<sub>5</sub>R<sub>4</sub>) specifies register A.

In the first and second operands are registers or register pairs; the higher 4 bits of the register specification byte define the first operand and the lower 4 bits define the second operand. For example, in the MOV A,L instruction (transfer L register contents to register A), the second byte of the opcode is obtained from figure 26 as shown below.

Instruction	Opcode, Bytes 1 and 2
MOV r,r	0 0 1 0 0 1 0 0 0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
MOVA,L	0 0 1 0 0 1 0 0 0 0 0 1 0 1 1 0

**Memory Addressing Modes.** The 3-bit mem code and the 5-bit mod code are selected from figure 27 according to the description of mem in the operand field (table 7).



A MOV instruction with register indirect mode specified for mem is a special 1-byte instruction. When base mode or indexed mode is specified for mem, the 8-bit or 16-bit offset data corresponding to byte and word, respectively, is added from the third byte onward.

The opcode for an &mem or &mem1 operand is modified by inserting a 01H code as the first byte preceding the first-byte code listed in the Instruction Set table. Subsequent bytes are as shown in the table.

Figure 26. Opcodes for Registers (r, r1, rp)

r				r1		rp				
R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>	reg		R <sub>0</sub>	reg	P <sub>1</sub>	P <sub>0</sub>	reg-pair	
R <sub>6</sub>	R <sub>5</sub>	R <sub>4</sub>			0	C	P <sub>2</sub>	P <sub>1</sub>		
0	0	0	R0	X			P <sub>6</sub>	P <sub>5</sub>		
0	0	1	R1	A			0	0	RP0	AX
0	1	0	R2	C			0	1	RP1	BC
0	1	1	R3	B			1	0	RP2	DE
1	0	0	R4	E			1	1	RP3	HL
1	0	1	R5	D						
1	1	0	R6	L						
1	1	1	R7	H						

83ML-9992A

**Figure 27. Opcodes for Memory Addressing Modes (mem, mod)**

Mem	Mod	1 0110	0 0110	0 1010
		Register Indirect Mode	Base Mode	Index Mode
0 0 0		[DE+]	[DE+byte]	word [DE]
0 0 1		[HL+]	[SP+byte]	word [A]
0 1 0		[DE-]	[HL+byte]	word [HL]
0 1 1		[HL-]	-	word [B]
1 0 0		[DE]	-	-
1 0 1		[HL]	-	-

83ML-5999A

**Table 9. Bytes and Clocks for Instructions With "mem" and "&mem" Operands; Internal ROM (IROM)**

Instruction	Mem	Register Indirect Mode		Base Mode		Indexed Mode	
		[DE+] [HL+] [DE-] [HL-]	[DE] [HL]	[DE + byte] [HL + byte]	[SP + byte]	word[A] word[B] word[DE] word[HL]	
<b>Bytes</b>	mem	1/2*	1/2*	3	3	4	
	&mem	2/3*	2/3*	4	4	5	
<b>Clock Cycles</b>	MOV	A, mem	6/8	6/8	8-11	9-12	8-11
		mem, A					
		A, &mem	8/10	8/10	10-13	11-14	10-13
	XCH	A, mem	11-15	9-13	10-15	11-16	10-15
		A, &mem	13-17	11-15	12-17	13-18	12-17
	ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP	A, mem	10/12	8/12	9/12	10-13	9-12
A, &mem		12/14	10/14	11/14	12-15	11-14	

\* When internal RAM is accessed with an instruction having a mem operand, the number of bytes is the number before the symbol (/).

When the external memory (including the SFR area) is accessed, the number of bytes is the number after the symbol (/).



μPD7822x

T-49-19-07

T-49-19-59

**Table 10. Bytes and Clocks for Instructions With "mem" and "&mem" Operands; External ROM (IRAM, SFR, EMEM)**

Instruction		Register Indirect Mode		Base Mode		Indexed Mode	
		[DE+] [HL+]	[DE] [HL]	[DE+ byte] [HL+ byte]	[SP+ byte]	word[A] word[B] word[DE] word[HL]	
		[DE-] [HL-]					
<b>Bytes</b>	mem	2*	2*	3	3	4	
	&mem	3*	3*	4	4	5	
<b>Clock Cycles</b>	MOV	A, mem	9/11	6/8	11/13	12/14	14/16
		mem, A					
		A, &mem	12/14	9/11	14/16	15/17	17/19
		&mem, A					
	XCH	A, mem	14/18	12/16	13/17	14/18	16/20
		A, &mem	17/21	15/19	16/20	17/21	19/23
	ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP	A, mem	13/15	11/13	12/14	13/15	15/17
		A, &mem	16/18	14/16	15/17	16/18	18/20

\* When [DE], [HL], [DE+], [HL+], [DE-], or [HL-] is specified as the mem operand of a MOV instruction, the instruction is used as a dedicated 1-byte type. When the operand is &mem, the instruction is 2-byte.



**Table 11. Opcode Symbols**

Symbol	Meaning
Bn	nth bit of immediate data B
Nn	nth bit of immediate data N
Data	8-bit immediate data corresponding to byte
Low/High Byte	16-bit immediate data corresponding to word
Saddr-offset	Lower 8-bit offset data of 16-bit address corresponding to saddr
Sfr-offset	Lower 8-bit offset data of 16-bit address of special function register (sfr)
Low/High Offset	16-bit offset data corresponding to word in Indexed addressing
Low/High Addr	16-bit immediate data corresponding to addr16
jdisp	Signed 2's complement data (8 bits) indicating relative address distance between first address of next instruction and branch destination address
fa	Lower 11 bits of immediate data corresponding to addr11
ta	Lower 5 bits of immediate data corresponding to (addr5xdis)



T-49-19-07

μPD7822x

Instruction Set

T-49-19-59

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>8-Bit Data Transfer</b>											
MOV	r,#byte	r ← byte	2	2	6					1 0 1 1 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
										Data	
	saddr,#byte	(saddr) ← byte	3	3/5	9	9	12			0 0 1 1 1 0 1 0	
										Saddr-offset	
										Data	
	sfr,#byte	sfr ← byte	3	5		9	12			0 0 1 0 1 0 1 1	
										Sfr-offset	
										Data	
	r,r	r ← r	2	2	6					0 0 1 0 0 1 0 0	
										0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	A,r	A ← r	1	2	3					1 1 0 1 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	A,saddr	A ← (saddr)	2	2/4	6	6	9			0 0 1 0 0 0 0 0	
										Saddr-offset	
	saddr,A	(saddr) ← A	2	3/5	6	8				0 0 1 0 0 0 1 0	
										Saddr-offset	
	saddr,saddr	(saddr) ← (saddr)	3	3-7	9					0 0 1 1 1 0 0 0	
										Saddr-offset	
										Saddr-offset	
	A,sfr	A ← sfr	2	4		6				0 0 0 1 0 0 0 0	
										Sfr-offset	
	sfr,A	sfr ← A	2	5		6				0 0 0 1 0 0 1 0	
										Sfr-offset	
	A,mem	A ← (mem)	1-4	6-12	6-14	8-16	8-16			* 0 1 0 1 1 mem	
										0 0 0 mod	
										0 mem 0 0 0 0	
										Low Offset	
										High Offset	
	A,&mem	A ← (&mem)	2-5	8-14	9-17	11-19	11-19			* 0 0 0 0 0 0 0 1	
										0 1 0 1 1 mem	
										0 0 0 0 0 0 0 1	
										0 0 0 mod	
										0 mem 0 0 0 0	
										Low Offset	
										High Offset	

Note:

\* If [DE], [HL], [DE+], [DE-], [HL+] or [HL-] is described as mem, these instructions are used as dedicated 1-byte codes. If the register name is described as &mem, the instructions are used as dedicated 2-byte codes.



μPD7822x

T-49-19-07

Instruction Set (cont)

T-49-19-59

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>8-Bit Data Transfer (cont)</b>											
MOV	mem,A	(mem) ← A	1-4	6-12	6-14	8-16	8-16				* 0 1 0 1 0 mem 0 0 0 mod 1 mem 0 0 0 0 Low Offset High Offset
&mem,A		(&mem) ← A	2-5	8-14	9-17	11-19	11-19				* 0 0 0 0 0 0 0 1 0 1 0 1 0 mem 0 0 0 0 0 0 0 1 0 0 0 mod 1 mem 0 0 0 0 Low Offset High Offset
A,laddr16		A ← (laddr16)	4	6/8	14		16				0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 Low Addr High Addr
A,&laddr16		A ← (&laddr16)	5	8/10			19				0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 Low Addr High Addr
laddr16,A		(laddr16) ← A	4	6/8	14		17				0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 1 Low Addr High Addr
&laddr16,A		(&laddr16) ← A	5	8/10			20				0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 1 Low Addr High Addr
PSW,#byte		PSW ← byte	3	3	9	9	9	x	x	x	0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0 Data
PSW,A		PSW ← A	2	2	6	6	6	x	x	x	0 0 0 1 0 0 1 0 1 1 1 1 1 1 1 0
A,PSW		A ← PSW	2	2	6	6	6				0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0



**NEC**

T-49-19-07

T-49-19-59

**μPD7822x**

**Instruction Set (cont)**

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>8-Bit Data Transfer (cont)</b>											
XCH	A,r	A ↔ r	1	4	4					1 1 0 1 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r,r	r ↔ r	2	3	6					0 0 1 1 0 1 0 1	
A,mem	A ↔ (mem)	A ↔ (mem)	2-4	9-16	12-16		16-20			0 0 0 mod	
										0 mem 0 1 0 0	
										Low Offset	
										High Offset	
A,&mem	A ↔ (&mem)	A ↔ (&mem)	3-5	11-18	15-19		19-23			0 0 0 0 0 0 0 1	
										0 0 0 mod	
										0 mem 0 1 0 0	
										Low Offset	
A,saddr	A ↔ (saddr)	A ↔ (saddr)	2	4:8	6					0 0 1 0 0 0 0 1	
										Saddr-offset	
										High Offset	
										High Offset	
A,sfr	A ↔ sfr	A ↔ sfr	3	6:10		13				0 0 0 0 0 0 0 1	
										0 0 1 0 0 0 0 1	
saddr,saddr	(saddr) ↔ (saddr)	(saddr) ↔ (saddr)	3	6-14		10				0 0 1 1 1 0 0 1	
										Saddr-offset	
<b>16-Bit Data Transfer</b>											
MOVW	rp,#word	rp ← word	3	3	9					0 1 1 0 0 P <sub>2</sub> P <sub>1</sub> 0	
										Low Byte	
										High Byte	
										High Byte	
saddrp,#word	(saddrp) ← word	(saddrp) ← word	4	4:8	12	12	18			0 0 0 0 1 1 0 0	
										Saddr-offset	
										Low Byte	
										Low Byte	
sfrp,#word	sfrp ← word	sfrp ← word	4	8		12				0 0 0 0 1 0 1 1	
										Saddr-offset	
										Low Byte	
										Low Byte	
rp,rp	rp ← rp	rp ← rp	2	4	6					0 0 1 0 0 1 0 0	
										0 P <sub>6</sub> P <sub>5</sub> 0 1 P <sub>2</sub> P <sub>1</sub> 0	
AX,saddrp	AX ← (saddrp)	AX ← (saddrp)	2	6/10	8	12				0 0 0 1 1 1 0 0	
										Saddr-offset	
saddrp,AX	(saddrp) ← AX	(saddrp) ← AX	2	5/9	8	12				0 0 0 1 1 0 1 0	
										Saddr-offset	

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>16-Bit Data Transfer (cont)</b>											
MOVW	AX,sfrp	AX ← sfrp	2	10		12				0 0 0 1 0 0 0 1	
										Sfr-offset	
	sfrp,AX	sfrp ← AX	2	9		12				0 0 0 1 0 0 1 1	
										Sfr-offset	
	AX,mem1	AX ← (mem1)	2	9-15	12	16	16			0 0 0 0 0 1 0 1	
										1 1 1 0 0 0 1 R <sub>0</sub>	
	AX,&mem1	AX ← (&mem1)	3	11-17	15	19	19			0 0 0 0 0 0 0 1	
										0 0 0 0 0 1 0 1	
										1 1 1 0 0 0 1 R <sub>0</sub>	
	mem1,AX	(mem1) ← AX	2	8-14	11	15	15			0 0 0 0 0 1 0 1	
										1 1 1 0 0 1 1 R <sub>0</sub>	
	&mem1,AX	(&mem1) ← AX	3	10-16	14	18	18			0 0 0 0 0 0 0 1	
										0 0 0 0 0 1 0 1	
										1 1 1 0 0 1 1 R <sub>0</sub>	
<b>8-Bit Operation</b>											
ADD	A,#byte	A,CY ← A + byte	2	2	6			x x x		1 0 1 0 1 0 0 0	
										Data	
	saddr,#byte	(saddr),CY ← (saddr) + byte	3	3/7	9	11		x x x		0 1 1 0 1 0 0 0	
										Saddr-offset	
										Data	
	sfr,#byte	sfr,CY ← sfr + byte	4	9		14		x x x		0 0 0 0 0 0 0 1	
										0 1 1 0 1 0 0 0	
										Sfr-offset	
										Data	
	r,r	r,CY ← r + r	2	3	7			x x x		1 0 0 0 1 0 0 0	
										0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	A,saddr	A,CY ← A + (saddr)	2	3/5	6	7	8	x x x		1 0 0 1 1 0 0 0	
										Saddr-offset	
	A,sfr	A,CY ← A + sfr	3	7		10		x x x		0 0 0 0 0 0 0 1	
										1 0 0 1 1 0 0 0	
										Sfr-offset	
	saddr,saddr	(saddr),CY ← (saddr) + (saddr)	3	3-9	9	11		x x x		0 1 1 1 1 0 0 0	
										Saddr-offset	
										Saddr-offset	





μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0)		
				IROM	IRAM	SFR	EMEM	Z	AC	CY	Bytes B1 thru B5		
<b>8-Bit Operation (cont)</b>													
ADD	A,mem	A,CY ← A + (mem)	2-4	8-13	11-15	13-17	13-17	x	x	x	0 0 0	mod	
											0	mem	1 0 0 0
												Low Offset	
												High Offset	
A,&mem	A,CY ← A + (&mem)	3-5	10-15	14-18	16-20	16-20	x	x	x	0 0 0 0	0 0 0 1		
										0 0 0	mod		
												Low Offset	
												High Offset	
ADDC	A,#byte	A,CY ← A + byte + CY	2	2	6				x	x	x	1 0 1 0	1 0 0 1
												Data	
	saddr,#byte	(saddr),CY ← (saddr) + byte + CY	3	3/7	9	11			x	x	x	0 1 1 0	1 0 0 1
												Saddr-offset	
											Data		
sfr,#byte	sfr,CY ← sfr + byte + CY	4	9		14			x	x	x	0 0 0 0	0 0 0 1	
											0 1 1 0	1 0 0 1	
												Sfr-offset	
												Data	
r,r	r,CY ← r + r + CY	2	3	7				x	x	x	1 0 0 1	1 0 0 1	
											0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
A,saddr	A,CY ← A + (saddr) + CY	2	2-5	6	7	8		x	x	x	1 0 0 1	1 0 0 1	
											Saddr-offset		
A,sfr	A,CY ← A + sfr + CY	3	7		10			x	x	x	0 0 0 0	0 0 0 1	
											1 0 0 1	1 0 0 1	
												Sfr-offset	
												Data	
saddr,saddr	(saddr),CY ← (saddr) + (saddr) + CY	3	3-9	9	11			x	x	x	0 1 1 1	1 0 0 1	
											Saddr-offset		
												Saddr-offset	
												Data	
A,mem	A,CY ← A + (mem) + CY	2-4	8-13	11-15	13-17	13-17	x	x	x	0 0 0	mod		
										0	mem	1 0 0 1	
												Low Offset	
												High Offset	
A,&mem	A,CY ← A + (&mem) + CY	3-5	10-15	14-18	16-20	16-20	x	x	x	0 0 0 0	0 0 0 1		
										0 0 0	mod		
												Low Offset	
												High Offset	





μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>8-Bit Operation (cont)</b>											
SUB	A,#byte	A,CY ← A-byte	2	2	6			x	x	x	1 0 1 0 1 0 1 0
											Data
	saddr,#byte	(saddr),CY ← (saddr)-(byte)	3	3/7	9	11		x	x	x	0 1 1 0 1 0 1 0
											Saddr-offset
											Data
	sfr,#byte	sfr,CY ← sfr-byte	4	9	14			x	x	x	0 0 0 0 0 0 0 1 0 1 1 0 1 0 1 0
											Sfr-offset
											Data
	r,r	r,CY ← r-r	2	3	7			x	x	x	1 0 0 0 1 0 1 0 0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
	A,saddr	A,CY ← A-(saddr)	2	3/5	6	7	8	x	x	x	1 0 0 1 1 0 1 0
											Saddr-offset
	A,sfr	A,CY ← A-sfr	3	7	10			x	x	x	0 0 0 0 0 0 0 1 1 0 0 1 1 0 1 0
											Sfr-offset
	saddr,saddr	(saddr),CY ← (saddr)-(saddr)	3	3-9	9	11		x	x	x	0 1 1 1 1 0 1 0
											Saddr-offset
											Saddr-offset
	A,mem	A,CY ← A-(&mem)	2-4	8-13	11-15	13-17	13-17	x	x	x	0 0 0 mod 0 mem 1 0 1 0
											Low Offset
											High Offset
	A,&mem	A,CY ← A-(&mem)	3-5	10-15	14-18	16-20	16-20	x	x	x	0 0 0 0 0 0 0 1 0 0 0 mod 0 mem 1 0 1 0
											Low Offset
											High Offset
SUBC	A,#byte	A,CY ← A-byte-CY	2	2	6			x	x	x	1 0 1 0 1 0 1 1
											Data
	saddr,#byte	(saddr),CY ← (saddr)-byte-CY	3	3/7	9	11		x	x	x	0 1 1 0 1 0 1 1
											Saddr-offset
											Data
	sfr,#byte	sfr,CY ← sfr-byte-CY	4	9	14			x	x	x	0 0 0 0 0 0 0 1 0 1 1 0 1 0 1 1
											Sfr-offset
											Data





μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>8-Bit Operation (cont)</b>											
SUBC	r,r	r,CY ← r-r-CY	2	3	7			x	x	x	1 0 0 0 1 0 1 1 0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
A,saddr	A,saddr	A,CY ← A-(saddr)-CY	2	3/5	6	7	8	x	x	x	1 0 0 1 1 0 1 1 Saddr-offset
A,sfr	A,sfr	A,CY ← A-sfr-CY	3	7		10		x	x	x	0 0 0 0 0 0 0 1 1 0 0 1 1 0 1 1 Sfr-offset
saddr,saddr	saddr,saddr	(saddr),CY ← (saddr)-(saddr)-CY	3	3-9	9	11		x	x	x	0 1 1 1 1 0 1 1 Saddr-offset Saddr-offset
A,mem	A,mem	A,CY ← A-(mem)-CY	2-4	8-13	11-15	13-17	13-17	x	x	x	0 0 0 mod 0 mem 1 0 1 1 Low Offset High Offset
A,&mem	A,&mem	A,CY ← A-(&mem)-CY	3-5	10-15	14-18	16-20	16-20	x	x	x	0 0 0 0 0 0 0 1 0 0 0 mod 0 mem 1 0 1 1 Low Offset High Offset
AND	A,#byte	A ← A∧byte	2	2	6					x	1 0 1 0 1 1 0 0 Data
saddr,#byte	saddr,#byte	(saddr) ← (saddr)∧byte	3	3/7	9	11				x	0 1 1 0 1 1 0 0 Saddr-offset Data
sfr,#byte	sfr,#byte	sfr ← sfr∧byte	4	9		14				x	0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 Sfr-offset Data
r,r	r,r	r ← r∧r	2	3	7					x	1 0 0 0 1 1 0 0 0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
A,saddr	A,saddr	A ← A∧(saddr)	2	3/5	6	7	8	x			1 0 0 1 1 1 0 0 Saddr-offset
A,sfr	A,sfr	A ← A∧(sfr)	3	7		10		x			0 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 Sfr-offset
saddr,saddr	saddr,saddr	(saddr) ← (saddr)∧(saddr)	3	3-9	9	11				x	0 1 1 1 1 1 0 0 Saddr-offset Saddr-offset



μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5					
				IROM	IRAM	SFR	EMEM	Z	AC	CY						
<b>8-Bit Operation (cont)</b>																
AND	A,mem	A ← A ∧ (mem)	2-4	8-13	11-15	13-17	13-17	x	0 0 0 mod							
									0 mem 1 1 0 0							
					Low Offset											
					High Offset											
A,&mem	A ← A ∧ (&mem)	3-5	10-15	14-18	16-20	16-20	x	0 0 0 0 0 0 1								
								0 0 0 mod								
					0 mem 1 1 0 0											
					Low Offset											
				High Offset												
OR	A,#byte	A ← A ∨ byte	2	2	6			x	1 0 1 0 1 1 1 0							
					Data											
	saddr,#byte	(saddr) ← (saddr) ∨ byte	3	3/7	9	11		x	0 1 1 0 1 1 1 0							
					Saddr-offset											
					Data											
	sfr,#byte	sfr ← sfr ∨ byte	4	9	14		x	0 0 0 0 0 0 0 1								
					0 1 1 0 1 1 1 0											
					Sfr-offset											
					Data											
	r,r	r ← r ∨ r	2	3	7		x	1 0 0 0 1 1 1 0								
					0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>											
	A,saddr	A ← A ∨ (saddr)	2	3/5	6	7	8	x	1 0 0 1 1 1 1 0							
					Saddr-offset											
	A,sfr	A ← A ∨ sfr	3	7	10		x	0 0 0 0 0 0 0 1								
				1 0 0 1 1 1 1 0												
				Sfr-offset												
saddr,saddr	(saddr) ← (saddr) ∨ (saddr)	3	3-9	9	11		x	0 1 1 1 1 1 1 0								
				Saddr-offset												
				Saddr-offset												
A,mem	A ← A ∨ (mem)	2-4	8-13	11-15	13-17	13-17	x	0 0 0 mod								
				0 mem 1 1 1 0												
				Low Offset												
				High Offset												
A,&mem	A ← A ∨ (&mem)	3-5	10-15	14-18	16-20	16-20	x	0 0 0 0 0 0 0 1								
				0 0 0 mod												
				0 mem 1 1 1 0												
				Low Offset												
				High Offset												





μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>8-Bit Operation (cont)</b>											
XOR	A,#byte	A ← A⊕byte	2	2	6			x		1 0 1 0 1 1 0 1	
										Data	
	saddr,#byte	(saddr) ← (saddr)⊕byte	3	3/5	9	11		x		0 1 1 0 1 1 0 1	
										Saddr-offset	
										Data	
	sfr,#byte	sfr ← sfr⊕byte	4	7	14			x		0 0 0 0 0 0 0 1	
										0 1 1 0 1 1 0 1	
										Sfr-offset	
										Data	
	r,r	r ← r⊕r	2	3	7			x		1 0 0 0 1 1 0 1	
										0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	A,saddr	A ← A⊕(saddr)	2	3/5	6	7	8	x		1 0 0 1 1 1 0 1	
										Saddr-offset	
	A,sfr	A ← A⊕(sfr)	3	7	10			x		0 0 0 0 0 0 0 1	
										1 0 0 1 1 1 0 1	
										Sfr-offset	
	saddr,saddr	(saddr) ← (saddr)⊕(saddr)	3	3-9	9	11		x		0 1 1 1 1 1 0 1	
										Saddr-offset	
										Saddr-offset	
	A,mem	A ← A⊕(mem)	2-4	8-13	11-15	13-17	13-17	x		0 0 0 mod	
										0 mem 1 1 0 1	
										Low Offset	
										High Offset	
	A,&mem	A ← A⊕(&mem)	3-5	10-15	14-18	16-20	16-20	x		0 0 0 0 0 0 0 1	
										0 0 0 mod	
										0 mem 1 1 0 1	
										Low Offset	
										High Offset	
CMP	A,#byte	A - byte	2	2	6			x x x		1 0 1 0 1 1 1 1	
										Data	
	saddr,#byte	(saddr) - byte	3	3/5	9	11		x x x		0 1 1 0 1 1 1 1	
										Saddr-offset	
										Data	
	sfr,#byte	sfr - byte	4	7	14			x x x		0 0 0 0 0 0 0 1	
										0 1 1 0 1 1 1 1	
										Sfr-offset	
										Data	



μPD7822x

T-49-19-07

Instruction Set (cont)

T-49-19-59

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5	
				IROM	IRAM	SFR	EMEM	Z	AC	CY		
<b>8-Bit Operation (cont)</b>												
CMP	r,r	r-r	2	3	7				x	x	x	1 0 0 0 1 1 1 1
	A,saddr	A-(saddr)	2	3/5	6	7	8		x	x	x	0 R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 1 0 0 1 1 1 1 1
												Saddr-offset
	A,sfr	A-sfr	3	7		10			x	x	x	0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1
												Sfr-offset
	saddr,saddr	(saddr)-(saddr)	3	3-7	9	11			x	x	x	0 1 1 1 1 1 1 1
												Saddr-offset
												Saddr-offset
	A,mem	A-(mem)	2-4	8-13	11-15	13-17	13-17		x	x	x	0 0 0 mod 0 mem 1 1 1 1
												Low Offset
												High Offset
	A,&mem	A-(&mem)	3-5	10-15	14-18	16-20	16-20		x	x	x	0 0 0 0 0 0 0 1 0 0 0 mod 0 mem 1 1 1 1
												Low Offset
												High Offset
<b>16-Bit Operation</b>												
ADDW	AX,#word	AX,CY ← AX + word	3	4	9				x	x	x	0 0 1 0 1 1 0 1
												Low Byte
												High Byte
	AX,rp	AX,CY ← AX + rp	2	6	8				x	x	x	1 0 0 0 1 0 0 0 0 0 0 0 1 P <sub>2</sub> P <sub>1</sub> 0
	AX,saddrp	AX,CY ← AX + (saddrp)	2	7/11	9	13			x	x	x	0 0 0 1 1 1 0 1
												Saddr-offset
	AX,sfrp	AX,CY ← AX + sfrp	3	13		16			x	x	x	0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1
												Sfr-offset
SUBW	AX,#word	AX,CY ← AX - word	3	4	9				x	x	x	0 0 1 0 1 1 1 0
												Low Byte
												High Byte
	AX,rp	AX,CY ← AX - rp	2	6	8				x	x	x	1 0 0 0 1 0 1 0 0 0 0 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>
	AX,saddrp	AX,CY ← AX - (saddrp)	2	7/11	9	13			x	x	x	0 0 0 1 1 1 1 0
												Saddr-offset



**NEC**

T-49-19-07

**μPD7822x**

T-49-19-59

**Instruction Set (cont)**

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>16-Bit Operation (cont)</b>											
SUBW	AX,sfp	AX,CY ← AX-sfp	3	13		16		x	x	x	0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0
Sfr-offset											
CMPW	AX,#word	AX-word	3	3	9			x	x	x	0 0 1 0 1 1 1 1
Low Byte											
High Byte											
	AX,rp	AX-rp	2	5	7			x	x	x	1 0 0 0 1 1 1 1 0 0 0 0 1 P <sub>2</sub> P <sub>1</sub> 0
	AX,saddrp	AX-(saddrp)	2	6/10	8	12		x	x	x	0 0 0 1 1 1 1 1
Saddr-offset											
	AX,sfp	AX-sfp	3	12		15		x	x	x	0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1
Sfr-offset											
<b>Multiplication/Division</b>											
MULU	r	AX ← Axr	2	22	24						0 0 0 0 0 1 0 1 0 0 0 0 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
DIVUW	r	AX(quotient), r (remainder) ← AX ÷ r	2	71	76						0 0 0 0 0 1 0 1 0 0 0 1 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>Increment/Decrement</b>											
INC	r	r ← r + 1	1	2	3			x	x		1 1 0 0 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
	saddr	(saddr) ← (saddr) + 1	2	2/6	6	7		x	x		0 0 1 0 0 1 1 0
Saddr-offset											
DEC	r	r ← r - 1	1	2	3			x	x		1 1 0 0 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
	saddr	(saddr) ← (saddr) - 1	2	2/6	6	7		x	x		0 0 1 0 0 1 1 1
Saddr-offset											
INCW	rp	rp ← rp + 1	1	3	3						0 1 0 0 0 1 P <sub>1</sub> P <sub>0</sub>
DECW	rp	rp ← rp - 1	1	3	3						0 1 0 0 1 1 P <sub>1</sub> P <sub>0</sub>
<b>Shift/Rotate</b>											
ROR	r,n	(CY,r <sub>7</sub> ← r <sub>0</sub> ,r <sub>m-1</sub> ← r <sub>m</sub> ) xn times, n=0-7	2	3+2n	5+2n			x			0 0 1 1 0 0 0 0 0 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
ROL	r,n	(CY,r <sub>0</sub> ← r <sub>7</sub> ,r <sub>m+1</sub> ← r <sub>m</sub> ) xn times, n=0-7	2	3+2n	5+2n			x			0 0 1 1 0 0 0 1 0 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>



μPD7822x

T-49-19-07

Instruction Set (cont)

T-49-19-59

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EEMEM	Z	AC	CY	
<b>Shift/Rotate (cont)</b>											
RORC	r,n	(CY ← r <sub>0</sub> , r <sub>7</sub> ← CY, r <sub>m-1</sub> ← r <sub>m</sub> ) xn times, n=0-7	2	3+2n	5+2n				x	0 0 1 1 0 0 0 0	
										0 0 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
ROLC	r,n	(CY ← r <sub>7</sub> , r <sub>0</sub> ← CY, r <sub>m+1</sub> ← r <sub>m</sub> ) xn times, n=0-7	2	3+2n	5+2n				x	0 0 1 1 0 0 0 1	
										0 0 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
SHR	r,n	(CY ← r <sub>0</sub> , r <sub>7</sub> ← 0, r <sub>m-1</sub> ← r <sub>m</sub> ) xn times, n=0-7	2	3+2n	5+2n			x 0 x		0 0 1 1 0 0 0 0	
										1 0 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
SHL	r,n	(CY ← r <sub>7</sub> , r <sub>0</sub> ← 0, r <sub>m+1</sub> ← r <sub>m</sub> ) xn times, n=0-7	2	3+2n	5+2n			x 0 x		0 0 1 1 0 0 0 1	
										1 0 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
SHRW	rp,n	(CY ← rp <sub>0</sub> , rp <sub>15</sub> ← 0, rp <sub>m-1</sub> ← rp <sub>m</sub> ) xn times, n=0-7	2	3+3n	5+3n			x 0 x		0 0 1 1 0 0 0 0	
										1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
SHLW	rp,n	(CY ← rp <sub>15</sub> , rp <sub>0</sub> ← 0, rp <sub>m+1</sub> ← rp <sub>m</sub> ) xn times, n=0-7	2	3+3n	5+3n			x 0 x		0 0 1 1 0 0 0 1	
										1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
ROR4	mem1	A <sub>3-0</sub> ← (mem1) <sub>3-0</sub> , (mem1) <sub>7-4</sub> ← A <sub>3-0</sub> , (mem1) <sub>3-0</sub> ← (mem1) <sub>7-4</sub>	2	24	26	34	34			0 0 0 0 0 1 0 1	
										1 0 0 0 1 1 R <sub>1</sub> 0	
	&mem1	A <sub>3-0</sub> ← (&mem1) <sub>3-0</sub> , (&mem1) <sub>7-4</sub> ← A <sub>3-0</sub> , (&mem1) <sub>3-0</sub> ← (&mem1) <sub>7-4</sub>	3	26	29	37	37			0 0 0 0 0 0 0 1	
										0 0 0 0 0 1 0 1	
										1 0 0 0 1 1 R <sub>1</sub> 0	
ROL4	mem1	A <sub>3-0</sub> ← (mem1) <sub>7-4</sub> , (mem1) <sub>3-0</sub> ← A <sub>3-0</sub> , (mem1) <sub>7-4</sub> ← (mem1) <sub>3-0</sub>	2	25	27	35	35			0 0 0 0 0 1 0 1	
										1 0 0 1 1 1 R <sub>1</sub> 0	
	&mem1	A <sub>3-0</sub> ← (&mem1) <sub>7-4</sub> , (&mem1) <sub>3-0</sub> ← A <sub>3-0</sub> , (&mem1) <sub>7-4</sub> ← (&mem1) <sub>3-0</sub>	3	27	30	38	38			0 0 0 0 0 0 0 1	
										0 0 0 0 0 1 0 1	
										1 0 0 1 1 1 R <sub>1</sub> 0	





T-49-19-07

T-49-19-59

μPD7822x

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0)							
				IROM	IRAM	SFR	EMEM	Z	AC	CY	Byte B1	thru	Byte B5					
<b>BCD Adjustment</b>																		
ADJBA		Decimal adjust accumulator after addition	1	3		3		x	x	x	0	0	0	0	1	1	1	0
ADJBS		Decimal adjust accumulator after addition	-1	3		3		x	x	x	0	0	0	0	1	1	1	1
<b>Bit Manipulation</b>																		
MOV1	CY,saddr.bit	CY ← (saddr bit)	3	5/7	9	9	11	x			0	0	0	0	1	0	0	0
											0	0	0	0	0	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Saddr-offset							
	CY,sfr.bit	CY ← sfr.bit	3	7		9		x			0	0	0	0	1	0	0	0
											0	0	0	0	1	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
	CY,A.bit	CY ← A.bit	2	5	7			x			0	0	0	0	0	1	1	
											0	0	0	0	1	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
	CY,X.bit	CY ← X.bit	2	5	7			x			0	0	0	0	0	1	1	
											0	0	0	0	0	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
	CY,PSW.bit	CY ← PSW.bit	2	5		7		x			0	0	0	0	0	1	0	
											0	0	0	0	0	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
	saddr.bit,CY	(saddr bit) ← CY	3	8-12	12	14	14				0	0	0	0	1	0	0	0
											0	0	0	1	0	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Saddr-offset							
	sfr.bit,CY	sfr.bit ← CY	3	12		14					0	0	0	0	1	0	0	0
											0	0	0	1	1	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
	A.bit,CY	A.bit ← CY	2	8	10						0	0	0	0	0	0	1	1
											0	0	0	1	1	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
	X.bit,CY	X.bit ← CY	2	8	10						0	0	0	0	0	0	1	1
											0	0	0	1	0	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
	PSW.bit,CY	PSW.bit ← CY	2	7		9		x	x		0	0	0	0	0	0	1	0
											0	0	0	1	0	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Sfr-offset							
AND1	CY,saddr.bit	CY ← CY ∧ (saddr bit)	3	5/7	9	11		x			0	0	0	0	1	0	0	0
											0	0	1	0	0	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
											Saddr-offset							





μPD7822x

T-49-19-07

Instruction Set (cont)

T-49-19-59

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>Bit Manipulation (cont)</b>											
AND1	CY, <i>saddr</i> .bit	CY ← CY ∧ ( <i>saddr</i> .bit)	3	5/7	9	11			x	0 0 0 0 1 0 0 0 0 0 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	
	CY, <i>sfr</i> .bit	CY ← CY ∧ <i>sfr</i> .bit	3	7		11			x	0 0 0 0 1 0 0 0 0 0 1 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	
	CY, <i>sfr</i> .bit	CY ← CY ∧ <i>sfr</i> .bit	3	7		11			x	0 0 0 0 1 0 0 0 0 0 1 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	
	CY, <i>A</i> .bit	CY ← CY ∧ <i>A</i> .bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 0 1 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, <i>A</i> .bit	CY ← CY ∧ <i>A</i> .bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 0 1 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, <i>X</i> .bit	CY ← CY ∧ <i>X</i> .bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 0 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, <i>X</i> .bit	CY ← CY ∧ <i>X</i> .bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 0 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, <i>PSW</i> .bit	CY ← CY ∧ <i>PSW</i> .bit	2	5		7			x	0 0 0 0 0 0 0 1 0 0 0 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, <i>PSW</i> .bit	CY ← CY ∧ <i>PSW</i> .bit	2	5		7			x	0 0 0 0 0 0 0 1 0 0 0 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
OR1	CY, <i>saddr</i> .bit	CY ← CY ∨ ( <i>saddr</i> .bit)	3	5/7	9	11			x	0 0 0 0 1 0 0 0 0 1 0 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Saddr-offset	
	CY, <i>saddr</i> .bit	CY ← CY ∨ ( <i>saddr</i> .bit)	3	5/7	9	11			x	0 0 0 0 1 0 0 0 0 1 0 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	
	CY, <i>sfr</i> .bit	CY ← CY ∨ <i>sfr</i> .bit	3	7		11			x	0 0 0 0 1 0 0 0 0 1 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	





T-49-19-07

T-49-19-59

μPD7822x

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>Bit Manipulation (cont)</b>											
OR1	CY,sfr.bit	CY ← CY V sfr.bit	3	7		11			x	0 0 0 0 1 0 0 0 0 1 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY,A.bit	CY ← CY V A.bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 1 0 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY,A.bit	CY ← CY V A̅.bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 1 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY,X.bit	CY ← CY V X.bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 1 0 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY,X.bit	CY ← CY V X̅.bit	2	5	7				x	0 0 0 0 0 0 0 1 1 0 1 0 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY,PSW.bit	CY ← CY V PSW.bit	2	5		7				x	0 0 0 0 0 0 0 1 0 0 1 0 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
	CY,PSW.bit	CY ← CY V PSW̅.bit	2	5		7				x	0 0 0 0 0 0 0 1 0 0 1 0 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
	XOR1	CY,saddr.bit	CY ← CY ⊕ (saddr.bit)	3	5:7	9	11			x	0 0 0 0 1 0 0 0 0 1 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Saddr-offset
CY,sfr.bit		CY ← CY ⊕ sfr.bit	3	7		11			x	0 0 0 0 1 0 0 0 0 1 1 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	
CY,A.bit		CY ← CY ⊕ A.bit	2	5	7				x	0 0 0 0 0 0 1 1 0 1 1 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
CY,X.bit		CY ← CY ⊕ X.bit	2	5	7				x	0 0 0 0 0 0 1 1 0 1 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
CY,PSW.bit		CY ← CY ⊕ PSW.bit	2	5		7			x	0 0 0 0 0 0 1 0 0 1 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
SET1		saddr.bit	(saddr.bit) ← 1	2	3/7	6					1 0 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Saddr-offset
	sfr.bit	sfr.bit ← 1	3	10		14				0 0 0 0 1 0 0 0 1 0 0 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	



μPD7822x

T-49-19-07

Instruction Set (cont)

T-49-19-59

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>Bit Manipulation (cont)</b>											
SET1	A.bit	A.bit ← 1	2	6	8					0 0 0 0 0 0 1 1 1 0 0 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	X.bit	X.bit ← 1	2	6	8					0 0 0 0 0 0 1 1 1 0 0 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSW.bit	PSW.bit ← 1	2	5		7		x x x		0 0 0 0 0 0 1 0 1 0 0 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
CLR1	saddr.bit	(saddr.bit) ← 0	2	6/10	6					1 0 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Saddr-offset	
	sfr.bit	sfr.bit ← 0	3	10		14				0 0 0 0 1 0 0 0 1 0 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	
	A.bit	A.bit ← 0	2	6	8					0 0 0 0 0 0 1 1 1 0 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	X.bit	X.bit ← 0	2	6	8					0 0 0 0 0 0 1 1 1 0 0 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSW.bit	PSW.bit ← 0	2	5		7		x x x		0 0 0 0 0 0 1 0 1 0 0 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	NOT1	saddr.bit	(saddr.bit) ← 0(saddr.bit)	3	6/10	10	14				0 0 0 0 1 0 0 0 0 1 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Saddr-offset
	sfr.bit	sfr.bit ← sfr.bit	3	10		14				0 0 0 0 1 0 0 0 0 1 1 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset	
	A.bit	A.bit ← A.bit	2	6	8					0 0 0 0 0 0 1 1 0 1 1 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	X.bit	X.bit ← X.bit	2	6	8					0 0 0 0 0 0 1 1 0 1 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSW.bit	PSW.bit ← PSW.bit	2	5		7		x x x		0 0 0 0 0 0 1 0 0 1 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
SET1	CY	CY ← 1	1	2		3				1 0 1 0 0 0 0 0 1	
CLR1	CY	CY ← 0	1	2		3				0 0 1 0 0 0 0 0 0	
NOT1	CY	CY ← $\overline{CY}$	1	2		3				x 0 1 0 0 0 0 0 1 0	





μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IROM	IRAM	SFR	EMEM	Z	AC	CY	
<b>Call/Return</b>											
CALL	laddr16	(SP-1) ← (PC+3) <sub>H</sub> , (SP-2) ← (PC+3) <sub>L</sub> , PC ← laddr16, SP ← SP-2	3	10-15	17		21				0 0 1 0 1 0 0 0
											Low Addr
											High Addr
	rp	(SP-1) ← (PC+2) <sub>H</sub> , (SP-2) ← (PC+2) <sub>L</sub> , PC <sub>H</sub> ← r <sub>pH</sub> , PC <sub>L</sub> ← r <sub>pL</sub> , SP ← SP-2	2	12-17	15		19				0 0 0 0 0 1 0 1 0 1 0 1 1 P <sub>2</sub> P <sub>1</sub> 0
CALLF	laddr11	(SP-1) ← (PC+2) <sub>H</sub> , (SP-2) ← (PC+2) <sub>L</sub> , PC <sub>15+11</sub> ← 00001, PC <sub>10-0</sub> ← laddr11, SP ← SP-2	2	10-15	14		18				1 0 0 1 0 ← fa →
CALLT	{addr5}	(SP-1) ← (PC+1) <sub>H</sub> , (SP-2) ← (PC+1) <sub>L</sub> , PC <sub>H</sub> ← (00000000, addr5+1), PC <sub>L</sub> ← (00000000, addr5), SP ← SP-2	1	14-20	20		24				1 1 1 ← ta →
BRK		(SP-1) ← PSW, (SP-2) ← (PC+1) <sub>H</sub> , (SP-3) ← (PC+1) <sub>L</sub> , PC <sub>H</sub> ← (003FH), PC <sub>L</sub> ← (003FH), SP ← SP-3, IE ← 0	1	16-26	22		28				0 1 0 1 1 1 1 0
RET		PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), SP ← SP+2	1	10-15	11		15				0 1 0 1 0 1 1 0
RETI		PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), PSW ← (SP+2), SP ← SP+3, NMIS ← 0	1	12-20	15		21	R R R			0 1 0 1 0 1 1 1
RETB		PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), PSW ← (SP+2), SP ← SP+3	1	12-20	13		19	R R R			0 1 0 1 1 1 1 1
<b>Stack Manipulation</b>											
PUSH	PSW	(SP-1) ← PSW, SP ← SP-1	1	4-8	5		7				0 1 0 0 1 0 0 1
	sfr	(SP-1) ← sfr, SP ← SP-1	2	7-9	9		12				0 0 1 0 1 0 0 1
											Sfr-offset
	rp	(SP-1) ← r <sub>pH</sub> (SP-2) ← r <sub>pL</sub> , SP ← SP-2	1	8-13	8		12				0 0 1 1 1 1 P <sub>1</sub> P <sub>0</sub>



μPD7822x

T-49-19-07

Instruction Set (cont)

T-49-19-59

Mnemonic	Operand	Operation	Bytes	Clocks				Flags			Operation Code (Bits 7-0)	
				IROM	IRAM	SFR	EMEM	Z	AC	CY	Bytes B1 thru B5	
<b>Stack Manipulation (cont)</b>												
POP	PSW	PSW ← (SP), SP ← SP + 1	1	4-8	6		8	R	R	R	0 1 0 0 1 0 0 0	
	sfr	sfr ← (SP), SP ← SP + 1	2	9-11	9		12				0 1 0 0 0 0 1 1	
	rp	rp <sub>L</sub> ← (SP), rp <sub>H</sub> ← (SP + 1), SP ← SP + 2	1	10-15	11		15				0 0 1 1 0 1 P <sub>1</sub> P <sub>0</sub>	
MOVW	SP, #word	SP ← word	4	8			12				0 0 0 0 1 0 1 1	
											1 1 1 1 1 1 0 0	
	Sfr-offset											
	Low Byte											
	High Byte											
	SP, AX	SP ← AX	2	9			11				0 0 0 1 0 0 1 1	
	AX, SP	AX ← SP	2	10			12				1 1 1 1 1 1 0 0	
INCW	SP	SP ← SP + 1	2	5			7				0 0 0 0 0 1 0 1	
											1 1 0 0 1 0 0 0	
DECW	SP	SP ← SP - 1	2	5			7				0 0 0 0 0 1 0 1	
											1 1 0 0 1 0 0 1	





μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks		Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				IntROM	Branch	No Branch	Z	AC	
<b>Unconditional Branch</b>									
BR	!addr16	PC ← !addr16	3	5	11				0 0 1 0 1 1 0 0
									Low Addr
	rp	PC <sub>H</sub> ← rp <sub>H</sub> , PC <sub>L</sub> ← rp <sub>L</sub>	2	6	10				0 0 0 0 0 1 0 1 0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> 0
	\$addr16	PC ← \$addr16	2	4	9				0 0 0 1 0 1 0 0
									jdisp
<b>Conditional Branch</b>									
BC	\$addr16	PC ← \$addr16 if CY = 1	2	2/4	9	6			1 0 0 0 0 0 1 1
									jdisp
BNC	\$addr16	PC ← \$addr16 if CY = 0	2	2/4	9	6			1 0 0 0 0 0 1 0
									jdisp
BZ	\$addr16	PC ← \$addr16 if Z = 1	2	2/4	9	6			1 0 0 0 0 0 0 1
									jdisp
BNZ	\$addr16	PC ← \$addr16 if Z = 0	2	2/4	9	6			1 0 0 0 0 0 0 0
									jdisp
BNE									jdisp
BT	saddr.bit, \$addr16	PC ← \$addr16 if (saddr.bit) = 1	3	5-9	12	9			0 1 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
									Saddr-offset
									jdisp
	sfr.bit, \$addr16	PC ← \$addr16 if sfr.bit = 1	4	7-9	16	13			0 0 0 0 1 0 0 0 1 0 1 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
									Sfr-offset
									jdisp
	A.bit, \$addr16	PC ← \$addr16 if A.bit = 1	3	5/7	12	9			0 0 0 0 0 0 1 1 1 0 1 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
									jdisp
	X.bit, \$addr16	PC ← \$addr16 if X.bit = 1	3	5/7	12	9			0 0 0 0 0 0 1 1 1 0 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
									jdisp
	PSW.bit, \$addr16	PC ← \$addr16 if PSW.bit = 1	3	5/7	12	9			0 0 0 0 0 0 1 0 1 0 1 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
									jdisp



μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks			Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				Int ROM	Branch	No Branch	Z	AC	CY	
<b>Conditional Branch (cont)</b>										
BF	saddr.bit,\$addr16	PC ← \$addr16 if (saddr.bit) = 0	4	5-9	15	12				0 0 0 0 1 0 0 0 1 0 1 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Saddr-offset jdisp
	sfr.bit,\$addr16	PC ← \$addr16 if sfr.bit = 0	4	7/9	16	13				0 0 0 0 1 0 0 0 1 0 1 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset jdisp
	A.bit,\$addr16	PC ← \$addr16 if A.bit = 0	3	5/7	12	9				0 0 0 0 0 0 1 1 1 0 1 0 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> jdisp
	X.bit,\$addr16	PC ← \$addr16 if X.bit = 0	3	5/7	12	9				0 0 0 0 0 0 1 1 1 0 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> jdisp
	PSW.bit,\$addr16	PC ← \$addr16 if PSW.bit = 0	3	5/7	12	9				0 0 0 0 0 0 1 0 1 0 1 0 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> jdisp
BTCLR	saddr.bit,\$addr16	PC ← \$addr16 if (saddr.bit) = 1 then reset (saddr.bit)	4	5-13	15	12				0 0 0 0 1 0 0 0 1 1 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Saddr-offset jdisp
	sfr.bit,\$addr16	PC ← \$addr16 if sfr.bit = 1 then reset sfr.bit	4	7/13	18	13				0 0 0 0 1 0 0 0 1 1 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> Sfr-offset jdisp
	A.bit,\$addr16	PC ← \$addr16 if A.bit = 1 then reset A.bit	3	5/9	12	9				0 0 0 0 0 0 1 1 1 1 0 1 1 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> jdisp
	X.bit,\$addr16	PC ← \$addr16 if X.bit = 1 then reset X.bit	3	5/9	12	9				0 0 0 0 0 0 1 1 1 1 0 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> jdisp
	PSW.bit,\$addr16	PC ← \$addr16 if PSW.bit = 1 then reset PSW.bit	3	5/8	12	9	x	x	x	0 0 0 0 0 0 1 0 1 1 0 1 0 B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> jdisp





μPD7822x

T-49-19-07

T-49-19-59

Instruction Set (cont)

Mnemonic	Operand	Operation	Bytes	Clocks		Flags			Operation Code (Bits 7-0) Bytes B1 thru B5
				Int ROM	Branch	No Branch	Z	AC	
<b>Conditional Branch (cont)</b>									
DBNZ	ri,\$addr16	ri ← ri-1, then PC ← \$addr16 if ri = 0	2	3/5	9	6			0 0 1 1 0 0 1 R <sub>0</sub> jdisp
	saddr,\$addr16	(saddr) ← (saddr) - 1, then PC ← \$addr16 if (saddr) = 0	3	4-10	12	9			0 0 1 1 1 0 1 1 Saddr-offset jdisp
<b>CPU Control</b>									
MOV	STBC,#byte	STBC ← byte	4	10		15			0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 Data Data
SEL	RBn	RBS1-0 ← n, n = 0-3	2	2		6			0 0 0 0 0 1 0 1 1 0 1 0 1 0 N <sub>1</sub> N <sub>0</sub>
NOP		No Operation	1	2		3			0 0 0 0 0 0 0 0
EI		IE ← 1 (Enable Interrupt)	1	2		3			0 1 0 0 1 0 1 1
DI		IE ← 0 (Disable Interrupt)	1	2		3			0 1 0 0 1 0 1 0