NEC

NEC Electronics Inc.

μPD7831xA/78P31xA 16-/8-Bit, Single-Chip CMOS Microcomputers, Real-Time Control Oriented

Description

The μPD7831xA family of microcomputers is designed for use in process control. They perform all the usual process control functions and are particularly well-suited for driving stepping motors and dc motors in servo loops. The processors include on-chip memory, timers, input/output registers, and a powerful interrupt handling facility. The μPD78310A/312A is constructed of high-speed CMOS circuitry and operates from a single +5-volt power supply.

The input frequency (maximum 12 MHz) is derived from an external crystal or an external oscillator. The internal processor clock is two-phase, and thus machine states are executed at a rate of 6 MHz. The shortest instructions require three states, making the minimum time 500 ns. The CPU contains a three-byte instruction prefetch queue, which allows a subsequent instruction to be fetched during execution of an instruction that does not reference memory.

Program memory is 8K bytes of mask-programmable ROM (μPD78312A only), and data memory is 256 bytes of static RAM. The μPD78310A is the ROMless version. μPD78P312A is a prototyping chip for μPD78312A. It has an on-chip 8K EPROM instead of a mask ROM.

Features

- □ Complete single-chip microcomputer
  - —16-bit ALU
  - —8K ROM (μPD78312A only)
  - —256 bytes RAM
  - —1-bit and 8-bit logic
- □ Instruction prefetch queue
- □ 16-bit unsigned multiply and divide
- □ String instructions
- □ Memory expansion
  - —8085A bus-compatible
  - —Total 64K address space
- □ Large I/O capacity: up to 32 I/O port lines
- □ Extensive timer/counter system
  - —Two 16-bit up/down counters
  - —Quadrature counting
  - —Two 16-bit timers
  - —Free-running counter with two 16-bit capture registers
  - —Pulse-width modulated outputs
  - —Timebase counter
- □ Four-channel 8-bit A/D converter
- □ Two 4-bit real-time output ports
- □ Two nonmaskable interrupts
- □ Eight hardware priority interrupt levels
- □ Macroservice facility for interrupts gives the effect of eight DMA channels
- □ Bidirectional serial port
  - —Either UART or interface mode
  - —Dedicated baud rate generator
- □ Watchdog timer
- □ Refresh output for pseudostatic RAM
- □ Programmable HALT and STOP modes
- □ One-byte call instruction
- □ On-chip clock generator
- □ CMOS silicon gate technology
- □ +5-volt power supply

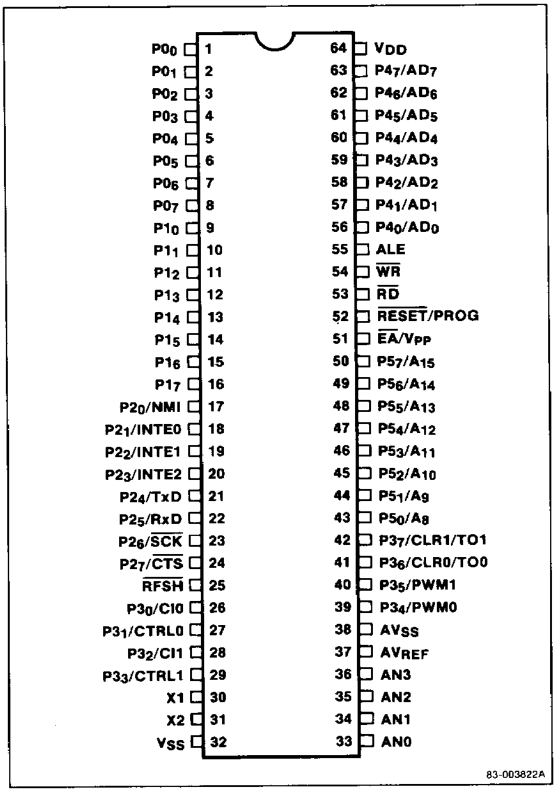
Ordering Information

Part Number	Package	ROM
μPD78310ACW	64-pin plastic shrink DIP	ROMless
μPD78310AGF-3BE	64-pin plastic QFP	
μPD78310AGQ-36	64-pin plastic QUIP	
μPD78310AL	68-pin plastic PLCC	
μPD78312ACW-xxx	64-pin plastic shrink DIP	Mask ROM
μPD78312AGF-xxx-3BE	64-pin plastic QFP	
μPD78312AGQ-xxx-36	64-pin plastic QUIP	
μPD78312AL-xxx	68-pin plastic PLCC	
μPD78P312ACW	64-pin plastic shrink DIP	OTP EPROM
μPD78P312AGF-3BE	64-pin plastic QFP	
μPD78P312AGQ-36	64-pin plastic QUIP	
μPD78P312AL	68-pin plastic PLCC	
μPD78P312ADW	64-pin ceramic shrink DIP with window (350 mil)	EPROM
μPD78P312AR	64-pin ceramic QUIP with window	

Notes: xxx is the ROM code number.

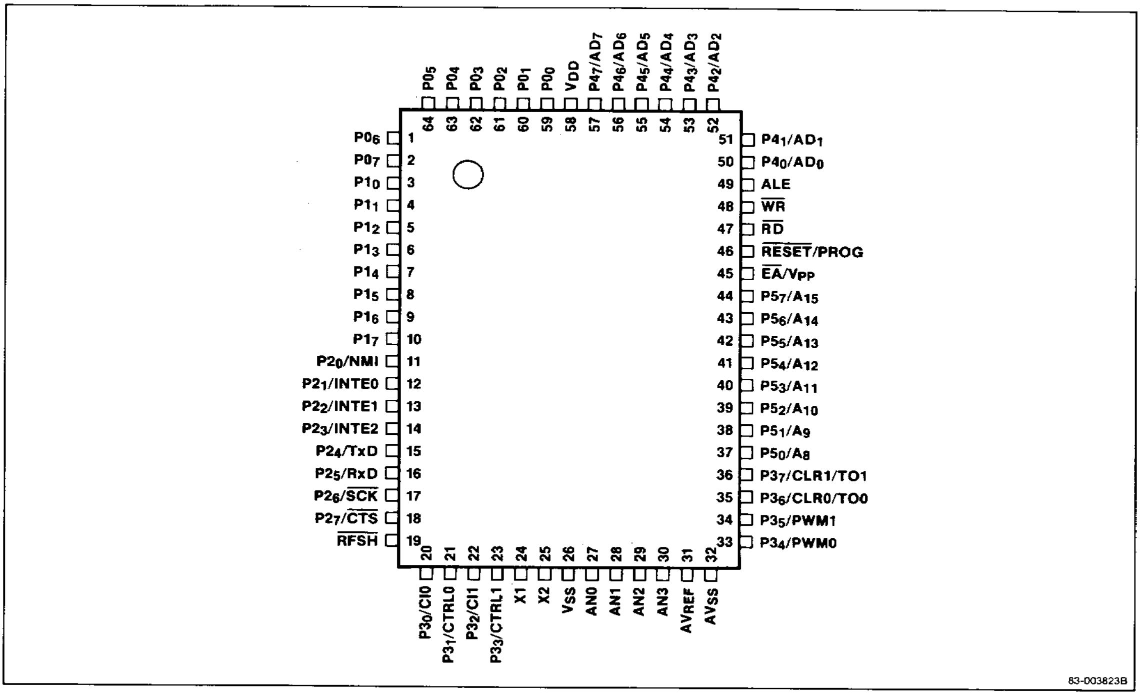
Pin Configurations

64-Pin Shrink DIP and QUIP, Plastic and Ceramic



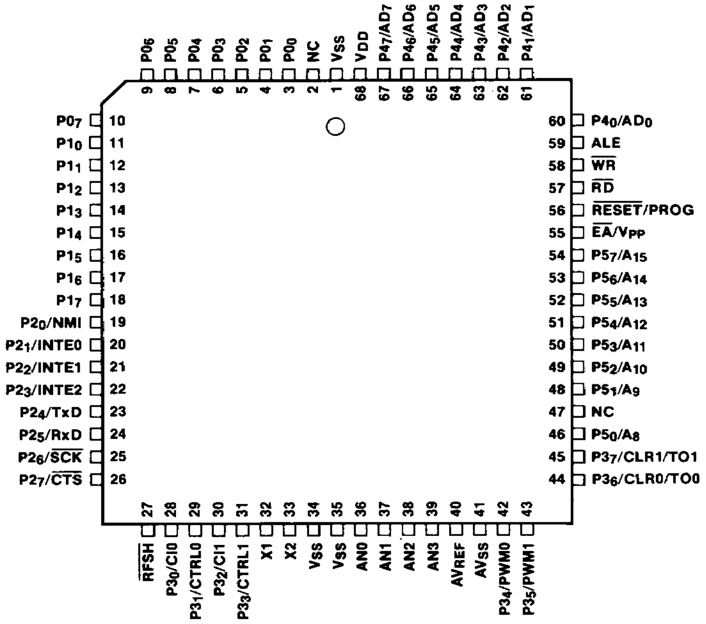
Pin Configurations (cont)

64-Pin Plastic QFP



Pin Configurations (cont)

68-Pin PLCC (Plastic Leaded Chip Carrier)



83-004442B

Pin Identification

Symbol	Function
AN0-AN3	A/D converter inputs
ALE	Address latch enable output
$\overline{\text{EA}}$/V$_{PP}$	External access control input; programming voltage
P0$_7$-P0$_0$	I/O port 0
P1$_7$-P1$_0$	I/O port 1
P2$_0$/NMI	Nonmaskable interrupt input
P2$_1$-P2$_3$/ INTE0-INTE2	Maskable interrupt inputs
P2$_4$/TxD	I/O port 2; serial transmit output
P2$_5$/RxD	I/O port 2; serial receive input
P2$_6$/$\overline{\text{SCK}}$	I/O port 2; serial clock output
P2$_7$/$\overline{\text{CTS}}$	I/O port 2; clear to send input
P3$_0$/CI0	Up/down counter 0 input
P3$_1$/CTRL0	Up/down counter 0 control input
P3$_2$/CI1	Up/down counter 1 input
P3$_3$/CTRL1	Up/down counter 1 control input
P3$_4$/PWM0	I/O port 3; pulse width modulated output 0
P3$_5$/PWM1	I/O port 3; pulse width modulated output 1
P3$_6$/CLR0/TO0	I/O port 3; counter 0 clear input; timer 0 output
P3$_7$/CLR1/TO1	I/O port 3; counter 1 clear input; timer 1 output
P4$_7$-P4$_0$/AD$_7$-AD$_0$	I/O port 4; external address; data bus
P5$_7$-P5$_0$/A$_{15}$-A$_8$	I/O port 5; high address byte output
$\overline{\text{RD}}$	Read strobe output
$\overline{\text{RESET}}$/PROG	External reset input; PROM programming mode
$\overline{\text{RFSH}}$	Refresh output
$\overline{\text{WR}}$	Write strobe output
X1	External crystal or external clock input
X2	External crystal
AV$_{REF}$	A/D reference voltage
AV$_{SS}$	Analog ground
V$_{DD}$	Power supply
V$_{SS}$	Power return

PIN FUNCTIONS

AN0-AN3 (A/D Converter Inputs)

AN0-AN3 are the four program selectable input channels for the A/D converter.

ALE (Address Latch Enable)

ALE is the address latch enable. It is to be used by external circuitry to latch the low-order 8 address bits during the first part of a read or write cycle.

$\overline{\text{EA}}$/V$_{PP}$

On μPD78312A, a low on $\overline{\text{EA}}$ enables use of external memory in place of on-chip ROM. The $\overline{\text{EA}}$ pin must be low on μPD78310A. On the μPD78P312A, this pin is used for programming voltage. In normal operation, it must be connected to V$_{DD}$.

P0$_7$-P0$_0$ (Port 0)

Port 0 consists of 8 bits, individually programmable for input/output or two 4-bit real-time (timer controlled) output ports.

P1$_7$-P1$_0$ (Port 1)

Port 1 consists of 8 bits, individually programmable for input/output.

P2$_0$/NMI (Port 2; Nonmaskable Interrupt)

Port P2$_0$ is dedicated to NMI, the nonmaskable external interrupt request.

P2$_1$-P2$_3$/INTE0-INTE2 (Port 2; Maskable Interrupts)

Ports P2$_1$-P2$_3$ are dedicated to INTE0, INTE1, and INTE2, the maskable external interrupt requests.

P2$_4$/TxD (Port 2; Serial Transmit)

P2$_4$ is an I/O port bit or the transmitted serial data output.

P2$_5$/RxD (Port 2; Serial Receive)

P2$_5$ is an I/O port bit or the received serial data input.

P2$_6$/$\overline{\text{SCK}}$ (Port 2; Serial Clock)

P2$_6$ is an I/O port bit or the serial shift clock output.

P2$_7$/$\overline{\text{CTS}}$ (Port 2; Clear to Send)

P2$_7$ is an I/O port bit or clear-to-send input (external serial transmission control) in the asynchronous communication mode. In the serial I/O interface mode, it becomes the serial receive clock I/O pin.

P3$_0$/CI0 (Port 3; Counter 0)

Port P3$_0$ is dedicated to CI0, the external count input for up/down counter 0.

P3$_1$/CTRL0 (Port 3; Counter 0 Control)

Port P3$_0$ is dedicated to CTRL0, the external control input for up/down counter 0.

$P3_2$/CI1 (Port 3; Counter 1)

Port $P3_2$ is dedicated to CI1, the external count input for up/down counter 1.

$P3_3$/CTRL1 (Port 3; Counter 1 Control)

Port $P3_3$ is dedicated to CTRL1, the external control input for up/down counter 1.

$P3_4$/PWM0 (Port 3; Pulse Width 0)

$P3_4$ is an I/O port bit or the pulse-width modulated output 0.

$P3_5$/PWM1 (Port 3; Pulse Width 1)

$P3_5$ is an I/O port bit or the pulse-width modulated output 1.

$P3_6$/CLR0/TO0 (Port 3; Counter 0 Clear; Timer 0)

$P3_6$ is an I/O port bit, or the clear input for up/down counter 0, or the timer 0 flip-flop output.

$P3_7$/CLR1/TO1 (Port 3; Counter 1 Clear; Timer 1)

$P3_7$ is an I/O port bit, or the clear input for up/down counter 1, or the timer 1 flip-flop output.

$P4_0$-$P4_7$/AD_0-AD_7 (Port 4; External Address/Data Bus)

Port 4 consists of 8 bits, programmable as a unit for input or output, or as the multiplexed address/data bus if external memory or external interface circuitry is used. The port is controlled by the memory mapping register. If the $\overline{EA}$ pin is low, port 4 is always an address/data bus.

$P5_0$-$P5_7$/A_8-A_{15} (Port 5; High-Address Byte)

Port 5 consists of 8 bits, individually programmable for input or output, or the high-order address bits for external memory. Under control of the memory mask register, bits $P5_3$-$P5_0$ are used for 4K memory expansion, bits $P5_5$-$P5_0$ for 16K memory expansion, or bits $P5_7$-$P5_0$ for 56K memory expansion. If the $\overline{EA}$ pin is low, port 5 is always the high-order address bus.

$\overline{RD}$ (Read Strobe)

$\overline{RD}$ is the read strobe output. It is to be used by external memory (or data registers) to place data on the I/O bus during a read operation.

$\overline{RESET}$/PROG

This pin is used for the external reset input. A low level sets all registers to their specified reset values. During programming of the µPD78P312A, this pin is used to place the device into PROM programming mode.

$\overline{RFSH}$ (Refresh)

$\overline{RFSH}$ is the refresh pulse output to be used for external pseudostatic DRAM.

$\overline{WR}$ (Write Strobe)

$\overline{WR}$ is the write strobe output. It is to be used by external memory (or data registers) to latch data from the I/O bus during a write operation.

X1, X2 (External Crystal or Clock Input)

X1 and X2 are the external oscillator inputs or the connections for an external crystal. If an external clock is used, it is connected to X1 and its inverse is connected to X2. The system clock frequency is half the input frequency.

AV_{REF} (A/D Reference Voltage)

AV_{REF} is the reference voltage input for the A/D converter.

AV_{SS} (Analog Ground)

AV_{SS} is the analog ground pin.

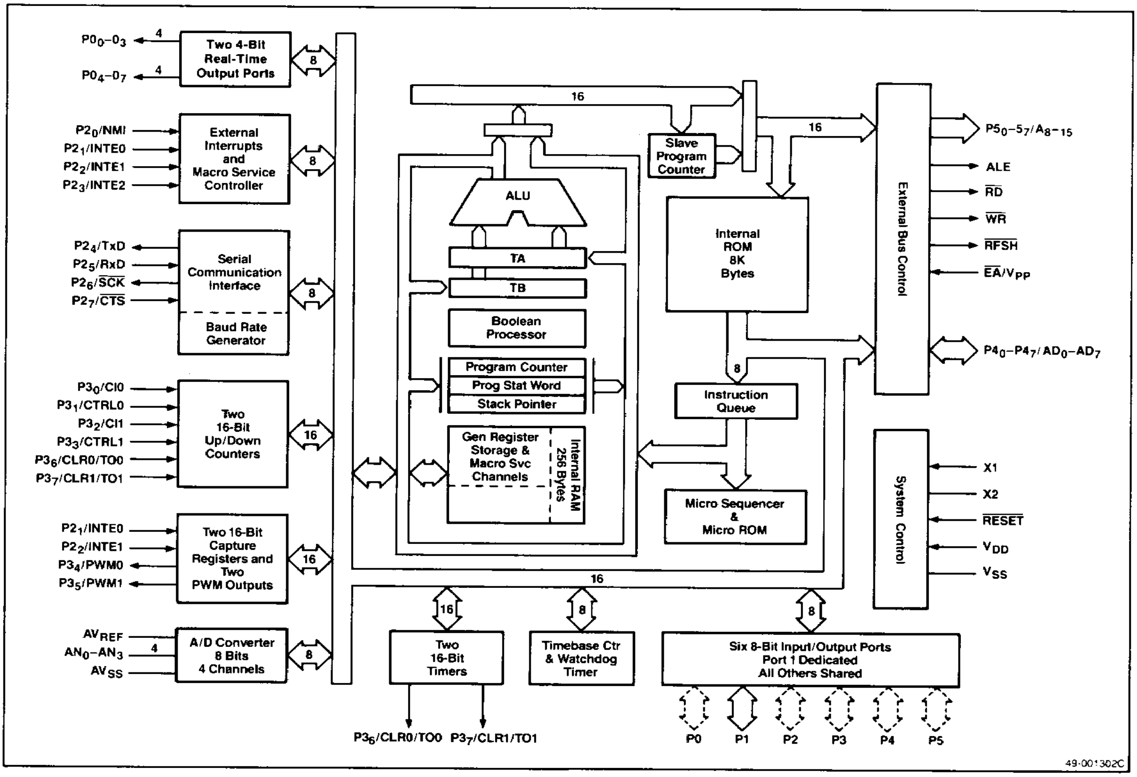
V_{DD} (Power Supply)

V_{DD} is the positive power supply input.

V_{SS} (Power Return)

V_{SS} is the power supply return, normally ground.

Block Diagram



FUNCTIONAL DESCRIPTION

On-chip features designed to facilitate process control include two 16-bit timers, quadrature counting, two 16-bit up/down counters, two pulse-width modulated outputs, a free-running counter with two capture registers, two 4-bit real-time (timer-controlled) output ports, an 8-bit A/D converter with four input channels, a timebase counter to generate widely spaced interrupts, and a watchdog timer to guard against infinite program loops.

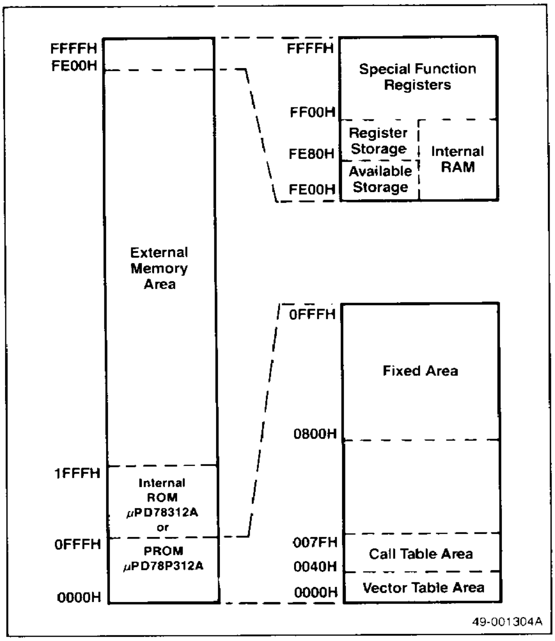
In addition, a serial I/O port can be used in either an interface mode or an asynchronous communication mode. HALT and STOP modes are provided to conserve power at times when CPU action is not required.

All I/O, timer, and control registers are defined as special function registers and assigned addresses in the top 256 bytes of memory. The special function registers may be operated on directly by many of the arithmetic, logic, and move instructions of the CPU. Table 1 describes the registers.

Addressing

The μPD783101xA features 1-byte addressing of the special function registers and 1-byte addressing of the internal RAM. There are nine modes of addressing main memory, including autoincrement, autodecrement, indexing, and double indexing. There are 8- and 16-bit immediate operands.

Figure 1. Memory Map



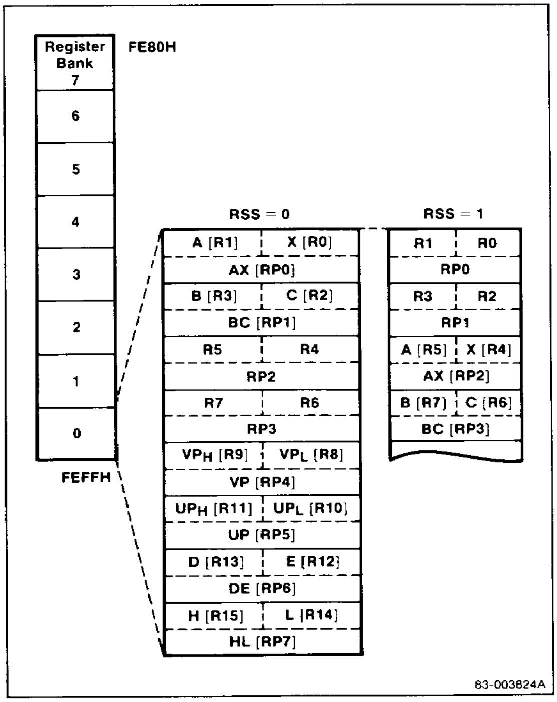
External Memory

External memory (figure 1) 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 additional wait states. High-order address bits are taken from I/O port 5 as required. No bits are required for 256 bytes of external memory; bits $P5_3$-$P5_0$ are used for 4K bytes, $P5_5$-$P5_0$ for 16K bytes, and $P5_7$-$P5_0$ for 56K bytes. Any remaining port 5 bits are available for I/O.

Refresh

The μPD7831xA has a refresh signal for use with the pseudostatic RAM. The refresh cycle can be set to one of four intervals ranging from 2.67 to 21.3 μs. The refresh is timed to follow a read or write operation so that there is no interference.

Figure 2. Register Designation and Storage



General Registers

The CPU has sixteen 8-bit registers (figure 2) that can also be used in pairs to function as 16-bit registers. A complete set of 16 general registers is mapped into each of 8 program-selectable register banks stored in RAM. Three bits in the PSW specify which of the register banks is active at any given time. Each register bank has two program-selectable accumulators.

The general registers of the μPD7831xA have both absolute and functinal names. AX is the functional name for the accumulator. Setting the RSS bit in the PSW to 1 transfers the AX and BC registers from their normal RP0 and RP1 positions to RP2 and RP3 as shown in figure 2. This adds considerable programming flexibility.

Program Status Word

Following is the program status word format.

0	RB_2	RB_1	RB_0	0	0	IE	0
15							8

S	Z	RSS	AC	UF	P/V	SUB	CY
7							0

RB_2-RB_0	Active register bank number
IE	Interrupt enable
S	Sign (1 if last result was negative)
Z	Zero (1 if last result was zero)
RSS	Register set select
AC	Auxiliary carry (carry out of 3rd bit)
UF	User flag
P/V	Parity or arithmetic overflow
SUB	Subtract (1 if last operation was subtract)
CY	Carry

Input/Output

All ports may be used for either latched output or high-impedance input. All ports except port 4 are bit-programmable for input or output. Port 0 is used for real-time or normal I/O. Port 1 is used for normal I/O. The low nibble of ports 2 and 3 is always used for control and the high nibble for control or normal I/O. Port 4 is used for the external address/data bus or byte-programmable I/O. Port 5 is used for the high bits of the external address or for normal I/O.

Real-Time Output Port

The real-time output port shares pins with I/O port 0. The high and low nibbles are treated separately or together. Data is transferred from a buffer to the port latches on either a timer or software command.

Serial Port

The serial port can operate in UART or interface mode with the baud rate and byte format under program control. The serial port also includes a dedicated baud rate generator.

Pulse-Width Modulated Outputs

The two independent pulse-width modulated outputs are controlled by two 16-bit modulus registers and counters. There are four programmable repetition rates ranging from 91.6 Hz to 23.4 MHz. Figure 3 shows one of these outputs.

Timers

The μPD7831xA has two 16-bit timers. The inputs to these timers may be the internal clock divided by 6 or by 128. Each timer has an associated modulus register to store the timer count. The timer counts down to zero, sets a flag, reloads from the modulus register, and then counts down again. The timer flags can be used under program control to generate interrupt requests and/or a square-wave output. TM0 also functions optionally as two one-shot timers.

Figure 4 is a diagram of the interval timers.

There is a free-running counter that counts the internal clock divided by 4 or by 16. The counter has two 16-bit capture registers. Capture is triggered by an external interrupt request or by the up/down counter clock.

The timebase counter generates a signal at one of four intervals ranging from 170 μs to 175 ms. The signal can be used to generate an interrupt request and/or an up/down counter capture.

Figure 3. Pulse-Width Modulated Output

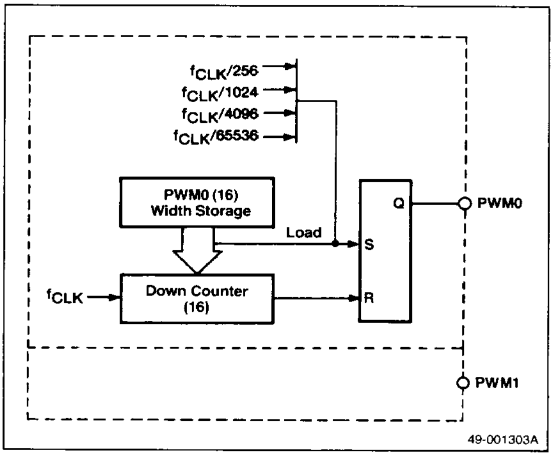
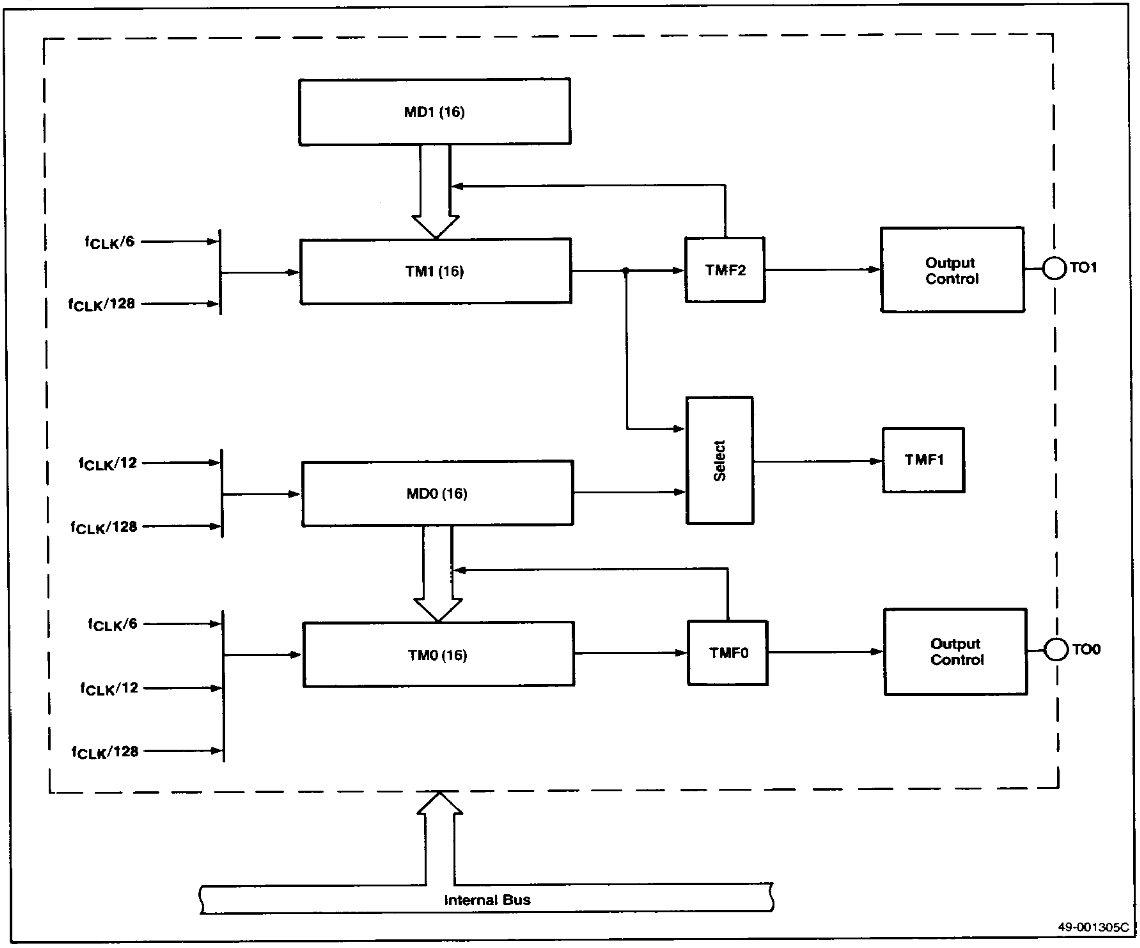


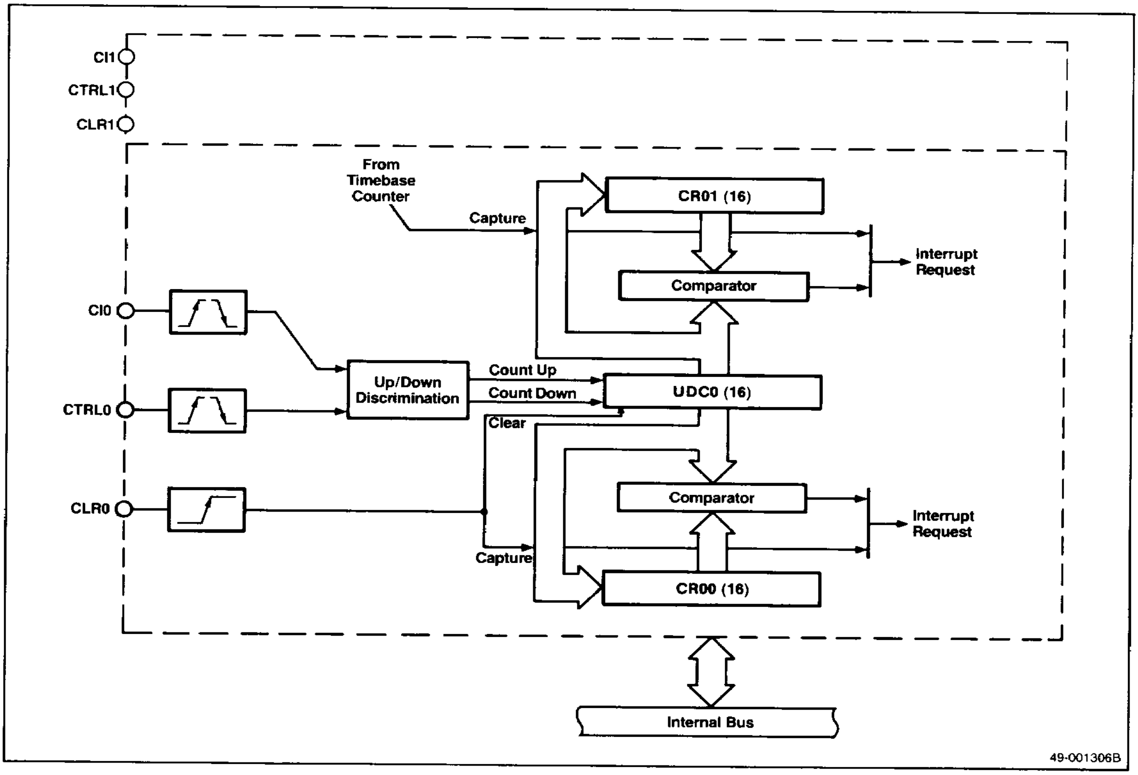
Figure 4. Timer Block Diagram



Up/Down Counters

The µPD7831xA has two 16-bit up/down counters, each of which has two capture/compare registers. There are three modes of operation: compare and interrupt, capture on external command, and capture on timebase counter command. There are five sources of counts: the internal clock divided by 3, the external clock, external independent up and down inputs, external clock with direction control, and external clock with automatic up/down discrimination. Figure 5 shows an up/down counter.

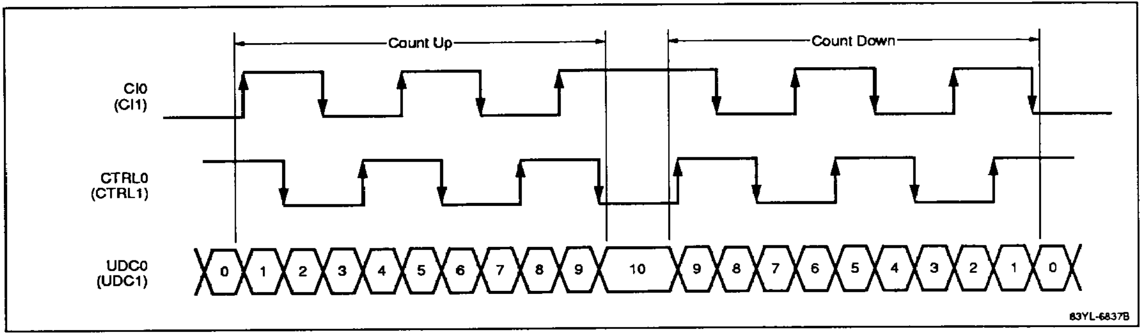
Figure 5. Up/Down Counter Block Diagram



Quadrature Counting

The two up/down counters, UDC0 and UDC1, have an optional quadrature counting mode, which is activated by specifying mode 4 in the counter unit input mode register, CUIM. It is designed to count the output of a two-phase pulsed optical shaft angle encoder. The input for phase A is the CI0 (or CI1) pin, and the input for phase B is the CTRL0 (or CTRL1) pin. The counter UDC0 (or UDC1) is incremented or decremented at both positive and negative transitions of both input signals. Whether it is incremented or decremented is dependent upon the relative phase of the two signals as illustrated in figure 6.

Figure 6. Counter Operation (Mode 4)



Standby Modes

HALT and STOP modes conserve power when CPU action is not required. In HALT mode, the CPU stops and the clock continues to run. Maskable interrupts can restart the CPU.

In STOP mode, the CPU and clock are both stopped. A $\overline{\text{RESET}}$ pulse or the nonmaskable external interrupt is required to restart them. There is also the option of slowing the system clock by a factor of four. The standby control register controls the standby modes and is a protected location written to only by a special instruction.

Watchdog Timer

The watchdog timer protects against inadvertent program loops. A nonmaskable interrupt occurs if the timer is not reset before a timeout occurs. There are four program-selectable intervals ranging from 5.5 to 349.3 ms. The watchdog timer can be disabled by software. The watchdog timer mode register controls the watchdog timer and is a protected location written to only by a special instruction.

A/D Converter

The A/D converter has four input channels and can operate in either scan or select mode. The A/D converter performs 8-bit successive approximation conversions, has a 30-µs conversion time, and is triggered either internally or externally. The A/D converter includes an on-chip sample and hold amplifier.

Interrupts

There are two nonmaskable interrupt sources: the external nonmaskable interrupt and the watchdog timer. Their relative priorities are software selectable.

There are eight hardware priority interrupt levels, level 0 having the highest priority and level 7 the lowest. The 15 maskable interrupt sources (table 2) are divided into five groups, and each group can, under program control, be assigned to any one of the priority levels.

Interrupts may be serviced by routines entered either by vectoring or by context switching. Context switching automatically saves all the general registers, the program status word, and the program counter. Figure 7 illustrates the mechanism of context switching.

Finally, an optional macroservice function transfers data between any one special function register and memory without program intervention.

Macroservice

The macroservice controller can be programmed to perform word or byte transfers. It can 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 eight macroservice channels; channel control information is stored in RAM. This information (figure 8) consists of a 16-bit memory address (optionally incremented at each transfer), and 8-bit special function register designator, and an 8-bit transfer counter (decremented at each transfer). When the count equals 0, a context switch or vectored interrupt occurs.

Figure 7. Hardware Context Switching

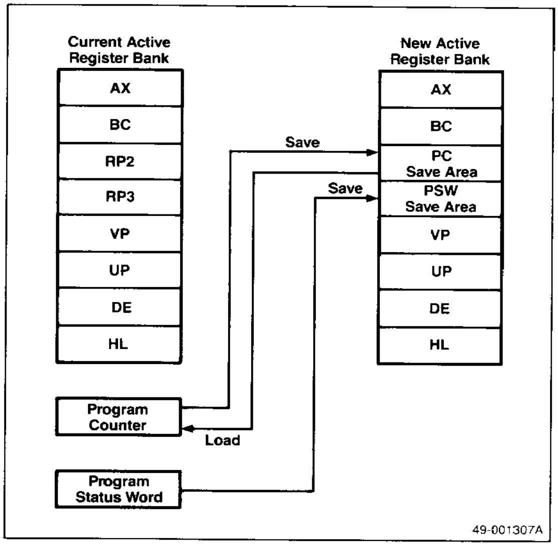
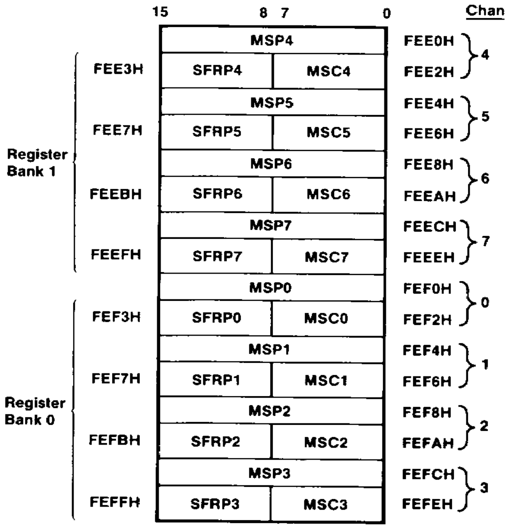


Figure 8. Macroservice Pointer Addresses



Note:

[1] The macroservice pointers share storage with register banks 0 and 1.

[2] MSP = Memory address pointer
SFRP = Special function register pointer
MSC = Transfer counter

83-003825A

Table 1. Special Function Registers

Address	Function	Mnemonic		Read/Write	16-Bit Transfer	Reset State
FF00H	I/O port 0	P0		R/W	No	Undefined
FF01H	I/O port 1	P1		R/W	No	Undefined
FF02H	I/O port 2	P2		R/W (Note 1)	No	Undefined
FF03H	I/O port 3	P3		R/W (Note 1)	No	Undefined
FF04H	I/O port 4	P4		R/W	No	Undefined
FF05H	I/O port 5	P5		R/W	No	Undefined
FF08H FF09H	Capture/compare register 00	CR00L CR00H	CR00	R/W	Yes	Undefined
FF0AH FF08H	Capture/compare register 01	CR01L CR01H	CR01	R/W	Yes	Undefined
FF0CH FF0DH	Capture/compare register 10	CR10L CR10H	CR10	R/W	Yes	Undefined
FF0EH FF0FH	Capture/compare register 11	CR11L CR11H	CR11	R/W	Yes	Undefined
FF10H F11H	Capture register 0 (from FRC)	CPT0L CPT0H	CPT0	R/W	Yes	Undefined
FF12H FF13H	Capture register 1 (from FRC)	CPT1L CPT1H	CPT1	R/W	Yes	Undefined

Table 1. Special Function Registers (cont)

Address	Function	Mnemonic		Read/Write	16-Bit Transfer	Reset State
FF14H FF15H	PWM register 0 (duration)	PWM0L PWM0H	PWM0	R/W	Yes	Undefined
FF16H FF17H	PWM register 1 (duration)	PWM1L PWM1H	PWM1	R/W	Yes	Undefined
FF1CH FF1DH	Presettable up/down counter 0	UDC0L UDC0H	UDC0	R/W	Yes	Undefined
FF1EH FF1FH	Presettable up/down counter 1	UDC1L UDC1H	UDC1	R/W	Yes	Undefined
FF20H	Port 0 mode register	PM0		R/W	No	FFH
FF21H	Port 1 mode register	PM1		R/W	No	FFH
FF22H	Port 2 mode register	PM2		R/W (Note 1)	No	FFH
FF23H	Port 3 mode register	PM3		R/W (Note 1)	No	FFH
FF25H	Port 5 mode register	PM5		R/W	No	FFH
FF32H	Port 2 mode control register	PMC2		R/W	No	0FH
FF33H	Port 3 mode control register	PMC3		R/W	No	0FH
FF38H	Real-time output port control register	RTPC		R/W	No	08H
FF3AH FF3BH	Port 0 buffer register (Note 2)	P0L P0H		R/W	No	Undefined
FF40H	Memory expansion mode register	MM		R/W	No	30H
FF41H	Refresh mode register	RFM		R/W	No	10H
FF42H	Watchdog timer mode register	WDM		R/W	No	00H
FF44H	Standby control register	STBC		R/W	No	2nH (Note 3)
FF46H	Timebase mode register	TBM		R/W	No	00H
FF48H	External interrupt mode register	INTM		R/W	No	00H
FF4AH	In-service priority register	ISPR		R	No	00H
FF4EH	CPU control word	CCW		R/W	No	00H
FF50H	Serial communication mode register	SCM		R/W	No	00H
FF52H	Serial communication control register	SCC		R/W	No	00H
FF53H	Baud rate generator	BRG		R/W	No	00H
FF56H	Serial communication receive buffer	RXB		R	No	Undefined
FF57H	Serial communication transmit buffer	TXB		W	No	Undefined
FF60H	Free-running counter control register	FRCC		R/W	No	00H

Table 1. Special Function Registers (cont)

Address	Function	Mnemonic	Read/Write	16-Bit Transfer	Reset State
FF64H	Capture mode register	CPTM	R/W	No	00H
FF66H	PWM mode register	PWMM	R/W	No	00H
FF68H	A/D converter mode register	ADM	R/W	No	00H
FF6AH	A/D converter result register	ADCR	R	No	Undefined
FF70H	Count unit input mode register	CUIM	R/W	No	00H
FF72H	Up/down counter control register 0	UDCC0	R/W	No	00H
FF74H	Capture/compare control register	CRC	R/W	No	00H
FF7AH	Up/down counter control register 1	UDCC1	R/W	No	00H
FF80H	Timer 0 control register	TMC0	R/W	No	00H
FF82H	Timer 1 control register	TMC1	R/W	No	00H
FF88H FF89H	Timer 0	TM0L TM0 TM0H	R/W	Yes	Undefined
FF8AH FF8BH	Modulus/timer register 0	MD0L MD0 MD0H	R/W	Yes	Undefined
FF8CH FF8DH	Timer 1	TM1L TM1 TM1H	R/W	Yes	Undefined
FF8EH FF8FH	Modulus register 1	MD1L MD1 TM1H	R/W	Yes	Undefined
FFB0H to FFBFH	External area (Note 4)				
FFC0H	CRF00 interrupt control Up/down counter 0	CRIC00	R/W	No	47H
FFC1H	CRF00 macroservice control Up/down counter 0	CRMS00	R/W	No	Undefined
FFC2H	CRF01 interrupt control Up/down counter 0	CRIC01	R/W	No	47H
FFC4H	CRF10 Interrupt control Up/down counter 1	CRIC10	R/W	No	47H
FFC5H	CRF10 macroservice control Up/down counter 1	CRMS10	R/W	No	Undefined
FFC6H	CRF11 interrupt control Up/down counter 1	CRIC11	R/W	No	47H
FFC8H	EXIF0 interrupt control External interrupt INTE0	EXIC0	R/W	No	47H
FFC9H	EXIF0 macroservice control External interrupt INTE0	EXMS0	R/W	No	Undefined
FFCAH	EXIF1 interrupt control External interrupt INTE1	EXIC1	R/W	No	47H
FFCBH	EXIF1 macroservice control External interrupt INTE1	EXMS1	R/W	No	Undefined
FFCCH	EXIF2 interrupt control External interrupt INTE2	EXIC2	R/W	No	47H

Table 1. Special Function Registers (cont)

Address	Function	Mnemonic	Read/Write	16-Bit Transfer	Reset State
FFCDH	EXIF2 macroservice control External interrupt INTE2	EXMS2	R/W	No	Undefined
FFCEH	TMF0 interrupt control Timer flag	TMIC0	R/W	No	47H
FFCFH	TMF0 macroservice control Timer flag	TMMS0	R/W	No	Undefined
FFD0H	TMF1 interrupt control Timer flag	TMIC1	R/W	No	47H
FFD1H	TMF1 macroservice control Timer flag	TMMS1	R/W	No	Undefined
FFD2H	TMF2 interrupt control Timer flag	TMIC2	R/W	No	47H
FFD3H	TMF2 macroservice control Timer flag	TMMS2	R/W	No	Undefined
FFDAH	Receive error interrupt control Serial port	SEIC	R/W	No	47H
FFDCH	Receive interrupt control Serial port	SRIC	R/W	No	47H
FFDDH	Receive macroservice control Serial port	SRMS	R/W	No	Undefined
FFDEH	Transmit interrupt control Serial port	STIC	R/W	No	47H
FFDFH	Transmit macroservice control Serial port	STMS	R/W	No	Undefined
FFE0H	A/D converter interrupt control	ADIC	R/W	No	47H
FFE1H	A/D converter macroservice control	ADMS	R/W	No	Undefined
FFE2H	Timebase counter interrupt control	TBIC	R/W	No	47H

Notes:

(1) Bits 0-3 of port 2 and of port 3 are read-only.

(2) P0H and P0L are 4-bit buffer registers used to store data to be loaded into the high and low nibbles of the real-time output (P0). The high order 4 bits of P0H and the low order 4 bits of P0L are used.

(3) Bit 3 of the STBC is not affected by $\overline{\text{RESET}}$ (n = 0 or 8).

(4) External registers interfaced with these addresses can be accessed by special function register addressing.

Table 2. Interrupt Sources and Vector Addresses

	Default Priority	Mnemonic	Interrupt Source	Macroservice	Vector
Software	—	BRK	Break instruction	No	003EH
Nonmaskable Interrupts		NMI	External nonmaskable interrupt	No	0002H
	—	WDT	Watchdog timer	No	000AH
Maskable interrupts	0	CRF00	Up/down counter 0	Yes	001AH
	1	CRF01	Up/down counter 0	No	001CH
	2	CRF10	Up/down counter 1	Yes	001EH
	3	CRF11	Up/down counter 1	No	0020H
	4	EXIF0	External interrupt 0	Yes	0004H
	5	EXIF1	External interrupt 1	Yes	0006H
	6	EXIF2	External interrupt 2	Yes	0008H
	7	TMF0	Timer flag 0	Yes	000EH
	8	TMF1	Timer flag 1	Yes	0010H
	9	TMF2	Timer flag 2	Yes	0012H
	10	SEF	Serial port error	No	0022H
	11	SRF	Serial port receive buffer	Yes	0024H
	12	STF	Serial port transmit buffer	Yes	0026H
	13	ADF	A/D converter done flag	Yes	0028H
	14	TBF	Timebase counter flag	No	000CH
Reset	—	$\overline{\text{RESET}}$	External reset line	—	0000H

ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings

T_A +25°C

Power supply voltage V_{DD}	−0.5 to +7.0 V
Reference voltage, AV_{REF}	−0.5 V to V_{DD} +0.3 V
Power supply return, AV_{SS}	−0.5 to +0.5 V
Input voltage, V_{I1} (except $\overline{\text{RESET}}$ of µPD78P312A)	−0.5 to +V_{DD} + 0.5
Input voltage, V_{I2} ($\overline{\text{RESET}}$ of µPD78P312A only)	−0.5 to +13.5 V
Output voltage, V_O	−0.5 to V_{DD} +0.5 V
Output current, low; I_{OL} (single pin)	4 mA
Output current, low; I_{OL}; total, all output pins (µPD78312/310A)	100 mA
Output current, low; I_{OL}; total, all output pins (µPD78P312A)	60 mA
Output current, high; I_{OH} (single pin)	−1 mA
Output current, high; I_{OH}; total, all output pins (µPD78312/310A)	−25 mA
Output current, high; I_{OH}; total, all output pins (µPD78P312A)	−15 mA
Operating temperature, T_{OPT}	−10 to +70 °C
Storage temperature, T_{STG}	−65 to +150 °C

Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage.

Operating Frequency

Oscillator Frequency f_{XX}	T_A	V_{DD}
4 MHz ≤ f_{XX} ≤ 12 MHz	−10 to +70°C	+5.0 V 10%

Capacitance

T_A = +25°C; V_{DD} = V_{SS} = 0 V

Parameter	Symbol	Max	Unit	Conditions
Input capacitance	C_I	10	pF	f = 1 MHz; unmeasured pins returned to 0 V.
Output capacitance	C_O	20	pF	
I/O capacitance	C_{IO}	20	pF	

DC Characteristics

T_A = −10 to +70°C; V_{DD} = +5.0 V ±5%; V_{SS} = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Input low voltage	V_{IL1}	0		0.8	V	Except $\overline{EA}$ on µPD78310A/312A
	V_{IL2}	0		0.5	V	$\overline{EA}$ on (µPD78310A/312A only)
Input high voltage	V_{IH1}	2.2		V_{DD}	V	Except $P2_0$/NMI, X1, X2, $\overline{RESET}$
	V_{IH2}	3.8		V_{DD}	V	$P2_0$/NMI X1, X2, $\overline{RESET}$
Output low voltage	V_{OL}			0.45	V	I_{OL} = 2.0 mA
Output high voltage	V_{OH}	V_{DD} −1			V	I_{OH} = −1 mA
Input current	L_I			±10	µA	$P2_0$/NMI, $\overline{RESET}$ V_I = 0.45 V to V_{DD}
Input leakage current	I_{LI}			±10	µA	
Input/output leakage current	I_{LO}			±10	µA	
AV_{REF} current	AI_{REF}		1.5	5	mA	f_{CLK} = 6 MHz
V_{DD} supply current	I_{DD1}		30	60	mA	Operating mode; f_{CLK} = 6 MHz
	I_{DD2}		5	15	mA	Halt mode; f_{CLK} = 6 MHz
Data retention voltage	V_{DDDR}	2.5			V	Stop mode
Stop mode supply current	I_{DDDR}		3	15	µA	Stop mode; V_{DDDR} = 2.5 V
			10	50	µA	Stop mode; V_{DDDR} = 5.0 V ±10%

AC Characteristics

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

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Read/Write Operation						
System clock cycle time	t_{CYK}	166		2000	ns	(Note 1)
Address setup time to ALE ↓	t_{SAL}	150			ns	
Address hold time after ALE ↓	t_{HLA}	30			ns	(Note 4)
Address to $\overline{RD}$ ↓ delay time	t_{DAR}	230			ns	
$\overline{RD}$ ↓ to address floating	t_{FRA}			0	ns	
Address to data input	t_{DAID}			410	ns	
ALE ↓ to data input	t_{DLID}			230	ns	
$\overline{RD}$ ↓ to data input	t_{DRID}			180	ns	
ALE ↓ to $\overline{RD}$ ↓ delay time	t_{DLR}	60			ns	
Data hold time after $\overline{RD}$ ↑	t_{HRID}	0			ns	
$\overline{RD}$ ↑ to address active	t_{DRA}	50			ns	
$\overline{RD}$ ↑ to ALE ↑ delay time	t_{DRL}	100			ns	
$\overline{RD}$ width low	t_{WRL}	200			ns	
ALE width high	t_{WLH}	120			ns	
Address to $\overline{WR}$ ↓ delay time	t_{DAW}	300			ns	
ALE ↓ to data output	t_{DLOD}			190	ns	
$\overline{WR}$ ↓ to data output	t_{DWOD}			100	ns	
ALE ↓ to $\overline{WR}$ ↓ delay time (Note 2)	t_{DLW}	30			ns	
		110			ns	During refresh mode
Data setup time to $\overline{WR}$ ↑	t_{SODWR}	150			ns	

AC Characteristics (cont)

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Read/Write Operation (cont)						
Data setup time to $\overline{WR}$ ↓ (Note 3)	t_{SODWF}	30			ns	During refresh mode
Data hold time to $\overline{WR}$ ↑	t_{HWOD}	20			ns	(Note 4)
$\overline{WR}$ ↑ to ALE ↑ delay time	t_{DWL}	110			ns	
$\overline{WR}$ width low	t_{WWL}	200			ns	
Serial Port						
Serial clock cycle time	t_{CYSK}	1.33			μs	$\overline{SCK}$ output (Note 5)
		1.33			μs	$\overline{CTS}$ output (Note 6)
		1			μs	$\overline{CTS}$ input (Note 7)
Serial clock low level width	t_{WSKL}	580			ns	$\overline{SCK}$ output (Note 5)
		580			ns	$\overline{CTS}$ output (Note 6)
		420			ns	$\overline{CTS}$ input (Note 7)
Serial clock high level width	t_{WSKH}	580			ns	$\overline{SCK}$ output (Note 5)
		580			ns	$\overline{CTS}$ output (Note 6)
		420			ns	$\overline{CTS}$ input (Note 7)
$\overline{CTS}$ high, low level	t_{WCSH}, t_{WCSL}	3			t_{CYK}	Asynchronous mode
RxD setup time to $\overline{CTS}$ ↑	t_{SRXSK}	80			ns	
RxD hold time after $\overline{CTS}$ ↑	t_{HSKRX}	80			ns	
$\overline{SCK}$ ↓ to TxD delay time	t_{DSKTX}			210	ns	
A/D Converter T_A = −10°C to +70°C; V_{DD} = +5 V ±10%; AV_{REF} = 4.0 V to V_{DD}; AV_{SS} = V_{SS} = 0 V						
Resolution		8			Bit	
Full scale error				0.4	%	t_{CYK} = 166 to 500 ns
Quantization error				±1/2	LSB	
Conversion time	t_{CONV}	180			t_{CYK}	t_{CYK} = 166 to 250 ns
		120			t_{CYK}	t_{CYK} = 250 to 500 ns
Sampling time	t_{SAMP}	36			t_{CYK}	t_{CYK} = 166 to 250 ns
		24			t_{CYK}	t_{CYK} = 250 to 500 ns
Analog input voltage	V_{IAN}	0		AV_{REF}	V	
Input impedance	R_{AN}		1000		mΩ	
Analog reference voltage	AV_{REF}	4.0		V_{DD}	V	
AV_{REF} current	AI_{REF}		1.5	5.0	mA	f_{CLK} = 6 MHz
Counter Operation						
CI0, CI1 high, low levels	t_{WCIH}, t_{WCIL}	3			t_{CYK}	
CTRL0, CRTL1 high, low levels	t_{WCTH}, t_{WCTL}	3			t_{CYK}	
CTRL0, CTRL1 setup time to CI ↑	t_{SCTCI}	2			t_{CYK}	Operating mode of count unit is set to mode 3. CI input is set to rising edge active.

AC Characteristics (cont)

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Counter Operation (cont)						
CTRL0, CTRL1 hold time after CI ↑	t_{HCICT}	5			t_{CYK}	
CLRO. CLR1 high, low level width	t_{WCRH}, t_{WCRL}	3			t_{CYK}	
CI0, CI1 setup time to CTRL	t_{S4CTCI}	6			t_{CYK}	Counter mode 4
CTRL0, CTRL1 setup time to CI	t_{H4CTCI}	6			t_{CYK}	Counter mode 4
CI0/CI1, CTRL0/CTRL1 cycle time	t_{CYC4}			250	KHz	Counter mode 4
External Interrupts and Reset						
NMI high, low level width	t_{WNIH}, t_{WNIL}	10			μs	
INTE0 high, low level width	t_{WI0H}, t_{WI0L}	3			t_{CYK}	
INTE1 high, low level width	t_{WI1H}, t_{WI1L}	3			t_{CYK}	
INTE2 high, low level width	t_{WI2H}, t_{WI2L}	3			t_{CYK}	
$\overline{\text{RESET}}$ high, low level width	t_{WRSH}, t_{WRSL}	10			μs	
V_{DD} rise, fall time	t_{RVD}, t_{FVD}	200			μs	

Notes:

(1) The internal clock (f_{CLK}) equals the oscillation clock (f_{XX}) divided by 2 or 8 as determined by bit 5 of the STBC. In this table, f_{XX} = 12 MHz and $f_{CLK} = f_{XX}/2$.

(2) During refresh operation, the $\overline{\text{WR}}$ signal falls to low level 1/2 clock cycle later than if there is no refresh.

(3) When accessing data from pseudostatic DRAMs (e.g. μPD4168) with the falling edge of the $\overline{\text{WR}}$ signal, the data setup time is t_{SODWF} instead of t_{SODWR}.

(4) Hold time is measured with C_L = 100 pF and R_L = 2 kΩ load, and includes the period necessary to guarantee V_{OH} and V_{OL}.

(5) I/O interface mode transmit data at a data rate of 750 kb/s.

(6) I/O interface mode receive data, internal clock, at a data rate of 750 kb/s.

(7) In the I/O interface mode this is the optional external clock for received data at a maximum rate of 1 MB/s.

Oscillator Characteristics

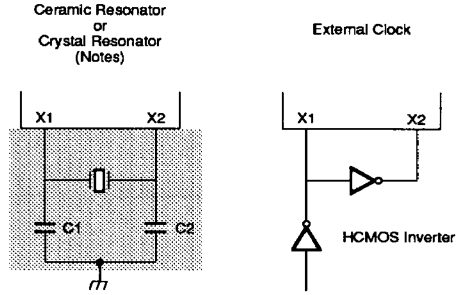
T_A = −10 to 70°C; V_{DD} = +5.0 V ±10%; V_{SS} = AV_{SS} = 0 V; 4 V ≤ AV_{REF} ≤ V_{DD}

Oscillator	Parameter	Symbol	Min	Max	Unit
Ceramic resonator or crystal resonator	Oscillation frequency	f_{XX}	4	12	MHz
External clock	X1 input frequency	f_X	4	12	MHz
	X1 input rise, fall time	t_{XR}, t_{XF}	0	30	ns
	X1 input high-low-level width	t_{WXH}, t_{WXL}	30	130	ns

Recommended Ceramic Resonators (μPD78310/312A)

			External Capacitance (pF)	
Manufacturer	Part No.	Frequency (MHz)	C1	C2
Murata Mfg. Co., Ltd.	CSA12.OMT	12.0	30	30
	CST12.OMT	12.0	Included	Included

Recommended Circuits



Notes:

1. When using a crystal resonator, the following external capacitor is recommended:
   C1 = C2 = 15pF
2. Oscillator circuit must be located as close as possible to the X1 and X2 pins.
3. To prevent noise from affecting operation, avoid locating other signal lines within the shaded area.

83YL-6838A

Timing Dependent on t_{CYK}

Symbol	Formula	Min/Max	Unit
t_{SAL}	1.5T − 100	Min	ns
t_{DAR}	2T − 100		
t_{DAID}	(3.5 + n) T − 170	Max	ns
t_{DLID}	(2 + n) T − 100		
t_{DRID}	(1.5 +n) T − 70		
t_{DLR}	0.5T − 20	Min	ns
t_{DRL}	T − 50		
t_{DRA}	0.5T − 30		
t_{WRL}	(1.5 + n) T − 50		
t_{WLH}	T − 40		
t_{DAW}	2T − 100		
t_{DLOD}	0.5T + 110	Max	ns
t_{DLW}	0.5T − 20 (normal operation)	Min	ns
	T − 50 (during refresh mode)		
t_{SODWR}	(1.5 +n) T − 100		
t_{SODWF}	0.5T − 50		
t_{DWL}	T − 50		
t_{WWL}	(1.5 + n) − 50		

Notes:

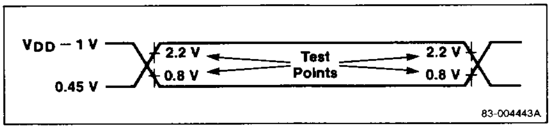
(1) n is the number of additional wait cycles specified by the MM register.

(2) T = t_{CYK} = $1/f_{CLK}$ = $2/f_{XX}$. f_{CLK} is the internal sytem clock frequency.

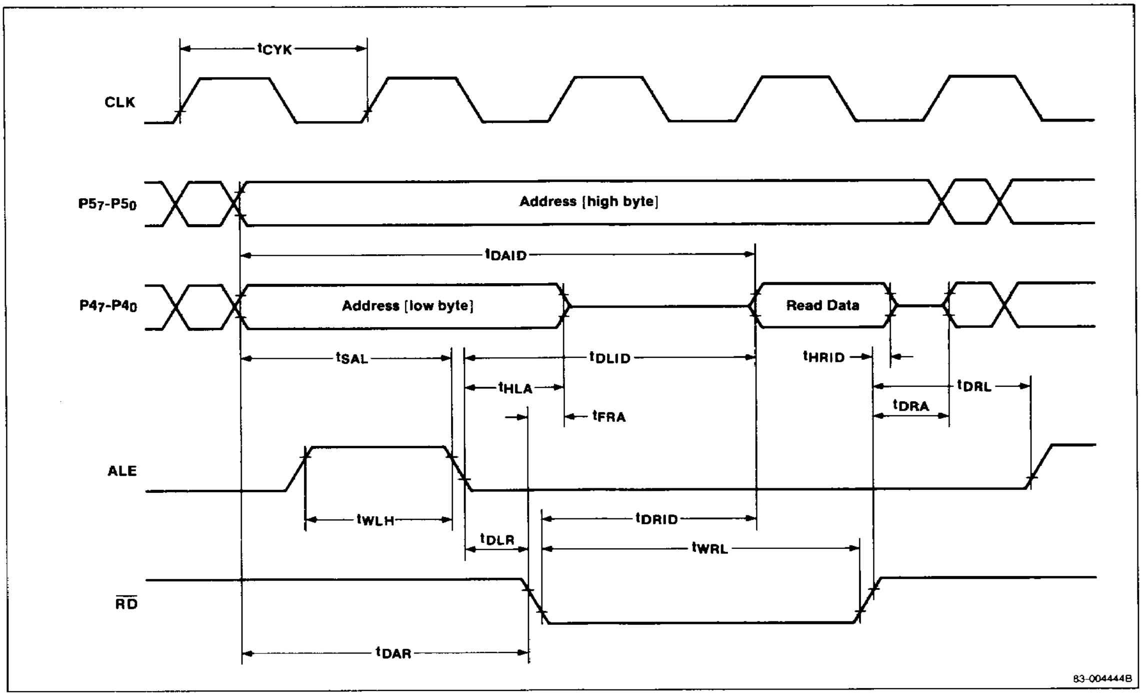
(3) Any parameter not included in this table is not dependent on f_{CLK}.

Timing Waveforms

AC Timing Test Points

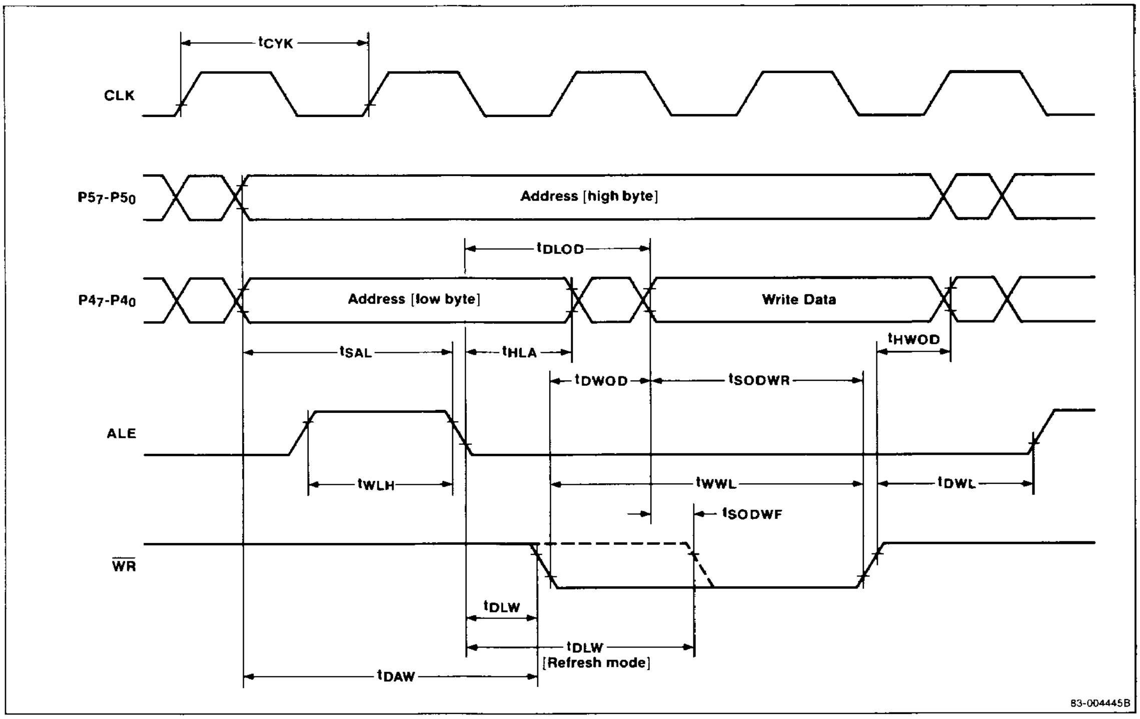


Read Operation



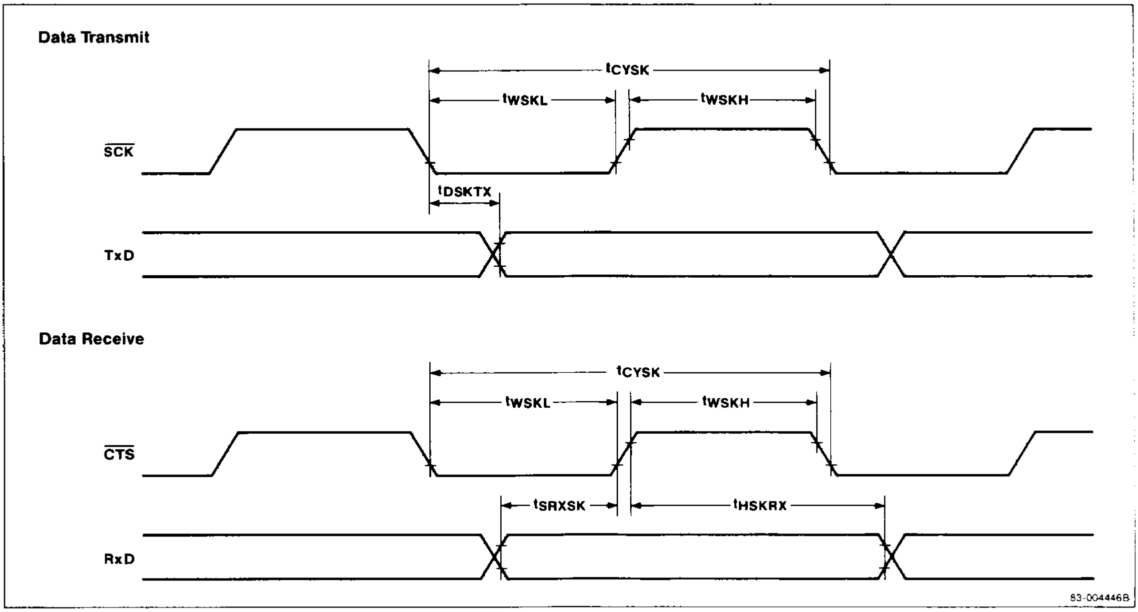
Timing Waveforms (cont)

Write Operation

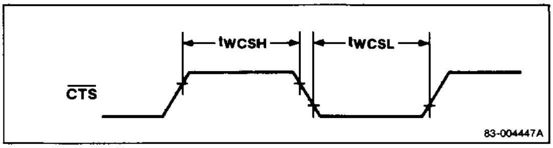


Timing Waveforms (cont)

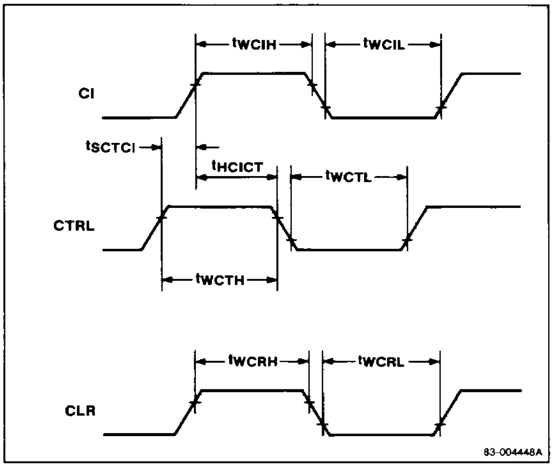
Serial Port, I/O Interface Mode



Serial Port, Asynchronous Mode Send Enable Input Timing

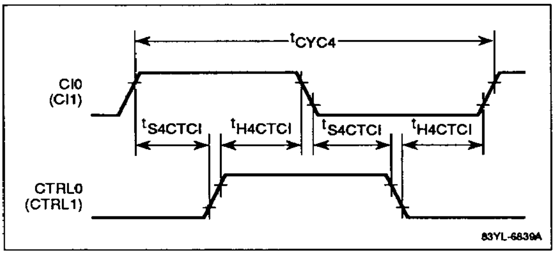


Counter Operation (Mode 3)

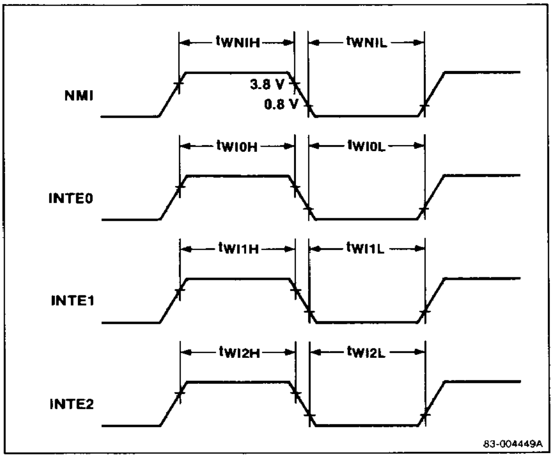


Timing Waveforms (cont)

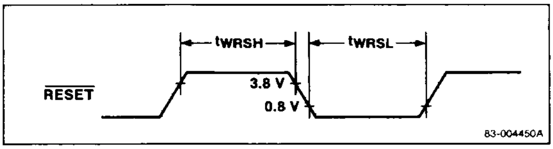
Count Timing Specification (Mode 4)



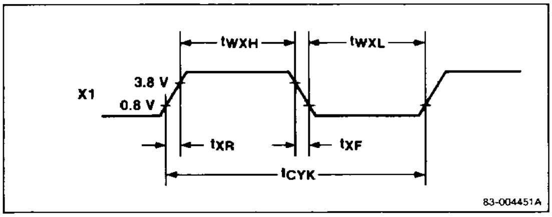
External Interrupts



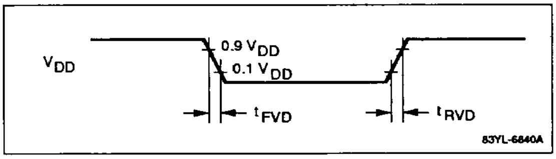
External Reset



External Clock



Data Retention Timing



PROM PROGRAMMING

The PROM in the µPD78P312A is an OTP or UVE EPROM with an 8,192 x 8-bit configuration. The pins listed in the table below are used to program the PROM.

When used in the normal operation mode, 5V ±10% is applied to the V_{DD} and V_{PP} pins. A voltage higher than V_{DD} should not be applied to other pins.

The programming characteristics of the µPD78P312A are identical to those of the µPD27C256A.

Pin	Function
V_{PP}	High voltage input (write/verify mode), high-level input (read mode)
PROG	High voltage input (write/verify mode, read mode)
A_0-A_7	Address input (lower 8 bits)
A_8-A_{12}	Address input (upper 8 bits)
D_0-D_7	Data input (write mode), data output (verify mode)
$\overline{CE}$	Program pulse input
$\overline{OE}$	Output enable input
V_{DD}	Power supply pin

Notes:

(1) Mask the window of the UVE EPROM version to protect the PROM from being erased accidentally.

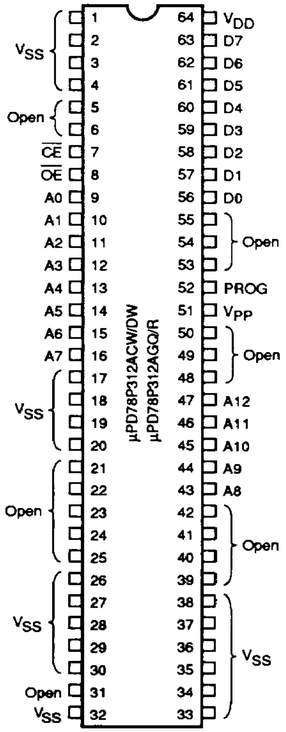
(2) The OTP EPROM version cannot be erased by ultraviolet rays because it does not have a window.

Programming Setup

Programming socket adaptors PA-78P312CW/GF/GQ/L are used to configure the µPD78P312A to fit a standard PROM socket. Set the PROM programmer to program the 27C256A. If the PROM programmer is an older model, check that the programming voltage does not exceed 12.5 volts.

Pin Functions, PROM Programming Mode

64-Pin Shrink DIP and QUIP, Plastic and Ceramic



Notes:

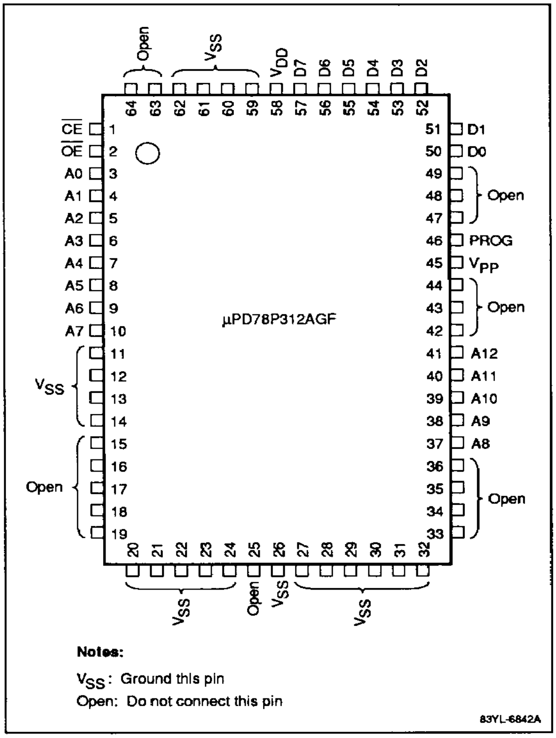
[1] V_{SS}: Ground this pin.

[2] Open: Do not connect this pin.

83YL-6841A

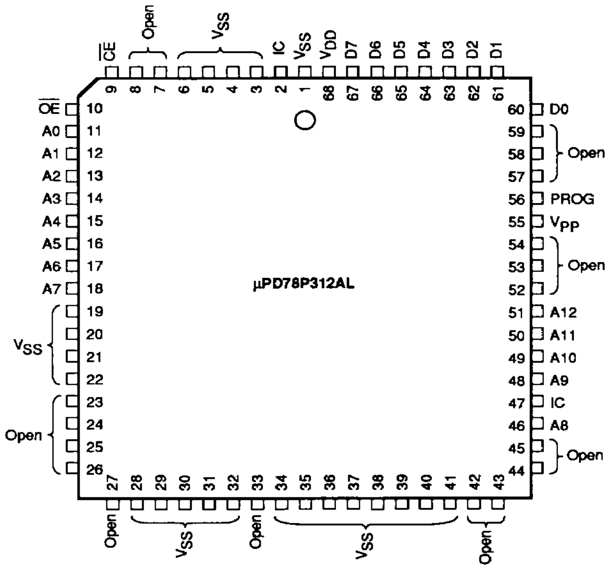
Pin Functions, PROM Programming Mode (cont)

64-Pin Plastic QFP (bent leads)



Pin Functions, PROM Programming Mode (cont)

68-Pin PLCC



Notes:

[1] V_{SS}: Ground this pin.

[2] Open: Do not connect this pin.

83YL-6843B

PROM Programming Mode

When + 6 V is applied to the V_{DD} pin and + 12.5 V is applied to the PROG pin and V_{PP} pin, the μPD78P312A enters the program write/verify mode. Operation in this mode is determined by the setting of $\overline{\text{CE}}$ and $\overline{\text{OE}}$ pins as indicated in the table below.

Mode	$\overline{\text{CE}}$	$\overline{\text{OE}}$	V_{PP}	V_{DD}	PROG
Write	L	H	+ 12.5 V	+6V	+ 12.5 V
Verify	H	L			
Program inhibit	H	H			
Read (Note 2)	L/H	L	+ 5 V	+ 5 V	+ 12.5 V
Read (Note 3)	L/H	H			

Notes:

(1) When + 12.5 V is applied to V_{PP} and + 6 V is applied to V_{DD}, both $\overline{\text{CE}}$ and $\overline{\text{OE}}$ must not be set to the low level (L) simultaneously.

(2) Data is output from the D_0-D_7 pins.

(3) D_0-D_7 are high impedance.

Recommended Conditions for Unused Pins

Table 3 describes how to set unused pins when programming the PROM.

Table 3. Recommended Conditions for Unused Pins

Pin	Recommended Connection
$P0_0$-$P0_3$	Connect to V_{SS}
$P0_4$, $P0_5$	Open
$P2_0$-$P2_3$	Connect to V_{SS}
$P2_5$-$P2_7$, $\overline{\text{RFSH}}$	Open
$P3_0$-$P3_3$, X1	Connect to V_{SS}
X2	Open
AN0-AN3, AV_{REF}, AV_{SS}	Connect to V_{SS}
$P3_4$-$P3_7$, $P5_5$-$P5_7$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, ALE	Open

PROM Write Procedure

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

(1) Set the pins not used for programming as indicated in table 3, and supply + 6 V to the V_{DD} pin, and + 12.5 V to the V_{PP} and PROG pins.

(2) Provide the initial address.

(3) Provide write data.

(4) Provide a 1 ms program pulse (active low) to the $\overline{\text{CE}}$ pin.

(5) Use the verify mode to test the data. If the data has been written, proceed to (7), if not, repeat steps (3) to (5). If the data cannot be correctly written in 25 attempts, go to step (6).

(6) Classify the PROM as defective and cease write operation.

(7) Provide write data and supply program pulse (for additional writing) for 3 ms times the number of repeats performed between steps (3) to (5).

(8) Increment the address.

(9) Repeat steps (3) to (8) until the last address is reached.

PROM Read Procedure

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

(1) Set the unused pins as indicated in table 3.

(2) Supply + 5 V to the V_{DD} pin and V_{PP} pin, and + 12.5 V to the PROG pin.

(3) Input the address of the data to be read to the A_0 to A_{12} pins.

(4) Put an active low pulse of at least 1 μs on the $\overline{\text{OE}}$ pin.

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

Erasure

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

Typically, data is erased by 254-nm ultraviolet rays. A minimum lighting level of 15 W s/cm^2 (ultraviolet ray intensity x exposure time) is required to completely erase written data. Erasure by an ultraviolet lamp rated at 12,000 μW/cm^2 takes approximately 15 to 20 minutes. Remove any filter on the lamp and place the device within 2.5 cm of the lamp tubes.

DC Programming Characteristics

T_A = 25 ±5°C; V_{IP} = 12.0 ±0.5 V; V_{SS} = 0 V

Parameter	Symbol	Symbol (Note)	Min	Typ	Max	Unit	Condition
High-level input voltage	V_{IH}	V_{IH}	2.2		V_{DDP} + 0.3	V	
Low-level input voltage	V_{IL}	V_{IL}	−0.3		0.8	V	
Input leakage current	V_{LIP}	V_{LI}			10	µA	$0 \le V_I \le V_{DDP}$
High-level output voltage	V_{OH}	V_{OH}	V_{DD}−1			V	I_{OH} = −1.0 mA
Low-level output voltage	V_{OL}	V_{OL}			0.45	V	I_{OL} = 2.0 mA
Output leakage current	I_{LO}	—			10	µA	$0 \le V_o \le V_{DDP}$, $\overline{OE} = V_{IH}$
PROG pin high voltage input current	I_{IP}	—			±10	µA	
V_{DDP} power supply voltage	V_{DDP}	V_{DD}	5.75	6.0	6.25	V	Program memory write mode
			4.5	5.0	5.5	V	Program memory read mode
V_{PP} power supply voltage	V_{PP}	V_{PP}	12.2	12.5	12.8	V	Program memory write mode
				$V_{PP} = V_{DDP}$		V	Program memory read mode
V_{DDP} power supply current	I_{DD}	I_{DD}		10	30	mA	Program memory write mode
				10	30	mA	Program memory read mode $\overline{CE} = V_{IL}$, $V_I = V_{IH}$
V_{PP} power supply current	I_{PP}	I_{PP}		10	30	mA	Program memory write mode $\overline{CE} = V_{IL}$, $\overline{OE} = V_{IH}$
				1	100	µA	Program memory read mode

Notes:

(1) Corresponding symbols for the µPD27C256A

AC Programming Characteristics

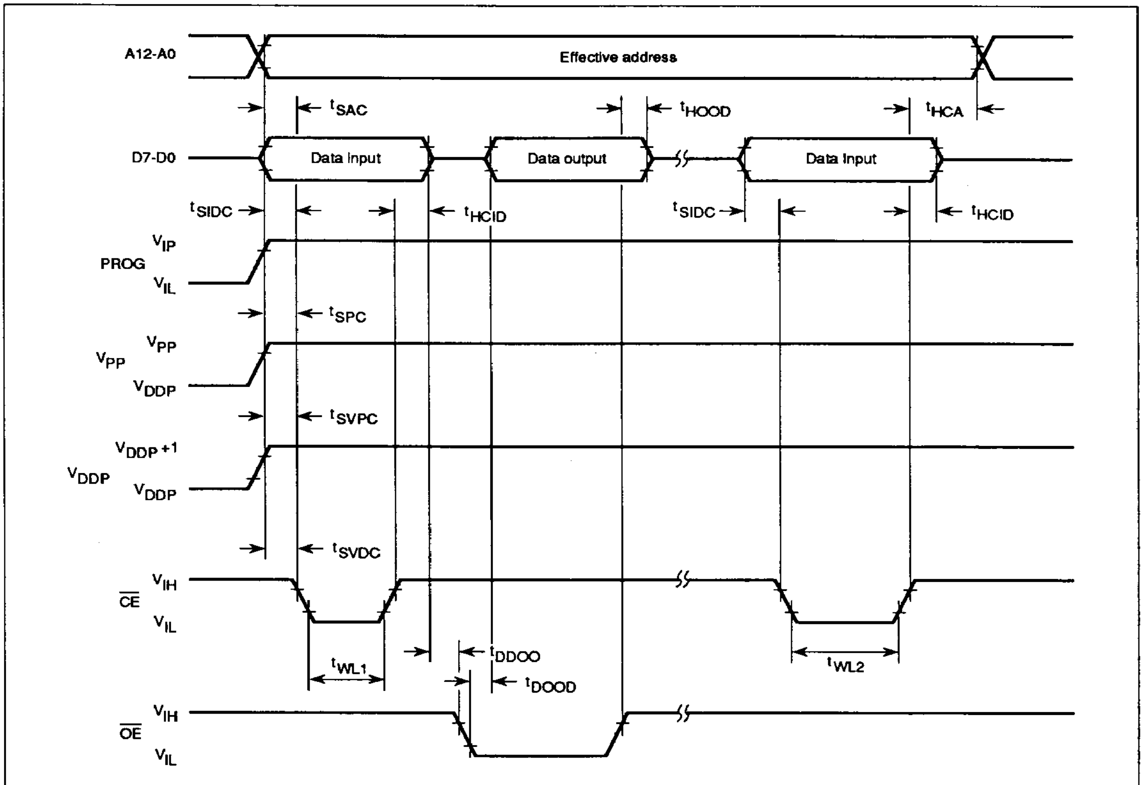
T_A = 25 ±5°C; V_{IP} = 12.0 ±0.5 V; V_{SS} = 0 V

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

Notes:

(1) Corresponding symbols for the µPD27C256A

PROM Write Mode Timing



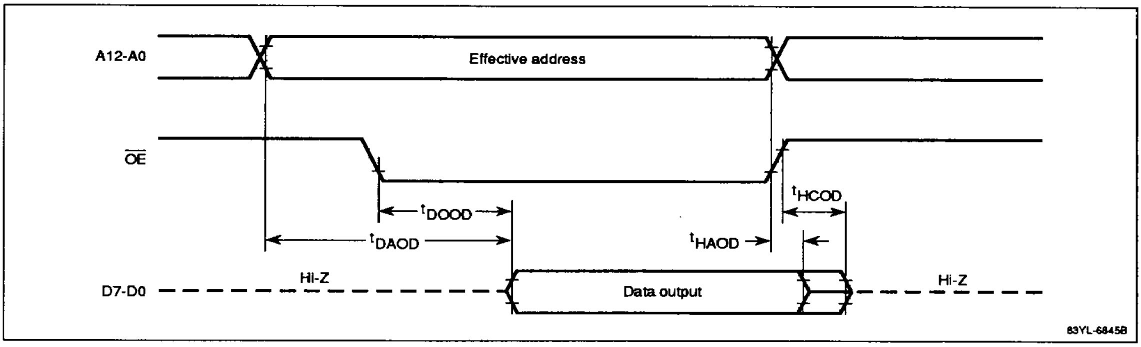
Notes:

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

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

83YL-6844B

PROM Read Mode Timing



83YL-6845B

INSTRUCTION SET

The instruction set for the μPD7831xA has 8- and 16-bit arithmetic instructions including: a 16 x 16-bit unsigned multiply with a 32-bit product; a 32 by 16-bit unsigned divide with a 32-bit quotient and a 16-bit remainder. The instruction set also executes an 8-bit and a 16-bit shift and rotate by count, 1-and 8-bit logic, and 1-, 2-, and 3-byte call instructions. String manipulation instructions are also included.

Branch

There are four addressing modes for unconditional branching. Branch instructions exist to test single bits in the program status word, the 16-bit accumulator, the special function registers, and internal RAM. The instruction set also includes multiple register PUSH and POP instructions.

Addressing

On-chip RAM locations FE20H through FEFFH can be addressed by "saddr" addressing, in which the machine code specifies the address by its low-order byte only. This mode is also used to address the first 32 special function registers, addresses FF00H through FF1FH.

Timing

Access to on-chip ROM requires one state per byte, on-chip RAM two states per byte, and external memory four states per byte minimum.

The States column of the instruction set listing indicates the number of states required to execute an instruction after it has been fetched. In "saddr" addressing, the number after the slash is applicable when addressing special function registers FF00H through FF1FH. In conditional branch instructions, the number in parentheses is applicable when the branch is not taken. String instructions are interruptable, and the number in parentheses applies if the instruction has been interrupted during its execution.

The Idle States column indicates the number of states during which the CPU does not use the peripheral bus. They are therefore available for fetching succeeding instructions. If sufficient idle states are available, prefetching will continue until the buffer is full, so as many as three bytes can be pre-fetched in this manner. If the instructions are stored in external memory, a minimum of four states is required for each byte. Idle states from each instruction are used in multiples of four, and any states in excess of multiples of four are lost.

Symbols

Symbols designations, and codes used in the instruction set are explained in the following tables.

In addition to the general register designations (such as $P_2P_1P_0$, $Q_2Q_1Q_0$ and $R_2R_1R_0$), the following designations appear in the Operation Code column.

$B_2B_1B_0$	Bit number (bit = 0 through 7) in single-bit instructions
$N_2N_1N_0$	Number of bits (n = 0 through 7) in shift and rotate instructions
$N_2N_1N_0$	Register bank number (n = 0 through 7) in BRKCS and SEL instructions

Symbols

Symbol	Meaning
r	R0-R15
r1	R0-R7
r2	C, B
rp	RP0-RP7*
rp1	RP0-RP7*
rp2	DE, HL, VP, UP
sfr	Special function register, 8 bits
sfrp	Special function register, 16 bits
post	RP0, RP1, RP2, RP3, RP4, RP5/PSW, RP6, RP7. Bits set to 1 indicate register pairs to be pushed/popped to/from the stack. RP5 pushed/popped by PUSH/POP: SP is stack pointer. PSW pushed/popped by PUSHU/POPU: RP5 is stack pointer
mem	Register indirect: [DE], [HL], [DE+], [HL+], [DE–], [HL–], [VP], [UP] Base index mode: [DE + A], [HL + A], [DE + B], [HL + B], [VP + DE], [VP + HL] Base Mode: [DE + byte], [HL + byte], [VP + byte], [UP + byte], [SP + byte] Index mode: Word [A], word [B], word [DE], word [HL]
saddr	FE20H-FF1FH: immediate byte addresses one byte in RAM, or label
saddrp	FE20H-FF1FH: immediate byte (bit 0 = 0) addresses one word in RAM or label
#word	16 bits of immediate data or label
#byte	8 bits of immediate data or label
jdisp	8-bit two's complement displacement (immediate data)
f_0-f_{10}	Eleven bits of immediate data corresponding to addr11
t_0-t_4	Five bits of immediate data corresponding to addr5

*rp and rp1 refer to the same register pairs, but generate different machine code.

Symbols

Symbol	Meaning
.bit	3 bits of immediate data (bit position in byte), or label
n	3 bits of immediate data
!addr16	16-bit absolute address specified by an immediate address or label
$addr16	Relative branch address {(PC) + jdisp} or label
addr16	16-bit address
!addr11	11-bit immediate address or label
addr11	0800H to 0FFFH; 0800H + 11-bit immediate address
addr5	Pointer into call table, 0040H-007EH: or 8040H-807EH, 5 bit immediate data or label
A	A register (8-bit accumulator)
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
R0-R15	Register 0-15
AX	Register pair AX (16-bit accumulator)
BC	Register pair BC
VP	Register pair VP
UP	Register pair UP (user stack pointer)
DE	Register pair DE
HL	Register pair HL
RP0-RP7	Register pair 0-7
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
P/V	Parity/overflow flag
S	Sign flag
SUB	Subtract flag
TPF	Table position flag
RBS	Register bank select flag
RSS	Register set select flag
IE	Interrupt enable flag
EOS	End of software interrupt flag
STBC	Standby control register
WDM	Watchdog timer mode register

Symbol	Meaning
/	Logical complement
( )	Contents of the location whose address is within ( ); (+) and (–) indicate that the address is incremented or decremented after it is used.
(( ))	Contents of the memory location defined by the contents of the location defined by the quantity within the (( )).
XXH	Hexadecimal number
XH, XL	High-order 8 bits and low-order 8 bits of X

Flag Indicators

Symbol	Meaning
(blank)	No change
0	Cleared to 0
1	Set to 1
X	Set or cleared according to result
P	Parity of result
V	Arithmetic overflow
U	Undefined
R	Restored from saved PSW

Execution Times of Memory Reference Instructions: Number of Processor States

		Memory Reference Mode			
Instruction		Register Indirect	Base Index	Base	Index
MOV	A, mem	5	6	6	6
	mem,A				
XCH	A, mem	7	8	8	8
	mem,A				
ADD, ADDC, SUB, SUBC, AND, OR, XOR	A, mem	6	7	7	7
	mem,A	7	8	8	8
CMP	A, mem	6	7	7	7
	mem,A				

Memory Addressing Modes

mod	1 0110	1 0111	0 0110	0 1010
mem	Register Indirect	Base Index	Base	Index
0 0 0	[DE+]*	[DE+A]	[DE+byte]	word [DE]
0 0 1	[HL+]*	[HL+A]	[SP+byte]	word [A]
0 1 0	[DE–]*	[DE+B]	[HL+byte]	word [HL]
0 1 1	[HL–]*	[HL+B]	[UP+byte]	word [B]
1 0 0	[DE]*	[VP+DE]	[VP+byte]	—
1 0 1	[HL]*	[VP+HL]	—	—
1 1 0	[VP]	—	—	—
1 1 1	[UP]	—	—	—

*1-byte instructions: defined by special opcode and mem only.

General Register Designation r, r1

R_3	R_2	R_1	R_0	Reg		
0	0	0	0	R0		
0	0	0	1	R1		
0	0	1	0	R2		
0	0	1	1	R3	r1	
0	1	0	0	R4		
0	1	0	1	R5		
0	1	1	0	R6		
0	1	1	1	R7		r
1	0	0	0	R8		
1	0	0	1	R9		
1	0	1	0	R10		
1	0	1	1	R11		
1	1	0	0	R12		
1	1	0	1	R13		
1	1	1	0	R14		
1	1	1	1	R15		

r2

C	Reg
0	C
1	B

rp

P_2	P_1	P_0	Reg Pair
0	0	0	RP0
0	0	1	RP1
0	1	0	RP2
0	1	1	RP3
1	0	0	RP4
1	0	1	RP5
1	1	0	RP6
1	1	1	RP7

rp1

Q_2	Q_1	Q_0	Reg Pair
0	0	0	RP0
0	0	1	RP4
0	1	0	RP1
0	1	1	RP5
1	0	0	RP2
1	0	1	RP6
1	1	0	RP3
1	1	1	RP7

rp2

S_1	S_0	Reg Pair
0	0	VP
0	1	UP
1	0	DE
1	1	HL

Instructions

Mnemonic	Operand	Operation	States	Idle States	Bytes	Flags						Operation Code (Bits 7-0) Bytes B1 thru B5
						S	Z	AC	P/V	SUB	CY	
Data Transfer												
MOV	r1,#byte	r1 ← byte	3	3	2							1 0 1 1 1 R_2 R_1 R_0 Data
	saddr,#byte	(saddr) ← byte	3/4	0	3							0 0 1 1 1 0 1 0 Saddr-offset Data
	sfr,#byte	sfr ← byte	4	0	3							0 0 1 0 1 0 1 1 Sfr-offset Data
	r,r1	r ← r1	3	3	2							0 0 1 0 0 1 0 0 R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,r1	A ← r1	3	3	1							1 1 0 1 0 R_2 R_1 R_0
	A,saddr	A ← (saddr)	3/4	1	2							0 0 1 0 0 0 0 0 Saddr-offset
	saddr,A	(saddr) ← A	3/4	0	2							0 0 1 0 0 0 1 0 Saddr-offset
	saddr,saddr	(saddr) ← (saddr)	4/6	0	3							0 0 1 1 1 0 0 0 Saddr-offset Saddr-offset
	A,sfr	A ← sfr	4	1	2							0 0 0 1 0 0 0 0 Sfr-offset
	sfr,A	sfr ← A	4	0	2							0 0 0 1 0 0 1 0 Sfr-offset
	A,mem*	A ← (mem)	5	3	1							0 1 0 1 1 mem
	A,mem	A ← (mem)	5-6	3-4	2-4							0 0 0 mod 0 mem 0 0 0 0 Low offset High offset
	mem,A*	(mem) ← A	5	2	1							0 1 0 1 0 mem
	mem,A	(mem) ← A	5-6	2	2-4							0 0 0 mod 1 mem 0 0 0 0 Low offset High offset

*When mem is [DE], [HL], [DE+], [DE−], [HL+], or [HL−]

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
Data Transfer (cont)												
MOV (cont)	A,[saddrp]	A ← ((saddrp))	5/6	1	2							0 0 0 1 1 0 0 0
												Saddr-offset
	[saddrp],A	((saddrp)) ← A	4/5	0	2							0 0 0 1 1 0 0 1
												Saddr-offset
	A,!addr16	A ← (addr16)	5	3	4							0 0 0 0 1 0 0 1
												1 1 1 1 0 0 0 0
												Low addr
												High addr
	!addr16,A	(addr16) ← A	4	2	4							0 0 0 0 1 0 0 1
												1 1 1 1 0 0 0 1
												Low addr
												High addr
	PSWL,#byte	PSWL ← byte	4	0	3	X	X	X	X	X	X	0 0 1 0 1 0 1 1
												1 1 1 1 1 1 1 0
												Data
	PSWH,#byte	PSWH ← byte	4	0	3							0 0 1 0 1 0 1 1
												1 1 1 1 1 1 1 1
												Data
	PSWL,A	PSWL ← A	4	0	2	X	X	X	X	X	X	0 0 0 1 0 0 1 0
												1 1 1 1 1 1 1 0
	PSWH,A	PSWH ← A	4	0	2							0 0 0 1 0 0 1 0
												1 1 1 1 1 1 1 1
	A,PSWL	A ← PSWL	4	1	2							0 0 0 1 0 0 0 0
												1 1 1 1 1 1 1 0
	A,PSWH	A ← PSWH	4	1	2							0 0 0 1 0 0 0 0
												1 1 1 1 1 1 1 1
XCH	A,r1	A ↔ r1	4	4	1							1 1 0 1 1 R_2 R_1 R_0
	r,r1	r ↔ r1	4	4	2							0 0 1 0 0 1 0 1
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,mem	A ↔ (mem)	7-8	3-4	2-4							0 0 0 mod
												0 mem 0 1 0 0
												Low offset
												High offset
	A,saddr	A ↔ (saddr)	4/6	0	2							0 0 1 0 0 0 0 1
												Saddr-offset
	A,sfr	A ↔ sfr	8	3	3							0 0 0 0 0 0 0 1
												0 0 1 0 0 0 0 1
												Sfr-offset

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	**Operand**	**Operation**	**States**	**States**	**Bytes**	**S**	**Z**	**AC**	**P/V**	**SUB**	**CY**	**Bytes B1 thru B5**
Data Transfer (cont)												
XCH (cont)	A,[saddrp]	A ←→ ((saddrp))	6/7	0	2							0 0 1 0 0 0 1 1 Saddr-offset
	saddr,saddr	(saddr) ←→ (saddr)	8/12	0	3							0 0 1 1 1 0 0 1 Saddr-offset Saddr-offset
MOVW	rp1,#word	rp1 ← word	3	3	3							0 1 1 0 0 Q_2 Q_1 Q_0 Low byte High byte
	saddrp,#word	(saddrp) ← word	3/4	0	4							0 0 0 0 1 1 0 0 Saddr-offset Low byte High byte
	sfrp,#word	sfrp ← word	4	0	4							0 0 0 0 1 0 1 1 Sfr-offset Low byte High byte
	rp,rp1	rp ← rp1	3	3	2							0 0 1 0 0 1 0 0 P_2 P_1 P_0 0 1 Q_2 Q_1 Q_0
	AX,saddrp	AX ← (saddrp)	3/4	1	2							0 0 0 1 1 1 0 0 Saddr-offset
	saddrp,AX	(saddrp) ← AX	3/4	0	2							0 0 0 1 1 0 1 0 Saddr-offset
	saddrp,saddrp	(saddrp) ← (saddrp)	4/6	0	3							0 0 1 1 1 1 0 0 Saddr-offset Saddr-offset
	AX,sfrp	AX ← sfrp	4	1	2							0 0 0 1 0 0 0 1 Sfr-offset
	sfrp,AX	sfrp ← AX	4	0	2							0 0 0 1 0 0 1 1 Sfr-offset
	rpl,!addr16	rpl ← (addr16)	10	6	4							0 0 0 0 1 0 0 1 1 0 0 0 0 Q_2 Q_1 Q_0 Low Addr High Addr
	!addr16,rpl	(addr16) ← rpl	8	4	4							0 0 0 0 1 0 0 1 1 0 0 1 0 Q_2 Q_1 Q_0 Low Addr High Addr

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
Data Transfer (cont)												
XCHW	AX,saddrp	AX ⟷ (saddrp)	4/6	0	2							0 0 0 1 1 0 1 1
												Saddr-offset
	AX,sfrp	AX ⟷ sfrp	9	3	3							0 0 0 0 0 0 0 1
												0 0 0 1 1 0 1 1
												Sfr-offset
	saddrp, saddrp	(saddrp) ⟷ (saddrp)	8/12	0	3							0 0 1 0 1 0 1 0
												Saddr-offset
												Saddr-offset
	rp,rp1	rp ⟷ rp1	5	5	2							0 0 1 0 0 1 0 1
												P_2 P_1 P_0 0 1 Q_2 Q_1 Q_0
8-Bit Operation												
ADD	A,#byte	A, CY ← A + byte	3	3	2	X	X	X	V	0	X	1 0 1 0 1 0 0 0
												Data
	saddr,#byte	(saddr), CY ← (saddr) + byte	5/7	0	3	X	X	X	V	0	X	0 1 1 0 1 0 0 0
												Saddr-offset
												Data
	sfr,#byte	sfr, CY ← sfr + byte	10	3	4	X	X	X	V	0	X	0 0 0 0 0 0 0 1
												0 1 1 0 1 0 0 0
												Sfr-offset
												Data
	r,r1	r, CY ← r + r1	3	3	2	X	X	X	V	0	X	1 0 0 0 1 0 0 0
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A, CY ← A + (saddr)	3/4	1	2	X	X	X	V	0	X	1 0 0 1 1 0 0 0
												Saddr-offset
	A,sfr	A, CY ← A + sfr	7	4	3	X	X	X	V	0	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)	6/9	0	3	X	X	X	V	0	X	0 1 1 1 1 0 0 0
												Saddr-offset
												Saddr-offset
	A,mem	A, CY ← A + (mem)	6-7	4-5	2-4	X	X	X	V	0	X	0 0 0 mod
												0 mem 1 0 0 0
												Low offset
												High offset
	mem,A	(mem), CY ← (mem) + A	7-8	2-3	2-4	X	X	X	V	0	X	0 0 0 mod
												1 mem 1 0 0 0
												Low offset
												High offset

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
8-Bit Operation (cont)												
ADDC	A,#byte	A, CY ← A + byte + CY	3	3	2	X	X	X	V	0	X	1 0 1 0 1 0 0 1
												Data
	saddr,#byte	(saddr), CY ← (saddr)	5/7	0	3	X	X	X	V	0	X	0 1 1 0 1 0 0 1
		+ byte + CY										Saddr-offset
												Data
	sfr,#byte	sfr, CY ← sfr + byte + CY	10	3	4	X	X	X	V	0	X	0 0 0 0 0 0 0 1
												0 1 1 0 1 0 0 1
												Sfr-offset
												Data
	r,r1	r, CY ← r + r1 + CY	3	3	2	X	X	X	V	0	X	1 0 0 0 1 0 0 1
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A, CY ← A + (saddr) + CY	3/4	1	2	X	X	X	V	0	X	1 0 0 1 1 0 0 1
												Saddr-offset
	A,sfr	A, CY ← A + sfr + CY	7	4	3	X	X	X	V	0	X	0 0 0 0 0 0 0 1
												1 0 0 1 1 0 0 1
												Sfr-offset
	saddr,saddr	(saddr), CY ← (saddr)	6/9	0	3	X	X	X	V	0	X	0 1 1 1 1 0 0 1
		+ (saddr) + CY										Saddr-offset
												Saddr-offset
	A,mem	A, CY ← A + (mem) + CY	6-7	4-5	2-4	X	X	X	V	0	X	0 0 0 mod
												0 mem 1 0 0 1
												Low offset
												High offset
	mem,A	(mem), CY ← (mem)	7-8	2-3	2-4	X	X	X	V	0	X	0 0 0 mod
		+ A + CY										1 mem 1 0 0 1
												Low offset
												High offset
SUB	A,#byte	A, CY ← A − byte	3	3	2	X	X	X	V	1	X	1 0 1 0 1 0 1 0
												Data
	saddr,#byte	(saddr), CY ← (saddr) − byte	5/7	0	3	X	X	X	V	1	X	0 1 1 0 1 0 1 0
												Saddr-offset
												Data
	sfr,#byte	sfr, CY ← sfr − byte	10	3	4	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												0 1 1 0 1 0 1 0
												Sfr-offset
												Data
	r,r1	r, CY ← r − r1	3	3	2	X	X	X	V	1	X	1 0 0 0 1 0 1 0
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A, CY ← A − (saddr)	3/4	1	2	X	X	X	V	1	X	1 0 0 1 1 0 1 0
												Saddr-offset

7

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
8-Bit Operation (cont)												
SUB (cont)	A,sfr	A, CY ← A − sfr	7	4	3	X	X	X	V	1	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)	6/9	0	3	X	X	X	V	1	X	0 1 1 1 1 0 1 0
												Saddr-offset
												Saddr-offset
	A,mem	A, CY ← A − (mem)	6-7	4-5	2-4	X	X	X	V	1	X	0 0 0 mod
												0 mem 1 0 1 0
												Low offset
												High offset
	mem, A	(mem), CY ← (mem) − A	7-8	2-3	2-4	X	X	X	V	1	X	0 0 0 mod
												1 mem 1 0 1 0
												Low offset
												High offset
SUBC	A,#byte	A, CY ← A − byte − CY	3	3	2	X	X	X	V	1	X	1 0 1 0 1 0 1 1
												Data
	saddr,#byte	(saddr), CY ← (saddr) − byte − CY	5/7	0	3	X	X	X	V	1	X	0 1 1 0 1 0 1 1
												Saddr-offset
												Data
	sfr,#byte	sfr, CY ← sfr − byte − CY	10	3	4	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												0 1 1 0 1 0 1 1
												Sfr-offset
												Data
	r,r1	r, CY ← r − r1 − CY	3	3	2	X	X	X	V	1	X	1 0 0 0 1 0 1 1
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A, CY ← A − (saddr) − CY	3/4	1	2	X	X	X	V	1	X	1 0 0 1 1 0 1 1
												Saddr-offset
	A,sfr	A, CY ← A − sfr − CY	7	4	3	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												1 0 0 1 1 0 1 1
												Sfr-offset
	saddr,saddr	(saddr), CY ← (saddr) − (saddr) − CY	6/9	0	3	X	X	X	V	1	X	0 1 1 1 1 0 1 1
												Saddr-offset
												Saddr-offset
	A,mem	A, CY ← A − (mem) − CY	6-7	4-5	2-4	X	X	X	V	1	X	0 0 0 mod
												0 mem 1 0 1 1
												Low offset
												High offset

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
8-Bit Operation (cont)												
SUBC	mem, A	(mem), CY ← (mem)	7-8	2-3	2-4	X	X	X	V	1	X	0 0 0 mod
(cont)		− A − CY										1 mem 1 0 1 1
												Low offset
												High offset
AND	A,#byte	A ← A ∧ byte	3	3	2	X	X		P	0		1 0 1 0 1 1 0 0
												Data
	saddr,#byte	(saddr) ← (saddr) ∧ byte	5/7	0	3	X	X		P	0		0 1 1 0 1 1 0 0
												Saddr-offset
												Data
	sfr,#byte	sfr ← sfr ∧ byte	10	3	4	X	X		P	0		0 0 0 0 0 0 0 1
												0 1 1 0 1 1 0 0
												Sfr-offset
												Data
	r,r1	r ← r ∧ r1	3	3	2	X	X		P	0		1 0 0 0 1 1 0 0
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A ← A ∧ (saddr)	3/4	1	2	X	X		P	0		1 0 0 1 1 1 0 0
												Saddr-offset
	A,sfr	A ← A ∧ sfr	7	4	3	X	X		P	0		0 0 0 0 0 0 0 1
												1 0 0 1 1 1 0 0
												Sfr-offset
	saddr,saddr	(saddr) ← (saddr) ∧ (saddr)	6/9	0	3	X	X		P	0		0 1 1 1 1 1 0 0
												Saddr-offset
												Saddr-offset
	A,mem	A ← A ∧ (mem)	6-7	4-5	2-4	X	X		P	0		0 0 0 mod
												0 mem 1 1 0 0
												Low offset
												High offset
	mem,A	(mem) ← (mem) ∧ A	7-8	2-3	2-4	X	X		P	0		0 0 0 mod
												1 mem 1 1 0 0
												Low offset
												High offset
OR	A,#byte	A ← A V byte	3	3	2	X	X		P	0		1 0 1 0 1 1 1 0
												Data
	saddr,#byte	(saddr) ← (saddr) V byte	5/7	0	3	X	X		P	0		0 1 1 0 1 1 1 0
												Saddr-offset
												Data

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	**Operand**	**Operation**	**States**	**States**	**Bytes**	**S**	**Z**	**AC**	**P/V**	**SUB**	**CY**	**Bytes B1 thru B5**
8-Bit Operation (cont)												
OR (cont)	sfr,#byte	sfr ← sfr V byte	10	3	4	X	X		P	0		0 0 0 0 0 0 0 1
												0 1 1 0 1 1 1 0
												Sfr-offset
												Data
	r,r1	r ← r V r1	3	3	2	X	X		P	0		1 0 0 0 1 1 1 0
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A ← A V (saddr)	3/4	1	2	X	X		P	0		1 0 0 1 1 1 1 0
												Saddr-offset
	A,sfr	A ← A V sfr	7	4	3	X	X		P	0		0 0 0 0 0 0 0 1
												1 0 0 1 1 1 1 0
												Sfr-offset
	saddr,saddr	(saddr) ← (saddr) V (saddr)	6/9	0	3	X	X		P	0		0 1 1 1 1 1 1 0
												Saddr-offset
												Saddr-offset
	A,mem	A ← A V (mem)	6-7	4-5	2-4	X	X		P	0		0 0 0 mod
												0 mem 1 1 1 0
												Low offset
												High offset
	mem,A	(mem) ← (mem) V A	7-8	2-3	2-4	X	X		P	0		0 0 0 mod
												1 mem 1 1 1 0
												Low offset
												High offset
XOR	A,#byte	A ← ⊻ byte	3	3	2	X	X		P	0		1 0 1 0 1 1 0 1
												Data
	saddr,#byte	(saddr) ← (saddr) ⊻ byte	5/7	0	3	X	X		P	0		0 1 1 0 1 1 0 1
												Saddr-offset
												Data
	sfr,#byte	sfr ← sfr ⊻ byte	10	3	4	X	X		P	0		0 0 0 0 0 0 0 1
												0 1 1 0 1 1 0 1
												Sfr-offset
												Data
	r,r1	r ← r ⊻ r1	3	3	2	X	X		P	0		1 0 0 0 1 1 0 1
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A ← A ⊻ (saddr)	3/4	1	2	X	X		P	0		1 0 0 1 1 1 0 1
												Saddr-offset
	A,sfr	A ← A ⊻ sfr	7	4	3	X	X		P	0		0 0 0 0 0 0 0 1
												1 0 0 1 1 1 0 1
												Sfr-offset

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
8-Bit Operation (cont)												
XOR (cont)	saddr,saddr	(saddr) ← (saddr) ⊻ (saddr)	6/9	0	3	X	X		P	0		0 1 1 1 1 1 0 1
												Saddr-offset
												Saddr-offset
	A,mem	A ← A ⊻ (mem)	6-7	4-5	2-4	X	X		P	0		0 0 0 mod
												0 mem 1 1 0 1
												Low offset
												High offset
	mem,A	(mem) ← (mem) ⊻ A	7-8	2-3	2-4	X	X		P	0		0 0 0 mod
												1 mem 1 1 0 1
												Low offset
												High offset
CMP	A,#byte	A − byte	3	3	2	X	X	X	V	1	X	1 0 1 0 1 1 1 1
												Data
	saddr,#byte	(saddr) − byte	5/7	1	3	X	X	X	V	1	X	0 1 1 0 1 1 1 1
												Saddr-offset
												Data
	sfr,#byte	sfr − byte	10	4	4	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												0 1 1 0 1 1 1 1
												Sfr-offset
												Data
	r,r1	r − r1	3	3	2	X	X	X	V	1	X	1 0 0 0 1 1 1 1
												R_3 R_2 R_1 R_0 0 R_2 R_1 R_0
	A,saddr	A − (saddr)	3/4	1	2	X	X	X	V	1	X	1 0 0 1 1 1 1 1
												Saddr-offset
	A,sfr	A − sfr	7	4	3	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												1 0 0 1 1 1 1 1
												Sfr-offset
	saddr,saddr	(saddr) − (saddr)	6/8	1	3	X	X	X	V	1	X	0 1 1 1 1 1 1 1
												Saddr-offset
												Saddr-offset
	A,mem	A − (mem)	6-7	4-5	2-4	X	X	X	V	1	X	0 0 0 mod
												0 mem 1 1 1 1
												Low offset
												High offset
	mem,A	(mem) − A	6-7	3-4	2-4	X	X	X	V	1	X	0 0 0 mod
												1 mem 1 1 1 1
												Low offset
												High offset

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
16-Bit Operation												
ADDW	AX,#word	AX, CY ← AX + word	4	4	3	X	X	X	V	0	X	0 0 1 0 1 1 0 1 Low byte High byte
	saddrp,#word	(saddrp), CY ← (saddrp) + word	5/7	0	4	X	X	X	V	0	X	0 0 0 0 1 1 0 1 Saddr-offset Low byte High byte
	sfrp,#word	sfrp, CY ← sfrp + word	10	3	5	X	X	X	V	0	X	0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 Sfr-offset Low byte High byte
	rp,rp1	rp, CY ← rp + rp1	4	4	2	X	X	X	V	0	X	1 0 0 0 1 0 0 0 P_2 P_1 P_0 0 1 Q_2 Q_1 Q_0
	AX,saddrp	AX, CY ← AX + (saddrp)	4/5	2	2	X	X	X	V	0	X	0 0 0 1 1 1 0 1 Saddr-offset
	AX,sfrp	AX, CY ← AX + sfrp	8	5	3	X	X	X	V	0	X	0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1 Sft-offset
	saddrp,saddrp	(saddrp), CY ← (saddrp) + (saddrp)	6/9	0	3	X	X	X	V	0	X	0 0 1 1 1 1 0 1 Saddr-offset Saddr-offset
SUBW	AX,#word	AX, CY ← AX − word	4	3	3	X	X	X	V	1	X	0 0 1 0 1 1 1 0 Low byte High byte
	saddrp,#word	(saddrp), CY ← (saddrp) − word	5/7	0	4	X	X	X	V	1	X	0 0 0 0 1 1 1 0 Saddr-offset Low byte High byte

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	**Operand**	**Operation**	**States**	**States**	**Bytes**	**S**	**Z**	**AC**	**P/V**	**SUB**	**CY**	**Bytes B1 thru B5**
16-Bit Operation (cont)												
SUBW (cont)	sfrp,#word	sfrp, CY ← sfrp − word	10	3	5	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												0 0 0 0 1 1 1 0
												Sfr-offset
												Low byte
												High byte
	rp,rp1	rp, CY ← rp − rp1	4	4	2	X	X	X	V	1	X	1 0 0 0 1 0 1 0
												P_2 P_1 P_0 0 1 Q_2 Q_1 Q_0
	AX,saddrp	AX, CY ← AX − (saddrp)	4/5	2	2	X	X	X	V	1	X	0 0 0 1 1 1 1 0
												Saddr-offset
	AX,sfrp	AX, CY ← AX − sfrp	8	5	3	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												0 0 0 1 1 1 1 0
												Sfr-offset
	saddrp,saddrp	(saddrp), CY ← (saddrp) − (saddrp)	6/9	0	3	X	X	X	V	1	X	0 0 1 1 1 1 1 0
												Saddr-offset
												Saddr-offset
CMPW	AX,#word	AX − word	4	3	3	X	X	X	V	1	X	0 0 1 0 1 1 1 1
												Low byte
												High byte
	saddrp,#word	(saddrp) − word	4/5	1	4	X	X	X	V	1	X	0 0 0 0 1 1 1 1
												Saddr-offset
												Low byte
												High byte
	sfrp,#word	sfrp − word	8	4	5	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												0 0 0 0 1 1 1 1
												Sfr-offset
												Low byte
												High byte
	rp,rp1	rp − rp1	4	4	2	X	X	X	V	1	X	1 0 0 0 1 1 1 1
												P_2 P_1 P_0 0 1 Q_2 Q_1 Q_0
	AX,saddrp	AX − (saddrp)	4/5	1	2	X	X	X	V	1	X	0 0 0 1 1 1 1 1
												Saddr-offset
	AX,sfrp	AX − sfrp	8	4	3	X	X	X	V	1	X	0 0 0 0 0 0 0 1
												0 0 0 1 1 1 1 1
												Sfr-offset
	saddrp,saddrp	(saddrp) − (saddrp)	5/7	1	3	X	X	X	V	1	X	0 0 1 1 1 1 1 1
												Saddr-offset
												Saddr-offset

Instructions (cont)

Mnemonic	Operand	Operation	States	Idle States	Bytes	S	Z	AC	P/V	SUB	CY	Operation Code (Bits 7-0) Bytes B1 thru B5
Multiplication/Division												
MULU	r1	AX ← A x r1	18	18	2							0 0 0 0 0 1 0 1 0 0 0 0 1 R_2 R_1 R_0
DIVUW	r1	AX (Quotient), r1 (Remainder) ← AX ÷ r1	26	26	2							0 0 0 0 0 1 0 1 0 0 0 1 1 R_2 R_1 R_0
MULUW	rp1	AX (High-order 16 bits), rp1 (Low-order 16 bits) ← AX x rp1	27	27	2							0 0 0 0 0 1 0 1 0 0 1 0 1 Q_2 Q_1 Q_0
DIVUX	rp1	AXDE (Quotient), rp1 (Remainder) ← AXDE ÷ rp1	50	50	2							0 0 0 0 0 1 0 1 1 1 1 0 1 Q_2 Q_1 Q_0
Increment/Decrement												
INC	r1	r1 ← r1 + 1	3	3	1	X	X	X	V	0		1 1 0 0 0 R_2 R_1 R_0
	saddr	(saddr) ← (saddr) + 1	4/6	0	2	X	X	X	V	0		0 0 1 0 0 1 1 0 Saddr-offset
DEC	r1	r1 ← r1 − 1	3	3	1	X	X	X	V	1		1 1 0 0 1 R_2 R_1 R_0
	saddr	(saddr) ← (saddr) − 1	4/6	0	2	X	X	X	V	1		0 0 1 0 0 1 1 1 Saddr-offset
INCW	rp2	rp2 ← rp2 + 1	3	3	1							0 1 0 0 0 1 S_1 S_0
	saddrp	(saddrp) ← (saddrp) + 1	6/8	2	3							0 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 Saddr-offset
DECW	rp2	rp2 ← rp2 − 1	3	3	1							0 1 0 0 1 1 S_1 S_0
	saddrp	(saddrp) ← (saddrp) − 1	6/8	2	3							0 0 0 0 0 1 1 1 1 1 1 0 1 0 0 1 Saddr-offset
Shift and Rotate												
ROR	r1,n	(CY,r1_7 ← r1_0, r1_{m-1} ← r1_m) x n	4+3n	4+3n	2				P	0	X	0 0 1 1 0 0 0 0 0 1 N_2 N_1 N_0 R_2 R_1 R_0
ROL	r1,n	(CY, r1_0 ← r1_7, r1_{m+1} ← r1_m) x n	4+3n	4+3n	2				P	0	X	0 0 1 1 0 0 0 1 0 1 N_2 N_1 N_0 R_2 R_1 R_0
RORC	r1,n	(CY ← r1_0, r1_7 ← CY, r1_{m-1} ← r1_m) x n	4+3n	4+3n	2				P	0	X	0 0 1 1 0 0 0 0 0 0 N_2 N_1 N_0 R_2 R_1 R_0
ROLC	r1,n	(CY ← r1_7, r1_0 ← CY, r1_{m+1} ← r1_m) x n	4+3n	4+3n	2				P	0	X	0 0 1 1 0 0 0 1 0 0 N_2 N_1 N_0 R_2 R_1 R_0

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
Shift and Rotate (cont)												
SHR	r1,n	$(CY \leftarrow r1_0, r1_7 \leftarrow 0,$ $r1_{m-1} \leftarrow r1_m)$ x n	4+3n	4+3n	2	X	X	0	P	0	X	0 0 1 1 0 0 0 0 1 0 N_2 N_1 N_0 R_2 R_1 R_0
SHL	r1,n	$(CY \leftarrow r1_7, r1_0 \leftarrow 0,$ $r1_{m+1} \leftarrow r1_m)$ x n	4+3n	4+3n	2	X	X	0	P	0	X	0 0 1 1 0 0 0 1 1 0 N_2 N_1 N_0 R_2 R_1 R_0
SHRW	rp1,n	$(CY \leftarrow rp1_0, rp1_{15} \leftarrow 0,$ $rp1_{m-1} \leftarrow rp1_m)$ x n	4+3n	4+3n	2	X	X	0	P	0	X	0 0 1 1 0 0 0 0 1 1 N_2 N_1 N_0 Q_2 Q_1 Q_0
SHLW	rp1,n	$(CY \leftarrow rp1_{15}, rp1_0 \leftarrow 0,$ $rp1_{m+1} \leftarrow rp1_m)$ x n	4+3n	4+3n	2	X	X	0	P	0	X	0 0 1 1 0 0 0 1 1 1 N_2 N_1 N_0 Q_2 Q_1 Q_0
ROR4	[rp1]	$A_{3-0} \leftarrow (rp1)_{3-0},$ $(rp1)_{7-4} \leftarrow A_{3-0},$ $(rp1)_{3-0} \leftarrow (rp1)_{7-4}$	7	3	2							0 0 0 0 0 1 0 1 1 0 0 0 1 Q_2 Q_1 Q_0
ROL4	[rp1]	$A_{3-0} \leftarrow (rp1)_{7-4},$ $(rp1)_{3-0} \leftarrow A_{3-0},$ $(rp1)_{7-4} \leftarrow (rp1)_{3-0}$	7	3	2							0 0 0 0 0 1 0 1 1 0 0 1 1 Q_2 Q_1 Q_0
BCD Adjustment												
ADJ4		Decimal adjust accumulator	3	3	1	X	X	X	P		X	0 0 0 0 0 1 0 0
Bit Manipulation												
MOV1	CY,saddr.bit	CY ← (saddr.bit)	6/7	4	3						X	0 0 0 0 1 0 0 0 0 0 0 0 0 B_2 B_1 B_0 Saddr-offset
	CY,sfr.bit	CY ← sfr.bit	7	4	3						X	0 0 0 0 1 0 0 0 0 0 0 0 1 B_2 B_1 B_0 Sfr-offset
	CY,A.bit	CY ← A.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 0 0 0 1 B_2 B_1 B_0
	CY,X.bit	CY ← X.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 0 0 0 0 B_2 B_1 B_0
	CY,PSWL.bit	CY ← PSWH.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 0 0 0 1 B_2 B_1 B_0
	CY,PSWL.bit	CY ← PSWL.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 0 0 0 0 B_2 B_1 B_0
	saddr.bit,CY	(saddr.bit) ← CY	7/8	3	3							0 0 0 0 1 0 0 0 0 0 0 1 0 B_2 B_1 B_0 Saddr-offset
	sfr.bit,CY	sfr.bit ← CY	8	3	3							0 0 0 0 1 0 0 0 0 0 0 1 1 B_2 B_1 B_0 Sfr-offset
	A.bit,CY	A.bit ← CY	8	8	2							0 0 0 0 0 0 1 1 0 0 0 1 1 B_2 B_1 B_0

Instructions (cont)

Mnemonic	Operand	Operation	States	Idle States	Bytes	Flags						Operation Code (Bits 7-0) Bytes B1 thru B5
						S	Z	AC	P/V	SUB	CY	
Bit Manipulation (cont)												
MOV1 (cont)	X.bit,CY	X.bit ← CY	8	8	2							0 0 0 0 0 0 1 1 0 0 0 1 0 B_2 B_1 B_0
	PSWH.bit,CY	PSW_H.bit ← CY	9	9	2							0 0 0 0 0 0 1 0 0 0 0 1 1 B_2 B_1 B_0
	PSWL.bit,CY	PSW_L.bit ← CY	9	9	2	X	X	X	X	X		0 0 0 0 0 0 1 0 0 0 0 1 0 B_2 B_1 B_0
AND1	CY,saddr.bit	CY ← CY ∧ (saddr.bit)	6/7	4	3						X	0 0 0 0 1 0 0 0 0 0 1 0 0 B_2 B_1 B_0 Saddr-offset
	CY,/saddr.bit	CY ← CY ∧ $(\overline{\text{saddr.bit}})$	6/7	4	3						X	0 0 0 0 1 0 0 0 0 0 1 1 0 B_2 B_1 B_0 Saddr-offset
	CY,sfr.bit	CY ← CY ∧ sfr.bit	7	4	3						X	0 0 0 0 1 0 0 0 0 0 1 0 1 B_2 B_1 B_0 Sfr-offset
	CY,/sfr.bit	CY ← CY ∧ $\overline{\text{sfr.bit}}$	7	4	3						X	0 0 0 0 1 0 0 0 0 0 1 1 1 B_2 B_1 B_0 Sfr-offset
	CY,A.bit	CY ← CY ∧ A.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 0 1 0 1 B_2 B_1 B_0
	CY,/A.bit	CY ← CY ∧ $\overline{\text{A.bit}}$	6	6	2						X	0 0 0 0 0 0 1 1 0 0 1 1 1 B_2 B_1 B_0
	CY,X.bit	CY ← CY ∧ X.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 0 1 0 0 B_2 B_1 B_0
	CY,/X.bit	CY ← CY ∧ $\overline{\text{X.bit}}$	6	6	2						X	0 0 0 0 0 0 1 1 0 0 1 1 0 B_2 B_1 B_0
	CY,PSWH.bit	CY ← CY ∧ PSWH.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 0 1 0 1 B_2 B_1 B_0
	CY,/PSWH.bit	CY ← CY ∧ $\overline{\text{PSWH.bit}}$	6	6	2						X	0 0 0 0 0 0 1 0 0 0 1 1 1 B_2 B_1 B_0
	CY,PSWL.bit	CY ← CY ∧ PSWL.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 0 1 0 0 B_2 B_1 B_0
	CY,/PSWL.bit	CY ← CY ∧ $\overline{\text{PSWL.bit}}$	6	6	2						X	0 0 0 0 0 0 1 0 0 0 1 1 0 B_2 B_1 B_0
OR1	CY,saddr.bit	CY ← CY V (saddr.bit)	6/7	4	3						X	0 0 0 0 1 0 0 0 0 1 0 0 0 B_2 B_1 B_0 Saddr-offset
	CY,/saddr.bit	CY ← CY V $(\overline{\text{saddr.bit}})$	6/7	4	3						X	0 0 0 0 1 0 0 0 0 1 0 1 0 B_2 B_1 B_0 Saddr-offset

Instructions (cont)

Mnemonic	Operand	Operation	States	Idle States	Bytes	Flags S	Z	AC	P/V	SUB	CY	Operation Code (Bits 7-0) Bytes B1 thru B5
Bit Manipulation (cont)												
OR1 (cont)	CY,sfr.bit	CY ← CY V sfr.bit	7	4	3						X	0 0 0 0 1 0 0 0 0 1 0 0 1 B_2 B_1 B_0 Sfr-offset
	CY,/sfr.bit	CY ← CY V $\overline{\text{sfr.bit}}$	7	4	3						X	0 0 0 0 1 0 0 0 0 1 0 1 1 B_2 B_1 B_0 Sfr-offset
	CY,A.bit	CY ← CY V A.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 1 0 0 1 B_2 B_1 B_0
	CY,/A.bit	CY ← CY V $\overline{\text{A.bit}}$	6	6	2						X	0 0 0 0 0 0 1 1 0 1 0 1 1 B_2 B_1 B_0
	CY,X.bit	CY ← CY V X.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 1 0 0 0 B_2 B_1 B_0
	CY,/X.bit	CY ← CY V $\overline{\text{X.bit}}$	6	6	2						X	0 0 0 0 0 0 1 1 0 1 0 1 0 B_2 B_1 B_0
	CY,PSWH.bit	CY ← CY V PSWH.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 1 0 0 1 B_2 B_1 B_0
	CY,/PSWH.bit	CY ← CY V $\overline{\text{PSWH.bit}}$	6	6	2						X	0 0 0 0 0 0 1 0 0 1 0 1 1 B_2 B_1 B_0
	CY,PSWL.bit	CY ← CY V PSWL.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 1 0 0 0 B_2 B_1 B_0
	CY,/PSWL.bit	CY ← CY V $\overline{\text{PSWL.bit}}$	6	6	2						X	0 0 0 0 0 0 1 0 0 1 0 1 0 B_2 B_1 B_0
XOR1	CY,saddr.bit	CY ← CY ⊻ (saddr.bit)	6/7	4	3						X	0 0 0 0 1 0 0 0 0 1 1 0 0 B_2 B_1 B_0 Saddr-offset
	CY,sfr.bit	CY ← CY ⊻ sfr.bit	7	4	3						X	0 0 0 0 1 0 0 0 0 1 1 0 1 B_2 B_1 B_0 Sfr-offset
	CY,A.bit	CY ← CY ⊻ A.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 1 1 0 1 B_2 B_1 B_0
	CY,X.bit	CY ← CY ⊻ X.bit	6	6	2						X	0 0 0 0 0 0 1 1 0 1 1 0 0 B_2 B_1 B_0
	CY,PSWH.bit	CY ← CY ⊻ PSWH.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 1 1 0 1 B_2 B_1 B_0
	CY,PSWL.bit	CY ← CY ⊻ PSWL.bit	6	6	2						X	0 0 0 0 0 0 1 0 0 1 1 0 0 B_2 B_1 B_0

Instructions (cont)

Mnemonic	Operand	Operation	States	Idle States	Bytes	Flags						Operation Code (Bits 7-0) Bytes B1 thru B5
						S	Z	AC	P/V	SUB	CY	
Bit Manipulation (cont)												
SET1	saddr.bit	(saddr.bit) ← 1	5/7	1	2							1 0 1 1 0 B_2 B_1 B_0
												Saddr-offset
	sfr.bit	sfr.bit ← 1	8	2	3							0 0 0 0 1 0 0 0
												1 0 0 0 1 B_2 B_1 B_0
												Sfr-offset
	A.bit	A.bit ← 1	7	7	2							0 0 0 0 0 0 1 1
												1 0 0 0 1 B_2 B_1 B_0
	X.bit	X.bit ← 1	7	7	2							0 0 0 0 0 0 1 1
												1 0 0 0 0 B_2 B_1 B_0
	PSWH.bit	PSWH.bit ← 1	8	8	2							0 0 0 0 0 0 1 0
												1 0 0 0 1 B_2 B_1 B_0
	PSWL.bit	PSWL.bit ← 1	8	8	2	X	X	X	X	X	X	0 0 0 0 0 0 1 0
												1 0 0 0 0 B_2 B_1 B_0
CLR1	saddr.bit	(saddr.bit) ← 0	5/7	1	2							1 0 1 0 0 B_2 B_1 B_0
												Saddr-offset
	sfr.bit	sfr.bit ← 0	8	2	3							0 0 0 0 1 0 0 0
												1 0 0 1 1 B_2 B_1 B_0
												Sfr-offset
	A.bit	A.bit ← 0	7	7	2							0 0 0 0 0 0 1 1
												1 0 0 1 1 B_2 B_1 B_0
	X.bit	X.bit ← 0	7	7	2							0 0 0 0 0 0 1 1
												1 0 0 1 0 B_2 B_1 B_0
	PSWH.bit	PSWH.bit ← 0	8	8	2							0 0 0 0 0 0 1 0
												1 0 0 1 1 B_2 B_1 B_0
	PSWL.bit	PSWL.bit ← 0	8	8	2	X	X	X	X	X	X	0 0 0 0 0 0 1 0
												1 0 0 1 0 B_2 B_1 B_0
NOT1	saddr.bit	(saddr.bit) ← $\overline{\text{(saddr.bit)}}$	6/8	2	3							0 0 0 0 1 0 0 0
												0 1 1 1 0 B_2 B_1 B_0
												Saddr-offset
	sfr.bit	sfr.bit ← $\overline{\text{sfr.bit}}$	8	2	3							0 0 0 0 1 0 0 0
												0 1 1 1 1 B_2 B_1 B_0
												Sfr-offset
	A.bit	A.bit ← $\overline{\text{A.bit}}$	7	7	2							0 0 0 0 0 0 1 1
												0 1 1 1 1 B_2 B_1 B_0
	X.bit	X.bit ← $\overline{\text{X.bit}}$	7	7	2							0 0 0 0 0 0 1 1
												0 1 1 1 0 B_2 B_1 B_0
	PSWH.bit	PSWH.bit ← $\overline{\text{PSWH.bit}}$	8	8	2							0 0 0 0 0 0 1 0
												0 1 1 1 1 B_2 B_1 B_0
	PSWL.bit	PSWL.bit ← $\overline{\text{PSWL.bit}}$	8	8	2	X	X	X	X	X	X	0 0 0 0 0 0 1 0
												0 1 1 1 0 B_2 B_1 B_0

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
Bit Manipulation (cont)												
SET1	CY	CY ← 1	3	3	1						1	0 1 0 0 0 0 0 1
CLR1	CY	CY ← 0	3	3	1						0	0 1 0 0 0 0 0 0
NOT1	CY	CY ← $\overline{CY}$	3	3	1						X	0 1 0 0 0 0 1 0
Call/Return												
CALL	!addr16	$(SP - 1) \leftarrow (PC + 3)_H$, $(SP - 2) \leftarrow (PC + 3)_L$, PC ← addr16, SP ← SP − 2	8	0	3							0 0 1 0 1 0 0 0 Low addr High addr
CALLF	!addr11	$(SP - 1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, PC ← addr11, SP ← SP − 2	8	0	2							1 0 0 1 0 f_{10} f_9 f_8 f_7 f_6 f_5 f_4 f_3 f_2 f_1 f_0
CALLT	[addr5]	$(SP - 1) \leftarrow (PC + 1)_H$, $(SP - 2) \leftarrow (PC + 1)_L$, $PC_H \leftarrow$ (TPF x 8000H + addr5 + 1), $PC_L \leftarrow$ (TPF x 8000H + addr5), SP ← SP − 2	13	0	1							1 1 1 t_4 t_3 t_2 t_1 t_0
CALL	rp1	$(SP - 1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, $PC_H \leftarrow rp1_H$, $PC_L \leftarrow rp1_L$, SP ← SP − 2	9	0	2							0 0 0 0 0 1 0 1 0 1 0 1 1 Q_2 Q_1 Q_0
	[rp1]	$(SP - 1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, $PC_H \leftarrow (rp1)_H$, $PC_L \leftarrow (rp1)_L$, SP ← SP − 2	11	0	2							0 0 0 0 0 1 0 1 0 1 1 1 1 Q_2 Q_1 Q_0
BRK		$(SP - 1) \leftarrow PSW_H$, $(SP - 2) \leftarrow PSW_L$, $(SP - 3) \leftarrow (PC + 1)_H$, $(SP - 4) \leftarrow (PC + 1)_L$, $PC_L \leftarrow$ (003EH), $PC_H \leftarrow$ (003FH), SP ← SP − 4 IE ← 0	20	0	1							0 1 0 1 1 1 1 0
RET		$PC_L \leftarrow$ (SP), $PC_H \leftarrow$ (SP + 1), SP ← SP + 2	8	0	1							0 1 0 1 0 1 1 0
RETI		$PC_L \leftarrow$ (SP), $PC_H \leftarrow$ (SP + 1), $PSW_L \leftarrow$ (SP + 2), $PSW_H \leftarrow$ (SP + 3), SP ← SP + 4, EOS ← 0	14	0	1	R	R	R	R	R	R	0 1 0 1 0 1 1 1
Stack Manipulation												
PUSH	post	$((SP - 1) \leftarrow rpp_H$,* $(SP - 2) \leftarrow rpp_L$, SP ← SP − 2) x n	41+4n	41	2							0 0 1 1 0 1 0 1 Post byte
	PSW	$(SP - 1) \leftarrow PSW_H$, $(SP - 2) \leftarrow PSW_L$, SP ← SP − 2	5	1	1							0 1 0 0 1 0 0 1

*rpp refers to register pairs specified in post byte. n is the number of register pairs specified in post byte.

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
Stack Manipulation (cont)												
PUSHU	post	$((UP-1) \leftarrow rpp_H$,* $(UP-2) \leftarrow rpp_L$, $UP \leftarrow UP-2) \times n$	42+4n	42	2							0 0 1 1 0 1 1 1 Post byte
POP	post	$(rpp_L \leftarrow (SP)$,* $rpp_H \leftarrow (SP+1)$, $SP \leftarrow SP+2) \times n$	41+5n	41+n	2							0 0 1 1 0 1 0 0 Post byte
	PSW	$PSW_L \leftarrow (SP)$, $PSW_H \leftarrow (SP+1)$, $SP \leftarrow SP+2$	6	2	1	R	R	R	R	R	R	0 1 0 0 1 0 0 0
POPU	post	$(rpp_L \leftarrow (UP)$,* $rpp_H \leftarrow (UP+1)$, $UP \leftarrow UP+2) \times n$	42+5n	42+n	2							0 0 1 1 0 1 1 0 Post byte
MOVW	SP,#word	SP ← word	4	0	4							0 0 0 0 1 0 1 1 1 1 1 1 1 1 0 0 Low byte High byte
	SP,AX	SP ← AX	4	0	2							0 0 0 1 0 0 1 1 1 1 1 1 1 1 0 0
	AX,SP	AX ← SP	4	1	2							0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0
INCW	SP	SP ← SP + 1	5	5	2							0 0 0 0 0 1 0 1 1 1 0 0 1 0 0 0
DECW	SP	SP ← SP − 1	5	5	2							0 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1
Unconditional Branch												
BR	!addr16	PC ← addr16	4	0	3							0 0 1 0 1 1 0 0 Low addr High addr
	rp1	$PC_H \leftarrow rp1_H, PC_L \leftarrow rp1_L$	5	0	2							0 0 0 0 0 1 0 1 0 1 0 0 1 Q_2 Q_1 Q_0
	[rp1]	$PC_H \leftarrow (rp1)_H, PC_L \leftarrow (rp1)_L$	8	0	2							0 0 0 0 0 1 0 1 0 1 1 0 1 Q_2 Q_1 Q_0
	$addr16	PC ← addr16	7	0	2							0 0 0 1 0 1 0 0 jdisp
Conditional Branch												
BC or BL**	$addr16	PC ← addr16 if CY = 1	7(3)	0(3)	2							1 0 0 0 0 0 1 1 jdisp
BNC or BNL**	$addr16	PC ← addr16 if CY = 0	7(3)	0(3)	2							1 0 0 0 0 0 1 0 jdisp
BZ or BE**	$addr16	PC ← addr16 if Z = 1	7(3)	0(3)	2							1 0 0 0 0 0 0 1 jdisp

*rpp refers to register pairs specified in post byte. n is the number of register pairs specified in post byte.

**Either of the two mnemonics may be used.

Instructions (cont)

Mnemonic	Operand	Operation	States	Idle States	Bytes	Flags S	Z	AC	P/V	SUB	CY	Operation Code (Bits 7-0) Bytes B1 thru B5
Conditional Branch (cont)												
BNZ or BNE**	$addr16	PC ← addr16 if Z = 0	7(3)	0(3)	2							1 0 0 0 0 0 0 0 jdisp
BV or BPE**	$addr16	PC ← addr16 if P/V = 1	7(3)	0(3)	2							1 0 0 0 0 1 0 1 jdisp
BNV or BPO**	$addr16	PC ← addr16 if P/V = 0	7(3)	0(3)	2							1 0 0 0 0 1 0 0 jdisp
BN	$addr16	PC ← addr16 if S = 1	7(3)	0(3)	2							1 0 0 0 0 1 1 1 jdisp
BP	$addr16	PC ← addr16 if S = 0	7(3)	0(3)	2							1 0 0 0 0 1 1 0 jdisp
BGT	$addr16	PC ← addr16 if (P/V ⊻ S) V Z = 0	9(5)	0(5)	3							0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 jdisp
BGE	$addr16	PC ← addr16 if P/V ⊻ S = 0	9(5)	0(5)	3							0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 1 jdisp
BLT	$addr16	PC ← addr16 if P/V ⊻ S = 1	9(5)	0(5)	3							0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 jdisp
BLE	$addr16	PC ← addr16 if (P/V ⊻ S) V Z = 1	9(5)	0(5)	3							0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 0 jdisp
BH	$addr16	PC ← addr16 if Z V CY = 0	9(5)	0(5)	3							0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 jdisp
BNH	$addr16	PC ← addr16 if Z V CY = 1	9(5)	0(5)	3							0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 jdisp
BT	saddr.bit,$addr16	PC ← addr16 if (saddr.bit) = 1	9(6)/ 10(7)	0(4)	3							0 1 1 1 0 B_2 B_1 B_0 Saddr-offset jdisp
	sfr.bit,$addr16	PC ← addr16 if sfr.bit = 1	11(8)	0(5)	4							0 0 0 0 1 0 0 0 1 0 1 1 1 B_2 B_1 B_0 Sfr-offset jdisp
	A.bit,$addr16	PC ← addr16 if A.bit = 1	10(7)	0(7)	3							0 0 0 0 0 0 1 1 1 0 1 1 1 B_2 B_1 B_0 jdisp

**Either of the two mnemonics may be used.

Instructions (cont)

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
Conditional Branch (cont)												
BT (cont)	X.bit,$addr16	PC ← addr16 if X.bit = 1	10(7)	0(7)	3							0 0 0 0 0 0 1 1 1 0 1 1 0 B_2 B_1 B_0 jdisp
	PSWH.bit,$addr16	PC ← addr16 if PSWH.bit = 1	10(7)	0(7)	3							0 0 0 0 0 0 1 0 1 0 1 1 1 B_2 B_1 B_0 jdisp
	PSWL.bit,$addr16	PC ← addr16 if PSWL.bit = 1	10(7)	0(7)	3							0 0 0 0 0 0 1 0 1 0 1 1 0 B_2 B_1 B_0 jdisp
BF	saddr.bit,$addr16	PC ← addr16 if (saddr.bit) = 0	10(7)/ 11(8)	0(5)	4							0 0 0 0 1 0 0 0 1 0 1 0 0 B_2 B_1 B_0 Saddr-offset jdisp
	sfr.bit,$addr16	PC ← addr16 if sfr.bit = 0	11(8)	0(5)	4							0 0 0 0 1 0 0 0 1 0 1 0 1 B_2 B_1 B_0 Sfr-offset jdisp
	A.bit,$addr16	PC ← addr16 if A.bit = 0	10(7)	0(7)	3							0 0 0 0 0 0 1 1 1 0 1 0 1 B_2 B_1 B_0 jdisp
	X.bit,$addr16	PC ← addr16 if X.bit = 0	10(7)	0(7)	3							0 0 0 0 0 0 1 1 1 0 1 0 0 B_2 B_1 B_0 jdisp
	PSWH.bit,$addr16	PC ← addr16 if PSWH.bit = 0	10(7)	0(7)	3							0 0 0 0 0 0 1 0 1 0 1 0 1 B_2 B_1 B_0 jdisp
	PSWL.bit,$addr16	PC ← addr16 if PSWL.bit = 0	10(7)	0(7)	3							0 0 0 0 0 0 1 0 1 0 1 0 0 B_2 B_1 B_0 jdisp
BTCLR	saddr.bit,$addr16	PC ← addr16 if (saddr.bit) = 1; then reset (saddr.bit)	12(7)/ 14(8)	0(5)	4							0 0 0 0 1 0 0 0 1 1 0 1 0 B_2 B_1 B_0 Saddr-offset jdisp
	sfr.bit,$addr16	PC ← addr16 if sfr.bit = 1; then reset sfr.bit	14(8)	0(5)	4							0 0 0 0 1 0 0 0 1 1 0 1 1 B_2 B_1 B_0 Sfr-offset jdisp
	A.bit,$addr16	PC ← addr16 if A.bit = 1; then reset A.bit	11(7)	0(7)	3							0 0 0 0 0 0 1 1 1 1 0 1 1 B_2 B_1 B_0 jdisp

Instructions (cont)

Mnemonic	Operand	Operation	States	Idle States	Bytes	Flags						Operation Code (Bits 7-0) Bytes B1 thru B5
						S	Z	AC	P/V	SUB	CY	
Conditional Branch (cont)												
BTCLR (cont)	X.bit,$addr16	PC ← addr16 if X.bit = 1; then reset X.bit	11(7)	0(7)	3							0 0 0 0 0 0 1 1 1 1 0 1 0 B_2 B_1 B_0 jdisp
	PSWH.bit,$addr16	PC ← addr16 if PSWH.bit = 1; then reset PSWH.bit	12(7)	0(7)	3							0 0 0 0 0 0 1 0 1 1 0 1 1 B_2 B_1 B_0 jdisp
	PSWL.bit,$addr16	PC ← addr16 if PSWL.bit = 1; then reset PSWL.bit	12(7)	0(7)	3	X	X	X	X	X	X	0 0 0 0 0 0 1 0 1 1 0 1 0 B_2 B_1 B_0 jdisp
BFSET	saddr.bit,$addr16	PC ← addr16 if (saddr.bit) = 0; then set (saddr.bit)	12(7)/ 14(8)	0(5)	4							0 0 0 0 1 0 0 0 1 1 0 0 0 B_2 B_1 B_0 Saddr-offset jdisp
	sfr.bit,$addr16	PC ← addr16 if sfr.bit = 0; then set sfr.bit	14(8)	0(5)	4							0 0 0 0 1 0 0 0 1 1 0 0 1 B_2 B_1 B_0 Sfr-offset jdisp
	A.bit,$addr16	PC ← addr16 if A.bit = 0; then set A.bit	11(7)	0(7)	3							0 0 0 0 0 0 1 1 1 1 0 0 1 B_2 B_1 B_0 jdisp
	X.bit,$addr16	PC ← addr16 if X.bit = 0; then set X.bit	11(7)	0(7)	3							0 0 0 0 0 0 1 1 1 1 0 0 0 B_2 B_1 B_0 jdisp
	PSWH.bit,$addr16	PC ← addr16 if PSWH.bit = 0; then set PSWH.bit	12(7)	0(7)	3							0 0 0 0 0 0 1 0 1 1 0 0 1 B_2 B_1 B_0 jdisp
	PSWL.bit,$addr16	PC ← addr16 if PSWL.bit = 0; then set PSWL.bit	12(7)	0(7)	3	X	X	X	X	X	X	0 0 0 0 0 0 1 0 1 1 0 0 0 B_2 B_1 B_0 jdisp
DBNZ	r2,$addr16	r2 ← r2 − 1; then PC ← addr16 if r2 ≠ 0	8(5)	0(5)	2							0 0 1 1 0 0 1 C_0 jdisp
	saddr,$addr16	(saddr) ← (saddr) − 1; then PC ← addr16 if saddr ≠ 0	9(6)/ 11(8)	0(2)	3							0 0 1 1 1 0 1 1 Saddr-offset jdisp

Instructions (cont)

Mnemonic	Operand	Operation	States	Idle States	Bytes	Flags S	Z	AC	P/V	SUB	CY	Operation Code (Bits 7-0) Bytes B1 thru B5
Context Switch												
BRKCS	RBn	PC_H ↔ R5, PC_L ↔ R4, R7 ← PSW_H, R6 ← PSW_L, RBS2-RBS0 ← n, RSS ← 0, IE ← 0	12	0	2							0 0 0 0 0 1 0 1 1 1 0 1 1 N_2 N_1 N_0
RETCS	!addr16	PC_H ← R5, PC_L ← R4, R5, R4 ← !addr16, PSW_H ← R7, PSW_L ← R6, EOS ← 0	6	0	3	R	R	R	R	R	R	0 0 1 0 1 0 0 1 Low addr High addr
String												
MOVM	[DE+],A	(DE+) ← A, C ← C − 1 End if C = 0	2+7n (4+7n)	2+5n (3+5n)	2							0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0
	[DE−],A	(DE−) ← A, C ← C − 1 End if C = 0	2+7n (4+7n)	2+5n (3+5n)	2							0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0
MOVBK	[DE+],[HL+]	(DE+) ← (HL+), C ← C − 1 End if C = 0	2+10n (4+10n)	2+6n (3+6n)	2							0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0
	[DE−],[HL−]	(DE−) ← (HL−), C ← C − 1 End if C = 0	2+10n (4+10n)	2+6n (3+6n)	2							0 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0
XCHM	[DE+],A	(DE+) ↔ A, C ← C − 1 End if C = 0	2+12n (4+12n)	2+6n (3+6n)	2							0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1
	[DE−],A	(DE−) ↔ A, C ← C − 1 End if C = 0	2+12n (4+12n)	2+6n (3+6n)	2							0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1
XCHBK	[DE+],[HL+]	(DE+) ↔ (HL+), C ← C − 1 End if C = 0	2+15n (4+15n)	2+7n (3+7n)	2							0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1
	[DE−],[HL−]	(DE−) ↔ (HL−), C ← C − 1 End if C = 0	2+15n (4+15n)	2+7n (3+7n)	2							0 0 0 1 0 1 0 1 0 0 1 1 0 0 0 1
CMPME	[DE+],A	(DE+) − A, C ← C − 1 End if C = 0 or Z = 0	2+7n (4+7n)	2+5n (3+5n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0
	[DE−],A	(DE−) − A, C ← C − 1 End if C = 0 or Z = 0	2+7n (4+7n)	2+5n (3+5n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 0
CMPBKE	[DE+],[HL+]	(DE+) − (HL+), C ← C − 1 End if C = 0 or Z = 0	2+10n (4+10n)	2+6n (3+6n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 1 0 0 1 0 0
	[DE−],[HL−]	(DE−) − (HL−), C ← C − 1 End if C = 0 or Z = 0	2+10n (4+10n)	2+6n (3+6n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 1 1 0 1 0 0
CMPMNE	[DE+],A	(DE+) − A, C ← C − 1 End if C = 0 or Z = 1	2+7n (4+7n)	2+5n (3+5n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1
	[DE−],A	(DE−) − A, C ← C − 1 End if C = 0 or Z = 1	2+7n (4+7n)	2+5n (3+5n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1
CMPBKNE	[DE+],[HL+]	(DE+) − (HL+), C ← C − 1 End if C = 0 or Z = 1	2+10n (4+10n)	2+6n (3+6n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 1 0 0 1 0 1
	[DE−],[HL−]	(DE−) − (HL−), C ← C − 1 End if C = 0 or Z = 1	2+10n (4+10n)	2+6n (3+6n)	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1 0 0 1 1 0 1 0 1

Instructions

				Idle		Flags						Operation Code (Bits 7-0)
Mnemonic	Operand	Operation	States	States	Bytes	S	Z	AC	P/V	SUB	CY	Bytes B1 thru B5
String (cont)												
CMPMC	[DE+],A	(DE+) − A, C ← C − 1	2+7n	2+5n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 0	(4+7n)	(3+5n)								0 0 0 0 0 1 1 1
	[DE−],A	(DE−) − A, C ← C − 1	2+7n	2+5n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 0	(4+7n)	(3+5n)								0 0 0 1 0 1 1 1
CMPBKC	[DE+],[HL+]	(DE+) − (HL+), C ← C − 1	2+10n	2+6n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 0	(4+10n)	(3+6n)								0 0 1 0 0 1 1 1
	[DE−],[HL−]	(DE−) − (HL−), C ← C − 1	2+10n	2+6n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 0	(4+10n)	(3+6n)								0 0 1 1 0 1 1 1
CMPMNC	[DE+],A	(DE+) − A, C ← C − 1	2+7n	2+5n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 1	(4+7n)	(3+5n)								0 0 0 0 0 1 1 0
	[DE−],A	(DE−) − A, C ← C − 1	2+7n	2+5n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 1	(4+7n)	(3+5n)								0 0 0 1 0 1 1 0
CMPBKNC	[DE+],[HL+]	(DE+) − (HL+), C ← C − 1	2+10n	2+6n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 1	(4+10n)	(3+6n)								0 0 1 0 0 1 1 0
	[DE−],[HL−]	(DE−) − (HL−), C ← C − 1	2+10n	2+6n	2	X	X	X	V	1	X	0 0 0 1 0 1 0 1
		End if C = 0 or CY = 1	(4+10n)	(3+6n)								0 0 1 1 0 1 1 0
CPU Control												
MOV	STBC,#byte	STBC ← byte	6	1	4							0 0 0 0 1 0 0 1
												0 1 0 0 0 1 0 0
												$\overline{\text{Data}}$
												Data
	WDM,#byte	WDM ← byte	6	1	4							0 0 0 0 1 0 0 1
												0 1 0 0 0 0 1 0
												$\overline{\text{Data}}$
												Data
SWRS		RSS ← $\overline{\text{RSS}}$	3	3	1							0 1 0 0 0 0 1 1
SEL	RBn	RSS ← 0, RBS2-RBS0 ← n	4	4	2							0 0 0 0 0 1 0 1
												1 0 1 0 1 N_2 N_1 N_0
	RBn,ALT	RSS ← 1, RBS2-RBS0 ← n	4	4	2							0 0 0 0 0 1 0 1
												1 0 1 1 1 N_2 N_1 N_0
NOP		No operation	3	3	1							0 0 0 0 0 0 0 0
EI		IE ← 1 (Enable interrupt)	3	3	1							0 1 0 0 1 0 1 1
DI		IE ← 0 (Disable interrupt)	3	3	1							0 1 0 0 1 0 1 0