NEC
NEC Electronics Inc.

μPD78310/312
8-BIT, SINGLE-CHIP CMOS MICROCOMPUTERS, REAL-TIME CONTROL ORIENTED

T-49-19-08

Description

The μPD78310 and μPD78312 microcomputers are designed for use in process control. They perform all the usual process control functions and are particularly well-suited for driving dc motors in servo loops and stepping motors. The processor includes on-chip memory, timers, input/output registers, and a powerful interrupt handling facility. The μPD78310/312 is constructed of high-speed CMOS circuitry and operates from a +5 V 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 (μPD78312 only), and data memory is 256 bytes of static RAM. The μPD78310 is the ROM-less version.

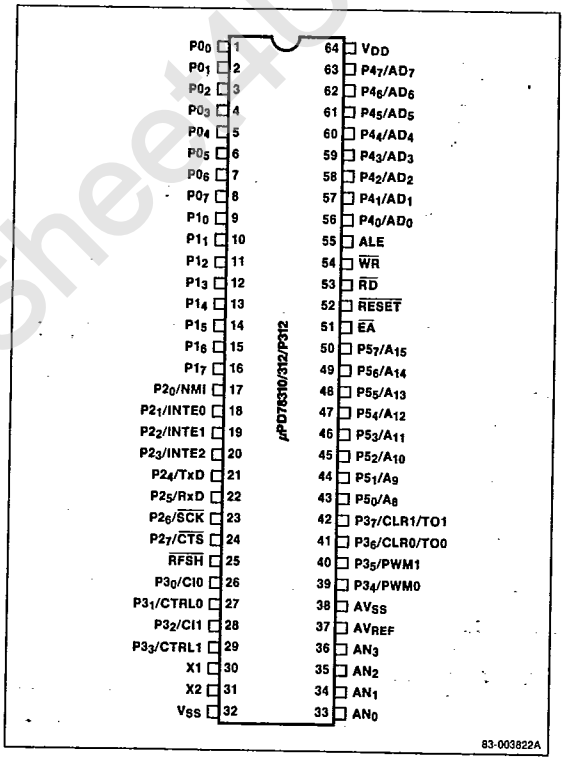
Note: μPD78P312, available in 3Q86, is a prototyping chip for μPD78312. It has an on-chip 8K EPROM instead of a mask ROM.

Features

- ☐ Complete single-chip microcomputer
  - — 16-bit ALU
  - — 8K ROM (μPD78312 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
  - — 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
- ☐ Macro service facility for interrupts
  - — Gives the effect of 8 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 V power supply

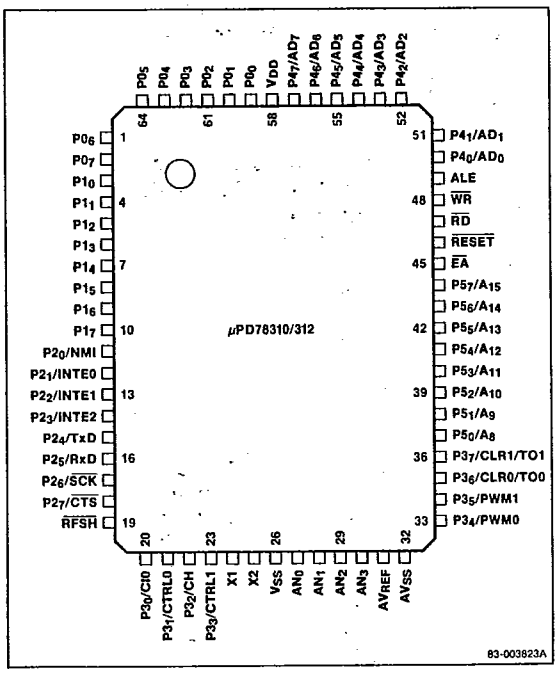
Pin Configurations

64-Pin DIP and QUIP



Pin Configurations (cont)

64-Pin Miniflat



Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD78310CW μPD78312CW	64-pin plastic shrink DIP	12 MHz
μPD78310G-36 μPD78312G-36 μPD78P312G-36	64-pin plastic QUIP	12 MHz
μPD78310G-1B μPD78312G-1B	64-pin plastic miniflat	12 MHz
μPD78310L μPD78312L	68-pin PLCC	12 MHz

Pin Identification

Symbol	Function
$P0_0$-$P0_7$	I/O port 0
$P1_0$-$P1_7$	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 transmit output
$P2_6$/$\overline{SCK}$	I/O port 2/Serial clock output

Pin Identification (cont)

Symbol	Function
$P2_7$/$\overline{CTS}$	I/O port 2/Clear to send input
$\overline{RFSH}$	Refresh output
$P3_0$/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
X1	External crystal/External clock input
X2	External crystal
V_{SS}	Power return
AN_0-AN_3	A/D converter inputs
AV_{REF}	A/D reference voltage
AV_{SS}	Analog ground
$P3_4$/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
$P5_0$-$P5_7$/A_8-A_{15}	I/O port 5/High address byte output
$\overline{EA}$	External access control input
$\overline{RESET}$	External reset input
$\overline{RD}$	Read strobe output
$\overline{WR}$	Write strobe output
ALE	Address latch enable output
$P4_0$-$P4_7$/AD_0-AD_7	I/O port 4/External address/Data bus
V_{DD}	Power supply

Pin Functions

$P0_0$-$P0_7$ [Port 0]

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

$P1_0$-$P1_7$ [Port 1]

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

$P2_0$/NMI

Port $P2_0$ is dedicated to NMI, the nonmaskable external interrupt request.

$P2_1$-$P2_3$/INTE0-INTE2

Ports $P2_1$-$P2_3$ are dedicated to INTE0, INTE1, and INTE2, the maskable external interrupt requests.

$P2_4$/TxD

$P2_4$ is an I/O port bit or the transmitted serial data output.

$P2_5$/RxD

$P2_5$ is an I/O port bit or the received serial data input.

$P2_6$/$\overline{SCK}$

$P2_6$ is an I/O port bit or the serial shift clock output.

$P2_7$/$\overline{CTS}$

$P2_7$ 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.

$\overline{RFSH}$

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

$P3_0$/CI0

Port $P3_0$ is dedicated to CI0, the external count input for up/down counter 0.

$P3_1$/CTRL0

Port $P3_1$ is dedicated to CTRL0, the external control input for up/down counter 0.

$P3_2$/CI1

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

$P3_3$/CTRL1

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

X1

X1 is the external oscillator input or one of the connections for an external crystal. It is used to generate the system clock. The system clock frequency is half the input frequency.

X2

X2 is the second connection for an external crystal.

V_{SS}

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

AN_0-AN_3

AN_0-AN_3 are the four program selectable input channels for the A/D converter.

AV_{REF}

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

AV_{SS}

AV_{SS} is the analog ground pin.

$P3_4$/PWM0

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

$P3_5$/PWM1

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

$P3_6$/CLR0/TO0

$P3_6$ 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

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

$P5_0$-$P5_7$/A_8-A_{15} [Port 5]

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 $P5_0$-$P5_3$ are used for 4K memory expansion, bits $P5_0$-$P5_5$ are used for 16K memory expansion, or bits $P5_0$-$P5_7$ are used for 56K memory expansion.

$\overline{EA}$ [External Access]

On μPD78312, a low on $\overline{EA}$ enables use of external memory in place of on-chip ROM. The $\overline{EA}$ pin must be low on μPD78310.

$\overline{RESET}$

This pin is used for the external reset input. A low level sets all registers to their specified reset values.

$\overline{RD}$

$\overline{RD}$ 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.

$\overline{WR}$

$\overline{WR}$ 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.

ALE

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.

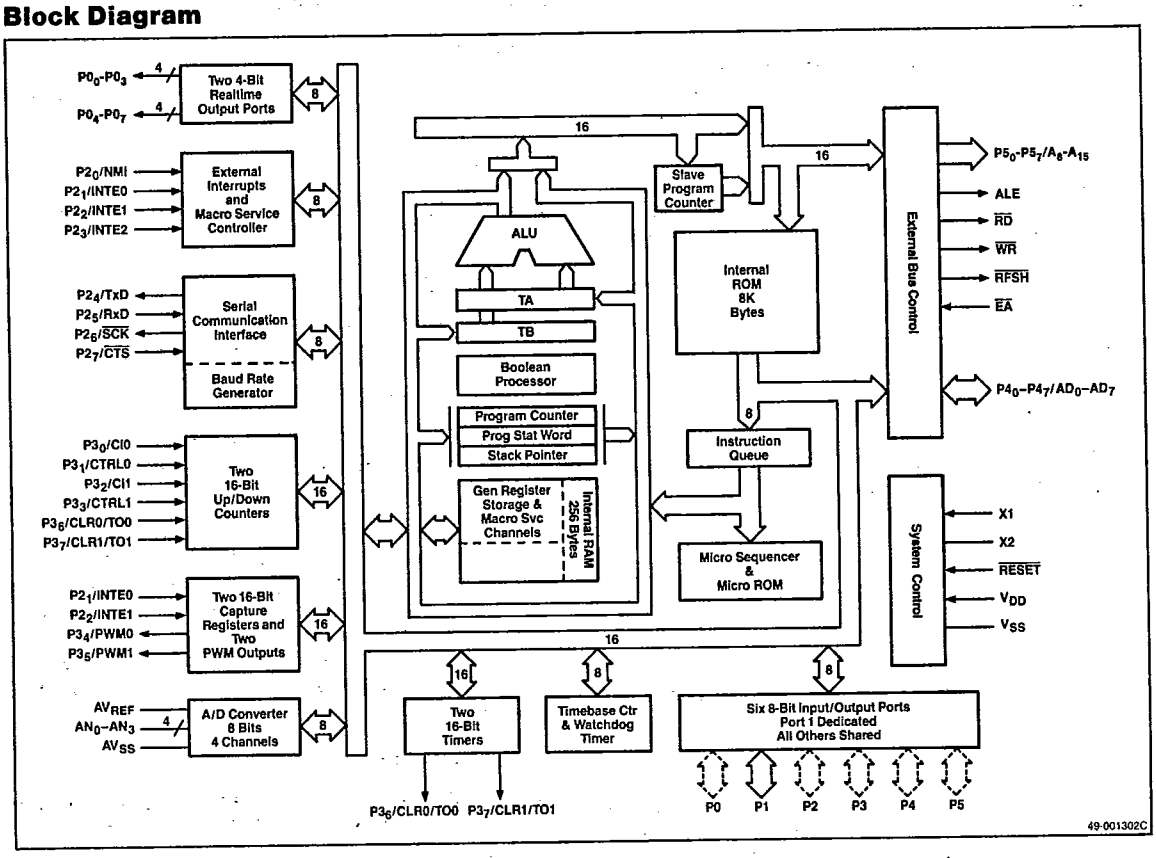
$P4_0$-$P4_7$/AD_0-AD_7 [Port 4]

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.

V_{DD}

V_{DD} is the positive power supply input.

Block Diagram



Functional Description

On-chip features designed to facilitate process control include two 16-bit timers, 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 4 input channels, a timebase counter to generate widely spaced interrupts, and a watchdog timer to guard against infinite program loops.

In addition there is a serial I/O port which 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 the action of the CPU 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 2 at the end of the Functional Description describes the registers.

Addressing

The μPD78310/312 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.

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 $P5_0$-$P5_3$ are used for 4K bytes, $P5_0$-$P5_5$ for 16K bytes, and $P5_0$-$P5_7$ for 56K bytes. Any remaining port 5 bits are available for I/O.

Refresh

The μPD78310/312 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 the CPU does not have to wait.

General Registers

The CPU has 16 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.

Figure 1. Memory Map

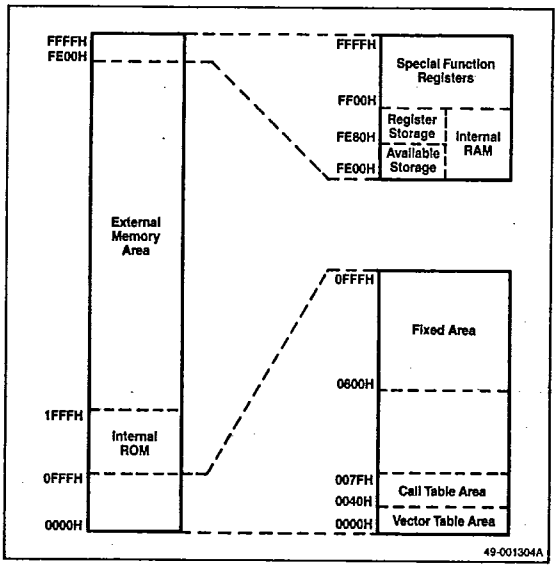
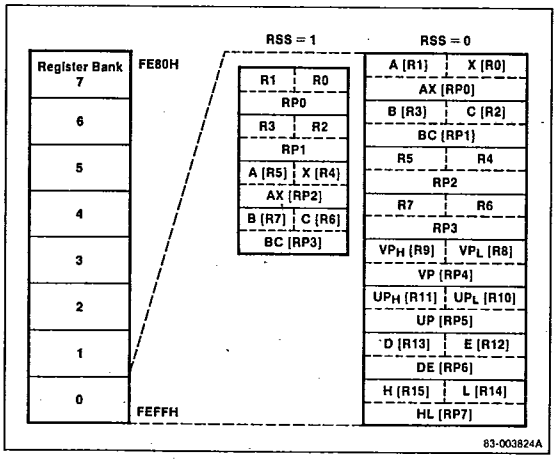


Figure 2. Register Designation and Storage



Program Status Word

Following is the program status word format.

15							8
0	RB_2	RB_1	RB_0	0	0	IE	0

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

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 3 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.

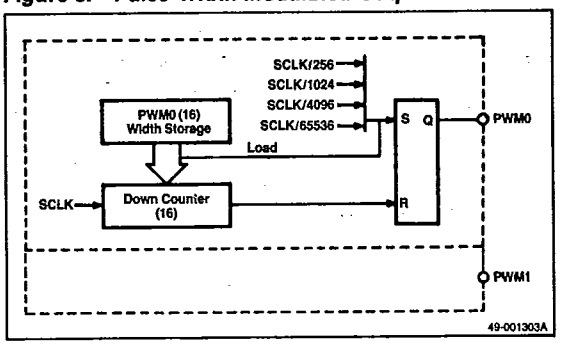
Serial Port

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 kHz. Figure 3 shows one of these outputs.

Figure 3. Pulse-Width Modulated Output



Timers

The μPD78310/312 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 triggerd by an external interrupt request or by the up/down counter clock.

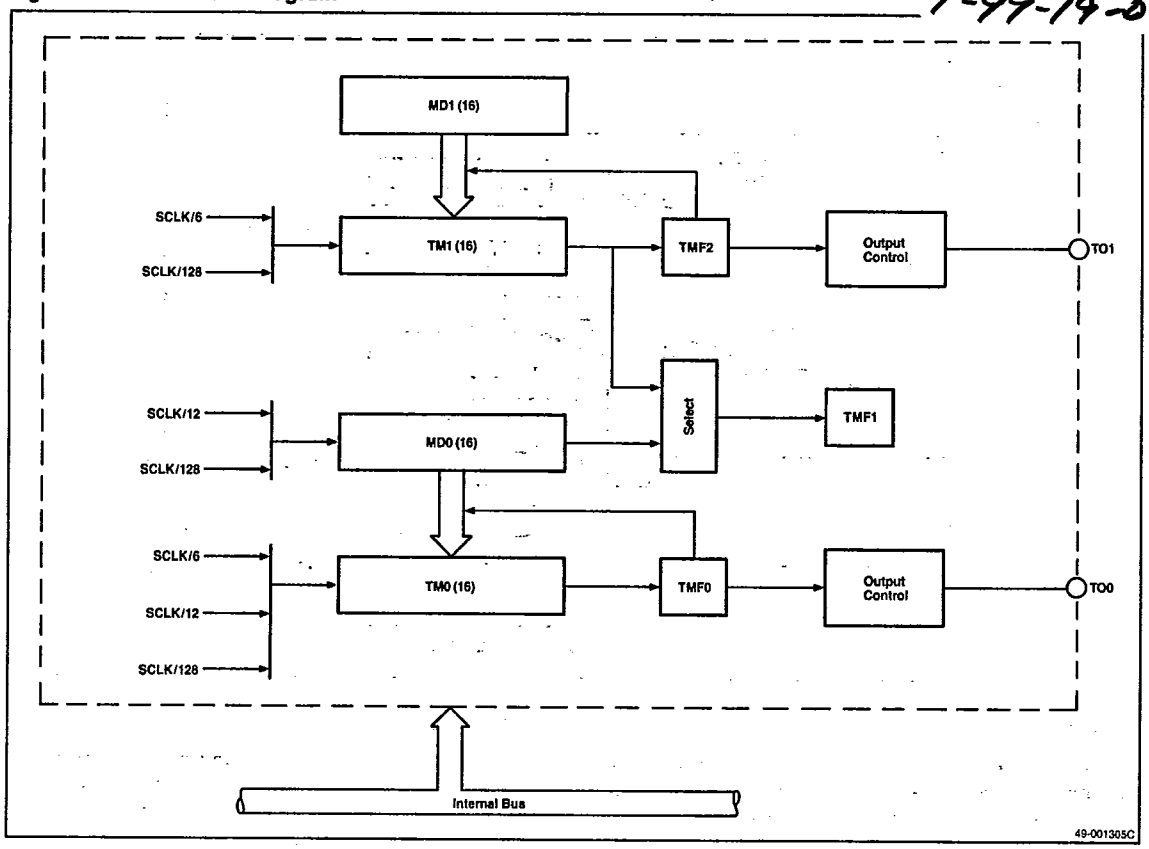
The timebase counter generates a signal at one of four intervals ranging from 170 μs to 175 ms. The signal can be used to generate an interrupt request and/or an up/down counter capture.

Up/Down Counters

The μPD78310/312 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 4. Timer Block Diagram

T-49-19-08



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.

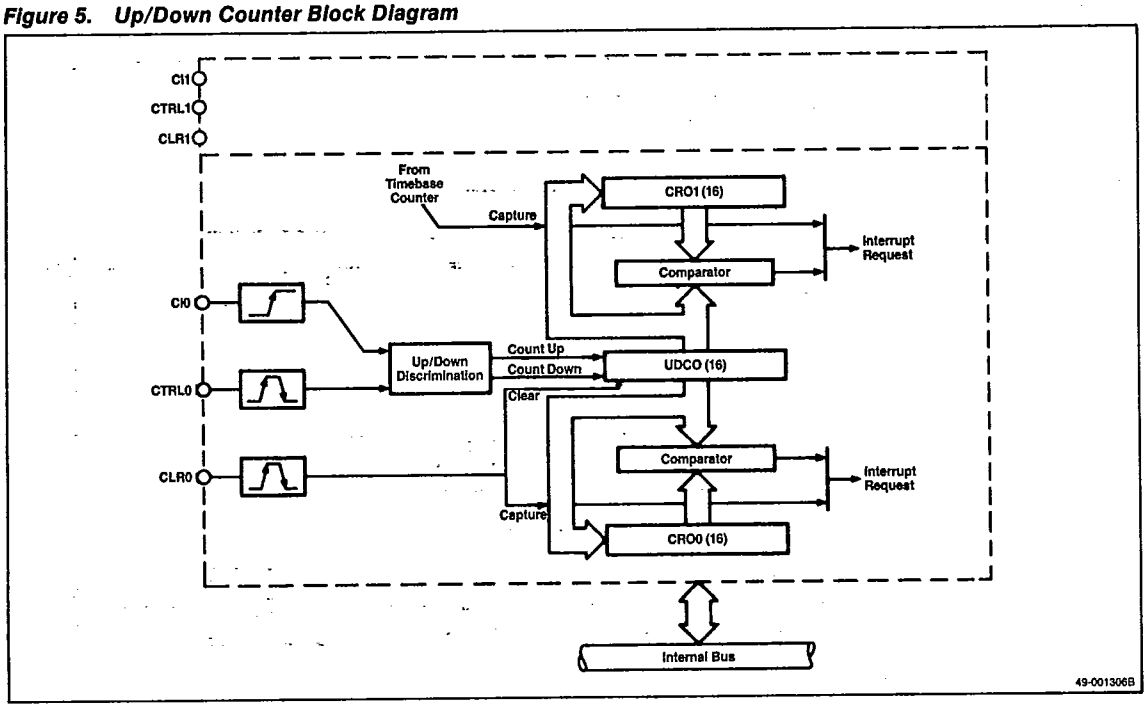
Watchdog Timer

The watchdog timer protects agains 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 ms 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 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.

Figure 5. Up/Down Counter Block Diagram



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 fifteen maskable interrupt sources (table 1) 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 6 illustrates the mechanism of context switching.

Finally, there is an optional macro service function that transfers data between any one special function register and memory without program intervention.

Macro Service

The macro service 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 macro service channels; channel control information is stored in RAM. This information (figure 7) consists of a 16-bit memory address (optionally incremented at each transfer), an 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.

Table 1. Interrupt Sources and Vector Addresses

	Default Priority	Source	Interrupt Service	Macro Service	Vector
Nonmaskable interrupts	—	BRK	Break instruction	No	003EH
	—	NMI	External nonmaskable interrupt	No	0002H
	—	WDT	Watchdog timer	No	000AH
Maskable interrupts	0	CRF00	Up/down counter	Yes	001AH
	1	CRF01	Up/down counter	No	001CH
	2	CRF10	Up/down counter	Yes	001EH
	3	CRF11	Up/down counter	No	0020H
	4	EXIF0	External interrupt 0	Yes	0004H
	5	EXIF1	External interrupt 1	Yes	0006H
	6	EXIF2	External interrupt 2	Yes	0008H
	7	TIMF0	Timer flag 0	Yes	000EH
	8	TIMF1	Timer flag 1	Yes	0010H
	9	TIMF2	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	External reset line	—	0000H

Figure 6. Hardware Context Switching

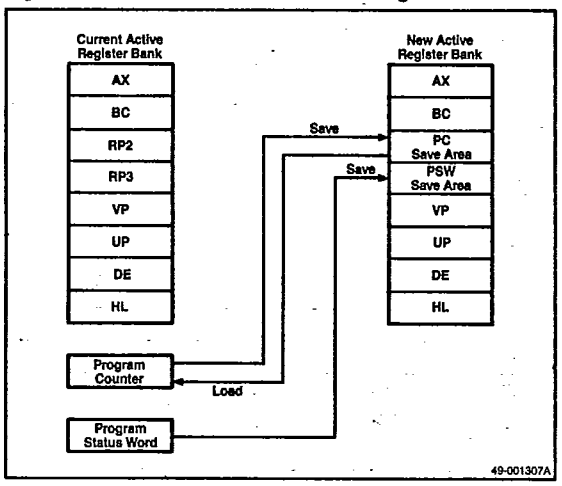
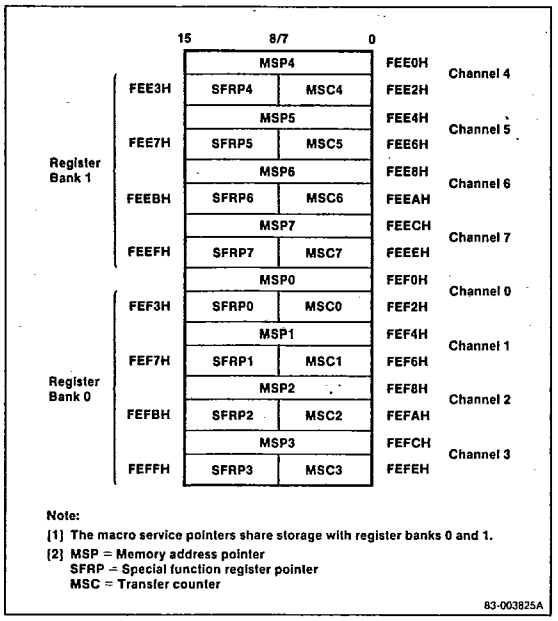


Figure 7. μPD78312 Macro Service Pointer Addresses



Note:
[1] The macro service 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 2. 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	No (Note 1)	Undefined
FF03H	I/O port 3	P3	R/W	No (Note 1)	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	R/W	Yes	Undefined
FF0AH FF0BH	Capture/compare register 01	CR01L CR01H	R/W	Yes	Undefined
FF0CH FF0DH	Capture/compare register 10	CR10L CR10H	R/W	Yes	Undefined
FF0EH FF0FH	Capture/compare register 11	CR11L CR11H	R/W	Yes	Undefined
FF10H FF11H	Capture register 0 (from FRC)	CPT0L CPT0H	R/W	Yes	Undefined
FF12H FF13H	Capture register 1 (from FRC)	CPT1L CPT1H	R/W	Yes	Undefined
FF14H FF15H	PWM register 0 (duration)	PWM0L PWM0H	R/W	Yes	Undefined
FF16H FF17H	PWM register 1 (duration)	PWM1L PWM1H	R/W	Yes	Undefined
FF1CH FF1DH	Presettable up/down counter 0	UDC0L UDC0H	R/W	Yes	Undefined
FF1EH FF1FH	Presettable up/down counter 1	UDC1L UDC1H	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	No	FFH
FF23H	Port 3 mode register	PM3	R/W	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

Table 2. Special Function Registers (cont)

Address	Function	Mnemonic	Read/ Write	16-Bit Transfer	Reset State
FF3AH FF3BH	Port 0 buffer register (Note 2)	P0L P0H	R/W R/W	No No	Undefined Undefined
FF40H	Memory mapping 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	TMB	R/W	No	00H
FF48H	Interrupt mode register	INTM	R/W	No	00H
FF4AH	In-service priority register	ISPR	R/W	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
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
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 TM0H	R/W	Yes	Undefined
FF8AH FF8BH	Modulus/timer register 0	MD0L MD0H	R/W	Yes	Undefined
FF8CH FF8DH	Timer 1	TM1L TM1H	R/W	Yes	Undefined
FF8EH FF8FH	Modulus register 1	MD1L MD1H	R/W	Yes	Undefined
FFB0H- FFBFH	External area (Note 4)				

Table 2. Special Function Registers (cont)

Address	Function		Mnemonic	Read/ Write	16-Bit Transfer	Reset State
FFC0H	Interrupt control 00	Up/down counter	CRIC00	R/W	No	47H
FFC1H	Macro service control 00	Up/down counter	CRMS00	R/W	No	Undefined
FFC2H	Interrupt control 01	Up/down counter	CRIC01	R/W	No	47H
FFC4H	Interrupt control 10	Up/down counter	CRIC10	R/W	No	47H
FFC5H	Macro service control 10	Up/down counter	CRMS10	R/W	No	Undefined
FFC6H	Interrupt control 11	Up/down counter	CRIC11	R/W	No	47H
FFC8H	EXIF0 interrupt control	External interrupt	EXIC0	R/W	No	47H
FFC9H	EXIF0 macro service control	External interrupt	EXMS0	R/W	No	Undefined
FFCAH	EXIF1 interrupt control	External interrupt	EXIC1	R/W	No	47H
FFCBH	EXIF1 macro service control	External interrupt	EXMS1	R/W	No	Undefined
FFCCH	EXIF2 interrupt control	External interrupt	EXIC2	R/W	No	47H
FFCDH	EXIF2 macro service control	External interrupt	EXMS2	R/W	No	Undefined
FFCEH	TMF0 interrupt control	Timer flag	TMIC0	R/W	No	47H
FFCFH	TMF0 macro service control	Timer flag	TMMS0	R/W	No	Undefined
FFD0H	TMF1 interrupt control	Timer flag	TMIC1	R/W	No	47H
FFD1H	TMF1 macro service control	Timer flag	TMMS1	R/W	No	Undefined
FFD2H	TMF2 interrupt control	Timer flag	TMIC2	R/W	No	47H
FFD3H	TMF2 macro service control	Timer flag	TMMS2	R/W	No	Undefined
FFDAH	Error interrupt control	Serial port	SEIC	R/W	No	47H
FFDCH	Receive interrupt control	Serial port	SRIC	R/W	No	47H
FFDDH	Receive macro service control	Serial port	SRMS	R/W	No	Undefined
FFDEH	Transmit interrupt control	Serial port	STIC	R/W	No	47H
FFDFH	Transmit macro service control	Serial port	STMS	R/W	No	Undefined
FFE0H	A/D converter interrupt control		ADIC	R/W	No	47H
FFE1H	A/D converter macro service control		ADMS	R/W	No	Undefined
FFE2H	Timebase counter interrupt control		TBIC	R/W	No	47H
FFFCH FFFDH	Stack pointer (Note 5)		SPL SPH	R/W	Yes	Undefined
FFFEH FFFFH	Program status word (Note 5)		PSWL PSWH	R/W	Yes	00H

Note:

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

(3) Bit 3 of the STBC is not affected by RESET (n = 0 or 8).

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

(5) SP and PSW do not have real SFR addresses and can be accessed only by special instructions.

NEC

T-49-19-08

μPD78310/312

Instruction Set

The instruction set for the μPD78310/312 has 8- and 16-bit arithmetic instructions including a 16 x 16-bit unsigned multiply with a 32-bit product and a 32 by 16-bit unsigned divide with a 32-bit quotient and a 16-bit remainder. The instruction set also excutes 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.

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.

Following are several tables explaining symbols, designations, and codes in the Instruction Set. Machine codes are omitted from the instructions but they are in the User's Manual.

Symbols in the Operand and Operation Columns

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	(DE), (HL), (DE+), (HL+), (DE−), (HL−), (VP), (UP); register indirect (DE + A), (HL + A), (DE + B), (HL + B), (VP + DE), (VP + HL); base/index mode (DE + byte), (HL + byte), (VP + byte), (UP + byte), (SP + byte); base mode Word (A), word (B), word (DE), word (HL); index mode
saddr	FF20H-FF1FH: immediate byte addresses one byte in RAM, or label
saddrp	FE20H-FF1FH: immediate byte (bit 0 = 0) addresses one word in RAM
word	16 bits of immediate data
byte	8 bits of immediate data
jdisp	8-bit two's complement displacement (immediate data)
bit	3 bits of immediate data (bit position in byte), or label
n	3 bits of immediate data
addr16	0000H-FEFFH: 16-bit immediate address (up to FFFFH in MOV instruction)
!addr16	0000H-FEFFH: 16-bit absolute branch address (immediate data)
$addr16	Relative branch address ((PC)+jdisp))
addr11	0800H-0FFFH: 0800H+ (11-bit immediate address), or label
addr5	0040H-007EH: 0040H + 2 x (5-bit immediate address), or label

T-49-19-08

Symbols in the Operand and Operation Columns (cont)

Symbol	Meaning
A	A register (8-bit accumulator)
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
R0-R15	Register 0-15
AX	Register pair AX (16-bit accumulator)
BC	Register pair BC
DE	Register pair DE
HL	Register pair HL
RP0-RP7	Register pair 0-7
PC	Program counter
SP	Stack pointer
UP	User stack pointer (RP5)
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
( )	Contents of the location whose address is within ( ); (+) and (−) indicate that the address is incremented after or decremented before 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

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

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

Instruction		Memory Reference Mode: 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.

NEC

T-49-19-08

μPD78310/312

General Register Designations

r, r1

R3	R2	R1	R0	reg	
0	0	0	0	R0	r1
0	0	0	1	R1	
0	0	1	0	R2	
0	0	1	1	R3	
0	1	0	0	R4	
0	1	0	1	R5	
0	1	1	0	R6	
0	1	1	1	R7	
1	0	0	0	R8	r
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

P2	P1	P0	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

Q2	Q1	Q0	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

S1	S0	reg-pair
0	0	VP
0	1	UP
1	0	DE
1	1	HL

4

T-49-19-08

Instruction Set

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
MOV	r1, #byte	r1 ← byte	3	2						
	saddr, #byte	(saddr) ← byte	3	3						
	sfr**, #byte	sfr ← byte	3	3						
	r, r1	r ← r1	3	2						
	A, r1	A ← r1	3	1						
	A, saddr	A ← (saddr)	3	2						
	saddr, A	(saddr) ← A	3	2						
	saddr, saddr	(saddr) ← (saddr)	4	3						
	A, sfr	A ← sfr	3	2						
	sfr, A	sfr ← A	3	2						
	A, mem*	A ← (mem)	5	1						
	A, mem	A ← (mem)	5-6	2-4						
	mem, A*	(mem) ← A	5	1						
	mem, A	(mem) ← A	5-6	2-4						
	A, (saddrp)	A ← ((saddrp))	5	2						
	(saddrp), A	((saddrp)) ← A	4	2						
	A, addr16	A ← (addr16)	4	4						
	addr16, A	(addr16) ← A	3	4						
	PSWL, #byte	PSW_L ← byte	3	3	X	X	X	X	X	X
	PSWH, #byte	PSW_H ← byte	3	3						
	PSWL, A	PSW_L ← A	3	2	X	X	X	X	X	X
	PSWH, A	PSW_H ← A	3	2						
	A, PSWL	A ← PSW_L	3	2						
	A, PSWH	A ← PSW_H	3	2						
XCH	A, r1	A ⟷ r1	4	1						
	r, r1	r ⟷ r1	4	2						
	A, mem	A ⟷ (mem)	7-8	2-4						
	A, saddr	A ⟷ (saddr)	4	2						
	A, sfr	A ⟷ sfr	7	3						
	A, (saddrp)	A ⟷ ((saddrp))	6	2						
	saddr, saddr	(saddr) ⟷ (saddr)	8	3						

** A special instruction is used to write to STBC and WDM (see below).

* One-byte move instruction.

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags					
					S	Z	AC	P/V	SUB	CY
MOVW	rp1, #word	rp1 ← word	3	3						
	saddrp, #word	(saddrp) ← word	4	4						
	sfrp, #word	sfrp ← word	3	4						
	rp, rp1	rp ← rp1	3	2						
	AX, saddrp	AX ← (saddrp)	3	2						
	saddrp, AX	(saddrp) ← AX	3	2						
	saddrp, saddrp	(saddrp) ← (saddrp)	4	3						
	AX, sfrp	AX ← sfrp	3	2						
	sfrp, AX	sfrp ← AX	3	2						
XCHW	AX, saddrp	AX ⟷ (saddrp)	4	2						
	AX, sfrp	AX ⟷ sfrp	7	3						
	saddrp, saddrp	(saddrp) ⟷ (saddrp)	8	3						
	rp rp1	rp ⟷ rp1	5	2						
ADD	A, #byte	A, CY ← A + byte	3	2	X	X	X	V	0	X
	saddr, #byte	(saddr), CY ← (saddr) + byte	4	3	X	X	X	V	0	X
	sfr, #byte	sfr, CY ← sfr + byte	7	4	X	X	X	V	0	X
	r, r1	r, CY ← r + r1	3	2	X	X	X	V	0	X
	A, saddr	A, CY ← A + (saddr)	3	2	X	X	X	V	0	X
	A, sfr	A, CY ← A + sfr	6	3	X	X	X	V	0	X
	saddr, saddr	(saddr), CY ← (saddr) + (saddr)	6	3	X	X	X	V	0	X
	A, mem	A, CY ← A + (mem)	6-7	2-4	X	X	X	V	0	X
	mem, A	(mem), CY ← (mem) + A	7-8	2-4	X	X	X	V	0	X
ADDC	A, #byte	A, CY ← A + byte + CY	3	2	X	X	X	V	0	X
	saddr, #byte	(saddr), CY ← (saddr) + byte + CY	4	3	X	X	X	V	0	X
	sfr, #byte	sfr, CY ← sfr + byte + CY	7	4	X	X	X	V	0	X
	r, r1	r, CY ← r + r1 + CY	3	2	X	X	X	V	0	X
	A, saddr	A, CY ← A + (saddr) + CY	3	2	X	X	X	V	0	X
	A, sfr	A, CY ← A + sfr + CY	6	3	X	X	X	V	0	X
	saddr, saddr	(saddr), CY ← (saddr) + (saddr) + CY	6	3	X	X	X	V	0	X
	A, mem	A, CY ← A + (mem) + CY	6-7	2-4	X	X	X	V	0	X
	mem, A	(mem), CY ← (mem) + A + CY	7-8	2-4	X	X	X	V	0	X

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
SUB	A, #byte	A, CY ← A – byte	3	2	X	X	X	V	1	X
	saddr, #byte	(saddr), CY ← (saddr) – byte	4	3	X	X	X	V	1	X
	sfr, #byte	sfr, CY ← sfr – byte	7	4	X	X	X	V	1	X
	r, r1	r, CY ← r – r1	3	2	X	X	X	V	1	X
	A, saddr	A, CY ← A – (saddr)	3	2	X	X	X	V	1	X
	A, sfr	A, CY ← A – sfr	6	3	X	X	X	V	1	X
	saddr, saddr	(saddr), CY ← (saddr) – (saddr)	6	3	X	X	X	V	1	X
	A, mem	A, CY ← A – (mem)	6-7	2-4	X	X	X	V	1	X
	mem, A	(mem), CY ← (mem) – A	7-8	2-4	X	X	X	V	1	X
SUBC	A, #byte	A, CY ← A – byte – CY	3	2	X	X	X	V	1	X
	saddr, #byte	(saddr), CY ← (saddr) – byte – CY	4	3	X	X	X	V	1	X
	sfr, #byte	sfr, CY ← sfr – byte – CY	7	4	X	X	X	V	1	X
	r, r1	r, CY ← r – r1 – CY	3	2	X	X	X	V	1	X
	A, saddr	A, CY ← A – (saddr) – CY	3	2	X	X	X	V	1	X
	A, sfr	A, CY ← A – sfr – CY	6	3	X	X	X	V	1	X
	saddr, saddr	(saddr), CY ← (saddr) – (saddr) – CY	6	3	X	X	X	V	1	X
	A, mem	A, CY ← A – (mem) – CY	6-7	2-4	X	X	X	V	1	X
	mem, A	(mem), CY ← (mem) – A – CY	7-8	2-4	X	X	X	V	1	X
AND	A, #byte	A ← A ∧ byte	3	2	X	X	U	P	0	0
	saddr, #byte	(saddr) ← (saddr) ∧ byte	4	3	X	X	U	P	0	0
	sfr, #byte	sfr ← sfr ∧ byte	7	4	X	X	U	P	0	0
	r, r1	r ← r ∧ r1	3	2	X	X	U	P	0	0
	A, saddr	A ← A ∧ (saddr)	3	2	X	X	U	P	0	0
	A, sfr	A → A ∧ sfr	6	3	X	X	U	P	0	0
	saddr, saddr	(saddr) ← (saddr) ∧ (saddr)	6	3	X	X	U	P	0	0
	A, mem	A ← A ∧ (mem)	6-7	2-4	X	X	U	P	0	0
	mem, A	(mem) ← (mem) ∧ A	7-8	2-4	X	X	U	P	0	0
OR	A, #byte	A ← A V byte	3	2	X	X	U	P	0	0
	saddr, #byte	(saddr) ← (saddr) V byte	4	3	X	X	U	P	0	0
	sfr, #byte	sfr ← sfr V byte	7	4	X	X	U	P	0	0
	r, r1	r ← r V r1	3	2	X	X	U	P	0	0
	A, saddr	A ← A V (saddr)	3	2	X	X	U	P	0	0
	A, sfr	A ← A V sfr	6	3	X	X	U	P	0	0
	saddr, saddr	(saddr) ← (saddr) V (saddr)	6	3	X	X	U	P	0	0
	A, mem	A ← A V (mem)	6-7	2-4	X	X	U	P	0	0
	mem, A	(mem) ← (mem) V A	7-8	2-4	X	X	U	P	0	0

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
XOR	A, #byte	A ← A ∀ byte	3	2	X	X	U	P	0	0
	saddr, #byte	(saddr) ← (saddr) ∀ byte	4	3	X	X	U	P	0	0
	sfr, #byte	sfr ← sfr ∀ byte	7	4	X	X	U	P	0	0
	r, r1	r ← r ∀ r1	3	2	X	X	U	P	0	0
	A, saddr	A ← A ∀ (saddr)	3	2	X	X	U	P	0	0
	A, sfr	A ← A ∀ sfr	6	3	X	X	U	P	0	0
	saddr, saddr	(saddr) ← (saddr) ∀ (saddr)	6	3	X	X	U	P	0	0
	A, mem	A ← A ∀ (mem)	6-7	2-4	X	X	U	P	0	0
	mem, A	(mem) ← (mem) ∀ A	7-8	2-4	X	X	U	P	0	0
CMP	A, #byte	A – byte	3	2	X	X	X	V	1	X
	saddr, #byte	(saddr) – byte	4	3	X	X	X	V	1	X
	sfr, #byte	sfr – byte	7	4	X	X	X	V	1	X
	r, r1	r – r1	3	2	X	X	X	V	1	X
	A, saddr	A – (saddr)	3	2	X	X	X	V	1	X
	A, sfr	A – sfr	6	3	X	X	X	V	1	X
	saddr, saddr	(saddr) – (saddr)	6	3	X	X	X	V	1	X
	A, mem	A – (mem)	6-7	2-4	X	X	X	V	1	X
	mem, A	(mem) – A	6-7	2-4	X	X	X	V	1	X
ADDW	AX, #word	AX, CY ← AX + word	4	3	X	X	U	V	0	X
	saddrp, #word	(saddrp), CY ← (saddrp) + word	5	4	X	X	U	V	0	X
	sfrp, #word	sfrp, CY ← sfrp + word	8	5	X	X	U	V	0	X
	rp, rp1	rp, CY ← rp + rp1	4	2	X	X	U	V	0	X
	AX, saddrp	AX, CY ← AX + (saddrp)	4	2	X	X	U	V	0	X
	AX, sfrp	AX, CY ← AX + sfrp	7	3	X	X	U	V	0	X
	saddrp, saddrp	(saddrp), CY ← (saddrp) + (saddrp)	6	3	X	X	U	V	0	X
SUBW	AX, #word	AX, CY ← AX – word	4	3	X	X	U	V	1	X
	saddrp, #word	(saddrp), CY ← (saddrp) – word	5	4	X	X	U	V	1	X
	sfrp, #word	sfrp, CY ← sfrp – word	8	5	X	X	U	V	1	X
	rp, rp1	rp, CY ← rp – rp1	4	2	X	X	U	V	1	X
	AX, saddrp	AX, CY ← AX – (saddrp)	4	2	X	X	U	V	1	X
	AX, sfrp	AX, CY ← AX – sfrp	7	3	X	X	U	V	1	X
	saddrp, saddrp	(saddrp), CY ← (saddrp) – (saddrp)	6	3	X	X	U	V	1	X
CMPW	AX, #word	AX – word	4	3	X	X	U	V	1	X
	saddrp, #word	(saddrp) – word	5	4	X	X	U	V	1	X
	sfrp, #word	sfrp – word	8	5	X	X	U	V	1	X
	rp, rp1	rp – rp1	4	2	X	X	U	V	1	X
	AX, saddrp	AX – (saddrp)	4	2	X	X	U	V	1	X
	AX, sfrp	AX – sfrp	7	3	X	X	U	V	1	X
	saddrp, saddrp	(saddrp) – (saddrp)	6	3	X	X	U	V	1	X

4

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
MULU	r1	AX ← A x r1	18	2						
DIVU	r1	AX (Quotient), r1 (Remainder) ← AX ÷ r1	18	2						
MULUW	rp1	AX (High Order 16 Bits), rp1 (Low Order 16 Bits), ← AX x rp1	27	2						
DIVUX	rp1	AXDE (Quotient), rp1 (Remainder) ← AXDE ÷ rp1	50	2						
INC	r1	r1 ← r1 + 1	3	1	X	X	X	V	0	
	saddr	(saddr) ← (saddr) + 1	4	2	X	X	X	V	0	
DEC	r1	r1 ← r1 − 1	3	1	X	X	X	V	1	
	saddr	(saddr) ← (saddr) − 1	4	2	X	X	X	V	1	
INCW	rp2	rp2 ← rp2 + 1	3	1						
	saddrp	(saddrp) ← (saddrp) + 1	6	3						
DECW	rp2	rp2 ← rp2 − 1	3	1						
	saddrp	(saddrp) ← (saddrp) − 1	6	3						
ROR	r1, n	(CY, $r1_7$ ← $r1_0$; $r1_{m-1}$ ← $r1_m$) x n	4 + 3n	2	X	X	0	P	0	X
ROL	r1, n	(CY, $r1_0$ ← $r1_7$; $r1_{m+1}$ ← $r1_m$) x n	4 + 3n	2	X	X	0	P	0	X
RORC	r1, n	(CY ← $r1_0$; $r1_7$ ← CY; $r1_{m-1}$ ← $r1_m$) x n	4 + 3n	2	X	X	0	P	0	X
ROLC	r1, n	(CY ← $r1_7$; $r1_0$ ← CY; $r1_{m+1}$ ← $r1_m$) x n	4 + 3n	2	X	X	0	P	0	X
SHR	r1, n	(CY ← $r1_0$; $r1_7$ ← 0; $r1_{m-1}$ ← $r1_m$) x n	4 + 3n	2	X	X	0	P	0	X
SHL	r1, n	(CY ← $r1_7$; $r1_0$ ← 0; $r1_{m+1}$ ← $r1_m$) x n	4 + 3n	2	X	X	0	P	0	X
SHRW	rp1, n	(CY ← rp_0; rp_{15} ← 0; rp_{m-1} ← rp_m) x n	4 + 3n	2	X	X	0	P	0	X
SHLW	rp1, n	(CY ← rp_{15}; rp_0 ← 0; rp_{m+1} ← rp_m) x n	4 + 3n	2	X	X	0	P	0	X
ROR4	(rp1)	A_{3-0} ← $(rp1)_{3-0}$; $(rp1)_{7-4}$ ← A_{3-0}; $(rp1)_{3-0}$ ← $(rp1)_{7-4}$	8	2	X	X	0	P	0	
ROL4	(rp1)	A_{3-0} ← $(rp1)_{7-4}$; $(rp1)_{3-0}$ ← A_{3-0}; $(rp1)_{7-4}$ ← $(rp1)_{3-0}$	8	2	X	X	0	P	0	
ADJ4		Decimal Adjust Accumulator	3	1	X	X	X	V	0	X

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
MOV1	CY, saddr.bit	CY ← (saddr.bit)	6	3						X
	CY, sfr.bit	CY ← sfr.bit	6	3						X
	CY, A.bit	CY ← A.bit	6	2						X
	CY, X.bit	CY ← X.bit	6	2						X
	CY, PSWH, bit	CY ← PSW_H.bit	6	2						X
	CY, PSWL, bit	CY ← PSW_L.bit	6	2						X
	saddr.bit, CY	(saddr.bit) ← CY	7	3						
	sfr.bit, CY	sfr.bit ← CY	7	3						
	A.bit, CY	A.bit ← CY	8	2						
	X.bit, CY	X.bit ← CY	8	2						
	PSWH.bit, CY	PSW_H.bit ← CY	8	2						
	PSWL.bit, CY	PSW_L.bit ← CY	8	2						
AND1	CY, saddr.bit	CY ← CY ∧ (saddr.bit)	6	3						X
	CY,/saddr.bit	CY ← CY ∧ $\overline{(saddr.bit)}$	6	3						X
	CY, sfr.bit	CY ← CY ∧ sfr.bit	6	3						X
	CY,/sfr.bit	CY ← CY ∧ $\overline{sfr.bit}$	6	3						X
	CY, A.bit	CY ← CY ∧ A.bit	6	2						X
	CY,/A.bit	CY ← CY ∧ $\overline{A.bit}$	6	2						X
	CY, X.bit	CY ← CY ∧ X.bit	6	2						X
	CY,/X.bit	CY ← CY ∧ $\overline{X.bit}$	6	2						X
	CY, PSWH.bit	CY ← CY ∧ PSW_H.bit	6	2						X
	CY,/PSWH.bit	CY ← CY ∧ $\overline{PSW_H.bit}$	6	2						X
	CY, PSWL.bit	CY ← CY ∧ PSW_L.bit	6	2						X
	CY,/PSWL.bit	CY ← CY ∧ $\overline{PSW_L.bit}$	6	2						X
OR1	CY, saddr.bit	CY ← CY V (saddr.bit)	6	3						X
	CY,/saddr.bit	CY ← CY V $\overline{(saddr.bit)}$	6	3						X
	CY, sfr.bit	CY ← CY V sfr.bit	6	3						X
	CY,/sfr.bit	CY ← CY V $\overline{sfr.bit}$	6	3						X
	CY, A.bit	CY ← CY V A.bit	6	2						X
	CY,/A.bit	CY ← CY V $\overline{A.bit}$	6	2						X
	CY, X.bit	CY ← CY V X.bit	6	2						X
	CY,/X.bit	CY ← CY V $\overline{X.bit}$	6	2						X
	CY, PSWH.bit	CY ← CY V PSW_H.bit	6	2						X
	CY,/PSWH.bit	CY ← CY V $\overline{PSW_H.bit}$	6	2						X
	CY, PSWL.bit	CY ← CY V PSW_L.bit	6	2						X
	CY,/PSWL.bit	CY ← CY V $\overline{PSW_L.bit}$	6	2						X

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
XOR1	CY, saddr.bit	CY ← CY ∀ (saddr.bit)	6	3						X
	CY, sfr.bit	CY ← CY ∀ sfr.bit	6	3						X
	CY, A.bit	CY ← CY ∀ A.bit	6	2						X
	CY, X.bit	CY ← CY ∀ X.bit	6	2						X
	CY, PSWH.bit	CY ← CY ∀ PSW_H.bit	6	2						X
	CY, PSWL.bit	CY ← CY ∀ PSW_L.bit	6	2						X
SET1	saddr.bit	(saddr.bit) ← 1	5	2						
	sfr.bit	sfr.bit ← 1	6	3						
	A.bit	A.bit ← 1	7	2						
	X.bit	X.bit ← 1	7	2						
	PSWH.bit	PSW_H.bit ← 1	7	2						
	PSWL.bit	PSW_L.bit ← 1	7	2						
CLR1	saddr.bit	(saddr.bit) ← 0	5	2						
	sfr.bit	sfr.bit ← 0	6	3						
	A.bit	A.bit ← 0	7	2						
	X.bit	X.bit ← 0	7	2						
	PSWH.bit	PSW_H.bit ← 0	7	2						
	PSWL.bit	PSW_L.bit ← 0	7	2						
NOT1	saddr.bit	(saddr.bit) ← $\overline{\text{(saddr.bit)}}$	6	3						
	sfr.bit	sfr.bit ← $\overline{\text{sfr.bit}}$	6	3						
	A.bit	A.bit ← $\overline{\text{A.bit}}$	7	2						
	X.bit	X.bit ← $\overline{\text{X.bit}}$	7	2						
	PSWH.bit	PSW_H.bit ← $\overline{PSW_H.bit}$	7	2						
	PSWL.bit	PSW_L.bit ← $\overline{PSW_L.bit}$	7	2						
SET1	CY	CY ← 1	3	1						1
CLR1	CY	CY ← 0	3	1						0
NOT1	CY	CY ← $\overline{CY}$	3	1						X
CALL	!addr16	(SP − 1) ← $(PC + 3)_H$; (SP − 2) ← $(PC + 3)_L$; PC ← addr16; SP ← SP − 2	8	3						
CALLF	!addr11	(SP − 1) ← $(PC + 2)_H$; (SP − 2) ← $(PC + 2)_L$; PC ← addr11; SP ← SP − 2	8	2						
CALLT	(addr5)	(SP − 1) ← $(PC + 1)_H$; (SP − 2) ← $(PC + 1)_L$; PC_H ← (TPFX8000H + addr5 + 1); PC_H ← (TPFX8000H + addr5); SP ← SP − 2	10	1						

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
CALL	rp1	$(SP-1) \leftarrow (PC+2)_H$; $(SP-2) \leftarrow (PC+2)_L$; $PC_H \leftarrow rp1_H$; $PC_L \leftarrow rp1_L$; $SP \leftarrow SP-2$	13	2						
	(rp1)	$(SP-1) \leftarrow (PC+2)_H$; $(SP-2) \leftarrow (PC+2)_L$; $PC_H \leftarrow (rp1)_H$; $PC_L \leftarrow (rp1)_L$; $SP \leftarrow SP-2$	11	2						
BRK		$(SP-1) \leftarrow PSW_H$; $(SP-2) \leftarrow PSW_L$; $(SP-3) \leftarrow (PC+1)_H$; $(SP-4) \leftarrow (PC+1)_L$; $PC_L \leftarrow (003EH)$; $PC_H \leftarrow (003FH)$; $SP \leftarrow SP-4$	16	1						
RET		$PC_L \leftarrow (SP)$; $PC_H \leftarrow (SP+1)$; $SP \leftarrow SP+2$	8	1						
RETI		$PC_L \leftarrow (SP)$; $PC_H \leftarrow (SP+1)$; $PSW_L \leftarrow (SP+2)$; $PSW_H \leftarrow (SP+3)$; $SP \leftarrow SP+4$; $EOS \leftarrow 0$	14	1	R	R	R	R	R	R
PUSH	post	$((SP-1) \leftarrow post_H$; $(SP-2) \leftarrow post_L$; $SP \leftarrow SP-2) \times n$.	7 + 8n	2						
	PSW	$(SP-1) \leftarrow PSW_H$; $(SP-2) \leftarrow PSW_L$; $SP \leftarrow SP-2$	5	1						
PUSHU	post	$((UP-1) \leftarrow post_H$; $(UP-2) \leftarrow post_L$; $UP \leftarrow UP-2) \times n$.	8 + 8n	2						
POP	post	$(post_L \leftarrow (SP)$; $(post_H \leftarrow (SP+1)$; $SP \leftarrow SP+2) \times n$.	7 + 8n	2						
	PSW	$PSW_L \leftarrow (SP)$; $PSW_H \leftarrow (SP+1)$; $SP \leftarrow SP+2$	5	1	R	R	R	R	R	R
POPU	post	$(post_L \leftarrow (UP)$; $post_H \leftarrow (UP+1)$; $UP \leftarrow UP+2) \times n$.	8 + 8n	2						
MOVW	SP, #word	$SP \leftarrow word$	3	4						
	SP, AX	$SP \leftarrow AX$	3	2						
	AX, SP	$AX \leftarrow SP$	3	2						
INCW	SP	$SP \leftarrow SP+1$	6	2						
DECW	SP	$SP \leftarrow SP-1$	6	2						

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
BR	laddr16	PC ← addr16	4	3						
	rp1	$PC_H \leftarrow rp1_H$; $PC_L \leftarrow rp1_L$;	6	2						
	(rp1)	$PC_H \leftarrow (rp1)_H$; $PC_L \leftarrow (rp1)_L$;	9	2						
	$addr16	PC ← addr16	7	2						
BC BL	$addr16	PC ← addr16 if CY = 1	7(3)	2						
BNC BNL	$addr16	PC ← addr16 if CY = 0	7(3)	2						
BZ BE	$addr16	PC ← addr16 if Z = 1	7(3)	2						
BNZ BNE	$addr16	PC ← addr16 if Z = 0	7(3)	2						
BV BPE	$addr16	PC ← addr16 if P/V = 1	7(3)	2						
BNV BPO	$addr16	PC ← addr16 if P/V = 0	7(3)	2						
BN	$addr16	PC ← addr16 if S = 1	7(3)	2						
BP	$addr16	PC ← addr16 if S = 0	7(3)	2						
BGT	$addr16	PC ← addr16 if (P/V ⊻ S) ∨ Z = 0	9(5)	3						
BGE	$addr16	PC ← addr16 if P/V ⊻ S = 0	9(5)	3						
BLT	$addr16	PC ← addr16 if P/V ⊻ S = 1	9(5)	3						
BLE	$addr16	PC ← addr16 if (P/V ⊻ S) ∨ Z = 1	9(5)	3						
BH	$addr16	PC ← addr16 if Z + CY = 0	9(5)	3						
BNH	$addr16	PC ← addr16 if Z + CY = 1	9(5)	3						
BT	saddr.bit, $addr16	PC ← addr16 if (saddr.bit) = 1	9(7)	3						
	sfr.bit, $addr16	PC ← addr16 if (sfr.bit) = 1	10(7)	4						
	A.bit, $addr16	PC ← addr16 if A.bit = 1	10(7)	3						
	X.bit, $addr16	PC ← addr16 if X.bit = 1	10(7)	3						
	PSWH.bit, $addr16	PC ← addr16 if PSW_H.bit = 1	10(7)	3						
	PSWL.bit, $addr16	PC ← addr16 if PSW_L.bit = 1	10(7)	3						
BF	saddr.bit, $addr16	PC ← addr16 if (saddr.bit) = 0	10(7)	4						
	sfr.bit, $addr16	PC ← addr16 if (sfr.bit) = 0	10(7)	4						
	A.bit, $addr16	PC ← addr16 if A.bit = 0	10(7)	3						
	X.bit, $addr16	PC ← addr16 if X.bit = 0	10(7)	3						
	PSWH.bit, $addr16	PC ← addr16 if PSW_H.bit = 0	10(7)	3						
	PSWL.bit, $addr16	PC ← addr16 if PSW_L.bit = 0	10(7)	3						
BTCLR	saddr.bit, $addr16	PC ← addr16 if (saddr.bit) = 1 then reset (saddr.bit)	12(7)	4						
	sfr.bit, $addr16	PC ← addr16 if (sfr.bit) = 1 then reset (sfr.bit)	12(7)	4						
	A.bit, $addr16	PC ← addr16 if A.bit = 1 then reset A.bit	11(7)	3						

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
BTCLR (cont)	X.bit, $addr16	PC ← addr16 if X.bit = 1 then reset X.bit	11(7)	3						
	PSWH.bit, $addr16	PC ← addr16 if PSW_H.bit = 1 then reset PSW_H.bit	12(7)	3						
	PSWL.bit, $addr16	PC ← addr16 if PSW_L.bit = 1 then reset PSW_L.bit	12(7)	3						
BFSET	saddr.bit, $addr16	PC ← addr16 if (saddr.bit) = 0 then set (saddr.bit)	12(7)	4						
	sfr.bit, $addr16	PC ← addr16 if (sfr.bit) = 0 then set (sfr.bit)	12(7)	4						
	A.bit, $addr16	PC ← addr16 if A.bit = 0 then set A.bit	11(7)	4						
	X.bit, $addr16	PC ← addr16 if X.bit = 0 then set X.bit	11(7)	3						
	PSWH.bit, $addr16	PC ← addr16 if PSW_H.bit = 0 then set PSW_H.bit	12(7)	3						
	PSWL.bit, $addr16	PC ← addr16 if PSW_L.bit = 0 then set PSW_L.bit	12(7)	3						
DBNZ	r2, $addr16	r2 ← r2 − 1; then PC ← addr16 if r2 ≠ 0	8(5)	2						
	saddr, $addr 16	(saddr) ← (saddr) − 1; then PC ← addr16 if saddr ≠ 0	7(6)	3						
BRKCS	RBn	PC_H ⟷ R5; PC_L ⟷ R4; R7 ⟷ PSW_H; R6 ⟷ PSW_L; RBS2-0 ← n; RSS ← 0; IE ← 0	13	2						
RETCS	!addr16	PC_H ← R5; PC_L ← R4; R4, R5 ← (addr16); PSW_H ← R7; PSW_L ← R6; EOS ← 0	6	3						
MOVM	(DE+), A	(DE+) ← A; C ← C − 1, End if C = 0	2 + 7n	2						
	(DE−), A	(DE−) ← A; C ← C − 1, End if C = 0	2 + 7n	2						
MOVBK	(DE+), (HL+)	(DE+) ← (HL+); C ← C − 1, End if C = 0	2 + 10n	2						
	(DE−), (HL−)	(DE−) ← (HL−); C ← C − 1, End if C = 0	2 + 10n	2						
XCHM	(DE+), A	(DE+) ⟷ A; C ← C − 1, End if C = 0	2 + 12n	2						
	(DE−), A	(DE−) ⟷ A; C ← C − 1, End if C = 0	2 + 12n	2						

4

NEC

μPD78310/312

T-49-19-08

Instruction Set (cont)

Mnemonic	Operand	Operation	States	Bytes	Flags S	Z	AC	P/V	SUB	CY
XCHBK	(DE+), (HL+)	(DE+) ⟷ (HL+); C ← C − 1, End if C = 0	2 + 15n	2						
	(DE−), (HL−)	(DE−) ⟷ (HL−); C ← C − 1, End if C = 0	2 + 15n	2						
CMPME	(DE+), A	(DE+) − A; C ← C − 1, End if C = 0 or Z = 0	2 + 8n	2	X	X	X	V	1	X
	(DE−), A	(DE−) − A; C ← C − 1, End if C = 0 or Z = 0	2 + 8n	2	X	X	X	V	1	X
CMPBKE	(DE+), (HL+)	(DE+) − (HL+); C ← C − 1, End if C = 0 or Z = 0	2 + 11n	2	X	X	X	V	1	X
	(DE−), (HL−)	(DE−) − (HL−); C ← C − 1, End if C = 0 or Z = 0	2 + 11n	2	X	X	X	V	1	X
CMPMNE	(DE+), A	(DE+) − A; C ← C − 1, End if C = 0 or Z = 1	2 + 8n	2	X	X	X	V	1	X
	(DE−), A	(DE−) − A; C ← C − 1, End if C = 0 or Z = 1	2 + 8n	2	X	X	X	V	1	X
CMPBKNE	(DE+), (HL+)	(DE+) − (HL+); C ← C − 1, End if C = 0 or Z = 1	2 + 11n	2	X	X	X	V	1	X
	(DE−), (HL−)	(DE−) − (HL−); C ← C − 1, End if C = 0 or Z = 1	2 + 11n	2	X	X	X	V	1	X
CMPMC	(DE+), A	(DE+) − A; C ← C − 1, End if C = 0 or CY = 0	2 + 8n	2	X	X	X	V	1	X
	(DE−), A	(DE−) − A; C ← C − 1, End if C = 0 or CY = 0	2 + 8n	2	X	X	X	V	1	X
CMPBKC	(DE+), (HL+)	(DE+) − (HL+); C ← C − 1, End if C = 0 or CY = 0	2 + 11n	2	X	X	X	V	1	X
	(DE−), (HL−)	(DE−) − (HL−); C ← C − 1, End if C = 0 or CY = 0	2 + 11n	2	X	X	X	V	1	X
CMPMNC	(DE+), A	(DE+) − A; C ← C − 1, End if C = 0 or CY = 1	2 + 8n	2	X	X	X	V	1	X
	(DE−), A	(DE−) − A; C ← C − 1, End if C = 0 or CY = 1	2 + 8n	2	X	X	X	V	1	X
CMPBKNC	(DE+), (HL+)	(DE+) − (HL+); C ← C − 1, End if C = 0 or CY = 1	2 + 11n	2	X	X	X	V	1	X
	(DE−), (HL−)	(DE−) − (HL−); C ← C − 1, End if C = 0 or CY = 1	2 + 11n	2	X	X	X	V	1	X
MOV	STBC, #byte	STBC ← byte	5	4						
	WDM, #byte	WDM ← byte	5	4						
SWRS		RSS ← $\overline{RSS}$	3	1						
SEL	RBn	RSS ← 0; RBS2-0 ← n	3	2						
	RBn, ALT	RSS ← 1; RBS2-0 ← n	3	2						
NOP		No Operation	3	1						
EI		IE ← 1 (Enable Interrupt)	3	1						
DI		IE ← 0 (Disable Interrupt	3	1						