Description

The family of single-chip microcomputers covered by this data sheet includes the following types:

μPD78C10	μPD78C10A	μPD78CG14
μPD78C11	μPD78C11A	μPD78CP14
μPD78C14	μPD78C12A	
	μPD78C14A	

These microcomputers integrate sophisticated on-chip peripheral functions normally provided by external components. Their internal 16-bit ALU and data paths, combined with a powerful instruction set and addressing, make the devices appropriate in data processing as well as control applications.

The devices integrate a 16-bit ALU, 4K-, 8K-, or 16K-byte ROM, 256-byte RAM, an eight channel A/D converter, a multifunction 16-bit timer/event counter, two 8-bit timers, a USART, and two zero-cross detect inputs on a single die, allowing their use in fast, high-end processing applications. This involves analog signal interface and processing.

The μPD78C1x/C1xA/Cx14 family includes: 4K-, 8K-, and 16K-byte mask ROM devices, embedded with a custom customer program; ROMless devices for use with up to 64K-bytes of external memory; 16K-byte piggyback EPROM device for prototyping; 16K-byte EPROM or OTP ROM devices for prototyping and low-volume production. The μPD78C11A/C12A/C14A also have mask optional pullup resistors available on ports A, B, and C.

Features

- CMOS technology
  - 25 mA operating current (78C10/C10A/C11/C11A/C12A)
  - 30 mA operating current (78C14/C14A)
- Complete single-chip microcomputer
  - 16-bit ALU
  - 4K, 8K, or 16K x 8 ROM
  - 256-byte RAM
- 44 I/O lines
- Mask optional pullup resistors
  - Ports A, B, and C
  - μPD78C11A/C12A/C14A only
- Two zero-cross detect inputs
- Two 8-bit timers
- Expansion capabilities
  - 8085A-like bus
  - 60K-byte external memory address range
- Eight-channel, 8-bit A/D converter
  - Autoscan mode
  - Channel select mode
- Full-duplex USART
  - Synchronous and asynchronous
- 159 instructions
  - 16-bit arithmetic, multiply, and divide
  - HALT and STOP instructions
- 0.8-μs instruction cycle time (15-MHz operation)
- Prioritized interrupt structure
  - Three external
  - Eight internal
- Standby function
- On-chip clock generator

5

Ordering Information

Part Number	Package	ROM
μPD78C10CW	64-pin plastic SDIP	ROMless
μPD78C10G-36	64-pin plastic QUIP	
μPD78C10G-1B	64-pin plastic QFP (Resin thickness 2.05 mm)	
μPD78C10GF-3BE	64-pin plastic QFP (Resin thickness 2.7 mm)	
μPD78C10L	68-pin PLCC	
μPD78C10ACW	64-pin plastic SDIP	ROMless
μPD78C10AGF-3BE	64-pin plastic QFP	
μPD78C10AGQ-36	64-pin plastic QUIP	
μPD78C10AL	68-pin PLCC	
μPD78C11CW-xxx	64-pin plastic SDIP	4K mask ROM
μPD78C11G-xxx-36	64-pin plastic QUIP	
μPD78C11G-xxx-1B	64-pin plastic QFP (Resin thickness 2.05 mm)	
μPD78C11GF-xxx-3BE	64-pin plastic QFP (Resin thickness 2.7 mm)	
μPD78C11L-xxx	68-pin PLCC	

Ordering Information (cont)

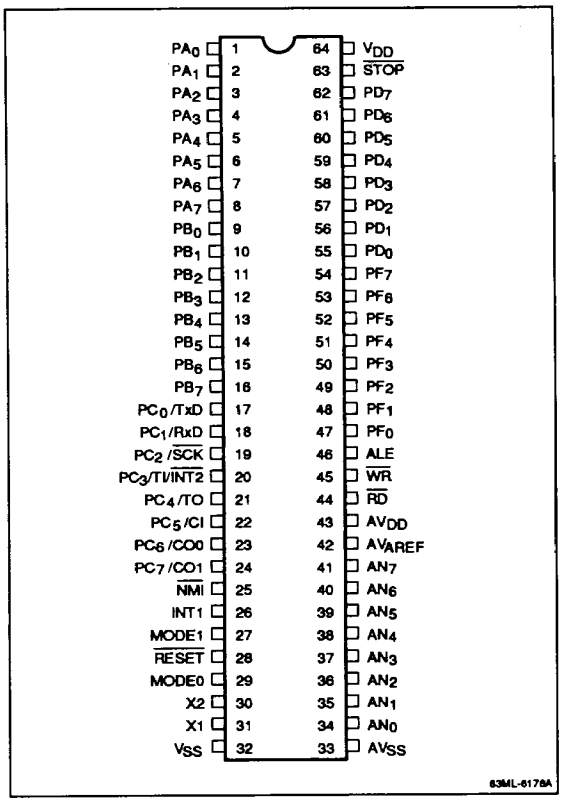
Part Number	Package	ROM
μPD78C11ACW-xxx	64-pin plastic SDIP	4K mask ROM
μPD78C11AGF-xxx-3BE	64-pin plastic QFP	
μPD78C11AGQ-xxx-36	64-pin plastic QUIP	
μPD78C11AL-xxx	68-pin PLCC	
μPD78C12ACW-xxx	64-pin plastic SDIP	8K mask ROM
μPD78C12AGF-xxx-3BE	64-pin plastic QFP	
μPD78C12AG-xxx-36	64-pin plastic QUIP	
μPD78C12AL-xxx	68-pin PLCC	
μPD78C14CW-xxx	64-pin plastic SDIP	16K mask ROM
μPD78C14G-xxx-36	64-pin plastic QUIP	
μPD78C14G-xxx-1B	64-pin plastic QFP (Resin thickness 2.05 mm)	
μPD78C14GF-xxx-3BE	64-pin plastic QFP (Resin thickness 2.7 mm)	
μPD78C14L-xxx	68-pin PLCC	
μPD78C14AG-xxx-AB8	64-pin plastic QFP (Interpin pitch 0.8 mm)	16K mask ROM
μPD78CG14E	64-pin ceramic piggyback QUIP	4/8/16K piggyback EPROM
μPD78CP14CW	64-pin plastic SDIP	16K OTP ROM
μPD78CP14G-36	64-pin plastic QUIP	
μPD78CP14GF-3BE	64-pin plastic QFP	
μPD78CP14L	68-pin PLCC	
μPD78CP14DW	64-pin ceramic SDIP with window	16K UV EPROM
μPD78CP14R	64-pin ceramic QUIP with window	

Notes:

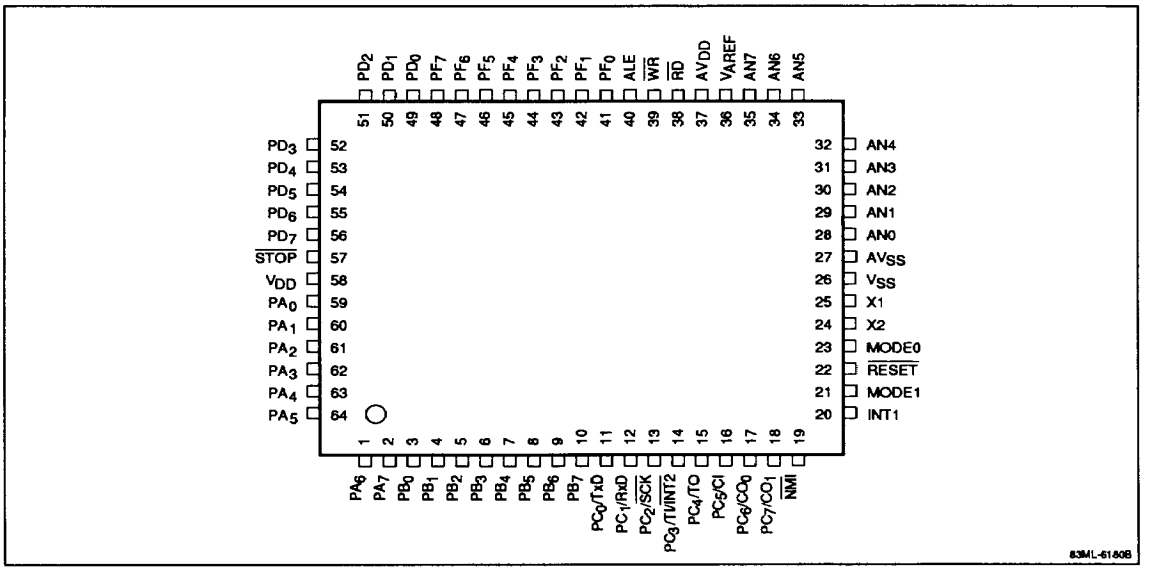
(1) xxx indicates ROM code suffix.

Pin Configurations

64-Pin QUIP or SDIP (Plastic or Ceramic)

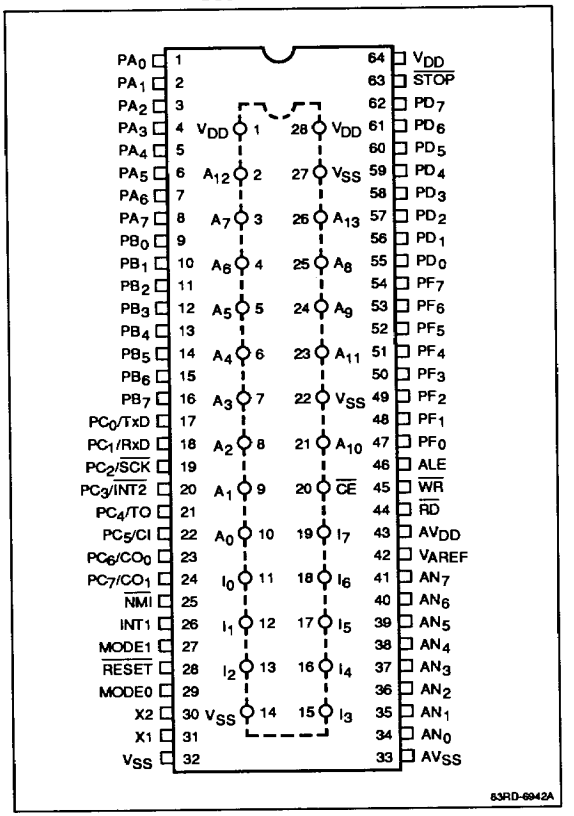


64-Pin Plastic QFP

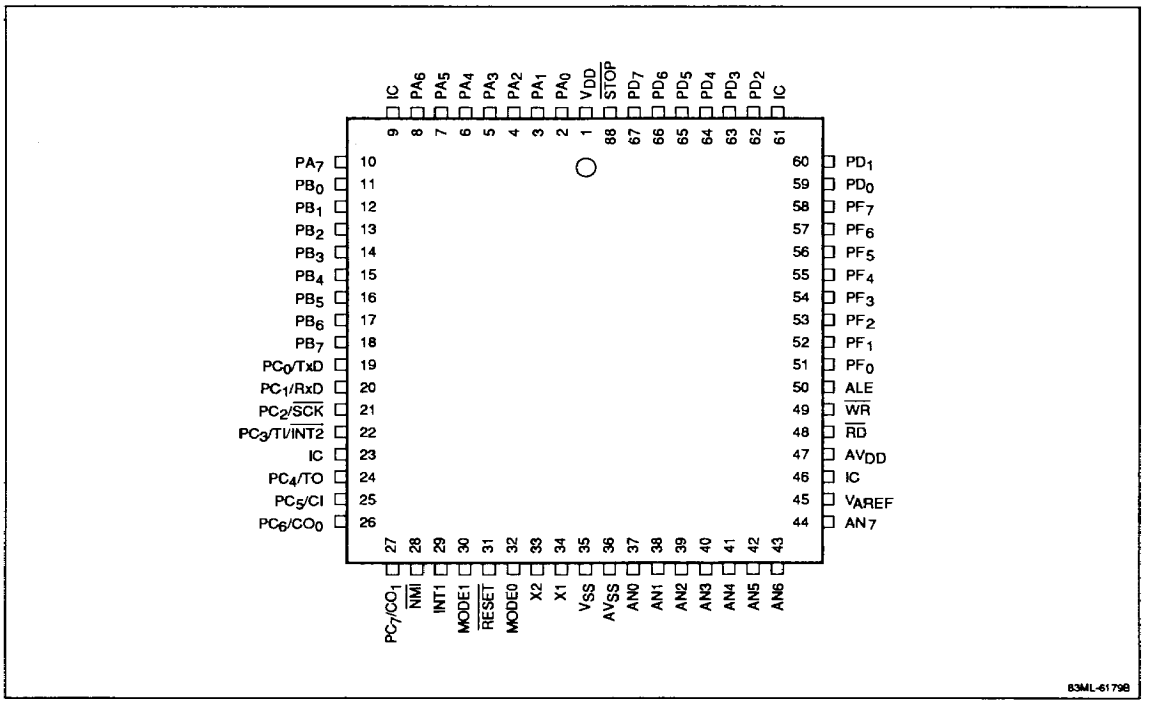


5

64-Pin Ceramic Piggyback QUIP



68-Pin PLCC



5

Pin Identification

Symbol	Function
ALE	Address latch enable output
AN0-AN7	A/D converter analog inputs 0-7
INT1	Interrupt request 1 input
MODE0	Mode 0 input; I/O memory output
MODE1	Mode 1 input
$\overline{NMI}$	Nonmaskable interrupt input
PA_0-PA_7	Port A I/O
PB_0-PB_7	Port B I/O
PC_0/TxD	Port C I/O line 0; transmit data output
PC_1/RxD	Port C I/O line 1; receive data input
$PC_2/\overline{SCK}$	Port C I/O line 2; serial clock I/O
PC_3/TI/$\overline{INT2}$	Port C I/O line 3; timer input; interrupt request 2 input
PC_4/TO	Port C I/O line 4; timer output
PC_5/CI	Port C I/O line 5; counter input
PC_6, PC_7/ CO_0,CO_1	Port C I/O lines 6, 7; counter outputs 0, 1
PD_0-PD_7	Port D I/O; expansion memory address, data bus (bits AD_0-AD_7)
PF_0-PF_7	Port F I/O; expansion memory address, (bits AB_8-AB_{15})
$\overline{RD}$	Read strobe output
$\overline{RESET}$	Reset input
$\overline{STOP}$	Stop mode control input
V_{AREF}	A/D converter reference voltage
$\overline{WR}$	Write strobe output
X1, X2	Crystal connections 1, 2
AV_{DD}	A/D converter power supply voltage
AV_{SS}	A/D converter power supply ground
V_{DD}	5 V power supply
V_{SS}	Ground
IC	Internal connection

Pin Identification
μPD78CG14E Upper EPROM Pins

Symbol	Pin	Function
A_0-A_{13}	2-10, 21 23-26	14-bit program counter (PC_0-PC_{13}) output used as 27C256/27C256A address signals
$\overline{CE}$	20	Chip enable signal for 27C256/27C256A; high-level output (during STOP or HALT), otherwise, low-level output
I_0-I_7	11-13 15-19	8-bit input of data read from 27C256/27C256A
V_{DD}	1	Same potential as lower V_{DD} pin; V_{CC} power supply line (V_{PP}) for 27C256/27C256A
V_{DD}	28	Same potential as lower V_{DD} pin; V_{CC} power supply line (V_{CC}) for 27C256/27C256A
V_{SS}	14	Same potential as lower V_{SS} pin connected to the 27C256/27C256A GND pin
V_{SS}	22	Same potential as lower V_{SS} pin; $\overline{OE}$ signal (always low) input to 27C256/27C256A
V_{SS}	27	Same potential as lower V_{SS} pin; A_{14} signal (always low) input to 27C256/27C256A

PIN FUNCTIONS

ALE (Address Latch Enable)

The ALE output is used to latch the address of PD_0-PD_7 into an external latch.

AN0-AN7 (Analog Inputs)

These are the eight analog inputs to the A/D converter. AN4-AN7 can also be used as a digital input for falling edge detection.

CI (Counter Input)

External pulse input to timer/event counter.

CO_0, CO_1 (Counter Outputs)

Programmable waveform outputs based on timer/event counter.

INT1 (Interrupt Request 1)

INT1 is a rising edge triggered, maskable interrupt input. It is also an ac-input, zero-cross detection terminal.

If the optional pullup resistor is specified for this pin on the μPD78C11A/C12A/C14A, the zero-cross detection circuitry will not function.

$\overline{INT2}$ (Interrupt Request 2)

$\overline{INT2}$ is a falling edge triggered, maskable interrupt input. It is also an ac-input, zero-cross detection terminal.

MODE0, MODE1 (Mode 0, 1)

The MODE0 and MODE1 inputs select the amount of external memory. MODE0 outputs the $\overline{IO}$ signal, and MODE1 outputs the $\overline{M1}$ signal. An external pullup resistor to V_{DD} is required if the input is to be a logic high.

The value of this pullup resistor, R, is dependent on t_{CYC} and is calculated as follows: R in KΩ is $4 \leq R \leq 0.4\, t_{CYC}$ where t_{CYC} is in ns units.

$\overline{NMI}$ (Nonmaskable Interrupt)

Falling edge, Schmitt triggered nonmaskable interrupt input.

PA_0-PA_7 (Port A)

Port A is an 8-bit three-state port. Each bit is independently programmable as either input or output. Reset makes all lines of port A inputs. Mask optional pullup resistors are available on the μPD78C11A/C12A/C14A.

PB_0-PB_7 (Port B)

Port B is an 8-bit three-state port. Each bit is independently programmable as either input or output. Reset makes all lines of port B inputs. Mask optional pullup resistors are available on the μPD78C11A/C12A/C14A.

PC_0-PC_7 (Port C)

Port C is an 8-bit three-state port. Each bit is independently programmable as either input or output. Alternatively, the lines of port C can be used as control lines for the USART, interrupts, and timer. Reset makes all lines of port C inputs. Mask optional pullup resistors are available on the μPD78C11A/C12A/C14A.

PD_0-PD_7 (Port D)

Port D is an 8-bit three-state port. It can be programmed as either 8 bits of input or 8 bits of output. When external expansion memory is used, port D acts as the multiplexed address/data bus.

PF_0-PF_7 (Port F)

Port F is an 8-bit three-state port. Each bit is independently programmable as either input or output. When external expansion memory is used, port F outputs the high-order address bits.

$\overline{RD}$ (Read Strobe)

The three-state $\overline{RD}$ output goes low to gate data from external devices onto the data bus. $\overline{RD}$ goes high during reset.

$\overline{RESET}$ (Reset)

When the Schmitt-triggered $\overline{RESET}$ input is brought low, it initializes the device.

5

RxD (Receive Data)

Serial data input terminal.

$\overline{SCK}$ (Serial Clock)

Output for the serial clock when internal clock is used. Input for serial clock when external clock is used.

$\overline{STOP}$ (STOP Mode Control Input)

A low-level input on $\overline{STOP}$ (Schmitt-triggered input) stops the system clock oscillator.

TI (Timer Input)

Timer input terminal.

TO (Timer Output)

The output of TO is a square wave with a frequency determined by the timer/counter.

TxD (Transmit Data)

Serial data output terminal.

V_{AREF} (A/D Converter Reference)

V_{AREF} sets the upper limit for the A/D conversion range.

$\overline{WR}$ (Write Strobe)

The three-state $\overline{WR}$ output goes low to indicate that the data bus holds valid data. It is a strobe signal for external memory or I/O write operations. $\overline{WR}$ goes high during reset.

X1, X2 (Crystal Connections)

X1 and X2 are the system clock crystal oscillator terminals. X1 is the input for an external clock.

AV_{DD} (A/D Converter Power)

This is the power supply voltage for the A/D converter.

AV_{SS} (A/D Converter Power Ground)

AV_{SS} is the ground potential for the A/D converter power supply.

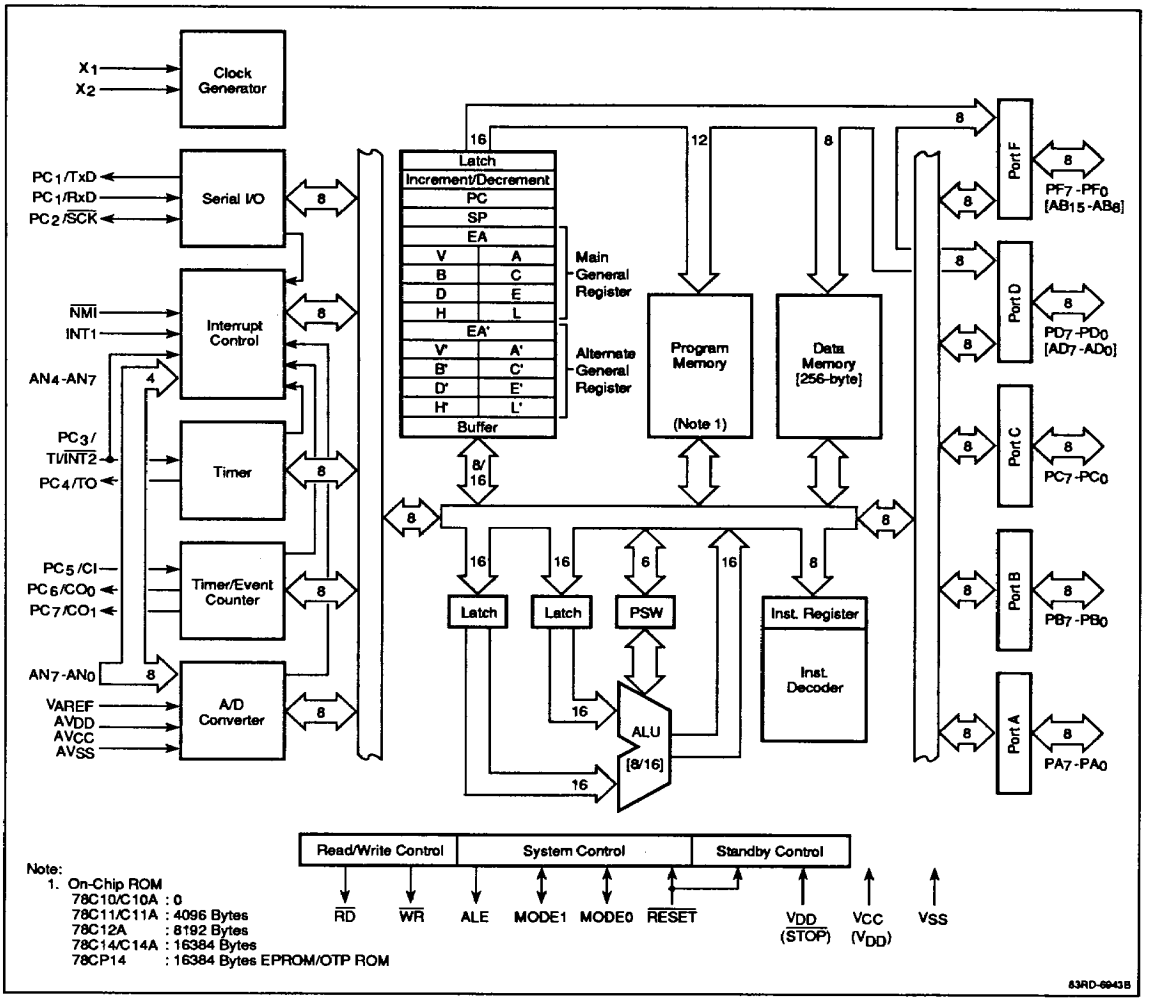
V_{DD} (Power Supply)

V_{DD} is the +5-volt power supply.

V_{SS} (Ground)

Ground potential.

Block Diagram



FUNCTIONAL DESCRIPTION

Memory Map

The μPD78C1x/C1xA/Cx14 family can directly address up to 64K bytes of memory. Except for the on-chip ROM (or PROM) and RAM (FF00H-FFFFH), any memory location can be used as ROM or RAM. The memory map, figure 1, defines the 0 to 64K-byte memory space for the μPD78C1x/C1xA/Cx14 family.

The μPD78CG14 and the μPD78CP14 can be programmed in software to have 4K, 8K, or 16K bytes of internal program memory. This programming is transparent to the ROM-based device, allowing easy transfer of code to a ROM-based device.

Input/Output

The μPD78C1x/C1xA/Cx14 family has 44 digital I/O lines, five 8-bit ports (ports A, B, C, D, F), and four digital input lines (AN4-AN7).

Analog Input Lines. AN0-AN7 are configured as analog input lines for the on-chip A/D converter. Lines AN4-AN7 can be used as digital input lines for falling edge detection.

Port A, Port B, Port C, Port F. Each line of these ports can be individually programmed as an input or output. When used as I/O ports, all have latched outputs and high-impedance inputs. On the μPD78C11A/C12A/C14A, mask optional pullup resistors are available for ports A, B, and C.

Port D. Port D can be programmed as a byte input or a byte output.

Control Lines. Under software control, each line of port C can be configured individually as a control line for the serial interface, timer, and timer/counter or as an I/O port.

Memory Expansion. In addition to the single-chip operation mode, the μPD78C1x/C1xA/Cx14 family has four memory expansion modes. Under software control, port D can provide a multiplexed low-order address and data bus; port F can provide a high-order address bus. Table 1 shows the relation between memory expansion modes and the pin configurations of port D and port F.

Table 1. Memory Expansion Modes and Port Configurations

Memory Expansion	Port	Port Configuration
None	Port D	I/O port
	Port F	I/O port
256 bytes	Port D	Multiplexed address/data bus
	Port F	I/O port
4K bytes	Port D	Multiplexed address/data bus
	Port F (PF_0-PF_3)	Address bus
	Port F (PF_4-PF_7)	I/O port
16K bytes	Port D	Multiplexed address/data bus
	Port F (PF_0-PF_5)	Address bus
	Port F (PF_6-PF_7)	I/O port
60K bytes	Port D	Multiplexed address/data bus
	Port F	Address bus

Timers

The two 8-bit timers may be programmed independently or cascaded as a 16-bit timer. The timer can be software set to increment at intervals of four machine cycles (0.8 μs at 15-MHz operation) or 128 machine cycles (25.6 μs at 15-MHz), or to increment on receipt of a pulse at TI. Figure 2 is the block diagram for the timer.

Timer/Event Counter

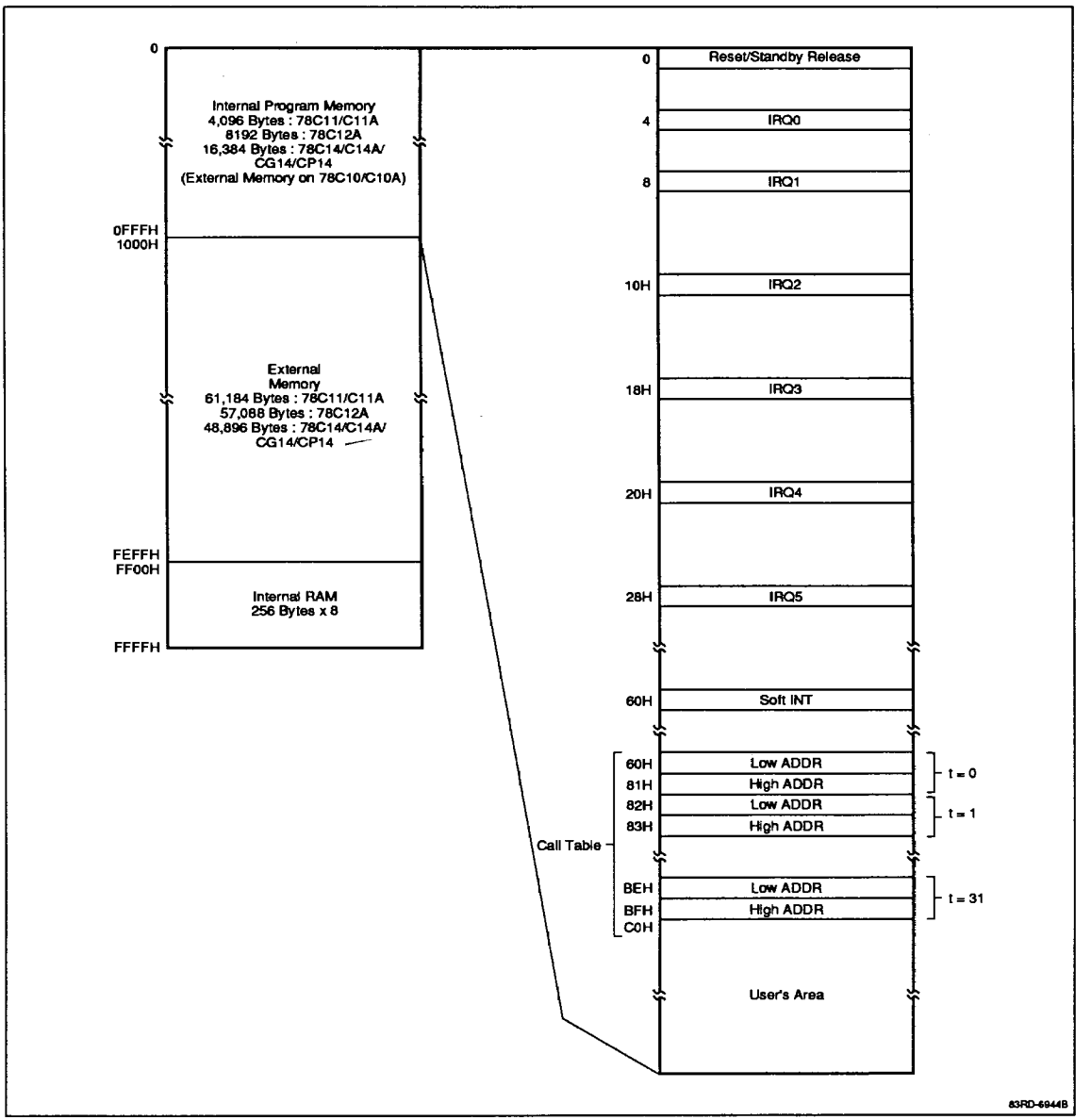
The 16-bit multifunctional timer/event counter (figure 3) can be used for the following operations:

- Interval timer
- External event counter
- Frequency measurement
- Pulse width measurement
- Programmable frequency and duty cycle waveform output
- Single pulse output

8-Bit A/D Converter

- 8 input channels
- 4 conversion result registers
- 2 powerful operation modes
  - — Autoscan mode
  - — Channel select mode
- Successive approximation technique
- Absolute accuracy: 0.6% FSR ± 1/2 LSB
- Conversion range: 0 to 5 V
- Conversion time: 38.4 μs
- Interrupt generation

Figure 1. Memory Map



5

Figure 2. Timer Block Diagram

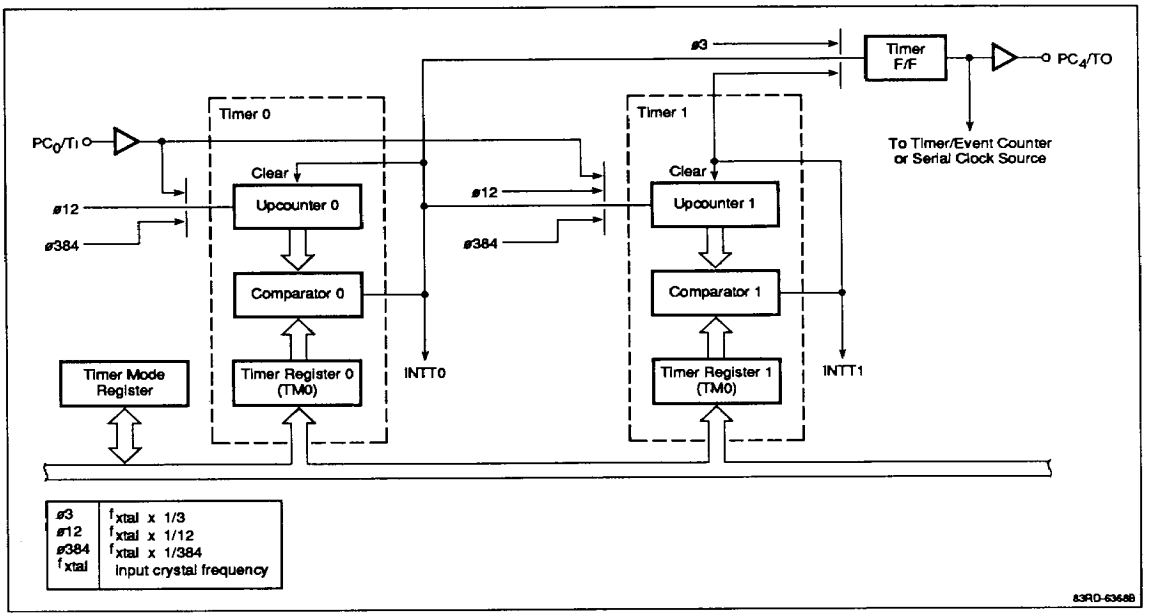
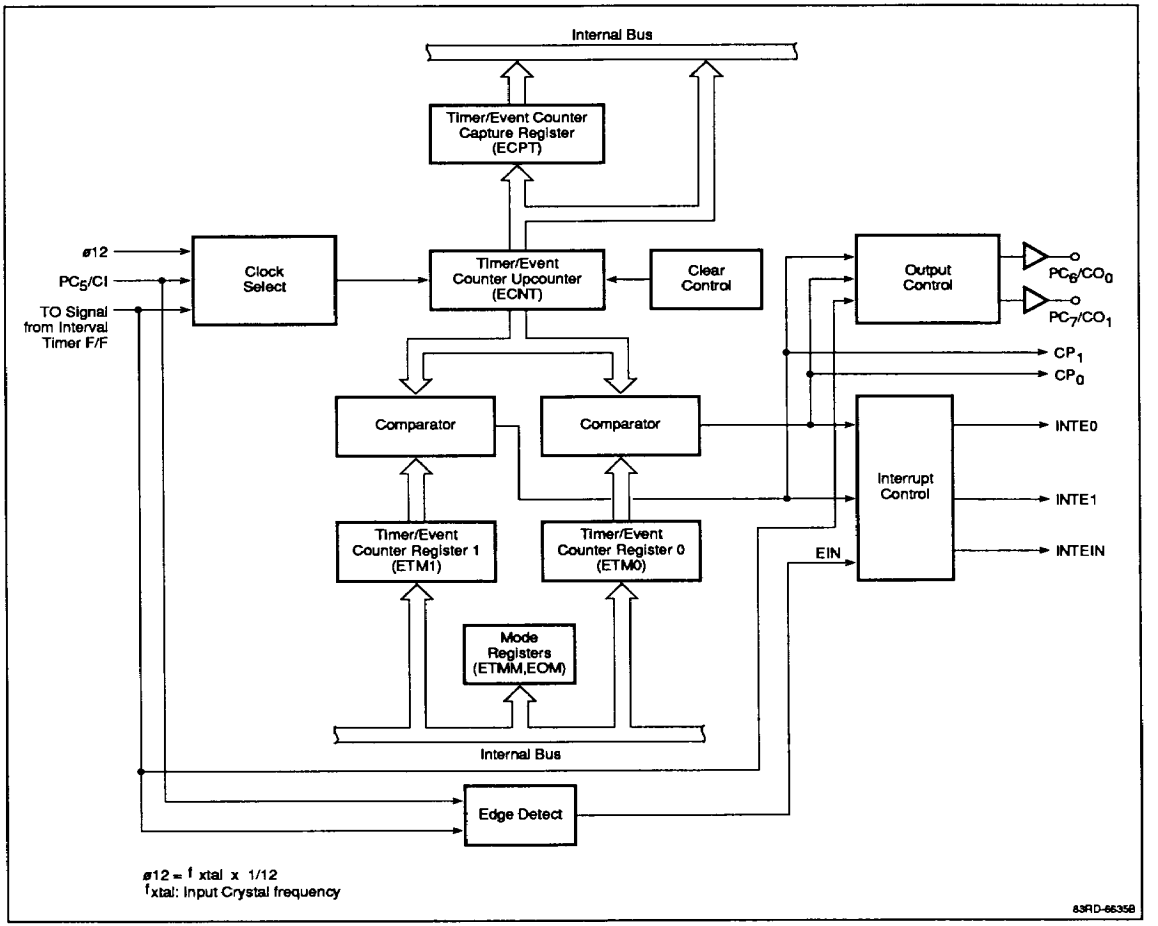


Figure 3. Block Diagram for the Timer/Event Counter



Analog/Digital Converter

The μPD78C1x/C1xA/Cx14 family features an 8-bit, high-speed, high accuracy A/D converter. The A/D converter is made up of a 256-resistor ladder and a successive approximation register (SAR). There are four conversion result registers (CR0-CR3).

The eight-channel analog input may be operated in either of two modes. In the select mode, the conversion value of one analog input is sequentially stored in CR0-CR3. In the scan mode, either the upper four channels or the lower four channels may be specified. Then those four channels will be consecutively selected and the conversion results stored sequentially in the four conversion result registers.

Figure 4 is the block diagram for the A/D converter. To stop the operation of the A/D converter and thus reduce power consumption, set $V_{AREF} = 0$ V.

Interrupt Structure

There are 12 interrupt sources in the μPD78C1x/C1xA/Cx14 family of chips. Three are external interrupts and nine are internal. Table 2 shows 11 interrupt sources divided into seven priority levels where IRQ0 is the highest and IRQ6 is the lowest. See figure 5.

Figure 4. A/D Converter Block Diagram

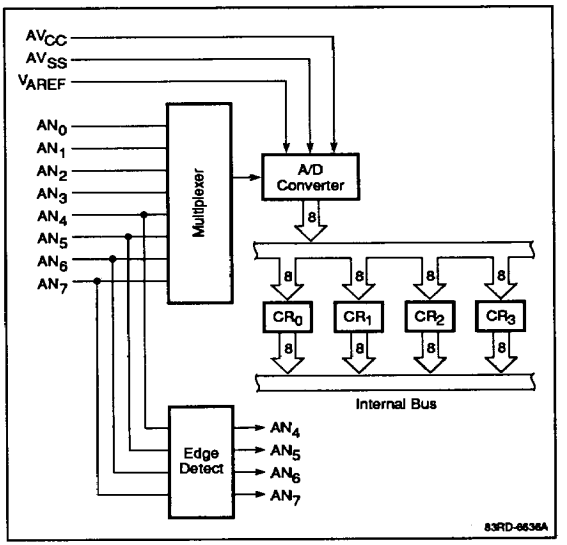
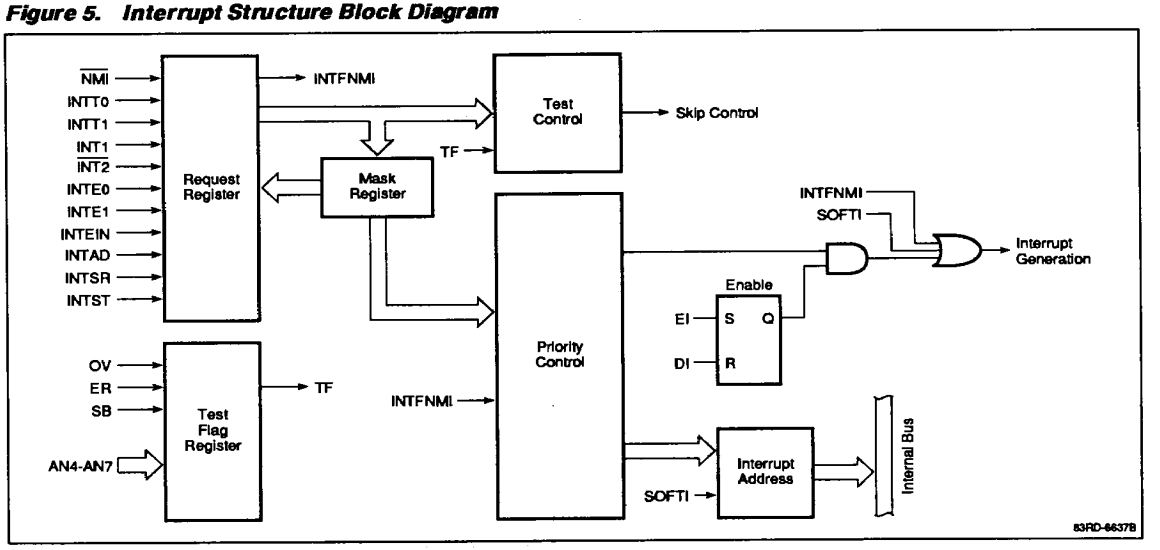


Figure 5. Interrupt Structure Block Diagram



Standby Functions

The μPD78C1x/C1xA/Cx14 family has two standby modes: HALT and STOP. The HALT mode reduces power consumption to 50% of normal operating requirements, while maintaining the contents of on-chip registers, RAM, and control status. The system clock and on-board peripherals continue to operate, but the CPU stops executing instructions. The HALT mode is initiated by executing the HLT instruction. The HALT mode can be released by any nonmasked interrupt or by $\overline{\text{RESET}}$.

The STOP mode reduces power consumption to less than 0.1% of normal operating requirements. There are two STOP modes: type A and type B.

Type A is initiated by executing a STOP instruction. If V_{DD} is held above 2.5 V, the on-board RAM is saved. The oscillator is stopped. The STOP mode can be released by an input on $\overline{\text{NMI}}$ or $\overline{\text{RESET}}$. The user can program oscillator stabilization time up to 52.4 ms via timer 1. By checking the standby flag (SB), the user can determine whether the processor has been in the standby mode or has been powered up.

Type B is initiated by inputting a low level on the $\overline{\text{STOP}}$ input. The RAM contents are saved if V_{DD} is held above 2.5 V. The oscillator is stopped. The STOP mode is released by raising $\overline{\text{STOP}}$ to a high level. The oscillator stabilization time is fixed at 52.4 ms; 52.4 ms after $\overline{\text{STOP}}$ is raised, instruction execution will automatically begin at location 0. You can increase the stabilization time by holding $\overline{\text{RESET}}$ low for the required time period.

Universal Serial Interface

The serial interface can operate in one of three modes: synchronous, asynchronous, and I/O interface. The I/O interface mode transfers data MSB first, for easy interfacing to certain NEC peripheral devices. Synchronous and asynchronous modes transfer data LSB first. Synchronous operation offers two modes of data reception: search and nonsearch. In the search mode, data is transferred one bit at a time from the serial register to the receive buffer. This allows a software search for a sync character. In the nonsearch mode, data transfer from the serial register to the transmit buffer occurs eight bits at a time. Figure 6 shows the universal serial interface block diagram.

Zero-Crossing Detector

The INT1 and $\overline{\text{INT2}}$ terminals (used common to TI and PC_3) can detect the zero-crossing point of low-frequency AC signals. When driven directly, these pins respond as a normal digital input. Figure 7 shows the zero-crossing detection circuitry.

5

Table 2. Interrupt Sources

Interrupt Request	Interrupt Address	Type of Interrupt	Internal/External
IRQ0	4	$\overline{\text{NMI}}$ (Nonmaskable interrupt)	External
IRQ1	8	INTT0, INTT1 (Coincidence signals from timers 0, 1)	Internal
IRQ2	16	INT1, $\overline{\text{INT2}}$ (Maskable interrupts)	External
IRQ3	24	INTE0, INTE1 (Coincidence signals from timer/ event counter)	Internal
IRQ4	32	INTEIN (Falling signal of CI or TO into the timer/ event counter)	Internal or External
		INTAD (A/D converter interrupt)	Internal
IRQ5	40	INTSR (Serial receive interrupt)	Internal
		INST (Serial send interrupt)	
IRQ6	96	SOFTI instruction	Internal

Figure 6. Universal Serial Interface Block Diagram

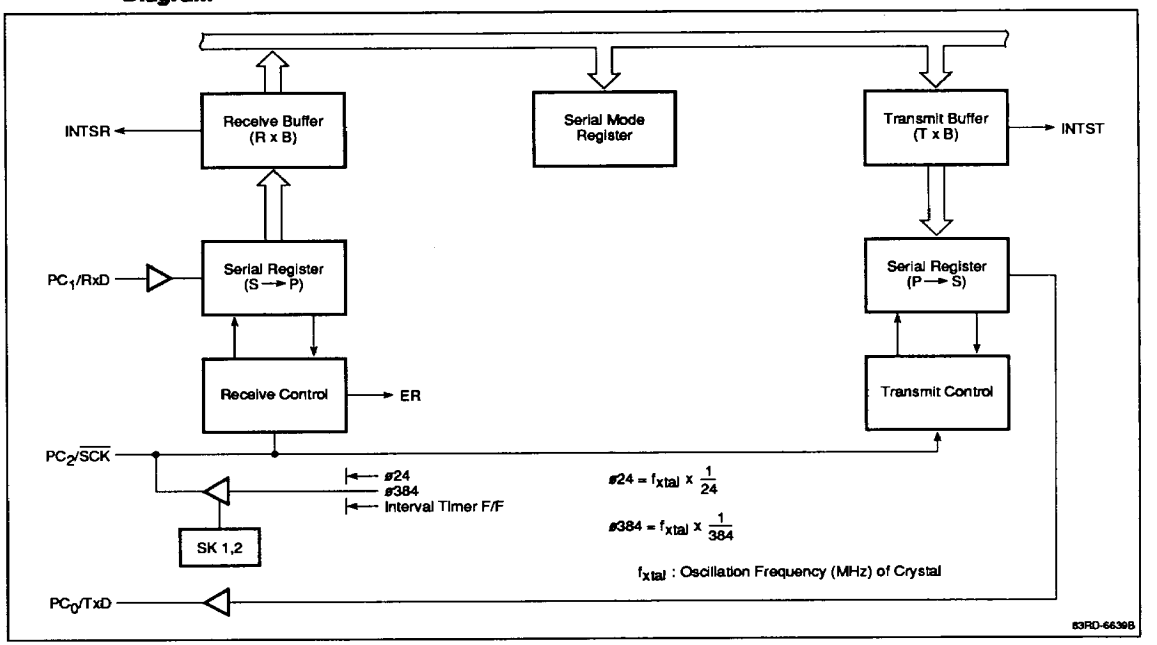
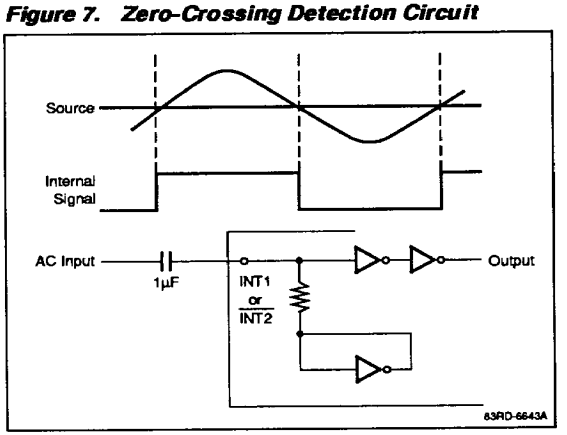


Figure 7. Zero-Crossing Detection Circuit



The zero-crossing detection capability allows you to make the 50-60 Hz power signal the basis for system timing and to control voltage phase-sensitive devices.

To use the zero-cross detection mode, an AC signal of 1.0 to 1.8 V (peak-to-peak) and a maximum frequency of 1 kHz is coupled through an external capacitor to the INT1 and $\overline{INT2}$ pins.

For the INT1 pin, the internal digital state is sensed as a 0 until the rising edge crosses the average DC level, when it becomes a 1 and an INT1 interrupt is generated.

For the $\overline{INT2}$ pin, the state is sensed as a 1 until the falling edge crosses the average DC level, when it becomes a 0 and $\overline{INT2}$ is generated.

ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings

T_A = 25°C

Power supply voltage, V_{DD}	−0.5 to +7.0 V
Power supply voltage, AV_{DD}	AV_{SS} to V_{DD} +0.5 V
Power supply voltage, AV_{SS}	−0.5 to +0.5 V
Power supply voltage, V_{PP} (μPD78CP14 only)	−0.5 to +13.5
Input voltage, V_I	−0.5 to V_{DD} +.5 V
$\overline{STOP}$ pin (μPD78CP14 only)	−0.5 to +13.5 V
Output voltage, V_O	−0.5 to V_{DD} +.5 V
Output current, low; I_{OL}	
Each output pin	4.0 mA
Total	100 mA
Output current, high; I_{OH}	
Each output pin	−2.0 mA
Total	−50 mA
Reference input voltage, VA_{REF}	−0.5 to AV_{DD} +0.3 V
Operating temperature, T_{OPR} $f_{XTAL} \leq$ 15 MHz	−40 to +85°C
Storage temperature, T_{STG}	−65 to +150°C

Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The device should be operated within the limits specified under DC and AC Characteristics.

Capacitance

T_A = 25°C; V_{DD} = V_{SS} = 0 V

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

5

Oscillation Characteristics

T_A = −40 to +85°C; V_{DD} = AV_{DD} = 5 V ± 10% (±5% μPD78CP14);
V_{SS} = AV_{SS} = 0 V; V_{DD} − 0.8 V ≤ AV_{DD} ≤ V_{DD}; 3.4 V ≤ V_{AREF} ≤ AV_{DD}

Resonator	Recommended Circuit	Parameter	Min	Typ	Max	Unit	Conditions
Ceramic resonator (Note 1) or XTAL (Note 2)	(Note 3)	Oscillation frequency (f_{XX})	4		15	MHz	A/D converter not used
			5.8		15	MHz	A/D converter used
			6		15	MHz	μPD78CP14 only
External clock	(Note 4)	X1 input frequency (f_X)	4		15	MHz	A/D converter not used
			5.8		15	MHz	A/D converter used
			6		15	MHz	μPD78CP14 only
		X1 input, rise, fall time (t_r, t_f)	0		20	ns	
		X_1 input low- and high-level width ($t_{\phi L}$, $t_{\phi H}$)	20		250	ns	
			20		167	ns	μPD78CP14

Notes:

(1) Refer to the Resonator and Capacitance Requirements table for the recommended ceramic resonators.

(2) For XTAL, the following external capacitances are recommended: C1 = C2 = 10 pF

(3) For XTAL, see the Recommended XTAL or Ceramic Resonator Oscillation Circuit Diagram.

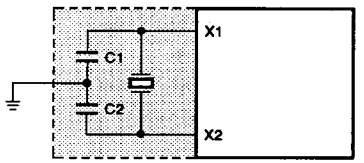
(4) See the following recommended external clock diagram.

When using an external crystal, it should be a parallel-resonant, fundamental mode, "AT cut" crystal. Capacitors C1 and C2 are required for frequency stability. The values of C1 and C2 (C1 = C2) can be calculated from the load capacitance (C_L), specified by the crystal manufacturer:

$$C_L = \frac{C1 \times C2}{C1 + C2} + C_S$$

Where C_S is any stray capacitance in parallel with the crystal such as the μPD78C10, μPD78C11, or μPD78C14 input capacitance between X1 and X2.

Recommended XTAL or Ceramic Resonator Oscillation Circuit Diagram

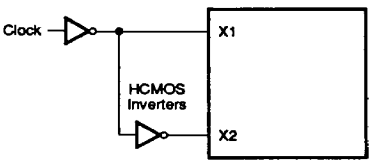


External oscillation circuit should be as close to the X1 and X2 pins as possible.

Do not place other signal lines in the shaded area.

83RD-6945A

Recommended External Clock Diagram



83RD-6946A

Resonator and Capacitance Requirements

T_A = −40 to +85°C

Manufacturer	Product Number	C1, C2 (pF)	Conditions
Murata	CSA15.0MX3	22	μPD78C10, 78C11, 78C14, 78C14A, 78CG14
	CSA10.0MT	30	
	CST10.0MT	Not required	
	CSA6.00MG	30	
	CST6.00MG	Not required	
	CSA12.0MT	30	Applies to all μPD78C1x/C1xA/CG14
	CST12.0MT	Not required	
	CSA15.00MX001	15	μPD78C10A/78C11A/78C12A
	CSA7.37MT	30	
	CST7.37MT	Not required	
TDK	FCR12.0MC	Not required	μPD78C10/78C11/78C14/ 78C14A/78CG14

5

DC Characteristics

T_A = −40° to +85°C; V_{DD} = +5.0 V ±10%; V_{DD} = +5.0 V ± 5% (μPD78C14 only); V_{SS} = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Input voltage, low	V_{IL1}	0		0.8	V	All except Note 1 inputs
	V_{IL2}	0		0.2 V_{DD}	V	Note 1 inputs
Input voltage, high	V_{IH1}	2.2		V_{DD}	V	All except X1, X2, and Note 1 inputs
	V_{IH2}	0.8 V_{DD}		V_{DD}	V	X1, X2, and Note 1 inputs
Output voltage, low	V_{OL}			0.45	V	I_{OL} = 2.0 mA
Output voltage, high	V_{OH}	V_{DD}−1.0			V	I_{OH} = 1.0 mA
		V_{DD}−0.5			V	I_{OH} = −100 μA
Data retention voltage	V_{DDDR}	2.5			V	STOP mode
Input current	I_{I1}			±200	μA	INT1 (Note 2); TI (PC_3) (Note 3); 0 V ≤ V_I ≤ V_{DD}
Input current (μPD78CG14 only)	I_{I2}			±200	μA	INT1 (Note 2); TI (PC_3) (Note 3); 0 V ≤ V_I ≤ V_{DD}
Input current (μPD78CG14 only)	I_{I3}			−300	μA	I_0–I_7 (upper input pin); V_I = 0
Input leakage current	I_{LI}			±10	μA	All except INT1, TI (PC_3), 0 V ≤ V_I ≤ V_{DD}
Output leakage current	I_{LO}			±10	μA	0 V ≤ V_O ≤ V_{DD}
AV_{DD} supply current	AI_{DD1}		0.5	1.3	mA	f = 15 MHz
	AI_{DD2}		10	20	μA	STOP mode
V_{DD} supply current	I_{DD1}		13	25	mA	Normal operation; f = 15 MHz; (μPD78C10/C10A/C11/C11A/C12A only)
	I_{DD2}		7	13	mA	HALT mode; f = 15 MHz; (μPD78C10/C10A/C11/C11A/C12A only)
	I_{DD3}		16	30	mA	Normal operation; f = 15 MHz (μPD78C14/C14A/CG14)
	I_{DD4}			32	mA	Normal operation; f = 15 MHz; (μPD78CP14 only)
	I_{DD5}		8	15	mA	HALT mode; f = 15 MHz; (μPD78C14/C14A/CG14/CP14 only)
Data retention current	I_{DDDR}		1	15	μA	V_{DDDR} = 2.5 V (Note 4)
				300		(μPD78CP14 only-Note 4)
			10	50	μA	V_{DDDR} = 5.0 V ±10% (Note 4)
				1	mA	(μPD78CP14 only-Note 4)
Pullup resistor	R_L	17	27	75	KΩ	Port A, B, C; 3.5 V ≤ V_{DD} ≤ 5.5 V; V_I = 0 V (μPD78C11A/C12A/C14A only)

Notes:

(1) Inputs $\overline{\text{RESET}}$, $\overline{\text{STOP}}$, $\overline{\text{NMI}}$, $\overline{\text{SCK}}$, INTP1, TI, and AN4-AN7.

(2) Assuming ZCM register is set to self-bias.

(3) Assuming ZCM register is set to self-bias and the MCC register is set to control mode.

(4) Hardware/software STOP mode and assuming ZCM register is set to self-bias not selected.

Serial Operation

Parameter	Symbol	Min	Max	Unit	Conditions
$\overline{SCK}$ cycle time	t_{CYK}	0.8		μs	$\overline{SCK}$ input (Notes 1, 3)
		0.4		μs	$\overline{SCK}$ input (Note 2)
		1.6		μs	$\overline{SCK}$ output (Note 3)
$\overline{SCK}$ width low	t_{KKL}	335		ns	$\overline{SCK}$ input (Notes 1, 3)
		160		ns	$\overline{SCK}$ input (Note 2)
		700		ns	$\overline{SCK}$ output (Note 3)
$\overline{SCK}$ width high	t_{KKH}	335		ns	$\overline{SCK}$ input (Notes 1, 3)
		160		ns	$\overline{SCK}$ input (Note 2)
		700		ns	$\overline{SCK}$ output (Note 3)
RxD setup time to $\overline{SCK}$ ↑	t_{RXK}	80		ns	(Note 1)
RxD hold time after $\overline{SCK}$ ↑	t_{KRX}	80		ns	(Note 1)
$\overline{SCK}$ ↓ TxD delay time	t_{KTX}		210	ns	(Note 1)

Notes:

(1) 1 x baud rate in synchronous or I/O interface mode.

(2) 16 x baud rate or 64 x baud rate in asynchronous mode.

(3) f_{XTAL} = 15 MHz.

5

Zero-Cross Characteristics

Parameter	Symbol	Min	Max	Unit	Condition
Zero-cross detection input	V_{ZX}	1	1.8	VAC_{p-p}	AC coupled 60 Hz sine wave
Zero-cross accuracy	A_{ZX}		±135	mV	
Zero-cross detection input frequency	f_{ZX}	0.05	1	kHz	

AC Characteristics (cont)

T_A = −40° to +85°C; V_{DD} = AV_{DD} = +5.0 V ±10% (±5% on μPD78CP14); V_{SS} = 0 V

Parameter	Symbol	Min	Max	Unit	Conditions
$\overline{RESET}$ pulse width high, low	t_{RSH}, t_{RSL}	10		μs	
$\overline{NMI}$ pulse width high, low	t_{NIH}, t_{NIH}	10		μs	
X1 input cycle time	t_{CYC}	66	250	ns	
			167	ns	(Note 1)
Address setup to ALE ↓	t_{AL}	30		ns	(Notes 2, 3)
Address hold to ALE ↓	t_{LA}	35		ns	(Notes 2, 3)
Address to $\overline{RD}$ ↓ delay time	t_{AR}	100		ns	(Notes 2, 3)
$\overline{RD}$ ↓ to address floating	t_{AFR}		20	ns	(Note 2)
Address to data input	t_{AD}		250	ns	(Notes 2, 3)
ALE ↓ to data input	t_{LDR}		135	ns	(Notes 2, 3)
$\overline{RD}$ ↓ to data input	t_{RD}		120	ns	(Notes 2, 3)
ALE ↓ to $\overline{RD}$ ↓ delay time	t_{LR}	15		ns	(Notes 2, 3)
Data hold time $\overline{RD}$ ↑	t_{RDH}	0		ns	(Note 2)
$\overline{RD}$ ↑ to ALE ↑ delay time	t_{RL}	80		ns	(Notes 2, 3)
$\overline{RD}$ width low	t_{RR}	215		ns	Data read (Notes 2, 3)
		415		ns	Opcode fetch (Notes 2, 3)

AC Characteristics (cont)

Parameter	Symbol	Min	Max	Unit	Conditions
ALE width high	t_{LL}	90		ns	(Notes 2, 3)
$\overline{M1}$ setup time to ALE ↓	tML	30		ns	(Note 3)
$\overline{M1}$ hold time after ALE ↓	tLM	35		ns	(Note 3)
$\overline{IO}$/M setup time to ALE ↓	tIL	30		ns	(Note 3)
$\overline{IO}$/M hold time after ALE ↓	tLI	35		ns	(Note 3)
Address to $\overline{WR}$ ↓ delay	t_{AW}	100		ns	(Notes 2, 3)
ALE ↓ to data output	t_{LDW}		180	ns	(Notes 2, 3)
$\overline{WR}$ ↓ to data output	t_{WD}		100	ns	(Note 2)
ALE ↓ to $\overline{WR}$ ↓ delay time	t_{LW}	15		ns	(Notes 2, 3)
Data setup time to $\overline{WR}$ ↑	t_{DW}	165		ns	(Notes 2, 3)
Data hold time to $\overline{WR}$ ↑	t_{WDH}	60		ns	(Notes 2, 3)
$\overline{WR}$ ↑ to ALE ↑ delay time	t_{WL}	80		ns	(Notes 2, 3)
$\overline{WR}$ width low	t_{WW}	215		ns	(Notes 2, 3)
Address to data input	t_{ACC}		250	ns	(Notes 2, 3)
Data hold time from address	t_{IH}	0		ns	(Note 2)

Notes:

(1) Applies to μPD78CP14 only.

(2) Load capacitance C_L = 150 pF.

(3) Values are for 15-MHz operation. For operation at other frequencies, refer to the table called Bus Timing Depending on t_{CYC}.

A/D Converter Characteristics

T_A = −40° to +85°C; V_{DD} = +5.0 V ±10% (±5% on μPD78CP14); V_{SS} = AV_{SS} 0 V;
V_{DD} −0.5 V ≤ AV_{DD} ≤ V_{DD}; 3.4 V ≤ V_{AREF} ≤ AV_{DD}

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Resolution		8			bits	
Absolute accuracy (Note 1)				±0.4	%FSR	T_A = −10 to +70°C; 66 ns ≤ t_{CYC} ≤ 170 ns; 4.0 V ≤ V_{AREF} ≤ AV_{DD}
				±0.6	%FSR	66 ns ≤ t_{CYC} ≤ 170 ns; 4.0 V ≤ V_{AREF} ≤ AV_{DD}
				±0.8	%FSR	66 ns ≤ t_{CYC} ≤ 170 ns; 3.4 V ≤ V_{AREF} ≤ AV_{DD}
Conversion time	t_{CONV}	576			t_{CYC}	66 ns ≤ t_{CYC} ≤ 110 ns
		432			t_{CYC}	110 ns ≤ t_{CYC} ≤ 170 ns
Sampling time	t_{SAMP}	96			t_{CYC}	66 ns ≤ t_{CYC} ≤ 110 ns
		72			t_{CYC}	110 ns ≤ t_{CYC} ≤ 170 ns
Analog input voltage	V_{IAN}	0		V_{AREF}	V	
Analog input impedance	R_{AN}		1000		MΩ	
Reference voltage	V_{AREF}	3.4		AV_{DD}	V	
V_{AREF} current	I_{AREF1}		1.5	3.0	mA	Operation mode
	I_{AREF2}		0.7	1.5	mA	STOP mode
AV_{DD} supply current	AI_{DD1}		0.5	1.3	mA	Operation mode
	AI_{DD2}		10	20	μA	STOP mode

Notes:

(1) Quantizing error (±1/2 LSB) is not included.

(2) FSR = Full-scale resolution.

Bus Timing Dependent on t_{CYK}

Symbol	Min/Max (ns)	Calculation Formula
t_{TIH}, t_{TIL}	Min	6T (TI input - PC_3)
t_{CI1H}, t_{CI1L} (Note 2)	Min	6T (TI input - PC_5)
t_{CI2H}, t_{CI2L} (Note 3)	Min	48T (TI input - PC_5)
t_{I1H}, t_{I1L}	Min	36T (INT1)
t_{I2H}, t_{I2L}	Min	36T ($\overline{INT2}$)
t_{ANH}, t_{ANL}	Min	36T (AN4-AN7)
t_{AL}	Min	2T – 100
t_{LA}	Min	T – 30
t_{AR}	Min	3T – 100
t_{AD}	Max	7T – 220
t_{LDR}	Max	5T – 200
t_{RD}	Max	4T – 150
t_{LR}	Min	T – 50
t_{RL}	Min	2T – 50
t_{RR}	Min	4T – 50 (Data read)
	Min	7T – 50 (Opcode fetch)
t_{LL}	Min	2T – 40
t_{ML}	Min	2T – 100
t_{LM}	Min	T – 30
t_{IL}	Min	2T – 100
t_{LI}	Min	T – 30
t_{AW}	Min	3T – 100
t_{LDW}	Max	T + 110
t_{LW}	Min	T – 50
t_{DW}	Min	4T – 100
t_{WDH}	Min	2T – 70
t_{WL}	Min	2T – 50
t_{WW}	Min	4T – 50
t_{CYK}	Min	12T ($\overline{SCK}$ input) (Note 1)
	Min	24T ($\overline{SCK}$ output)
t_{KKL}	Min	5T + 5 ($\overline{SCK}$ input) (Note 1)
	Min	12T – 100 ($\overline{SCK}$ output)
t_{KKH}	Min	5T + 5 ($\overline{SCK}$ input) (Note 1)
	Min	12T – 100 ($\overline{SCK}$ output)

Notes:

(1) 1 x baud rate in synchronous or I/O interface mode; T = t_{CYC} = $1/f_{XTAL}$.
The items not included in this list are independent of oscillator frequency (f_{XTAL}).

(2) Event counter mode.

(3) Pulse width measurement mode.

5

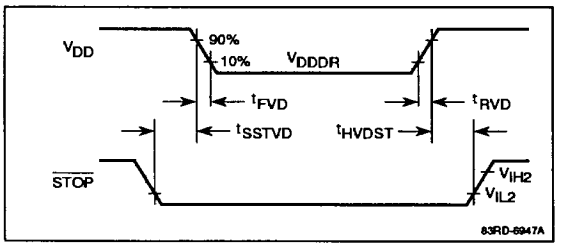
Data Memory STOP Mode Data Retention Characteristics

T_A = −40 to 85°C

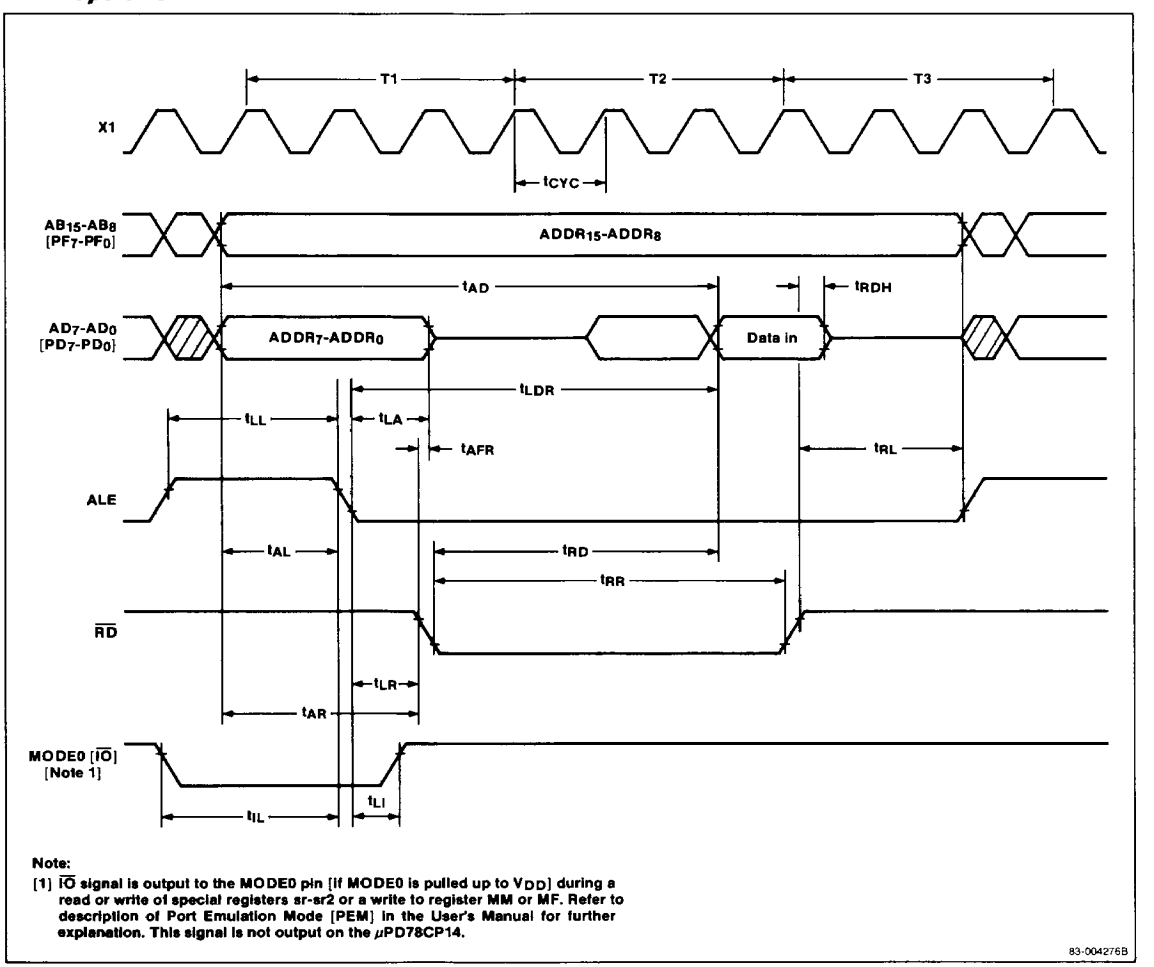
Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Data retention power supply voltage	V_{DDDR}	2.5		5.5	V	
Data retention power supply current	I_{DDDR}		1	15	μA	V_{DDDR} = 2.5 V
			15	50	μA	V_{DDDR} = 5.0 V ±10%
				300	μA	V_{DDDR} = 2.4 V (μPD78CP14)
				1	mA	V_{DDDR} = 5.0 V ±5% (μPD78CP14)
V_{DD} rise, fall time	t_{RVD}, t_{FVD}	200			μs	
$\overline{STOP}$ setup time to V_{DD}	t_{SSTVD}	12T+0.5			μs	
$\overline{STOP}$ hold time from V_{DD}	t_{HVDST}	12T+0.5			μs	

Timing Waveforms

Data Retention Timing



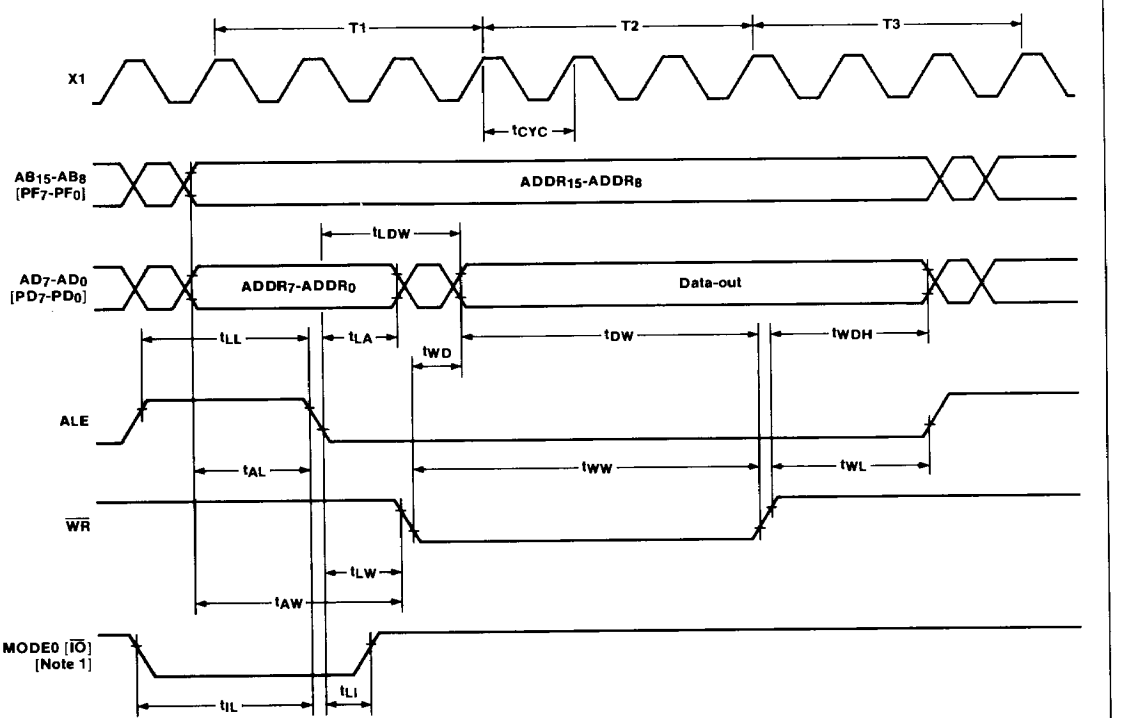
Read Operation



Note:
[1] $\overline{IO}$ signal is output to the MODE0 pin [if MODE0 is pulled up to V_{DD}] during a read or write of special registers sr-sr2 or a write to register MM or MF. Refer to description of Port Emulation Mode [PEM] in the User's Manual for further explanation. This signal is not output on the μPD78CP14.

83-004276B

Write Operation



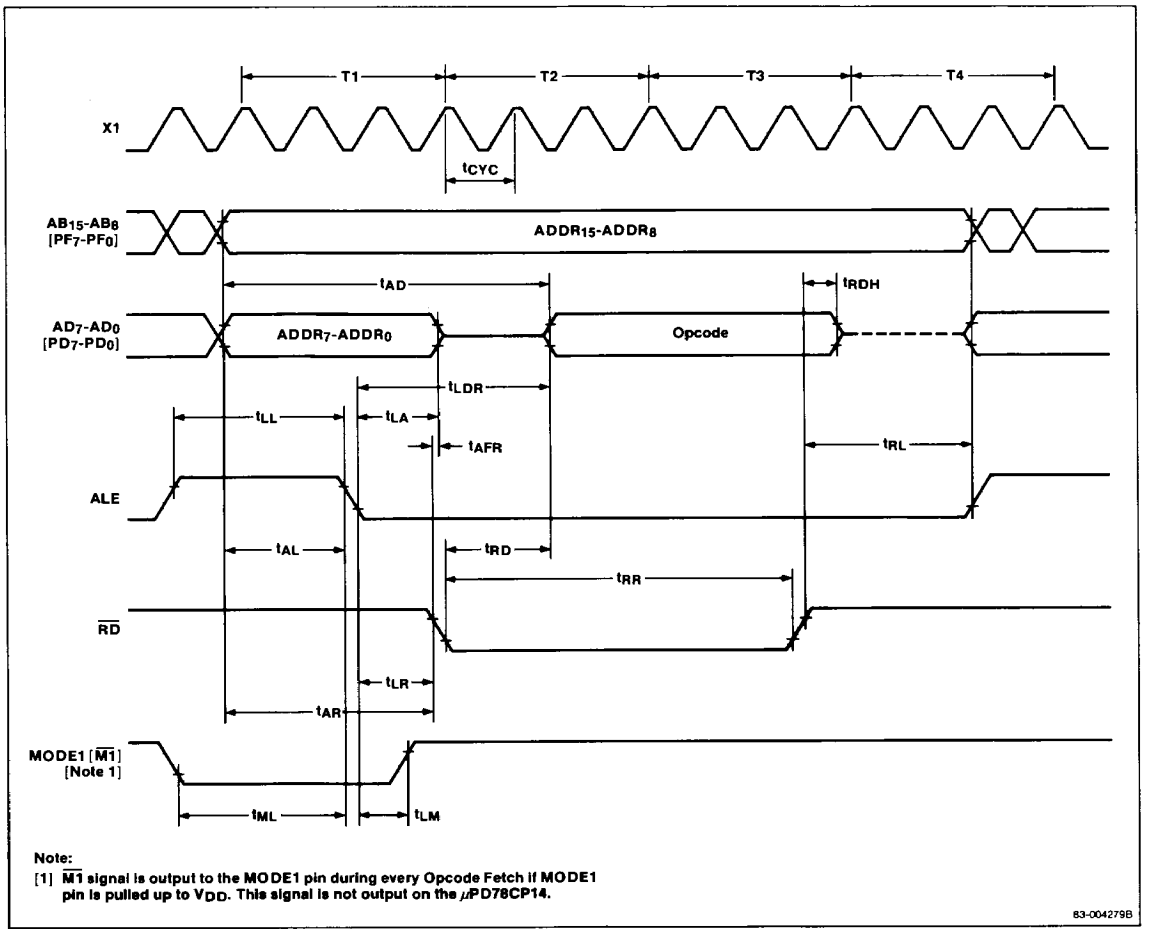
Note:

[1] $\overline{IO}$ signal is output to the MODE0 pin [if MODE0 is pulled up to V_{DD}] during a read or write of special registers sr-sr2 or a write to register MM or MF. Refer to description of Port Emulation Mode [PEM] in the User's Manual for further explanation. This signal is not output on the μPD78CP14.

83-004278B

5

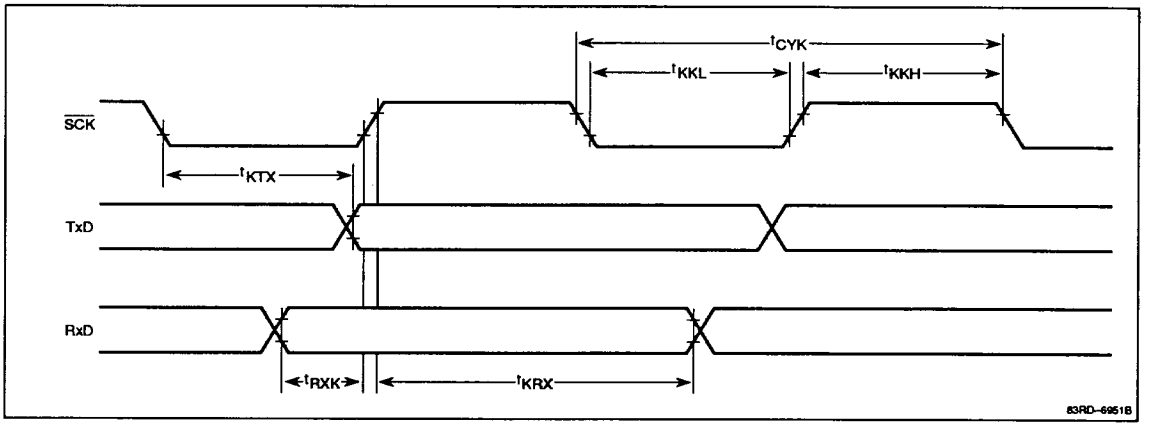
Opcode Fetch Operation



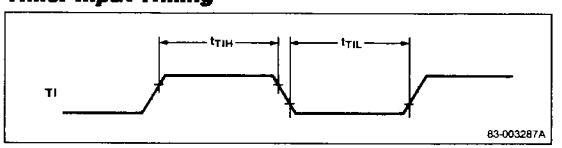
Note:
[1] $\overline{M1}$ signal is output to the MODE1 pin during every Opcode Fetch if MODE1 pin is pulled up to V_{DD}. This signal is not output on the μPD78CP14.

83-004279B

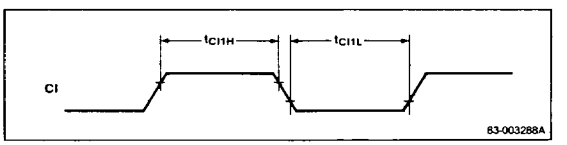
Serial Operation Transmit/Receive Timing



Timer Input Timing

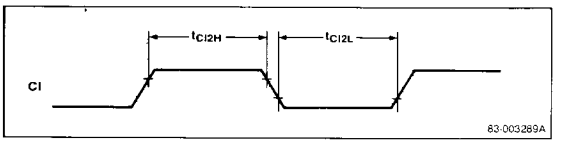


Timer/Event Counter Input Timing: Event Counter Mode

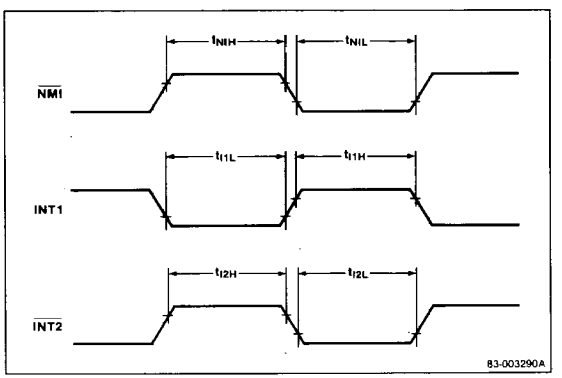


5

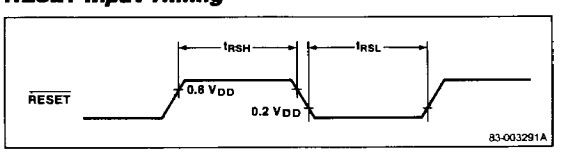
Timer/Event Counter Input Timing: Pulse Width Measurement Mode



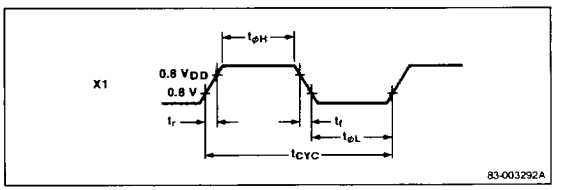
Interrupt Input Timing



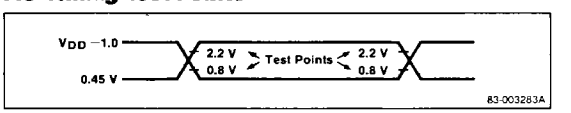
RESET Input Timing



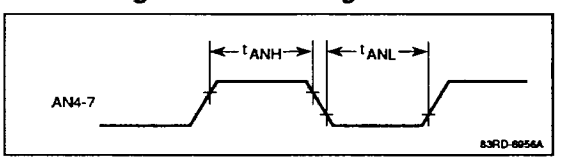
External Clock Timing



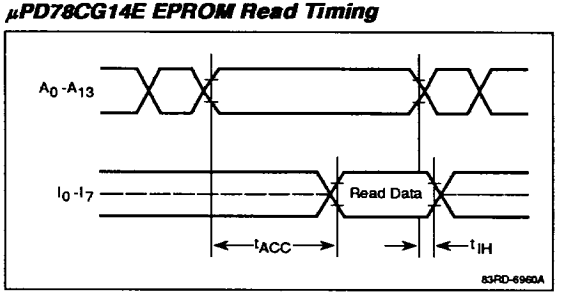
AC Timing Test Points



AN4-AN7 Edge Detection Timing



μPD78CG14E EPROM Read Timing



μPD78CP14 PROGRAMMING

In the μPD78CP14, the mask ROM of the μPD78C1X/C1XA is replaced by a one-time programmable ROM (OTP ROM) or a reprogrammable, ultraviolet erasable ROM (UV EPROM). The ROM is 16,384 by 8 bits and can be programmed using a general-purpose PROM writer with a μPD27C256A programming mode. Refer to tables 3 through 5 and the AC and DC Programming Characteristics for specific information applicable to programming the μPD78CP14.

The PA-78CP14CW/GF/GQ/L are the socket adapters used for configuring the μPD78CP14 to fit a standard μPD27C256A PROM socket.

Table 3. Pin Functions during EPROM Programming

Pin	Function	Description
PA_0-PA_7	A_0-A_7	Low-order 8-bit address
PF_0	A_8	High-order 7-bit address
$\overline{NMI}$	A_9	
PF_2-PF_6	A_{10}-A_{14}	
PD_0-PD_7	D_0-D_7	Data input/output
PB_6	$\overline{CE}$	Chip enable input
PB_7	$\overline{OE}$	Output enable input
$\overline{RESET}$	$\overline{RESET}$	PROM programming mode requires a low voltage on this pin
Mode 0	Mode 0	Enter PROM programming mode by applying a high voltage to this pin
Mode 1	Mode 1	Enter PROM programming mode by applying a low voltage to this pin
$\overline{STOP}$	V_{PP}	High-voltage input (write/verify) high level (read)

Table 4. Summary of Operation Modes for EPROM Programming

Operation Mode	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{DD}	$\overline{RESET}$	MODE0	MODE1	A_{14}
Program write	L	H	+12.5 V	+6 V	L	H	L	L
Program verify	H	L	+12.5 V	+6 V	L	H	L	L
Program inhibit	H	H	+12.5 V	+6 V	L	H	L	L
Read	L	L	+5 V	+5 V	L	H	L	L
Output disable	L	H	+5 V	+5 V	L	H	L	L
Standby	H	L/H	+5 V	+5 V	L	H	L	L

Notes:

(1) The $\overline{CE}$, $\overline{OE}$, V_{pp}, and V_{DD} pins are all compatible with the μPD27C256A pins.

Caution: When Vpp is set to +12.5 V and V_{DD} is set to +6 V, you cannot set both $\overline{CE}$ and $\overline{OE}$ to low level (L).

5

Table 5. Recommended Connections for Unused Pins (EPROM Programming Mode)

Pin	Recommended Connection Method
INT1	Connect to V_{SS}
X1	Connect to V_{SS}
X2	Leave this pin disconnected
AN0-AN7	Connect to V_{SS}
VA_{REF}	Connect to V_{SS}
AV_{DD}	Connect to V_{SS}
AV_{SS}	Connect to V_{SS}
Remaining pins	Connect each pin via a resistor to V_{SS}

PROM Write Procedure

(1) Connect the $\overline{\text{RESET}}$ pin, the MODE1 pin, and A_{14} pin to a low level and connect the MODE0 pin to a high level. Connect all unused pins as recommended in Table 5.

(2) Apply +6 V to the V_{DD} pin and +12.5 V to the V_{pp} pin.

(3) Provide the initial address.

(4) Provide write data.

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

(6) This bit is now verified with a pulse (active low) to the $\overline{\text{OE}}$ pin. If the data has been written, proceed to step 8; if not, repeat steps 4 to 6. If the data cannot be correctly written after 25 attempts, go to step 7.

(7) Classify as defective and stop write operation.

(8) Provide write data and supply program pulse (for additional writing) for 3 ms times the number of repeats performed between steps 4 to 6.

(9) Increment the address.

(10) Repeat steps 4 to 9 until the end address.

PROM Read Procedure

(1) FIX the $\overline{\text{RESET}}$ pin, the MODE1 pin, and A_{14} pin to a low level and connect the MODE0 pin to a high level.

(2) Apply +5 V to the V_{DD} and V_{pp} pins.

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

(4) Read mode is entered with a pulse (active low) on both the $\overline{\text{CE}}$ and $\overline{\text{OE}}$ pins.

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

EPROM Erasure

Data in an EPROM is erased by exposing the quartz window in the ceramic package 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 15W-s/cm^2 (ultraviolet ray intensity x exposure time) is required to completely erase written data. Erasure by an ultraviolet lamp rated at 12 mW/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.

μPD78CP14 DC Programming Characteristics

T_A = 25 ±5°C; MODE1 = V_{IL}; MODE0 = V_{IH}; V_{SS} = 0 V

Parameter	Symbol	Symbol*	Min	Typ	Max	Unit	Condition
High-level input voltage	V_{IH}	V_{IH}	2.2		V_{DDP}+0.3	V	
Low-level input voltage	V_{IL}	V_{IL}	−0.3		0.8	V	
Input leakage current	I_{LIP}	I_{LI}			±10	μA	$0 \le V_I \le V_{DDP}$
High-level output voltage	V_{OH}	V_{OH}	V_{DD} − 1.0			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}
V_{DDP} power voltage	V_{DDP}	V_{CC}	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 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 current	I_{DD}	I_{CC}			30	mA	Program memory write mode
					30	mA	Program memory read mode; $\overline{CE}$ = V_{IL}; V_I = V_{IH}
V_{PP} power current	I_{PP}	I_{PP}			30	mA	Program memory read mode; $\overline{CE}$ = V_{IL}; $\overline{OE}$ = V_{IH}
				1	100	μA	Program memory write mode

* Corresponding symbols of the μPD27C256A.

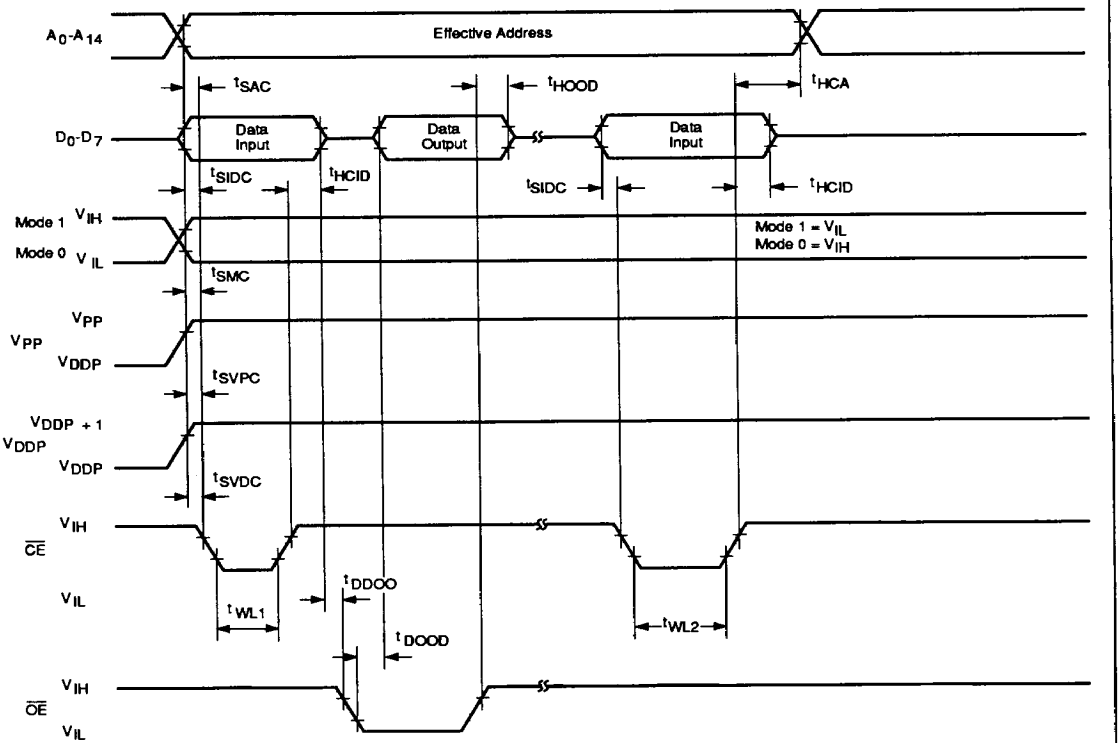
μPD78CP14 AC Programming Characteristics

T_A = 25 ±5°C; MODE1 = V_{IL}; V_{SS} = 0 V

Parameter	Symbol	Symbol*	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 from $\overline{CE}$ ↑	t_{HCA}	t_{AH}	2			μs	
Input data hold time from $\overline{CE}$ ↑	t_{HCID}	t_{DH}	2			μs	
Output data hold time from $\overline{OE}$ ↑	t_{HOOD}	t_{DF}	0		130	ns	
V_{pp} setup time to $\overline{CE}$ ↓	t_{SVPC}	t_{VPS}	2			μs	
V_{DDP} setup time to $\overline{CE}$ ↓	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	
MODE0/MODE1 setup time vs. $\overline{CE}$ ↓	t_{SMC}		2			μs	MODE1 = V_{IL} and MODE0 = V_{IH}
Address to data output time	t_{DAOD}	t_{ACC}			2	μs	$\overline{OE}$ = V_{IL}
$\overline{CE}$ ↓ to data output time	t_{DCOD}	t_{CE}			1	μs	
$\overline{OE}$ ↓ to data output time	t_{DOOD}	t_{OE}			1	μs	
Data hold time from $\overline{OE}$ ↑ or $\overline{CE}$ ↑	t_{HCOD}	t_{DF}	0		130	ns	
Data hold time from address	t_{HAOD}	t_{OH}	0			ns	$\overline{OE}$ = V_{IL}

* Corresponding symbols of the μPD27C256A.

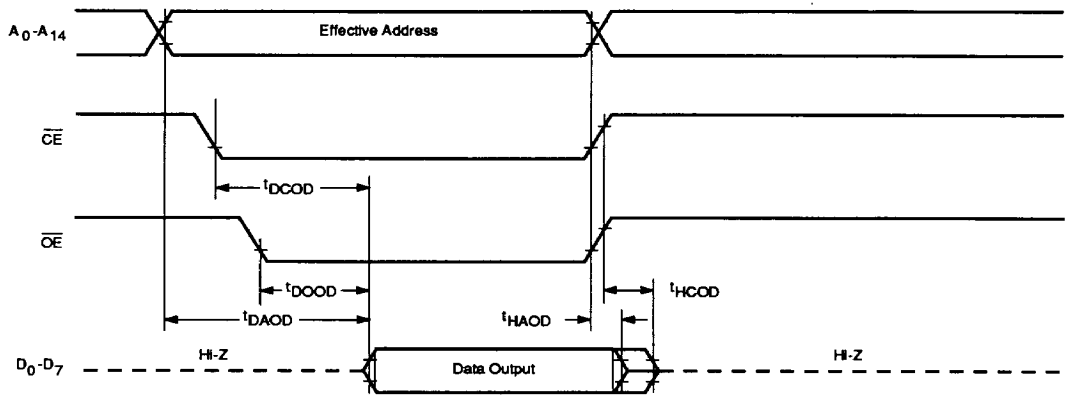
μPD78CP14 PROM Write Mode Timing



Notes:
(1) V_{DDP} must be applied before applying V_{PP}. It should be removed after removing V_{PP}.
(2) V_{PP} must not exceed +13 V, including overshoot.

83RD-6948B

μPD78CP14 PROM Read Mode Timing



Notes:

(1) To read PROM within the t_{DAOD} range, the delay of $\overline{OE}\downarrow$ from $\overline{CE}\downarrow$ must be within $t_{DAOD} - t_{DOOD}$.

(2) t_{HCOD} is the time from the state in which either $\overline{OE}$ or $\overline{CE}$ first becomes V_{IH}.

83RD-69498

5

Operand Symbols

Symbol	Allowable Operands
Registers	
r	V, A, B, C, D, E, H, L
r1	EAH, EAL, B, C, D, E, H, L
r2	A, B, C
Special Registers	
sr	PA, PB, PC, PD, PF, MKH, MKL, ANM, SMH, SML, EOM, ETMM, TMM, MM, MCC, MA, MB, MC, MF, TXB, TM0, TM1, ZCM
sr1	PA, PB, PC, PD, PF, MKH, MKL, ANM, SMH, EOM, TMM, RXB, CR0, CR1, CR2, CR3
sr2	PA, PB, PC, PD, PF, MKH, ANM, MKL, SMH, EOM, TMM
sr3	ETM0, ETM1
sr4	ECNT, ECPT
Register Pairs	
rp	SP, B, D, H
rp1	V, B, D, H, EA
rp2	SP, B, D, H, EA
rp3	B, D, H
Register Pair Addressing	
rpa	B, D, H, D+ , H+ , D−, H−
rpa1	B, D, H
rpa2	B, D, H, D+, H+, D−, H−, D+byte, H+A, H+B, H+EA, H+byte
rpa3	D, H, D++, H++, D+byte, H+A, H+B, H+EA, H+byte
Flags	
f	CY, HC, Z
Interrupt Flags	
irf	INTFNMI, INTFT0, INTFT1, INTF1, INTF2, INTFE0, INTFE1, INTFEIN, INTFAD, INTFSR, INTFST, ER, OV, AN4, AN5, AN6, AN7, SB
Immediate Data	
wa	8-bit immediate data (low byte of working register address)
word	16-bit immediate data
byte	8-bit immediate data
bit	3-bit immediate data (b_2, b_1, b_0)

Operand Definitions

Special Registers (sr-sr4)

PA = Port A
PB = Port B
PC = Port C
PD = Port D
PF = Port F
MA = Mode A
MB = Mode B
MC = Mode C
MCC = Mode control C
MF = Mode F

MM = Memory mapping
TM0 = Timer register 0
TM1 = Timer register 1
TMM = Timing mode
ETM0 = Timer/event counter register 0
ETM1 = Timer/event counter register 1
ZCM = Zero-cross mode control register

ECNT = Timer/event counter upcounter
ECPT = Timer/event counter capture
ETMM = Timer/event counter mode

EOM = Timer/event counter output mode

TXB = Transmit buffer
RXB = Receive buffer
SMH = Serial mode high
SML = Serial mode low
MKH = Mask high
MKL = Mask low
ANM = A/D channel mode
CR0 to CR3= A/D conversion result 0-3

Register Pairs (rp-rp3)

SP = Stack pointer
B = BC
D = DE
H = HL
V = VA
EA = Extended accumulator

Register Pair Addressing (rpa-rpa3)

B = (BC)
D = (DE)
H = (HL)
D+ = (DE)+
H+ = (HL)+
D− = (DE)−
H− = (HL)−

D++ = (DE)++
H++ = (HL)++
D+byte = (DE+byte)
H+byte = (HL+byte)
H+A = (HL+A)
H+B = (HL+B)
H+EA = (HL+EA)

Flags (f)

CY = Carry
HC = Half-carry
Z = Zero

Interrupt Flags (irf)

INTFNMI = NMI interrupt flag

INTFT0 = FT0
INTFT1 = FT1
INTF1 = F1
INTF2 = F2
INTFE0 = FE0
INTFE1 = FE1

INTFEIN = FEIN
INTFAD = FAD
INTFSR = FSR
INTFST = FST
ER = Error
OV = Overflow
AN4 to AN7 = Analog input 4-7
SB = Standby

Operand Codes

Registers (r, r2)

R_2	R_1	R_0	Reg	r	r2
0	0	0	V		
0	0	1	A		
0	1	0	B		
0	1	1	C		
1	0	0	D		
1	0	1	E		
1	1	0	H		
1	1	1	L		

Registers (r1)

T_2	T_1	T_0	Reg
0	0	0	EAH
0	0	1	EAL
0	1	0	B
0	1	1	C
1	0	0	D
1	0	1	E
1	1	0	H
1	1	1	L

Special Registers (sr, sr1, sr2)

S_5	S_4	S_3	S_2	S_1	S_0	Special Reg	sr	sr1	sr2
0	0	0	0	0	0	PA			
0	0	0	0	0	1	PB			
0	0	0	0	1	0	PC			
0	0	0	0	1	1	PD			
0	0	0	1	0	1	PF			
0	0	0	1	1	0	MKH			
0	0	0	1	1	1	MKL			
0	0	1	0	0	0	ANM			
0	0	1	0	0	1	SMH			
0	0	1	0	1	0	SML			
0	0	1	0	1	1	EOM			
0	0	1	1	0	0	ETMM			
0	0	1	1	0	1	TMM			
0	1	0	0	0	0	MM			
0	1	0	0	0	1	MCC			
0	1	0	0	1	0	MA			
0	1	0	0	1	1	MB			
0	1	0	1	0	0	MC			
0	1	0	1	1	1	MF			
0	1	1	0	0	0	TXB			
0	1	1	0	0	1	RXB			
0	1	1	0	1	0	TM0			
0	1	1	0	1	1	TM1			
1	0	0	0	0	0	CR0			
1	0	0	0	0	1	CR1			
1	0	0	0	1	0	CR2			
1	0	0	0	1	1	CR3			
1	0	1	0	0	0	ZCM			

Special Registers (sr3)

U_0	Special Reg
0	ETM0
1	ETM1

Special Registers (sr4)

V_0	Special Reg
0	ECNT
1	ECPT

Register Pairs (rp, rp2, rp3)

P_2	P_1	P_0	Reg Pair	rp	rp2	rp3
0	0	0	SP			
0	0	1	BC			
0	1	0	DE			
0	1	1	HL			
1	0	0	EA			

Register Pairs (rp1)

Q_2	Q_1	Q_0	Reg Pair
0	0	0	VA
0	0	1	BC
0	1	0	DE
0	1	1	HL
1	0	0	EA

Register Pair Addressing (rpa, rpa1, rpa2)

A_3	A_2	A_1	A_0	Addressing	rpa	rpa1	rpa2
0	0	0	0	—			
0	0	0	1	(BC)			
0	0	1	0	(DE)			
0	0	1	1	(HL)			
0	1	0	0	(DE)+			
0	1	0	1	(HL)+			
0	1	1	0	(DE)−			
0	1	1	1	(HL)−			
1	0	1	1	(DE+byte)			
1	1	0	0	(HL+A)			
1	1	0	1	(HL+B)			
1	1	1	0	(HL+EA)			
1	1	1	1	(HL+byte)			

Register Pair Addressing (rpa3)

C_3	C_2	C_1	C_0	Addressing
0	0	1	0	(DE)
0	0	1	1	(HL)
0	1	0	0	(DE)++
0	1	0	1	(HL)++
1	0	1	1	(DE+byte)
1	1	0	0	(HL+A)
1	1	0	1	(HL+B)
1	1	1	0	(HL+EA)
1	1	1	1	(HL+byte)

5

Operand Codes (cont)

Flags (f)

F_2	F_1	F_0	Flag
0	0	0	—
0	1	0	CY
0	1	1	HC
1	0	0	Z

Interrupt Flags (irf)

I_4	I_3	I_2	I_1	I_0	Flag
0	0	0	0	0	NMI
0	0	0	0	1	FT0
0	0	0	1	0	FT1
0	0	0	1	1	F1
0	0	1	0	0	F2
0	0	1	0	1	FE0
0	0	1	1	0	FE1
0	0	1	1	1	FEIN
0	1	0	0	0	FAD
0	1	0	0	1	FSR
0	1	0	1	0	FST
0	1	0	1	1	ER
0	1	1	0	0	OV
1	0	0	0	0	AN4
1	0	0	0	1	AN5
1	0	0	1	0	AN6
1	0	0	1	1	AN7
1	0	1	0	0	SB

Graphic Symbols

Symbol	Description
←	Transfer direction, result
∧	Logical product (logical AND)
∨	Logical sum (logical OR)
∀	Exclusive-OR
—	Complement
•	Concatenation

Instruction Set

Mnemonic	Operand	Operation	Operation Code B1 / B3								Operation Code B2 / B4								State (Note 1)	Bytes	Skip Condition
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
8-Bit Data Transfer																					
MOV	r1,A	(r1) ← (A)	0	0	0	1	1	T_2	T_1	T_0									4	1	
	A, r1	(A) ← (r1)	0	0	0	0	1	T_2	T_1	T_0									4	1	
	*sr,A	(sr) ← (A)	0	1	0	0	1	1	0	1	1	1	S_5	S_4	S_3	S_2	S_1	S_0	10	2	
	*A,sr1	(A) ← (sr1)	0	1	0	0	1	1	0	0	1	1	S_5	S_4	S_3	S_2	S_1	S_0	10	2	
	r,word	(r) ← (word)	0	1	1	1	0	0	0	0	0	1	1	0	1	R_2	R_1	R_0	17	4	
			Low addr								High addr										
	word,r	(word) ← (r)	0	1	1	1	0	0	0	0	0	1	1	1	1	R_2	R_1	R_0	17	4	
			Low addr								High addr										
MVI	*r,byte	(r) ← byte	0	1	1	0	1	R_2	R_1	R_0	Data								7	2	
	sr2,byte	(sr2) ← byte	0	1	1	0	0	1	0	0	S_3	0	0	0	0	S_2	S_1	S_0	14	3	
			Data																		
MVIW	*wa, byte	((V)•(wa)) ← byte	0	1	1	1	0	0	0	1	Offset								13	3	
			Data																		
MVIX	*rpa1,byte	(rpa1) ← byte	0	1	0	0	1	0	A_1	A_0	Data								10	2	
STAW	*wa	((V)•(wa)) ← (A)	0	1	1	0	0	0	1	1	Offset								10	2	
LDAW	*wa	(A) ← ((V)•(wa))	0	0	0	0	0	0	0	1	Offset								10	2	
STAX	*rpa2	((rpa2)) ← (A)	A_3	0	1	1	1	A_2	A_1	A_0	Data (Note 2)								7/13 (Note 3)	2	
LDAX	*rpa2	(A) ← ((rpa2))	A_3	0	1	0	1	A_2	A_1	A_0	Data (Note 2)								7/13 (Note 3)	2	
EXX		(B) ↔ (B'), (C) ↔ (C'), (D) ↔ (D') (E) ↔ (E'), (H) ↔ (H'), (L) ↔ (L')	0	0	0	1	0	0	0	1									4	1	
EXA		(V) ↔ (V'), (A) ↔ (A'), (EA) ↔ (EA')	0	0	0	1	0	0	0	0									4	1	
EXH		(H) ↔ (H'), (L) ↔ (L')	0	1	0	1	0	0	0	0									4	1	
BLOCK		((DE)) ← ((HL)), (DE) ← (DE) + 1, (HL) ← (HL) + 1, (C) ← (C) − 1 End if borrow	0	0	1	1	0	0	0	1									13 x (C + 1)	1	
16-Bit Data Transfer																					
DMOV	rp3, EA	($rp3_L$) ← (EAL), ($rp3_H$) ← (EAH)	1	0	1	1	0	1	P_1	P_0									4	1	
	EA,rp3	(EAL) ← ($rp3_L$), (EAH) ← ($rp3_H$)	1	0	1	0	0	1	P_1	P_0									4	1	

Notes:

(1) For the skip condition, the idle states are as follows:
- 1-byte instruction: 4 states
- 2-byte instruction: 8 states
- 3-byte instruction: 11 states
- 2-byte instruction (with *): 7 states
- 3-byte instruction (with *): 10 states
- 4-byte instruction: 14 states

(2) B2 (Data): rpa2 = D+byte or H+byte.

(3) Right side of slash (/) in states indicates case rpa2 or rpa3 = D+byte, H+A, H+B, H+EA, or H+byte.

(4) B3 (Data): rpa3 = D+byte or H+byte.

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1 / B3								Operation Code B2 / B4								State (Note 1)	Bytes	Skip Condition
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
16-Bit Data Transfer (cont)																					
DMOV	sr3, EA	(sr3) ← (EA)	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	U_0	14	2	
	EA,sr4	(EA) ← (sr4)	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	V_0	14	2	
SBCD	word	(word) ← (C), (word + 1) ← (B)	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0	20	4	
			Low addr								High addr										
SDED	word	(word) ← (E), (word + 1) ← (D)	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1	0	20	4	
			Low addr								High addr										
SHLD	word	(word) ← (L), (word + 1) ← (H)	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	20	4	
			Low addr								High addr										
SSPD	word	(word) ← (SP_L), (word + 1) ← (SP_H)	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	20	4	
			Low addr								High addr										
STEAX	rpa3	((rpa3)) ← (EAL), (((rpa3)) + 1)) ← (EAH)	0	1	0	0	1	0	0	0	1	0	0	1	C_3	C_2	C_1	C_0	14/20 (Note 3)	3	
			Data (Note 4)																		
LBCD	word	(C) ← (word), (B) ← (word + 1)	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	20	4	
			Low addr								High addr										
LDED	word	(E) ← (word), (D) ← (word + 1)	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1	1	20	4	
			Low addr								High addr										
LHLD	word	(L) ← (word), (H) ← (word + 1)	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	20	4	
			Low addr								High addr										
LSPD	word	(SP_L) ← (word), (SP_H) ← (word + 1)	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	20	4	
			Low addr								High addr										
LDEAX	rpa3	(EAL) ← ((rpa3)), (EAH) ← (((rpa3) + 1))	0	1	0	0	1	0	0	0	1	0	0	0	C_3	C_2	C_1	C_0	14/20 (Note 3)	3	
			Data (Note 4)																		
PUSH	rp1	(((SP) − 1)) ← ($rp1_H$), (((SP) − 2)) ← ($rp1_L$), (SP) ← (SP) − 2	1	0	1	1	0	Q_2	Q_1	Q_0									13	1	
POP	rp1	($rp1_L$) ← ((SP)), ($rp1_H$) ← (((SP) + 1)), (SP) ← (SP) + 2	1	0	1	0	0	Q_2	Q_1	Q_0									10	1	
LXI	*rp2,word	(rp2) ← (word)	0	P_2	P_1	P_0	0	1	0	0	Low byte								10	3	
			High byte																		
TABLE		(C) ← (((PC) + 3 + (A))), (B) ← (((PC) + 3 + (A) + 1))	0	1	0	0	1	0	0	0	1	0	1	0	1	0	0	0	17	2	
8-Bit Arithmetic (Register)																					
ADD	A,r	(A) ← (A) + (r)	0	1	1	0	0	0	0	0	1	1	0	0	0	R_2	R_1	R_0	8	2	
	r,A	(r) ← (r) + (A)	0	1	1	0	0	0	0	0	0	1	0	0	0	R_2	R_1	R_0	8	2	
ADC	A,r	(A) ← (A) + (r) + (CY)	0	1	1	0	0	0	0	0	1	1	0	1	0	R_2	R_1	R_0	8	2	
	r,A	(r) ← (r) + (A) + (CY)	0	1	1	0	0	0	0	0	0	1	0	1	0	R_2	R_1	R_0	8	2	

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1 / B3								Operation Code B2 / B4								State (Note 1)	Bytes	Skip Condition
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
8-Bit Arithmetic [Register] (cont)																					
ADDNC	A,r	(A) ← (A) + (r)	0	1	1	0	0	0	0	0	1	0	1	0	0	R_2	R_1	R_0	8	2	No carry
	r,A	(r) ← (r) + (A)	0	1	1	0	0	0	0	0	0	0	1	0	0	R_2	R_1	R_0	8	2	No carry
SUB	A,r	(A) ← (A) − (r)	0	1	1	0	0	0	0	0	1	1	1	0	0	R_2	R_1	R_0	8	2	
	r,A	(r) ← (r) − (A)	0	1	1	0	0	0	0	0	0	1	1	0	0	R_2	R_1	R_0	8	2	
SBB	A,r	(A) ← (A) − (r) − (CY)	0	1	1	0	0	0	0	0	1	1	1	1	0	R_2	R_1	R_0	8	2	
	r,A	(r) ← (r) − (A) − (CY)	0	1	1	0	0	0	0	0	0	1	1	1	0	R_2	R_1	R_0	8	2	
SUBNB	A,r	(A) ← (A) − (r)	0	1	1	0	0	0	0	0	1	0	1	1	0	R_2	R_1	R_0	8	2	No borrow
	r,A	(r) ← (r) − (A)	0	1	1	0	0	0	0	0	0	0	1	1	0	R_2	R_1	R_0	8	2	No borrow
ANA	A,r	(A) ← (A) ∧ (r)	0	1	1	0	0	0	0	0	1	0	0	0	1	R_2	R_1	R_0	8	2	
	r,A	(r) ← (r) ∧ (A)	0	1	1	0	0	0	0	0	0	0	0	0	1	R_2	R_1	R_0	8	2	
ORA	A,r	(A) ← (A) ∨ (r)	0	1	1	0	0	0	0	0	1	0	0	1	1	R_2	R_1	R_0	8	2	
	r,A	(r) ← (r) ∨ (A)	0	1	1	0	0	0	0	0	0	0	0	1	1	R_2	R_1	R_0	8	2	
XRA	A,r	(A) ← (A) ∀ (r)	0	1	1	0	0	0	0	0	1	0	0	1	0	R_2	R_1	R_0	8	2	
	r,A	(r) ← (r) ∀ (A)	0	1	1	0	0	0	0	0	0	0	0	1	0	R_2	R_1	R_0	8	2	
GTA	A,r	(A) − (r) − 1	0	1	1	0	0	0	0	0	1	0	1	0	1	R_2	R_1	R_0	8	2	No borrow
	r,A	(r) − (A) − 1	0	1	1	0	0	0	0	0	0	0	1	0	1	R_2	R_1	R_0	8	2	No borrow
LTA	A,r	(A) − (r)	0	1	1	0	0	0	0	0	1	0	1	1	1	R_2	R_1	R_0	8	2	Borrow
	r,A	(r) − (A)	0	1	1	0	0	0	0	0	0	0	1	1	1	R_2	R_1	R_0	8	2	Borrow
NEA	A,r	(A) − (r)	0	1	1	0	0	0	0	0	1	1	1	0	1	R_2	R_1	R_0	8	2	No zero
	r,A	(r) − (A)	0	1	1	0	0	0	0	0	0	1	1	0	1	R_2	R_1	R_0	8	2	No zero
EQA	A,r	(A) − (r)	0	1	1	0	0	0	0	0	1	1	1	1	1	R_2	R_1	R_0	8	2	Zero
	r,A	(r) − (A)	0	1	1	0	0	0	0	0	0	1	1	1	1	R_2	R_1	R_0	8	2	Zero
ONA	A,r	(A) ∧ (r)	0	1	1	0	0	0	0	0	1	1	0	0	1	R_2	R_1	R_0	8	2	No zero
OFFA	A,r	(A) ∧ (r)	0	1	1	0	0	0	0	0	1	1	0	1	1	R_2	R_1	R_0	8	2	Zero
8-Bit Arithmetic (Memory)																					
ADDX	rpa	(A) ← (A) + ((rpa))	0	1	1	1	0	0	0	0	1	1	0	0	0	A_2	A_1	A_0	11	2	
ADCX	rpa	(A) ← (A) + ((rpa)) + (CY)	0	1	1	1	0	0	0	0	1	1	0	1	0	A_2	A_1	A_0	11	2	
ADDNCX	rpa	(A) ← (A) + ((rpa))	0	1	1	1	0	0	0	0	1	0	1	0	0	A_2	A_1	A_0	11	2	No carry
SUBX	rpa	(A) ← (A) − ((rpa))	0	1	1	1	0	0	0	0	1	1	1	0	0	A_2	A_1	A_0	11	2	
SBBX	rpa	(A) ← (A) − ((rpa)) − (CY)	0	1	1	1	0	0	0	0	1	1	1	1	0	A_2	A_1	A_0	11	2	
SUBNBX	rpa	(A) ← (A) − ((rpa))	0	1	1	1	0	0	0	0	1	0	1	1	0	A_2	A_1	A_0	11	2	No borrow
ANAX	rpa	(A) ← (A) ∧ ((rpa))	0	1	1	1	0	0	0	0	1	0	0	0	1	A_2	A_1	A_0	11	2	
ORAX	rpa	(A) ← (A) ∨ ((rpa))	0	1	1	1	0	0	0	0	1	0	0	1	1	A_2	A_1	A_0	11	2	

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1 B3	Operation Code B2 B4	State (Note 1)	Bytes	Skip Condition
8-Bit Arithmetic (Memory) (cont)							
XRAX	rpa	(A) ← (A) ∀ ((rpa))	0 1 1 1 0 0 0 0	1 0 0 1 0 A_2 A_1 A_0	11	2	
GTAX	rpa	(A) − ((rpa)) − 1	0 1 1 1 0 0 0 0	1 0 1 0 1 A_2 A_1 A_0	11	2	No borrow
LTAX	rpa	(A) − ((rpa))	0 1 1 1 0 0 0 0	1 0 1 1 1 A_2 A_1 A_0	11	2	Borrow
NEAX	rpa	(A) − ((rpa))	0 1 1 1 0 0 0 0	1 1 1 0 1 A_2 A_1 A_0	11	2	No zero
EQAX	rpa	(A) − ((rpa))	0 1 1 1 0 0 0 0	1 1 1 1 1 A_2 A_1 A_0	11	2	Zero
ONAX	rpa	(A) ∧ ((rpa))	0 1 1 1 0 0 0 0	1 1 0 0 1 A_2 A_1 A_0	11	2	No zero
OFFAX	rpa	(A) ∧ ((rpa))	0 1 1 1 0 0 0 0	1 1 0 1 1 A_2 A_1 A_0	11	2	Zero
Immediate Data							
ADI	*A,byte	(A) ← (A) + byte	0 1 0 0 0 1 1 0	Data	7	2	
	r,byte	(r) ← (r) + byte	0 1 1 1 0 1 0 0	0 1 0 0 0 R_2 R_1 R_0	11	3	
			Data				
	sr2, byte	(sr2) ← (sr2) + byte	0 1 1 0 0 1 0 0	S_3 1 0 0 0 S_2 S_1 S_0	20	3	
			Data				
ACI	*A,byte	(A) ← (A) + byte + (CY)	0 1 0 1 0 1 1 0	Data	7	2	
	r,byte	(r) ← (r) + byte + (CY)	0 1 1 1 0 1 0 0	0 1 0 1 0 R_2 R_1 R_0	11	3	
			Data				
	sr2,byte	(sr2) ← (sr2) + byte + (CY)	0 1 1 0 0 1 0 0	S_3 1 0 1 0 S_2 S_1 S_0	20	3	
			Data				
ADINC	*A,byte	(A) ← (A) + byte	0 0 1 0 0 1 1 0	Data	7	2	No carry
	r,byte	(r) ← (r) + byte	0 1 1 1 0 1 0 0	0 0 1 0 0 R_2 R_1 R_0	11	3	No carry
			Data				
	sr2,byte	(sr2) ← (sr2) + byte	0 1 1 0 0 1 0 0	S_3 0 1 0 0 S_2 S_1 S_0	20	3	No carry
			Data				
SUI	*A,byte	(A) ← (A) − byte	0 1 1 0 0 1 1 0	Data	7	2	
	r,byte	(r) ← (r) − byte	0 1 1 1 0 1 0 0	0 1 1 0 0 R_2 R_1 R_0	11	3	
			Data				
	sr2,byte	(sr2) ← (sr2) − byte	0 1 1 0 0 1 0 0	S_3 1 1 0 0 S_2 S_1 S_0	20	3	
			Data				
SBI	*A,byte	(A) ← (A) − byte − (CY)	0 1 1 1 0 1 1 0	Data	7	2	
	r,byte	(r) ← (r) − byte − (CY)	0 1 1 1 0 1 0 0	0 1 1 1 0 R_2 R_1 R_0	11	3	
			Data				
	sr2,byte	(sr2) ← (sr2) − byte − (CY)	0 1 1 0 0 1 0 0	S_3 1 1 1 0 S_2 S_1 S_0	20	3	
			Data				

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1 / B3								Operation Code B2 / B4								State (Note 1)	Bytes	Skip Condition
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
Immediate Data (cont)																					
SUINB	*A,byte	(A) ← (A) – byte	0	0	1	1	0	1	1	0	Data								7	2	No borrow
	r,byte	(r) ← (r) – byte	0	1	1	1	0	1	0	0	0	0	1	1	0	R_2	R_1	R_0	11	3	No borrow
			Data																		
	sr2,byte	(sr2) ← (sr2) – byte	0	1	1	0	0	1	0	0	S_3	0	1	1	0	S_2	S_1	S_0	20	3	No borrow
			Data																		
ANI	*A,byte	(A) ← (A) ∧ byte	0	0	0	0	0	1	1	1	Data								7	2	
	r,byte	(r) ← (r) ∧ byte	0	1	1	1	0	1	0	0	0	0	0	0	1	R_2	R_1	R_0	11	3	
			Data																		
	sr2,byte	(sr2) ← (sr2) ∧ byte	0	1	1	0	0	1	0	0	S_3	0	0	0	1	S_2	S_1	S_0	20	3	
			Data																		
ORI	*A,byte	(A) ← (A) ∨ byte	0	0	0	1	0	1	1	1	Data								7	2	
	r,byte	(r) ← (r) ∨ byte	0	1	1	1	0	1	0	0	0	0	0	1	1	R_2	R_1	R_0	11	3	
			Data																		
	sr2,byte	(sr2) ← (sr2) ∨ byte	0	1	1	0	0	1	0	0	S_3	0	0	1	1	S_2	S_1	S_0	20	3	
			Data																		
XRI	*A,byte	(A) ← (A) ∀ byte	0	0	0	1	0	1	1	0	Data								7	2	
	r,byte	(r) ← (r) ∀ byte	0	1	1	1	0	1	0	0	0	0	0	1	0	R_2	R_1	R_0	11	3	
			Data																		
	sr2,byte	(sr2) ← (sr2) ∀ byte	0	1	1	0	0	1	0	0	S_3	0	0	1	0	S_2	S_1	S_0	20	3	
			Data																		
GTI	*A,byte	(A) – byte – 1	0	0	1	0	0	1	1	1	Data								7	2	No borrow
	r,byte	(r) – byte – 1	0	1	1	1	0	1	0	0	0	0	1	0	1	R_2	R_1	R_0	11	3	No borrow
			Data																		
	sr2,byte	(sr2) – byte – 1	0	1	1	0	0	1	0	0	S_3	0	1	0	1	S_2	S_1	S_0	14	3	No borrow
			Data																		
LTI	*A,byte	(A) – byte	0	0	1	1	0	1	1	1	Data								7	2	Borrow
	r,byte	(r) – byte	0	1	1	1	0	1	0	0	0	0	1	1	1	R_2	R_1	R_0	11	3	Borrow
			Data																		
	sr2,byte	(sr2) – byte	0	1	1	0	0	1	0	0	S_3	0	1	1	1	S_2	S_1	S_0	14	3	Borrow
			Data																		
NEI	*A,byte	(A) – byte	0	1	1	0	0	1	1	1	Data								7	2	No zero
	r,byte	(r) – byte	0	1	1	1	0	1	0	0	0	1	1	0	1	R_2	R_1	R_0	11	3	No zero
			Data																		

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1 / B3								Operation Code B2 / B4								State (Note 1)	Bytes	Skip Condition
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
Immediate Data (cont)																					
NEI	sr2,byte	(sr2) – byte	0	1	1	0	0	1	0	0	S_3	1	1	0	1	S_2	S_1	S_0	14	3	No zero
			Data																		
EQI	*A,byte	(A) – byte	0	1	1	1	0	1	1	1	Data								7	2	Zero
	r,byte	(r) – byte	0	1	1	1	0	1	0	0	0	1	1	1	1	R_2	R_1	R_0	11	3	Zero
			Data																		
	sr2,byte	(sr2) – byte	0	1	1	0	0	1	0	0	S_3	1	1	1	1	S_2	S_1	S_0	14	3	Zero
			Data																		
ONI	*A,byte	(A) ∧ byte	0	1	0	0	0	1	1	1	Data								7	2	No zero
	r,byte	(r) ∧ byte	0	1	1	1	0	1	0	0	0	1	0	0	1	R_2	R_1	R_0	11	3	No zero
			Data																		
	sr2,byte	(sr2) ∧ byte	0	1	1	0	0	1	0	0	S_3	1	0	0	1	S_2	S_1	S_0	14	3	No zero
			Data																		
OFFI	*A,byte	(A) ∧ byte	0	1	0	1	0	1	1	1	Data								7	2	Zero
	r,byte	(r) ∧ byte	0	1	1	1	0	1	0	0	0	1	0	1	1	R_2	R_1	R_0	11	3	Zero
			Data																		
	sr2,byte	(sr2) ∧ byte	0	1	1	0	0	1	0	0	S_3	1	0	1	1	S_2	S_1	S_0	14	3	Zero
			Data																		
Working Register																					
ADDW	wa	(A) ← (A) + ((V)•(wa))	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	14	3	
			Offset																		
ADCW	wa	(A) ← (A) + ((V)•(wa)) + (CY)	0	1	1	1	0	1	0	0	1	1	0	1	0	0	0	0	14	3	
			Offset																		
ADDNCW	wa	(A) ← (A) + ((V)•(wa))	0	1	1	1	0	1	0	0	1	0	1	0	0	0	0	0	14	3	No carry
			Offset																		
SUBW	wa	(A) ← (A) – ((V)•(wa))	0	1	1	1	0	1	0	0	1	1	1	0	0	0	0	0	14	3	
			Offset																		
SBBW	wa	(A) ← (A) – ((V)•(wa)) – (CY)	0	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	14	3	
			Offset																		
SUBNBW	wa	(A) ← (A) – ((V)•(wa))	0	1	1	1	0	1	0	0	1	0	1	1	0	0	0	0	14	3	No borrow
			Offset																		
ANAW	wa	(A) ← (A) ∧ ((V)•(wa))	0	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	14	3	
			Offset																		

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code (B1 / B3) 7	6	5	4	3	2	1	0	Operation Code (B2 / B4) 7	6	5	4	3	2	1	0	State (Note 1)	Bytes	Skip Condition
Working Register (cont)																					
ORAW	wa	(A) ← (A) V ((V)•(wa))	0	1	1	1	0	1	0	0	1	0	0	1	1	0	0	0	14	3	
			Offset																		
XRAW	wa	(A) ← (A)⊻((V)•(wa))	0	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	14	3	
			Offset																		
GTAW	wa	(A) – ((V)•(wa)) – 1	0	1	1	1	0	1	0	0	1	0	1	0	1	0	0	0	14	3	No borrow
			Offset																		
LTAW	wa	(A) – ((V)•(wa))	0	1	1	1	0	1	0	0	1	0	1	1	1	0	0	0	14	3	Borrow
			Offset																		
NEAW	wa	(A) – ((V)•(wa))	0	1	1	1	0	1	0	0	1	1	1	0	1	0	0	0	14	3	No zero
			Offset																		
EQAW	wa	(A) – ((V)•(wa))	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	14	3	Zero
			Offset																		
ONAW	wa	(A) ∧ ((V)•(wa))	0	1	1	1	0	1	0	0	1	1	0	0	1	0	0	0	14	3	No zero
			Offset																		
OFFAW	wa	(A) ∧ ((V)•(wa))	0	1	1	1	0	1	0	0	1	1	0	1	1	0	0	0	14	3	Zero
			Offset																		
ANIW	*wa,byte	((V)•(wa)) ← ((V)•(wa)) ∧ byte	0	0	0	0	0	1	0	1	Offset								19	3	
			Data																		
ORIW	*wa,byte	((V)•(wa)) ← ((V)•(wa)) V byte	0	0	0	1	0	1	0	1	Offset								19	3	
			Data																		
GTIW	*wa,byte	((V)•(wa)) – byte – 1	0	0	1	0	0	1	0	1	Offset								13	3	No borrow
			Data																		
LTIW	*wa,byte	((V)•(wa)) – byte	0	0	1	1	0	1	0	1	Offset								13	3	Borrow
			Data																		
NEIW	*wa,byte	((V)•(wa)) – byte	0	1	1	0	0	1	0	1	Offset								13	3	No zero
			Data																		
EQIW	*wa,byte	((V)•(wa)) – byte	0	1	1	1	0	1	0	1	Offset								13	3	Zero
			Data																		
ONIW	*wa,byte	((V)•(wa)) ∧ byte	0	1	0	0	0	1	0	1	Offset								13	3	No zero
			Data																		
OFFIW	*wa,byte	((V)•(wa)) ∧ byte	0	1	0	1	0	1	0	1	Offset								13	3	Zero
			Data																		

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1 / B3 7	6	5	4	3	2	1	0	Operation Code B2 / B4 7	6	5	4	3	2	1	0	State (Note 1)	Bytes	Skip Condition
16-Bit Arithmetic																					
EADD	EA,r2	(EA) ← (EA) + (r2)	0	1	1	1	0	0	0	0	0	1	0	0	0	0	R_1	R_0	11	2	
DADD	EA,rp3	(EA) ← (EA) + (rp3)	0	1	1	1	0	1	0	0	1	1	0	0	0	1	P_1	P_0	11	2	
DADC	EA,rp3	(EA) ← (EA) + (rp3) + (CY)	0	1	1	1	0	1	0	0	1	1	0	1	0	1	P_1	P_0	11	2	
DADDNC	EA,rp3	(EA) ← (EA) + (rp3)	0	1	1	1	0	1	0	0	1	0	1	0	0	1	P_1	P_0	11	2	No carry
ESUB	EA,r2	(EA) ← (EA) − (r2)	0	1	1	1	0	0	0	0	0	1	1	0	0	0	R_1	R_0	11	2	
DSUB	EA,rp3	(EA) ← (EA) − (rp3)	0	1	1	1	0	1	0	0	1	1	1	0	0	1	P_1	P_0	11	2	
DSBB	EA,rp3	(EA) ← (EA) − (rp3) − (CY)	0	1	1	1	0	1	0	0	1	1	1	1	0	1	P_1	P_0	11	2	
DSUBNB	EA,rp3	(EA) ← (EA) − (rp3)	0	1	1	1	0	1	0	0	1	0	1	1	0	1	P_1	P_0	11	2	No borrow
DAN	EA,rp3	(EA) ← (EA) ∧ (rp3)	0	1	1	1	0	1	0	0	1	0	0	0	1	1	P_1	P_0	11	2	
DOR	EA,rp3	(EA) ← (EA) ∨ (rp3)	0	1	1	1	0	1	0	0	1	0	0	1	1	1	P_1	P_0	11	2	
DXR	EA,rp3	(EA) ← (EA) ∀ (rp3)	0	1	1	1	0	1	0	0	1	0	0	1	0	1	P_1	P_0	11	2	
DGT	EA,rp3	(EA) − (rp3) − 1	0	1	1	1	0	1	0	0	1	0	1	0	1	1	P_1	P_0	11	2	No borrow
DLT	EA,rp3	(EA) − (rp3)	0	1	1	1	0	1	0	0	1	0	1	1	1	1	P_1	P_0	11	2	Borrow
DNE	EA,rp3	(EA) − (rp3)	0	1	1	1	0	1	0	0	1	1	1	0	1	1	P_1	P_0	11	2	No zero
DEQ	EA,rp3	(EA) − (rp3)	0	1	1	1	0	1	0	0	1	1	1	1	1	1	P_1	P_0	11	2	Zero
DON	EA,rp3	(EA) ∧ (rp3)	0	1	1	1	0	1	0	0	1	1	0	0	1	1	P_1	P_0	11	2	No zero
DOFF	EA,rp3	(EA) ∧ (rp3)	0	1	1	1	0	1	0	0	1	1	0	1	1	1	P_1	P_0	11	2	Zero
Multiply/Divide																					
MUL	r2	(EA) ← (A) x (r2)	0	1	0	0	1	0	0	0	0	0	1	0	1	1	R_1	R_0	32	2	
DIV	r2	(EA) ← (EA) ÷ (r2), (r2) ← Remainder	0	1	0	0	1	0	0	0	0	0	1	1	1	1	R_1	R_0	59	2	
Increment/Decrement																					
INR	r2	(r2) ← (r2) + 1	0	1	0	0	0	0	R_1	R_0									4	1	Carry
INRW	*wa	((V)•(wa)) ← ((V)•(wa)) + 1	0	0	1	0	0	0	0	0	Offset								16	2	Carry
INX	rp	(rp) ← (rp) + 1	0	0	P_1	P_0	0	0	1	0									7	1	
	EA	(EA) ← (EA) + 1	1	0	1	0	1	0	0	0									7	1	
DCR	r2	(r2) ← (r2) − 1	0	1	0	1	0	0	R_1	R_0									4	1	Borrow
DCRW	*wa	((V)•(wa)) ← ((V)•(wa)) − 1	0	0	1	1	0	0	0	0	Offset								16	2	Borrow
DCX	rp	(rp) ← (rp) − 1	0	0	P_1	P_0	0	0	1	1									7	1	
	EA	(EA) ← (EA) − 1	1	0	1	0	1	0	0	1									7	1	
Others																					
DAA		Decimal Adjust Accumulator	0	1	1	0	0	0	0	1									4	1	
STC		(CY) ← 1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	1	1	8	2	
CLC		(CY) ← 0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	1	0	8	2	

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1/B3 7	6	5	4	3	2	1	0	Operation Code B2/B4 7	6	5	4	3	2	1	0	State (Note 1)	Bytes	Skip Condition
Others (cont)																					
NEGA		(A) ← $\overline{(A)}$ + 1	0	1	0	0	1	0	0	0	0	0	1	1	1	0	1	0	8	2	
Rotate and Shift																					
RLD		Rotate left digit $(A_{3-0}) \leftarrow ((HL))_{7-4}$, $((HL))_{7-4} \leftarrow ((HL))_{3-0}$, $((HL))_{3-0} \leftarrow (A_{3-0})$	0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	17	2	
RRD		Rotate right digit $((HL))_{7-4} \leftarrow (A_{3-0})$, $((HL))_{3-0} \leftarrow ((HL))_{7-4}$, $(A_{3-0}) \leftarrow ((HL))_{3-0}$	0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	1	17	2	
RLL	r2	$(r2_{m+1}) \leftarrow (r2_m)$, $(r2_0) \leftarrow$ (CY), (CY) ← $(r2_7)$	0	1	0	0	1	0	0	0	0	0	1	1	0	1	R_1	R_0	8	2	
RLR	r2	$(r2_{m-1}) \leftarrow (r2_m)$, $(r2_7) \leftarrow$ (CY), (CY) ← $(r2_0)$	0	1	0	0	1	0	0	0	0	0	1	1	0	0	R_1	R_0	8	2	
SLL	r2	$(r2_{m+1}) \leftarrow (r2_m)$, $(r2_0) \leftarrow 0$, (CY) ← $(r2_7)$	0	1	0	0	1	0	0	0	0	0	1	0	0	1	R_1	R_0	8	2	
SLR	r2	$(r2_{m-1}) \leftarrow (r2_m)$, $(r2_7) \leftarrow 0$, (CY) ← $(r2_0)$	0	1	0	0	1	0	0	0	0	0	1	0	0	0	R_1	R_0	8	2	
SLLC	r2	$(r2_{m+1}) \leftarrow (r2_m)$, $(r2_0) \leftarrow 0$, (CY) ← $(r2_7)$	0	1	0	0	1	0	0	0	0	0	0	0	0	1	R_1	R_0	8	2	Carry
SLRC	r2	$(r2_{m-1}) \leftarrow (r2_m)$, $(r2_7) \leftarrow 0$, (CY) ← $(r2_0)$	0	1	0	0	1	0	0	0	0	0	0	0	0	0	R_1	R_0	8	2	Carry
DRLL	EA	$(EA_{n+1}) \leftarrow (EA_n)$, $(EA_0) \leftarrow$ (CY), (CY) ← (EA_{15})	0	1	0	0	1	0	0	0	1	0	1	1	0	1	0	0	8	2	
DRLR	EA	$(EA_{n-1}) \leftarrow (EA_n)$, $(EA_{15}) \leftarrow$ (CY), (CY) ← (EA_0)	0	1	0	0	1	0	0	0	1	0	1	1	0	0	0	0	8	2	
DSLL	EA	$(EA_{n+1}) \leftarrow (EA_n)$, $(EA_0) \leftarrow 0$, (CY) ← (EA_{15})	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0	8	2	
DSLR	EA	$(EA_{n-1}) \leftarrow (EA_n)$, $(EA_{15}) \leftarrow 0$, (CY) ← (EA_0)	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	8	2	
Jump																					
JMP	*word	(PC) ← word	0	1	0	1	0	1	0	0	Low addr								10	3	
			High addr																		
JB		$(PC_H) \leftarrow (B)$, $(PC_L) \leftarrow (C)$	0	0	1	0	0	0	0	1									4	1	
JR	word	(PC) ← (PC) + 1 + jdisp 1	1	1	← jdisp1 →														10	1	
JRE	*word	(PC) ← (PC) + 2 + jdisp	0	1	0	0	1	1	1	← jdisp →									10	2	
JEA		(PC) ← (EA)	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	8	2	
Call																					
CALL	*word	$((SP) - 1) \leftarrow ((PC) + 3)_H$, $((SP) - 2) \leftarrow ((PC) + 3)_L$, (PC) ← word, (SP) ← (SP) − 2	0	1	0	0	0	0	0	0	Low addr								16	3	
			High addr																		
CALB		$((SP) - 1) \leftarrow ((PC) + 2)_H$, $((SP) - 2) \leftarrow ((PC) + 2)_L$, $(PC_H) \leftarrow (B)$, $(PC_L) \leftarrow (C)$, (SP) ← (SP) − 2	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	17	2	

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code B1/B3 7	6	5	4	3	2	1	0	Operation Code B2/B4 7	6	5	4	3	2	1	0	State (Note 1)	Bytes	Skip Condition
Call (cont)																					
CALF	*word	$((SP)-1) \leftarrow ((PC)+2)_H$, $((SP)-2) \leftarrow ((PC)+2)_L$, $(PC_{15\text{-}11}) \leftarrow 00001$, $(PC_{10\text{-}0}) \leftarrow fa$, $(SP) \leftarrow (SP)-2$	0	1	1	1	1	←			fa			→					13	2	
CALT	word	$((SP)-1) \leftarrow ((PC)+1)_H$, $((SP)-2) \leftarrow ((PC)+1)_L$, $(PC_L) \leftarrow (128+2ta)$, $(PC_H) \leftarrow (129+2ta)$, $(SP) \leftarrow (SP)-2$	1	0	0	←		ta		→									16	1	
SOFTI		$((SP)-1) \leftarrow (PSW)$, $((SP)-2) \leftarrow ((PC)+1)_H$, $((SP)-3) \leftarrow ((PC)+1)_L$, $(PC) \leftarrow 0060H$, $(SP) \leftarrow (SP)-3$	0	1	1	1	0	0	1	0									16	1	
Return																					
RET		$(PC_L) \leftarrow ((SP)), (PC_H) \leftarrow ((SP)+1)$, $(SP) \leftarrow (SP)+2$	1	0	1	1	1	0	0	0									10	1	
RETS		$(PC_L) \leftarrow ((SP)), (PC_H) \leftarrow ((SP)+1)$, $(SP) \leftarrow (SP)+2$, $(PC) \leftarrow (PC)+n$	1	0	1	1	1	0	0	1									10	1	Unconditional Skip
RETI		$(PC_L) \leftarrow ((SP)), (PC_H) \leftarrow ((SP)+1)$, $(PSW) \leftarrow ((SP)+2)$, $(SP) \leftarrow (SP)+3$	0	1	1	0	0	0	1	0									13	1	
Skip																					
BIT	*bit, wa	Skip if ((V)•(wa)) bit = 1	0	1	0	1	1	B_2	B_1	B_0	←				Offset			→	10	2	Bit Test
SK	f	Skip if f = 1	0	1	0	0	1	0	0	0	0	0	0	0	1	F_2	F_1	F_0	8	2	f = 1
SKN	f	Skip if f = 0	0	1	0	0	1	0	0	0	0	0	0	1	1	F_2	F_1	F_0	8	2	f = 0
SKIT	irf	Skip if irf = 1, then reset irf	0	1	0	0	1	0	0	0	0	1	0	I_4	I_3	I_2	I_1	I_0	8	2	irf = 1
SKNIT	irf	Skip if irf = 0 Reset irf if irf = 1 and don't skip	0	1	0	0	1	0	0	0	0	1	1	I_4	I_3	I_2	I_1	I_0	8	2	irf = 0
CPU Control																					
NOP		No operation	0	0	0	0	0	0	0	0									4	1	
EI		Enable interrupt	1	0	1	0	1	0	1	0									4	1	
DI		Disable interrupt	1	0	1	1	1	0	1	0									4	1	
HLT		Set HALT mode	0	1	0	0	1	0	0	0	0	0	1	1	1	0	1	1	12	2	
STOP		Set STOP mode	0	1	0	0	1	0	0	0	1	0	1	1	1	0	1	1	12	2	