

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

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)

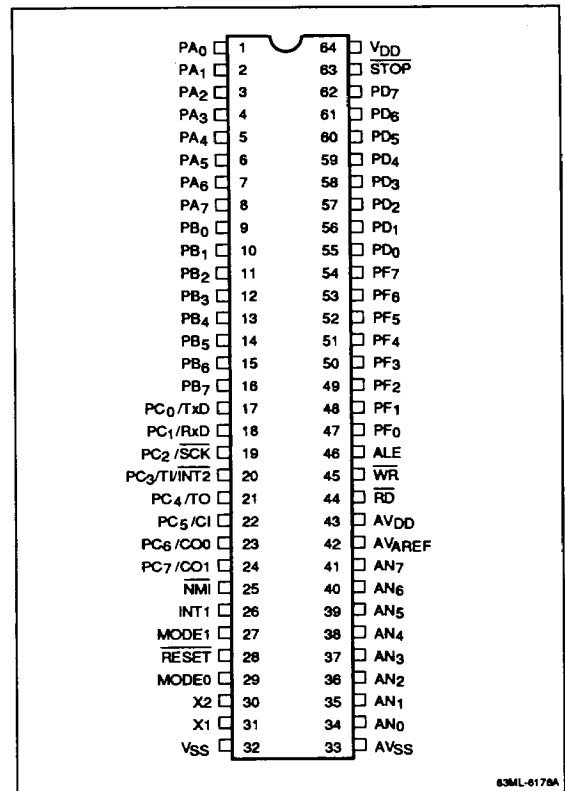
Part Number	Package	ROM
μPD78C11ACW-xxx	64-pin plastic SDIP	4K mask ROM
μPD78C11AGF-xxx-3BE	64-pin plastic QFP	
μPD78C11AGQ-xxx-36	64-pin plastic QUIP	
μPD78C11AL-xxx	68-pin PLCC	
μPD78C12ACW-xxx	64-pin plastic SDIP	8K mask ROM
μPD78C12AGF-xxx-3BE	64-pin plastic QFP	
μPD78C12AGQ-xxx-36	64-pin plastic 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.

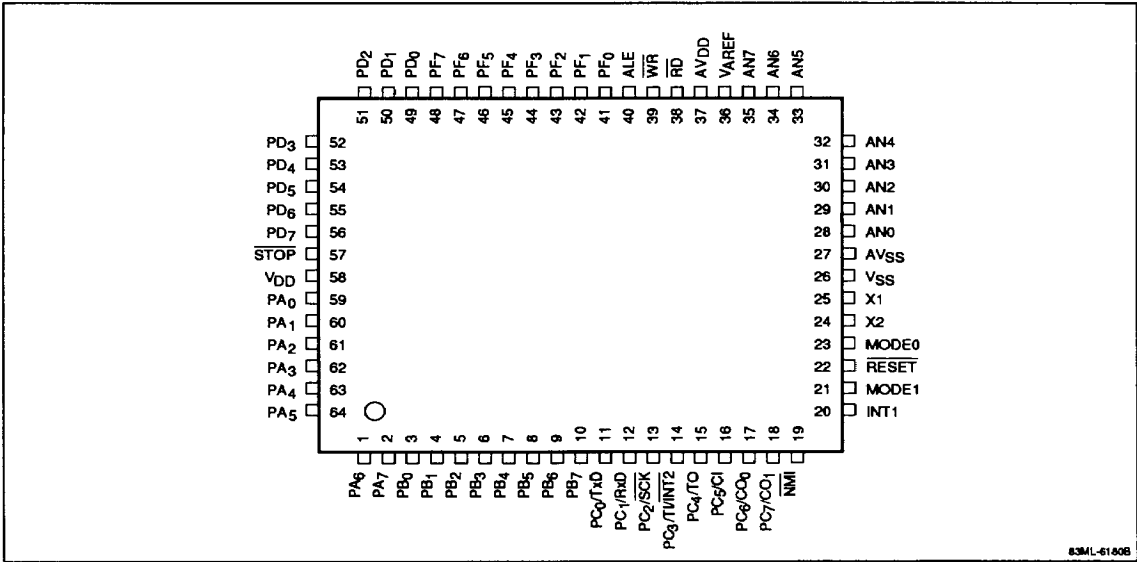
Pin Configurations

64-Pin QUIP or SDIP (Plastic or Ceramic)



63ML-6178A

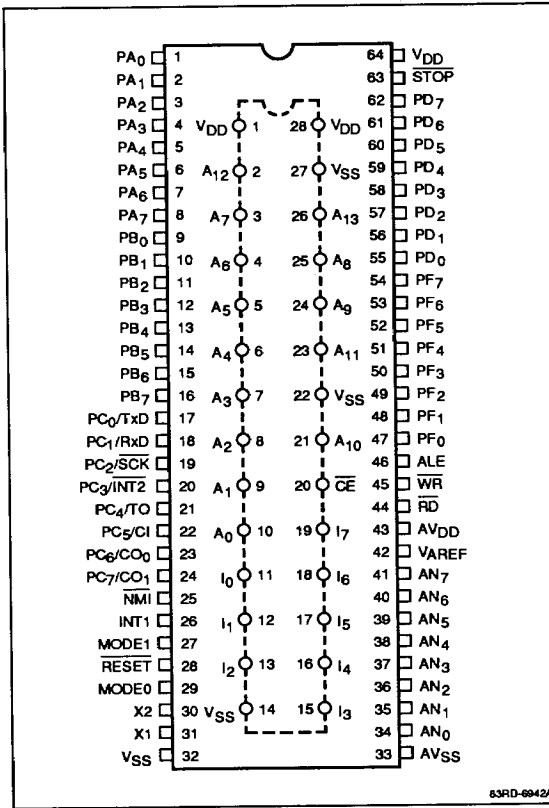
64-Pin Plastic QFP



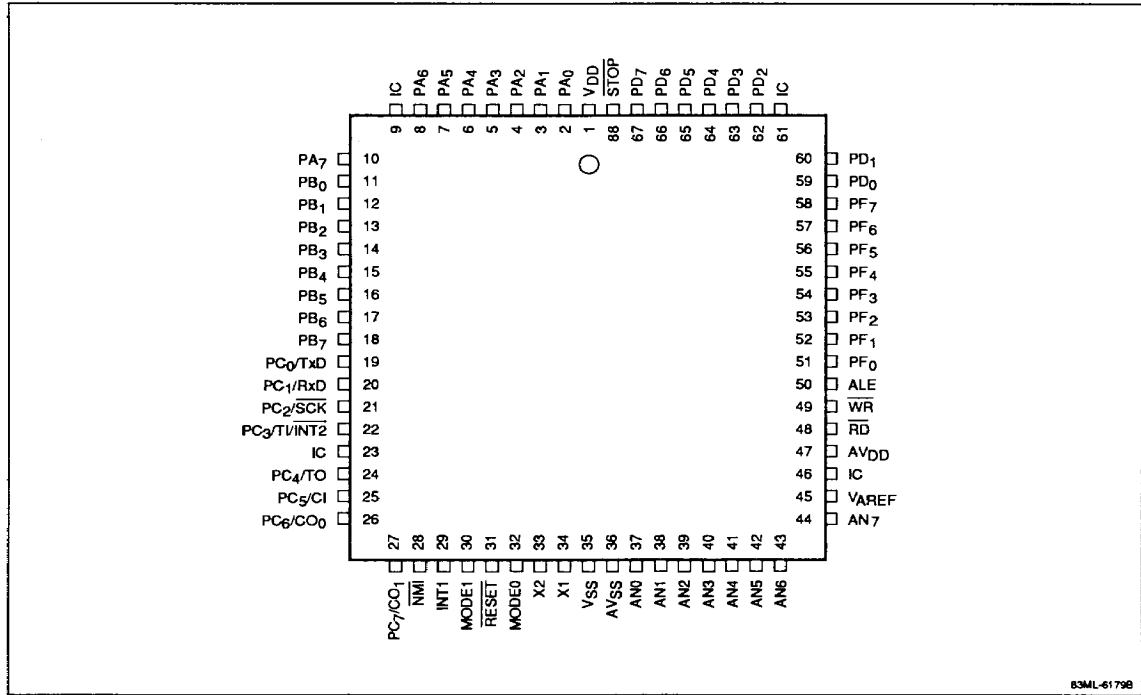
5

63ML-6160B

64-Pin Ceramic Piggyback QUIP



68-Pin PLCC



5

63ML-6179B

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
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 /SCR	Port C I/O line 2; serial clock I/O
PC ₃ /TI/INT ₂	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)
RD	Read strobe output
RESET	Reset input
STOP	Stop mode control input
V _{AREF}	A/D converter reference voltage
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
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; 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 PD₀-PD₇ 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.

CO₀, CO₁ (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 μ PD78C11A/C12A/C14A, the zero-cross detection circuitry will not function.

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

$\overline{\text{INT2}}$ 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 $\overline{\text{IO}}$ signal, and MODE1 outputs the $\overline{\text{M1}}$ 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 K Ω is $4 \leq R \leq 0.4 t_{\text{CYC}}$ where t_{CYC} is in ns units.

$\overline{\text{NMI}}$ (Nonmaskable Interrupt)

Falling edge, Schmitt triggered nonmaskable interrupt input.

PA₀-PA₇ (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 μ PD78C11A/C12A/C14A.

PB₀-PB₇ (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 μ PD78C11A/C12A/C14A.

PC₀-PC₇ (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 μ PD78C11A/C12A/C14A.

PD₀-PD₇ (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.

PF₀-PF₇ (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.

$\overline{\text{RD}}$ (Read Strobe)

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

$\overline{\text{RESET}}$ (Reset)

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

RxD (Receive Data)

Serial data input terminal.

SCK (Serial Clock)

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

STOP (STOP Mode Control Input)

A low-level input on $\overline{\text{STOP}}$ (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.

$\overline{\text{WR}}$ (Write Strobe)

The three-state $\overline{\text{WR}}$ 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. $\overline{\text{WR}}$ 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.

V_{DD} (A/D Converter Power)

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

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

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

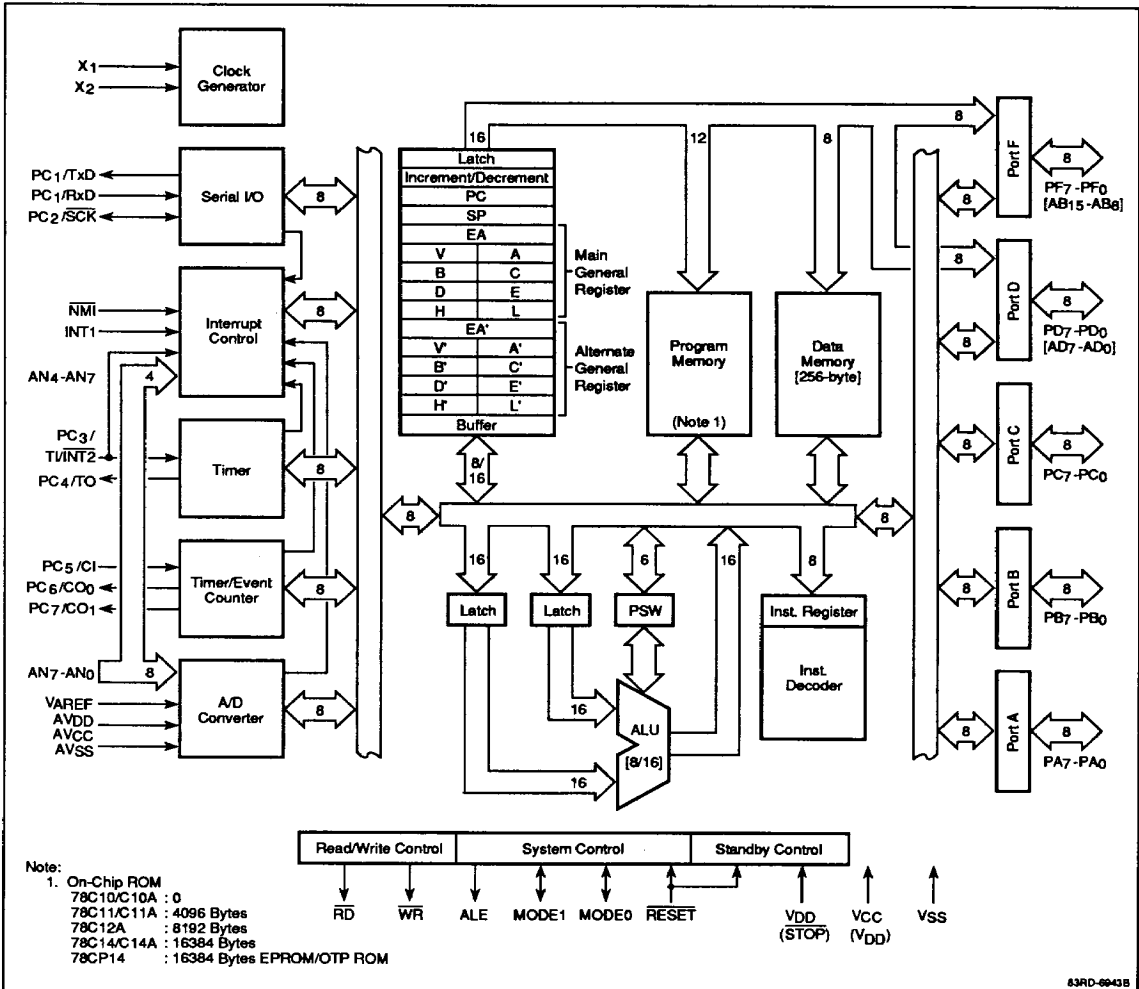
V_{DD} (Power Supply)

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

V_{SS} (Ground)

Ground potential.

Block Diagram



5

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

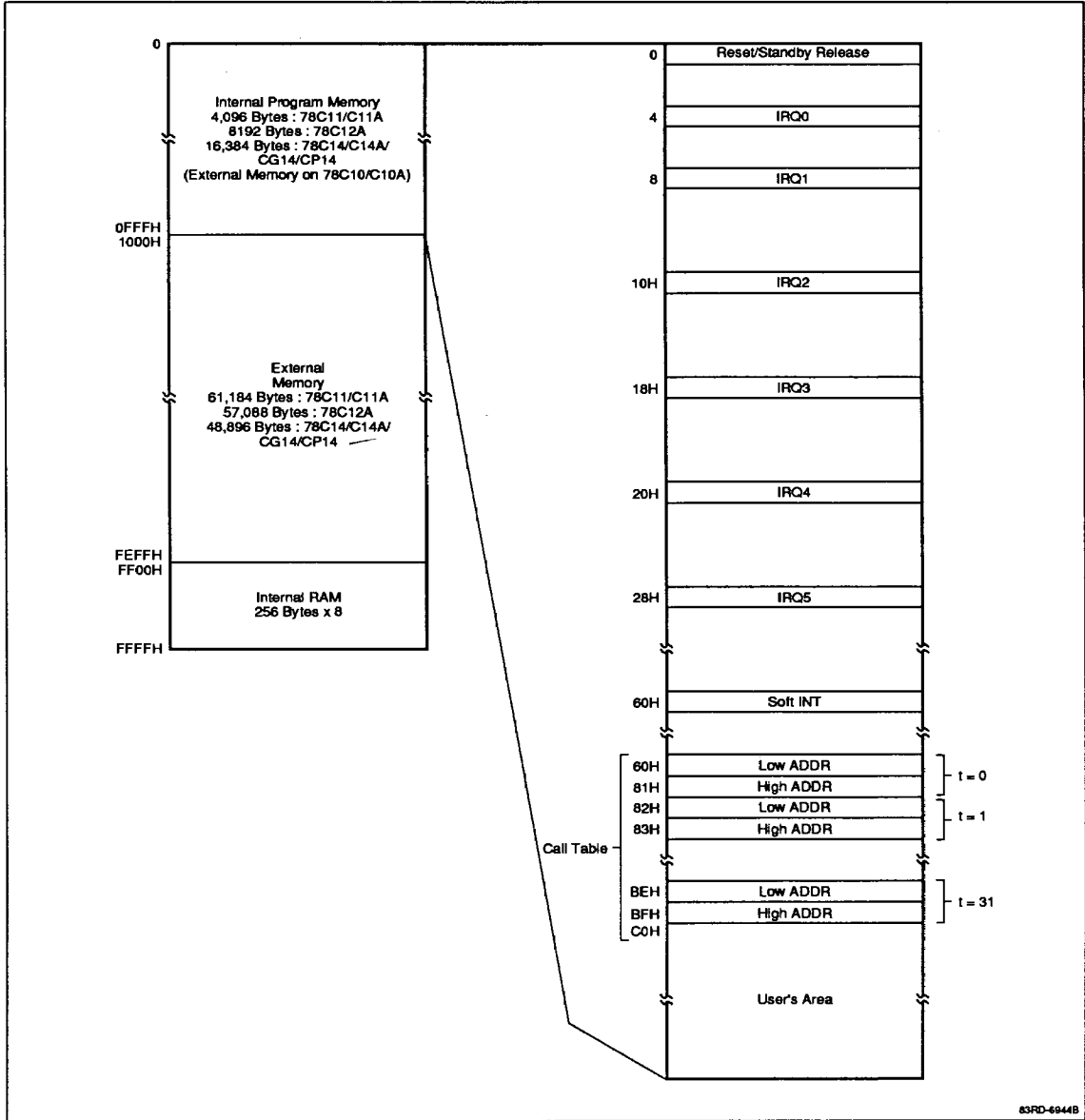
The 16-bit multifunctional timer/event counter (figure 3) can be used for the following operations:

- 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

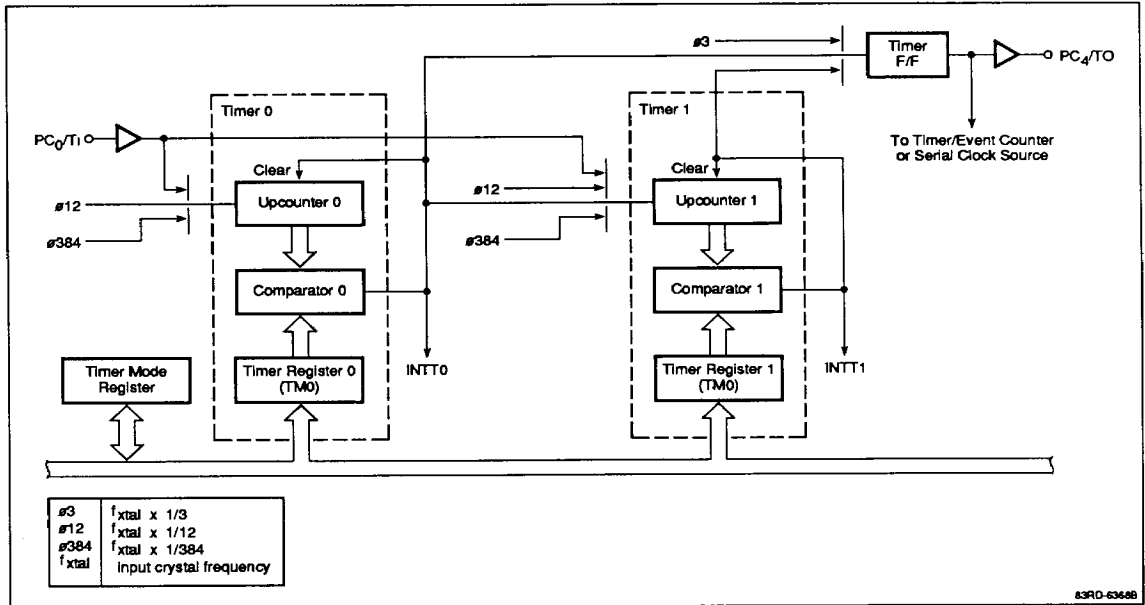
Figure 1. Memory Map



5

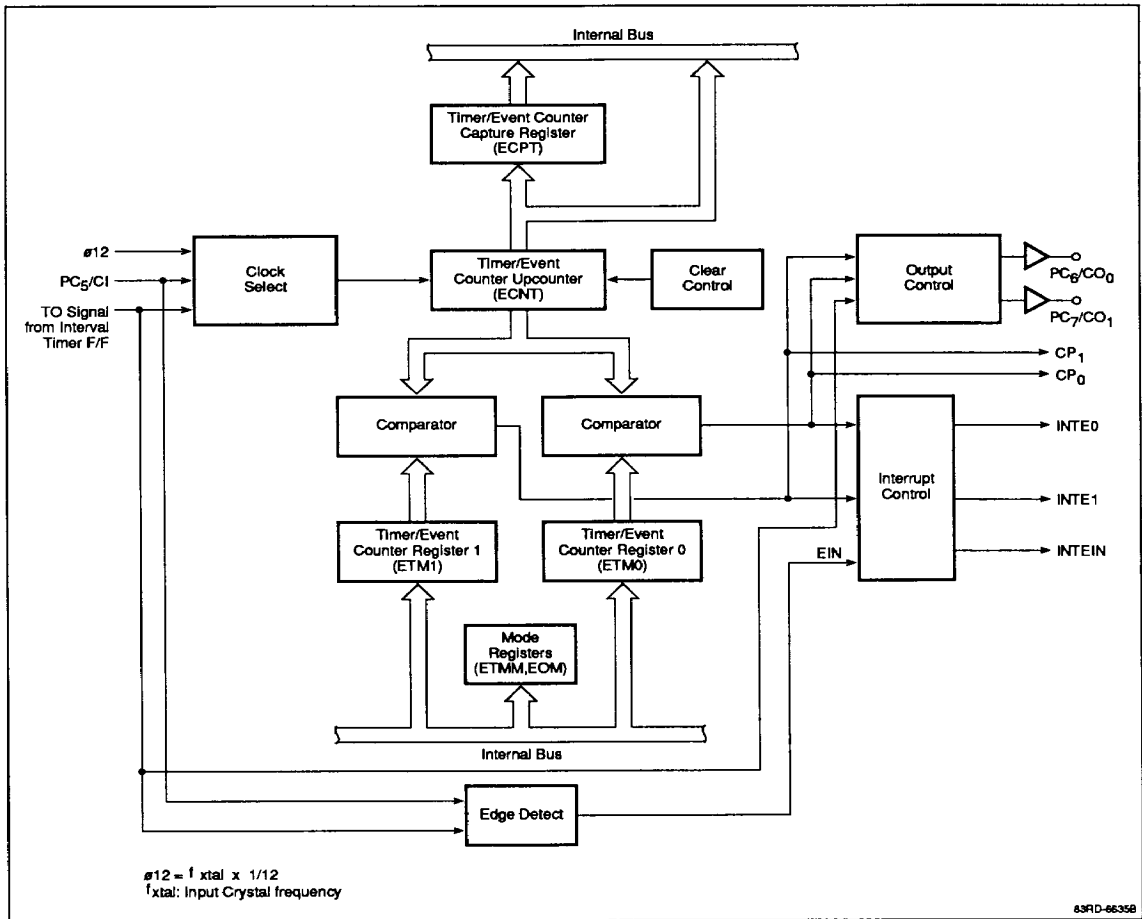
83RD-6944B

Figure 2. Timer Block Diagram



83RD-6368B

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 $V_{AREF} = 0V$.

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

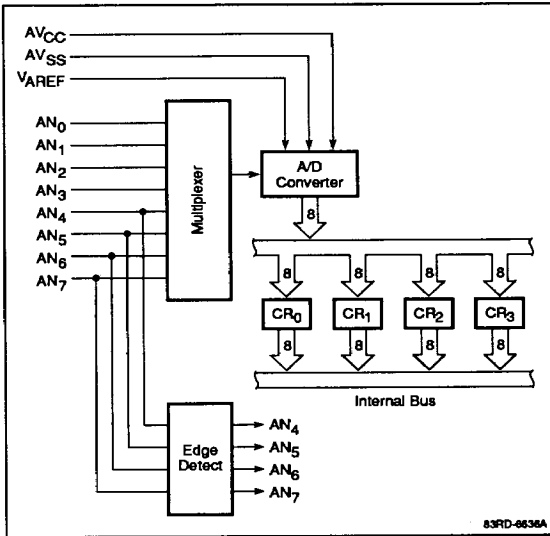
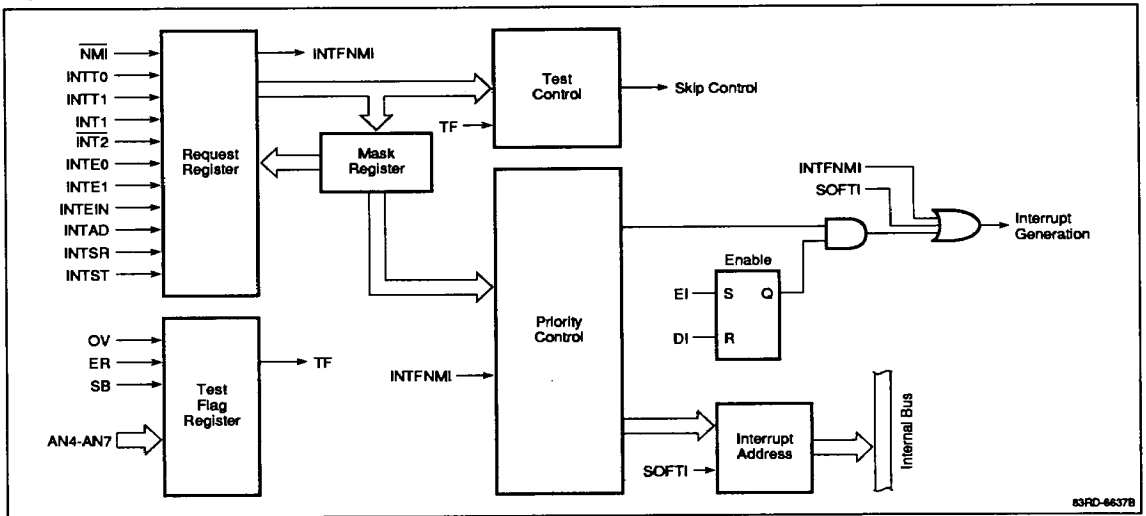


Figure 5. Interrupt Structure Block Diagram



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 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 NMI or 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 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 STOP to a high level. The oscillator stabilization time is fixed at 52.4 ms; 52.4 ms after STOP is raised, instruction execution will automatically begin at location 0. You can increase the stabilization time by holding 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 INT2 terminals (used common to TI and PC₃) 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.

Table 2. Interrupt Sources

Interrupt Request	Interrupt Address	Type of Interrupt	Internal/External
IRQ0	4	NMI (Nonmaskable interrupt)	External
IRQ1	8	INTT0, INTT1 (Coincidence signals from timers 0, 1)	Internal
IRQ2	16	INT1, 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)	Internal
IRQ6	96	SOFTI instruction	Internal

Figure 6. Universal Serial Interface Block Diagram

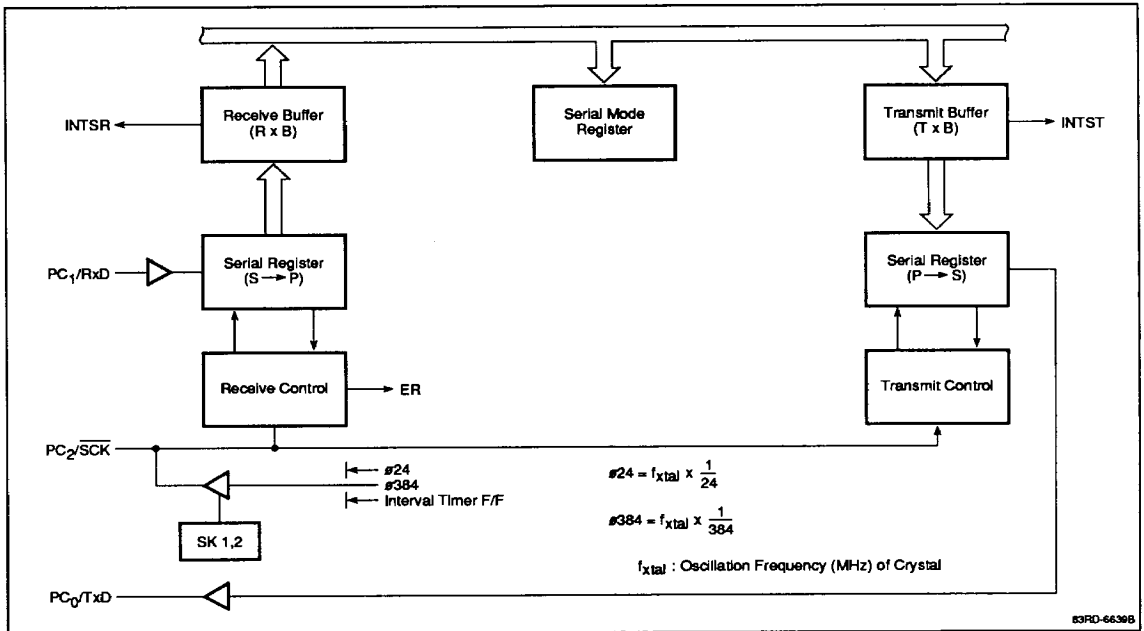
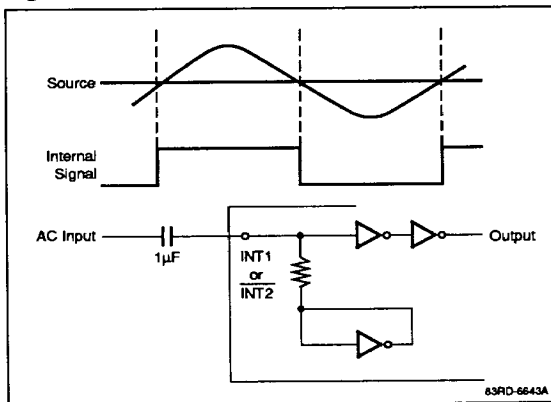


Figure 7. Zero-Crossing Detection Circuit



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

ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

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 $V_{DD} + 0.5$ V
STOP pin (μPD78CP14 only)	-0.5 to +13.5 V
Output voltage, V_O	-0.5 to $V_{DD} + 0.5$ 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, V_{REF}	-0.5 to $AV_{DD} + 0.3$ V
Operating temperature, T_{OPR}	-40 to +85°C
$f_{XTAL} \leq 15$ MHz	
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

$T_A = 25^\circ\text{C}; V_{DD} = V_{SS} = 0$ 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	

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 _{osc})	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 _{φL} , t _{φH})	20		250	ns	
			20		167	ns	μPD78CP14

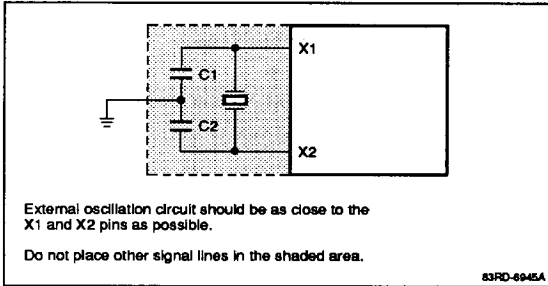
Notes:

- (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:

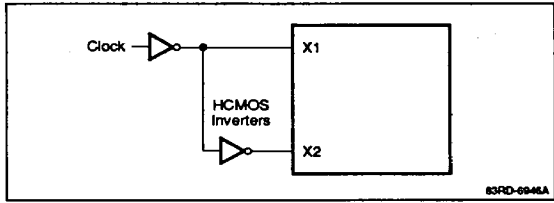
$$C_L = \frac{C_1 \times C_2}{C_1 + C_2} + C_S$$

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



Recommended External Clock Diagram



Resonator and Capacitance Requirements

$T_A = -40$ to $+85^\circ\text{C}$

Manufacturer	Product Number	C1, C2 (pF)	Conditions
Murata	CSA15.0MX3	22	μPD78C10, 78C11, 78C14, 78C14A, 78CG14
	CSA10.0MT	30	
	CST10.0MT	Not required	
	CSA8.00MG	30	Applies to all μPD78C1x/C1xA/CG14
	CST6.00MG	Not required	
	CSA12.0MT	30	
	CST12.0MT	Not required	
	CSA15.00MX001	15	
	CSA7.37MT	30	μPD78C10A/78C11A/78C12A
CST7.37MT	Not required		
TDK	FCR12.0MC	Not required	μPD78C10/78C11/78C14/ 78C14A/78CG14

DC Characteristics

$T_A = -40^\circ$ to $+85^\circ\text{C}$; $V_{DD} = +5.0\text{ V} \pm 10\%$; $V_{DD} = +5.0\text{ V} \pm 5\%$ (μPD78C14 only); $V_{SS} = 0\text{ V}$

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	$I_{OL} = 2.0\text{ mA}$
Output voltage, high	V_{OH}	$V_{DD}-1.0$			V	$I_{OH} = 1.0\text{ mA}$
		$V_{DD}-0.5$			V	$I_{OH} = -100\ \mu\text{A}$
Data retention voltage	V_{DDDR}	2.5			V	STOP mode
Input current	I_{I1}			± 200	μA	INT1 (Note 2); T1 (PC ₃) (Note 3); $0\text{ V} \leq V_i \leq V_{DD}$
Input current (μPD78CG14 only)	I_{I2}			± 200	μA	INT1 (Note 2); T1 (PC ₃) (Note 3); $0\text{ V} \leq V_i \leq V_{DD}$
Input current (μPD78CG14 only)	I_{I3}			-300	μA	I_{0-7} (upper input pin); $V_i = 0$
Input leakage current	I_{LI}			± 10	μA	All except INT1, T1 (PC ₃), $0\text{ V} \leq V_i \leq V_{DD}$
Output leakage current	I_{LO}			± 10	μA	$0\text{ V} \leq V_O \leq V_{DD}$
A_{VDD} supply current	A_{DD1}		0.5	1.3	mA	$f = 15\text{ MHz}$
	A_{DD2}		10	20	μA	STOP mode
V_{DD} supply current	I_{DD1}		13	25	mA	Normal operation; $f = 15\text{ MHz}$; (μPD78C10/C10A/C11/C11A/C12A only)
	I_{DD2}		7	13	mA	HALT mode; $f = 15\text{ MHz}$; (μPD78C10/C10A/C11/C11A/C12A only)
	I_{DD3}		16	30	mA	Normal operation; $f = 15\text{ MHz}$; (μPD78C14/C14A/CG14)
	I_{DD4}			32	mA	Normal operation; $f = 15\text{ MHz}$; (μPD78CP14 only)
	I_{DD5}		8	15	mA	HALT mode; $f = 15\text{ MHz}$; (μPD78C14/C14A/CG14/CP14 only)
Data retention current	I_{DDDR}		1	15	μA	$V_{DDDR} = 2.5\text{ V}$ (Note 4)
				300		(μPD78CP14 only-Note 4)
			10	50	μA	$V_{DDDR} = 5.0\text{ V} \pm 10\%$ (Note 4)
				1	mA	(μPD78CP14 only-Note 4)
Pullup resistor	R_L	17	27	75	kΩ	Port A, B, C; $3.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$; $V_i = 0\text{ V}$ (μPD78C11A/C12A/C14A only)

Notes:

- (1) Inputs RESET, STOP, NMI, SCK, INTP1, T1, 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
SCK cycle time	t_{CYK}	0.8		μs	SCK input (Notes 1, 3)
		0.4		μs	SCK input (Note 2)
		1.6		μs	SCK output (Note 3)
SCK width low	t_{KLL}	335		ns	SCK input (Notes 1, 3)
		160		ns	SCK input (Note 2)
		700		ns	SCK output (Note 3)
SCK width high	t_{KHH}	335		ns	SCK input (Notes 1, 3)
		160		ns	SCK input (Note 2)
		700		ns	SCK output (Note 3)
RxD setup time to SCK ↑	t_{RXK}	80		ns	(Note 1)
RxD hold time after SCK ↑	t_{KRX}	80		ns	(Note 1)
SCK ↓ TxD delay time	t_{KTX}		210	ns	(Note 1)

Notes:

- (1) 1 x baud rate in synchronous or I/O interface mode. (3) $f_{XTAL} = 15$ MHz.
 (2) 16 x baud rate or 64 x baud rate in asynchronous mode.

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)

$T_A = -40^\circ$ to $+85^\circ\text{C}$; $V_{DD} = AV_{DD} = +5.0\text{ V} \pm 10\%$ ($\pm 5\%$ on μPD78CP14); $V_{SS} = 0\text{ V}$

Parameter	Symbol	Min	Max	Unit	Conditions
RESET pulse width high, low	t_{RSH} , t_{RSL}	10		μs	
NMI pulse width high, low	t_{NIH} , t_{NIL}	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)
M ₁ setup time to ALE ↓	t _{ML}	30		ns	(Note 3)
M ₁ hold time after ALE ↓	t _{LM}	35		ns	(Note 3)
I/O/M setup time to ALE ↓	t _{IL}	30		ns	(Note 3)
I/O/M hold time after ALE ↓	t _{LI}	35		ns	(Note 3)
Address to WR ↓ delay	t _{AW}	100		ns	(Notes 2, 3)
ALE ↓ to data output	t _{LDW}		180	ns	(Notes 2, 3)
WR ↓ to data output	t _{WD}		100	ns	(Note 2)
ALE ↓ to WR ↓ delay time	t _{LW}	15		ns	(Notes 2, 3)
Data setup time to WR ↑	t _{DW}	165		ns	(Notes 2, 3)
Data hold time to WR ↑	t _{WDH}	60		ns	(Notes 2, 3)
WR ↑ to ALE ↑ delay time	t _{WL}	80		ns	(Notes 2, 3)
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 _{C11H} , t _{C11L} (Note 2)	Min	6T (TI input - PC ₃)
t _{C12H} , t _{C12L} (Note 3)	Min	48T (TI input - PC ₃)
t _{I1H} , t _{I1L}	Min	36T (INT1)
t _{I2H} , t _{I2L}	Min	36T (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

Symbol	Min/Max (ns)	Calculation Formula
t _{IL}	Min	2T - 100
t _{LI}	Min	T - 30
t _{AW}	Min	3T - 100
t _{LDW}	Max	T + 110
t _W	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 (SCK input) (Note 1)
	Min	24T (SCK output)
t _{KKL}	Min	5T + 5 (SCK input) (Note 1)
	Min	12T - 100 (SCK output)
t _{KKH}	Min	5T + 5 (SCK input) (Note 1)
	Min	12T - 100 (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.

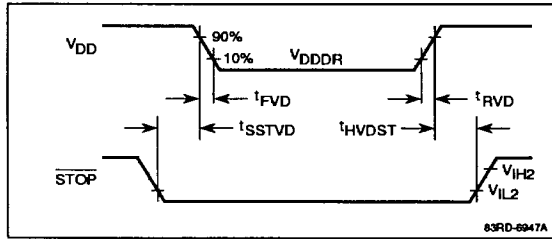
Data Memory STOP Mode Data Retention Characteristics

T_A = -40 to 85°C

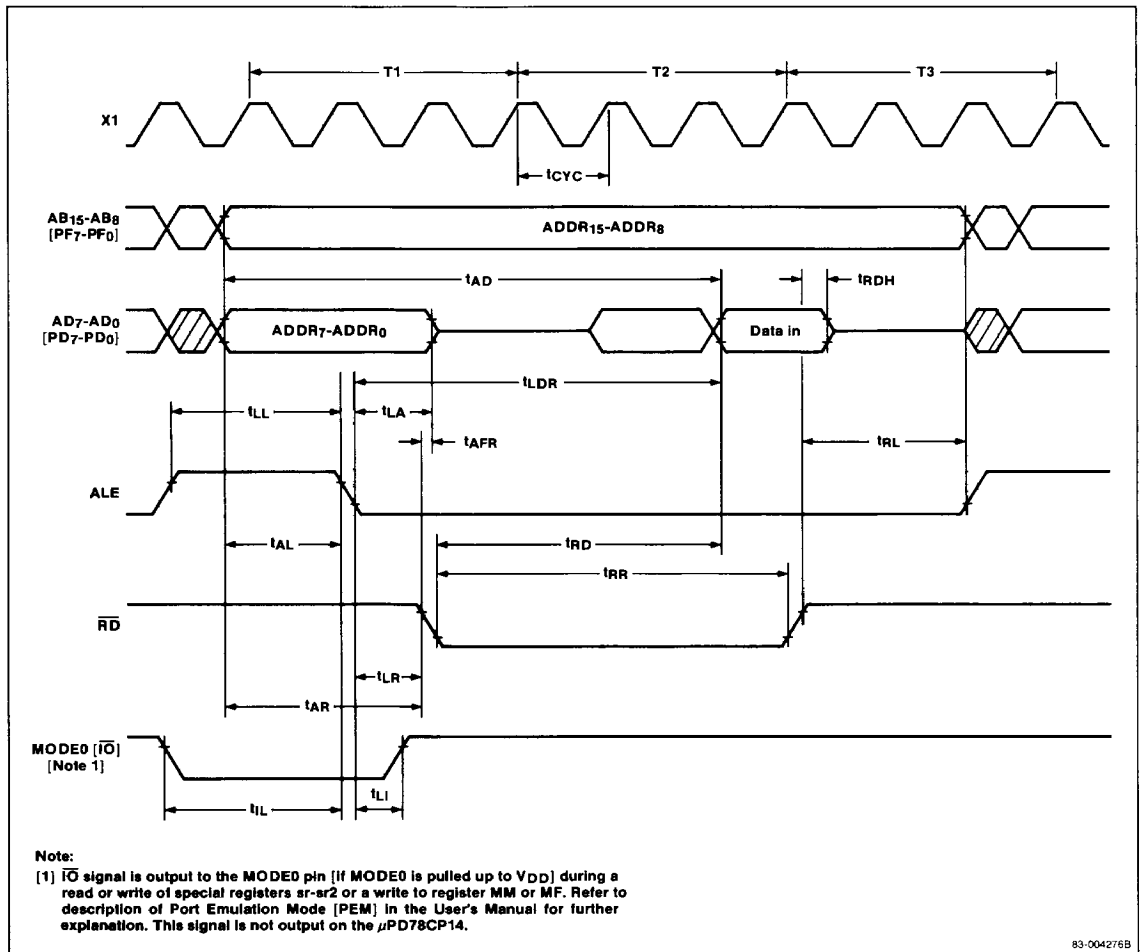
Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Data retention power supply voltage	V _{DDDR}	2.5		5.5	V	
Data retention power supply current	I _{DDDR}		1	15	μA	V _{DDDR} = 2.5 V
			15	50	μA	V _{DDDR} = 5.0 V ± 10%
				300	μA	V _{DDDR} = 2.4 V (μPD78CP14)
				1	mA	V _{DDDR} = 5.0 V ± 5% (μPD78CP14)
V _{DD}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

Timing Waveforms

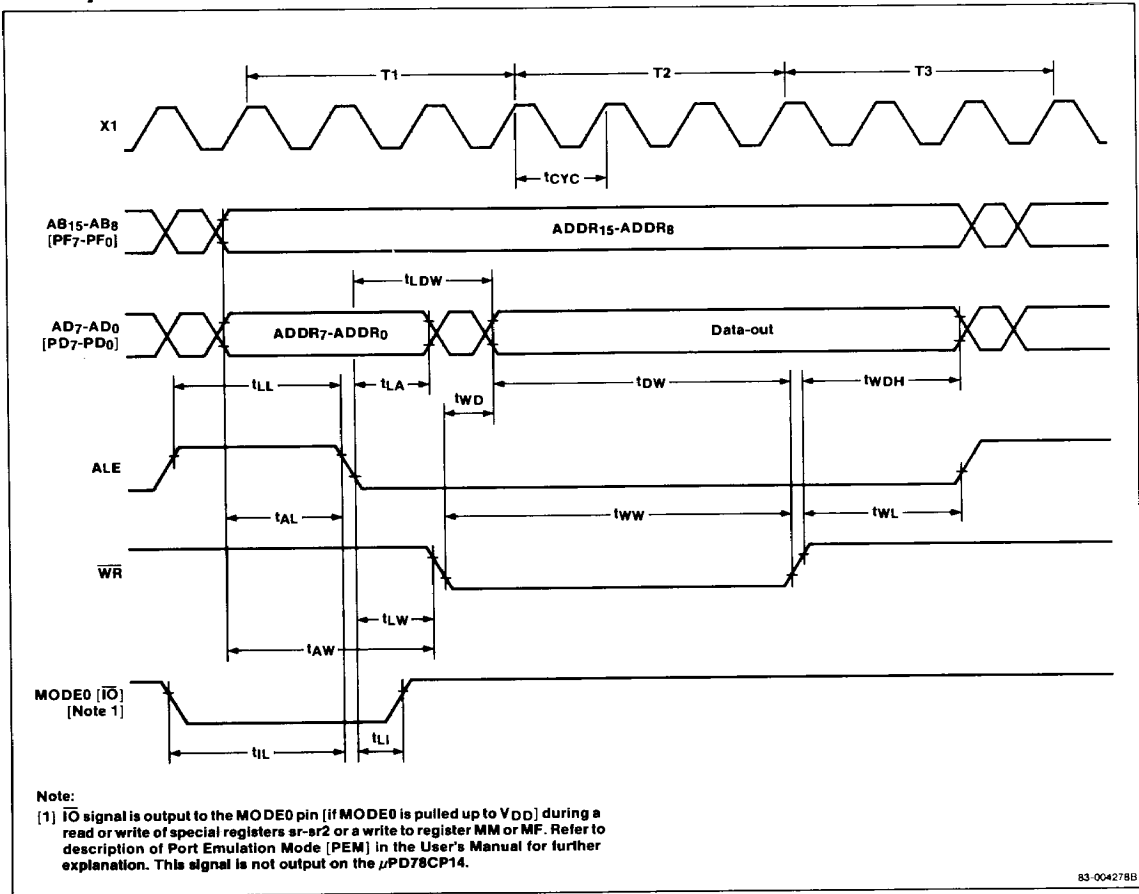
Data Retention Timing



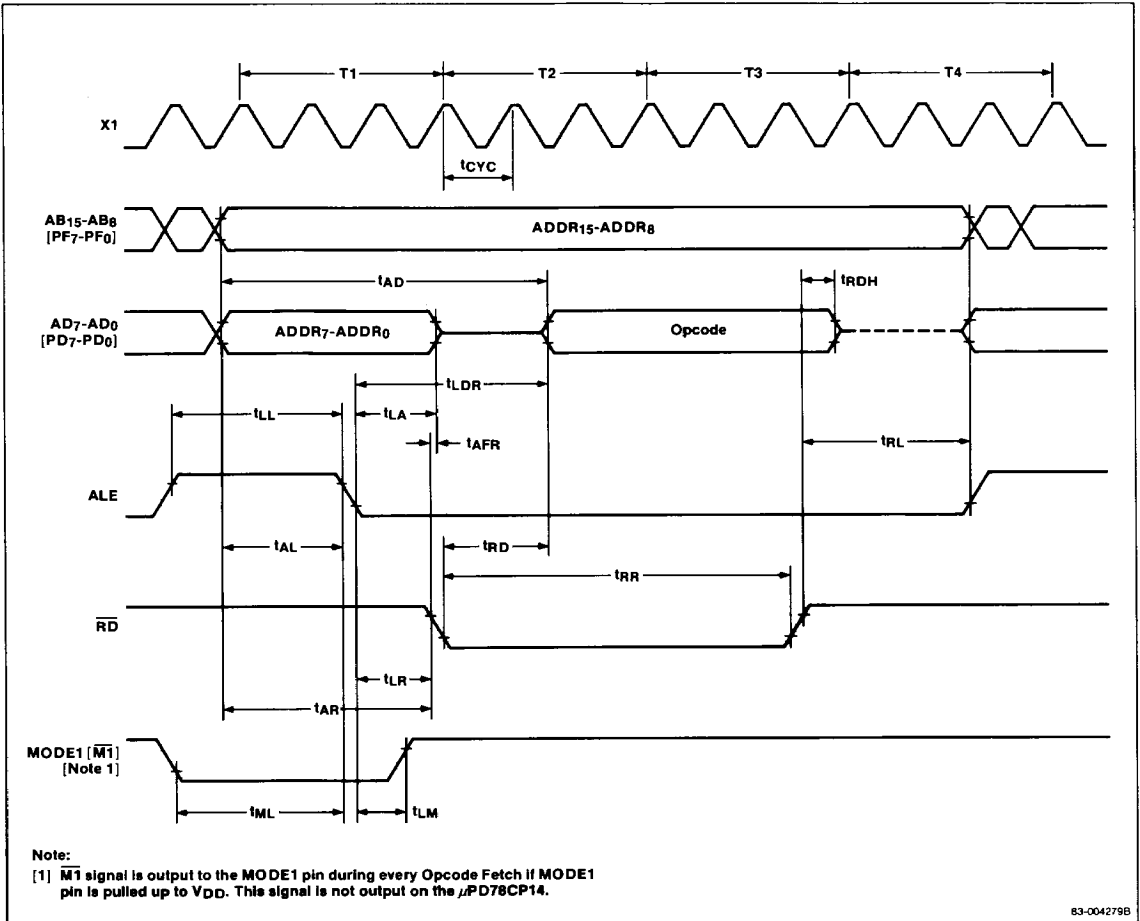
Read Operation



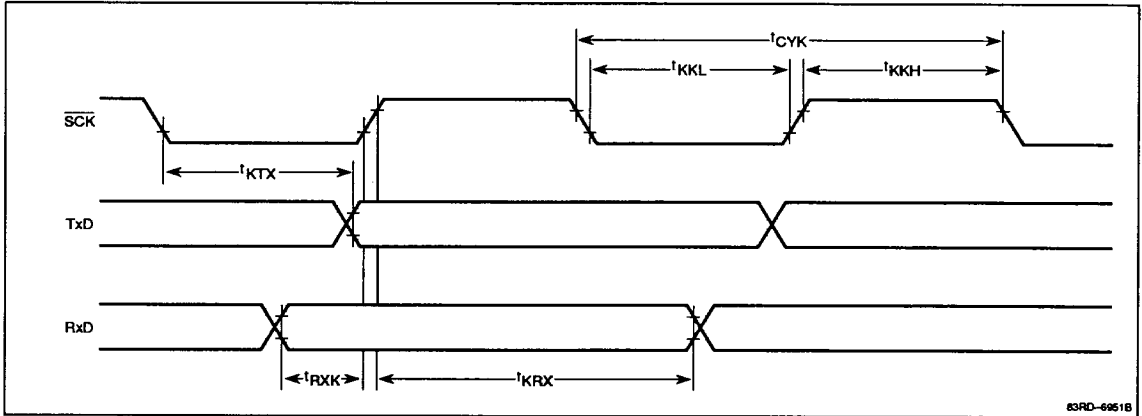
Write Operation



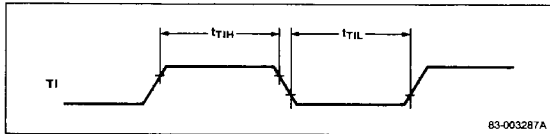
Opcode Fetch Operation



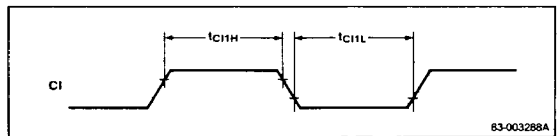
Serial Operation Transmit/Receive Timing



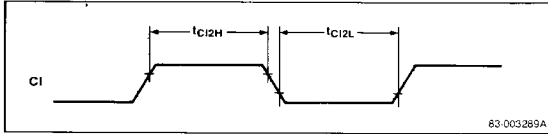
Timer Input Timing



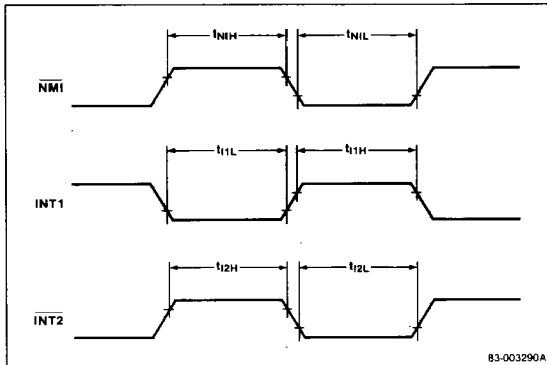
Timer/Event Counter Input Timing: Event Counter Mode



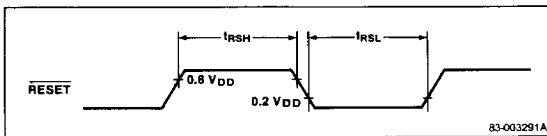
**Timer/Event Counter Input Timing:
Pulse Width Measurement Mode**



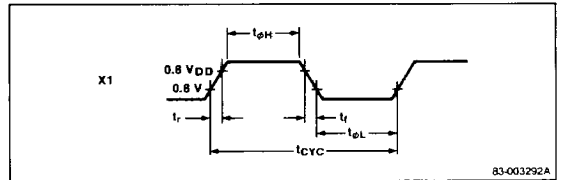
Interrupt Input Timing



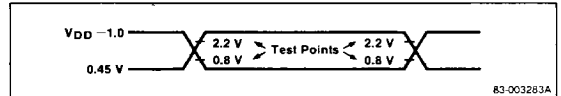
RESET Input Timing



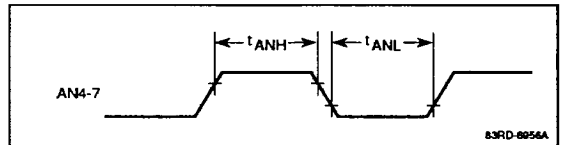
External Clock Timing



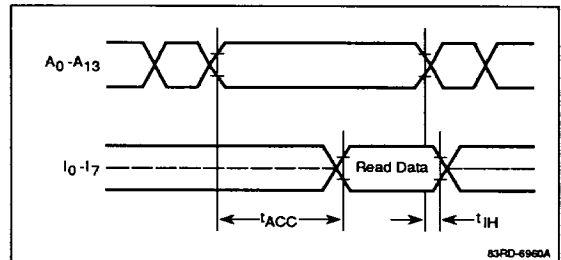
AC Timing Test Points



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
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
RESET	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
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 V_{PP} 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).

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}
V _{REF}	Connect to V _{SS}
V _{DD}	Connect to V _{SS}
V _{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 A₁₄ 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 A₁₄ 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 A₀-A₁₄.
- (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 D₀-D₇ 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/cm² (ultraviolet ray intensity x exposure time) is required to completely erase written data. Erasure by an ultraviolet lamp rated at 12 mW/cm² 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 ≤ V _I ≤ 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 ≤ V _O ≤ 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; CE = V _{IH} ; V _I = V _{IH}
V _{PP} power current	I _{PP}	I _{PP}			30	mA	Program memory read mode; CE = V _{IL} ; \overline{OE} = V _{IH}
					1	100	μA

* Corresponding symbols of the μPD27C256A.

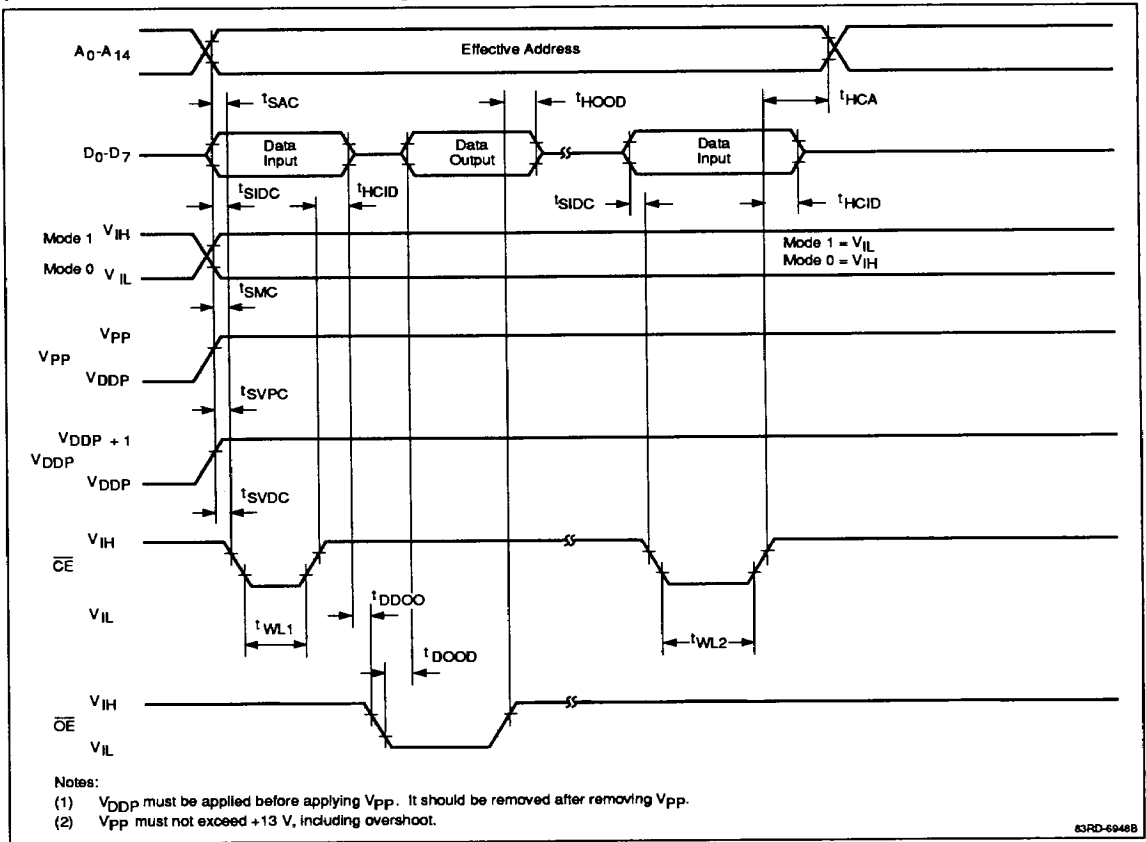
μPD78CP14 AC Programming Characteristics

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

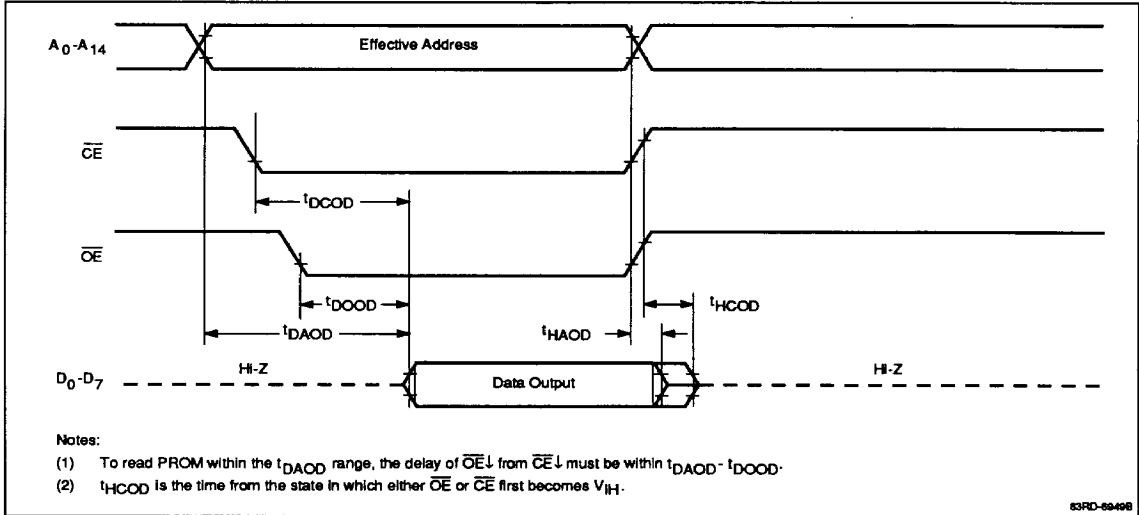
Parameter	Symbol	Symbol*	Min	Typ	Max	Unit	Condition
Address setup time to \overline{CE} ↓	t _{SAC}	t _{AS}	2			μs	
Data to \overline{OE} ↓ delay time	t _{DDO0}	t _{OES}	2			μs	
Input data setup time to \overline{CE} ↓	t _{SIDC}	t _{DS}	2			μs	
Address hold time from \overline{CE} ↑	t _{HCA}	t _{AH}	2			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



μPD78CP14 PROM Read Mode Timing



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, TMO, 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	ETMO, 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	ECNT = Timer/event counter upcounter
PB = Port B	ECPT = Timer/event counter capture
PC = Port C	ETMM = Timer/event counter mode
PD = Port D	
PF = Port F	
MA = Mode A	
MB = Mode B	
MC = Mode C	EOM = Timer/event counter output mode
MCC = Mode control C	
MF = Mode F	
	TXB = Transmit buffer
	RXB = Receive buffer
MM = Memory mapping	SMH = Serial mode high
TM0 = Timer register 0	SML = Serial mode low
TM1 = Timer register 1	MKH = Mask high
TMM = Timing mode	MKL = Mask low
ETM0 = Timer/event counter register 0	ANM = A/D channel mode
ETM1 = Timer/event counter register 1	CR0 to CR3 = A/D conversion result 0-3
ZCM = Zero-cross mode control register	

Register Pairs (rp-rp3)

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

Register Pair Addressing (rpa-rpa3)

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

Flags (f)

CY = Carry	HC = Half-carry	Z = Zero
------------	-----------------	----------

Interrupt Flags (irf)

INTFNMI = NMI interrupt flag	INTFEIN = FEIN
	INTFAD = FAD
INTFT0 = FT0	INTFSR = FSR
INTFT1 = FT1	INTFST = FST
INTF1 = F1	ER = Error
INTF2 = F2	OV = Overflow
INTFE0 = FE0	AN4 to AN7 = Analog input 4-7
INTFE1 = FE1	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																Skip Condition								
			B1	B2	B3	B4	7	6	5	4	3	2	1	0	7	6	5	4		3	2	1	0	State (Note 1)	Bytes		
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	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	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	0	0	0	0	0	1	1	0	1	0	1	R ₂	R ₁	R ₀							17	4
	word,r (word) ← (r)		0	1	1	0	0	0	0	0	1	1	1	1	1	R ₂	R ₁	R ₀							17	4	
MVI																											
	r1,byte (r) ← byte		0	1	1	0	1	R ₂	R ₁	R ₀														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	
MVIW																											
	*wa,byte ((V)←(wa)) ← byte		0	1	1	0	0	0	1														Offset	3			
MVIH																											
	*rpa1,byte (rpa1) ← byte		0	1	0	0	1	0	A ₁	A ₀														Data	2		
	*wa ((V)←(wa)) ← (A)		0	1	0	0	0	1	1														Offset	2			
	*wa (A) ← ((V)←(wa))		0	0	0	0	0	0	1														Offset	2			
	*rpa2 ((rpa2)) ← (A)		A ₃	0	1	1	1	A ₂	A ₁	A ₀														Data (Note 2)	2		
	*rpa2 (A) ← ((rpa2))		A ₃	0	1	0	1	A ₂	A ₁	A ₀														Data (Note 2)	2		
	(B) ↔ (B), (C) ↔ (C), (D) ↔ (D)		0	0	0	1	0	0	0														4	1			
	(E) ↔ (E), (H) ↔ (H), (L) ↔ (L)		0	0	0	1	0	0	0														4	1			
	(V) ↔ (V), (A) ↔ (A), (EA) ↔ (EA)		0	0	0	1	0	0	0														4	1			
	(H) ↔ (H), (L) ↔ (L)		0	1	0	1	0	0	0														4	1			
	((DE)) ← ((HL)), (DE) ← ((DE) + 1, HL) ← (HL) + 1, (C) ← (C) - 1		0	0	1	1	0	0	0														13 x (C + 1)	1			
End if borrow																											
16-Bit Data Transfer																											
DMOV	rp3,EA (rp3) ← (EAL), (rp3H) ← (EAH)		1	0	1	1	0	1	P ₁	P ₀														4	1		
	EA,rp3 (EAL) ← (rp3L), (EAH) ← (rp3H)		1	0	1	0	0	1	P ₁	P ₀														4	1		

Notes:

- 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
- B2 (Data): rpa2 = D+byte or H+byte.
- Right side of slash (/) in states indicates case rpa2 or rpa3 = D+byte, H+A, H+B, H+EA, or H+byte.
- B3 (Data): rpa3 = D+byte or H+byte.

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																State (Note 1)	Bytes	Skip Condition								
			B1				B2				B3				B4														
			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)																											
LBGD	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 (Registers)																													
ADD	r, A (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	r, A (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																State (Note 1)	Bytes	Skip Condition								
			B1				B3				B2											B4							
8-Bit Arithmetic (Register) (cont)			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0		
ADDNC	r_A, r (A) ← (A) + (r)		0	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	8									2	No carry
	r_A, r (r) ← (r) + (A)		0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	8									2	No carry
SUB	r_A, r (A) ← (A) - (r)		0	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	8									2	
	r_A, r (r) ← (r) - (A)		0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	8									2	
SBB	r_A, r (A) ← (A) - (r) - (CY)		0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	8									2	
	r_A, r (r) ← (r) - (A) - (CY)		0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	8									2	
SUBNB	r_A, r (A) ← (A) - (r)		0	1	1	0	0	0	0	0	1	0	1	1	0	0	0	0	8									2	No borrow
	r_A, r (r) ← (r) - (A)		0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	8									2	No borrow
ANA	r_A, r (A) ← (A) ∧ (r)		0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0	8									2	
	r_A, r (r) ← (r) ∧ (A)		0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	8									2	
ORA	r_A, r (A) ← (A) ∨ (r)		0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0	8									2	
	r_A, r (r) ← (r) ∨ (A)		0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	8									2	
XRA	r_A, r (A) ← (A) ⊘ (r)		0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	8									2	
	r_A, r (r) ← (r) ⊘ (A)		0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	8									2	No borrow
GTA	r_A, r (A) - (r) - 1		0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	0	8									2	No borrow
	r_A, r (r) - (A) - 1		0	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	8									2	No borrow
LTA	r_A, r (A) - (r)		0	1	1	0	0	0	0	0	1	0	1	1	1	0	1	0	8									2	Borrow
	r_A, r (r) - (A)		0	1	1	0	0	0	0	0	0	0	1	1	1	0	1	0	8									2	Borrow
NEA	r_A, r (A) - (r)		0	1	1	0	0	0	0	0	1	1	1	0	1	0	1	0	8									2	No zero
	r_A, r (r) - (A)		0	1	1	0	0	0	0	0	0	1	1	0	1	0	1	0	8									2	No zero
EQA	r_A, r (A) - (r)		0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	0	8									2	Zero
	r_A, r (r) - (A)		0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	8									2	Zero
ONA	r_A, r (A) ∧ (r)		0	1	1	0	0	0	0	0	1	1	0	0	1	1	0	0	8									2	No zero
OFFA	r_A, r (A) ∧ (r)		0	1	1	0	0	0	0	0	1	1	0	1	1	0	1	0	8									2	Zero
8-Bit Arithmetic (Memory)																													
ADDX	rpa, rpa (A) ← (A) + ((rpa))		0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	11									2	
ADDCX	rpa, rpa (A) ← (A) + ((rpa)) + (CY)		0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	0	11									2	
ADDNCX	rpa, rpa (A) ← (A) + ((rpa))		0	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	11									2	No carry
SUBX	rpa, rpa (A) ← (A) - ((rpa))		0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	11									2	
SBBX	rpa, rpa (A) ← (A) - ((rpa)) - (CY)		0	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	11									2	
SUBNBX	rpa, rpa (A) ← (A) - ((rpa))		0	1	1	1	0	0	0	0	1	0	1	1	0	0	0	0	11									2	No borrow
ANAX	rpa, rpa (A) ← (A) ∧ ((rpa))		0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	11									2	
ORAX	rpa, rpa (A) ← (A) ∨ ((rpa))		0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0	11									2	

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																State (Note 1)	Bytes	Skip Condition
			B1 7	B1 6	B1 5	B1 4	B1 3	B1 2	B1 1	B1 0	B2 7	B2 6	B2 5	B2 4	B2 3	B2 2	B2 1	B2 0			
8-Bit Arithmetic (Memory) (cont)																					
XRAX	rpa (A) ← (A) ∨ ((rpa))		0	1	1	0	0	0	0	1	0	0	1	0	0	1	0	A ₂ A ₁ A ₀	11	2	No borrow
GTAX	rpa (A) ← ((rpa)) - 1		0	1	1	0	0	0	0	1	0	1	0	1	0	1	A ₂ A ₁ A ₀	11	2	Borrow	
LJAX	rpa (A) ← ((rpa))		0	1	1	0	0	0	0	1	0	1	1	1	A ₂ A ₁ A ₀	11	2	No zero			
NEAX	rpa (A) ← ((rpa))		0	1	1	0	0	0	0	1	1	1	0	1	A ₂ A ₁ A ₀	11	2	Zero			
EOAX	rpa (A) ← ((rpa))		0	1	1	0	0	0	0	1	1	1	1	1	A ₂ A ₁ A ₀	11	2	No zero			
ONAX	rpa (A) ∧ ((rpa))		0	1	1	0	0	0	0	1	1	0	0	1	A ₂ A ₁ A ₀	11	2	Zero			
OFFAX	rpa (A) ∨ ((rpa))		0	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	0	1	0	0	Data							11	3			
	sr2.byte (sr2) ← (sr2) + byte		0	1	1	0	0	1	0	0	Data							20	3		
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	0	1	0	0	Data							11	3			
	sr2.byte (sr2) ← (sr2) + byte + (CY)		0	1	1	0	0	1	0	0	Data							20	3		
ADJNC	*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	0	1	0	0	Data							11	3	No carry		
	sr2.byte (sr2) ← (sr2) + byte		0	1	1	0	0	1	0	0	Data							20	3	No carry	
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	0	1	0	0	Data							11	3			
	sr2.byte (sr2) ← (sr2) - byte		0	1	1	0	0	1	0	0	Data							20	3		
SBI	*A.byte (A) ← (A) - byte - (CY)		0	1	1	0	1	1	0	Data							7	2			
	r.byte (r) ← (r) - byte - (CY)		0	1	1	0	1	0	0	Data							11	3			
	sr2.byte (sr2) ← (sr2) - byte - (CY)		0	1	1	0	0	1	0	0	Data							20	3		

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Skip Condition			
			B1				B2				B3				B4					State (Note 1)	Bytes	
Immediate Data (cont)			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
SUMB	*A,byte (A) ← (A) - byte r,byte (r) ← (r) - byte		0	0	1	1	0	1	1	0	Data	7								7	2	No borrow
	sr2,byte (sr2) ← (sr2) - byte		0	1	1	0	0	1	0	0	Data	0	0	1	1	0	0	0	0	R ₂ R ₁ R ₀	3	No borrow
			0	1	1	0	0	1	0	0	S ₃ 0 1 1 0 0 S ₂ S ₁ S ₀	20									3	No borrow
ANI	*A,byte (A) ← (A) ∧ byte r,byte (r) ← (r) ∧ byte		0	0	0	0	1	1	1	1	Data	7								7	2	
	sr2,byte (sr2) ← (sr2) ∧ byte		0	1	1	0	1	0	0	0	Data	0	0	0	0	1	0	0	0	R ₂ R ₁ R ₀	3	
			0	1	1	0	0	1	0	0	S ₃ 0 0 0 0 1 S ₂ S ₁ S ₀	20									3	
ORI	*A,byte (A) ← (A) ∨ byte r,byte (r) ← (r) ∨ byte		0	0	0	1	0	1	1	1	Data	7								7	2	
	sr2,byte (sr2) ← (sr2) ∨ byte		0	1	1	0	1	0	0	0	Data	0	0	0	1	1	0	0	0	R ₂ R ₁ R ₀	3	
			0	1	1	0	0	1	0	0	S ₃ 0 0 0 1 1 S ₂ S ₁ S ₀	20									3	
XRI	*A,byte (A) ← (A) ⊕ byte r,byte (r) ← (r) ⊕ byte		0	0	0	1	0	1	1	0	Data	7								7	2	
	sr2,byte (sr2) ← (sr2) ⊕ byte		0	1	1	0	1	0	0	0	Data	0	0	0	1	0	0	0	0	R ₂ R ₁ R ₀	3	
			0	1	1	0	0	1	0	0	S ₃ 0 0 1 1 0 S ₂ S ₁ S ₀	20									3	
GTI	*A,byte (A) ← byte - 1 r,byte (r) ← byte - 1		0	0	1	0	0	1	1	1	Data	7								7	2	No borrow
	sr2,byte (sr2) ← byte - 1		0	1	1	0	0	1	0	0	Data	0	0	1	0	1	0	0	0	R ₂ R ₁ R ₀	3	No borrow
			0	1	1	0	0	1	0	0	S ₃ 0 1 0 1 0 S ₂ S ₁ S ₀	14									3	No borrow
LTI	*A,byte (A) ← byte r,byte (r) ← byte		0	0	1	1	0	1	1	1	Data	7								7	2	Borrow
	sr2,byte (sr2) ← byte		0	1	1	0	1	0	0	0	Data	0	0	1	1	1	0	0	0	R ₂ R ₁ R ₀	3	Borrow
			0	1	1	0	0	1	0	0	S ₃ 0 1 1 1 1 S ₂ S ₁ S ₀	14									3	Borrow
NEI	*A,byte (A) ← byte r,byte (r) ← byte		0	1	1	0	0	1	1	1	Data	7								7	2	No zero
	sr2,byte (sr2) ← byte		0	1	1	0	1	0	0	0	Data	0	1	1	0	1	0	0	0	R ₂ R ₁ R ₀	3	No zero

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code														State (Note 1)	Bytes	Skip Condition				
			B1	B2	B3	B4	7	6	5	4	3	2	1	0	S ₃	S ₂				S ₁	S ₀		
Immediate Data (cont)																							
NEI	sr2.byte (sr2) - byte		0	1	1	0	0	1	0	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	0	1	1	1	1	Data										7	2	Zero
	r.byte (r) - byte		0	1	1	0	1	0	0	0	0	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	1	1	1	1	Data										7	2	No zero
	r.byte (r) ^ byte		0	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	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	0	1	0	0	1	1	0	0	0	0	0	0	0	0	14	3		
Offset																							
ADCW	wa (A) ← (A) + (V) * (wa) + (CY)		0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	14	3		
Offset																							
ADDNCW	wa (A) ← (A) + (V) * (wa)		0	1	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	14	3	No carry	
Offset																							
SUBW	wa (A) ← (A) - (V) * (wa)		0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	14	3		
Offset																							
SBBW	wa (A) ← (A) - (V) * (wa) - (CY)		0	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	14	3		
Offset																							
SUBNBW	wa (A) ← (A) - (V) * (wa)		0	1	1	0	1	0	0	1	0	1	1	0	0	0	0	0	0	14	3	No borrow	
Offset																							
ANAW	wa (A) ← (A) ^ (V) * (wa)		0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	14	3		
Offset																							

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																State (Note 1)	Bytes	Skip Condition
			B1	B3	B2				B4				B5								
Working Register (cont)	wa	(A) ← (A) V ((V) ← (wa))	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
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																			
E0AW	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	1	0	1										19	3	
		Data																			
ORIW	*wa.byte	((V) ← (wa)) ← ((V) ← (wa)) V byte	0	0	0	1	0	1	0	1									19	3	
		Data																			
GTIW	*wa.byte	((V) ← (wa)) - byte - 1	0	0	1	0	0	1	0	1									13	3	No borrow
		Data																			
LTIW	*wa.byte	((V) ← (wa)) - byte	0	0	1	1	0	1	0	1									13	3	Borrow
		Data																			
NEIW	*wa.byte	((V) ← (wa)) - byte	0	1	1	0	0	1	0	1									13	3	No zero
		Data																			
E0IW	*wa.byte	((V) ← (wa)) - byte	0	1	1	1	0	1	0	1									13	3	Zero
		Data																			
ONIW	*wa.byte	((V) ← (wa)) ∧ byte	0	1	0	0	0	1	0	1									13	3	No zero
		Data																			
OFFIW	*wa.byte	((V) ← (wa)) ∧ byte	0	1	0	1	0	1	0	1									13	3	Zero
		Data																			

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																State (Note 1)	Bytes	Skip Condition
			B1				B2				B3				B4						
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
16-Bit Arithmetic																					
EADD	EA,r2 (EA) ← (EA) + (r2)		0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	R1 R0	2	
DADD	EA,rp3 (EA) ← (EA) + (rp3)		0	1	1	1	0	1	0	0	1	1	0	0	0	1	1	0	P1 P0	2	
DADC	EA,rp3 (EA) ← (EA) + (rp3) + (CY)		0	1	1	1	0	1	0	0	1	1	0	1	0	1	1	0	P1 P0	2	No carry
DADDNC	EA,rp3 (EA) ← (EA) + (rp3)		0	1	1	1	0	1	0	0	1	0	1	0	0	1	1	0	P1 P0	2	
ESUB	EA,r2 (EA) ← (EA) - (r2)		0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	R1 R0	2	
DSUB	EA,rp3 (EA) ← (EA) - (rp3)		0	1	1	1	0	1	0	0	1	1	1	0	0	1	1	0	P1 P0	2	
DSBB	EA,rp3 (EA) ← (EA) - (rp3) - (CY)		0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	0	P1 P0	2	No borrow
DSUBNB	EA,rp3 (EA) ← (EA) - (rp3)		0	1	1	1	0	1	0	0	1	0	1	1	0	1	1	0	P1 P0	2	
DAN	EA,rp3 (EA) ← (EA) ∧ (rp3)		0	1	1	1	0	1	0	0	1	0	0	1	1	1	1	0	P1 P0	2	
DOR	EA,rp3 (EA) ← (EA) ∨ (rp3)		0	1	1	1	0	1	0	0	1	0	0	1	1	1	1	0	P1 P0	2	
DXR	EA,rp3 (EA) ← (EA) ∨ (rp3)		0	1	1	1	0	1	0	0	1	0	0	1	1	1	1	0	P1 P0	2	
DXR	EA,rp3 (EA) ← (EA) ∨ (rp3)		0	1	1	1	0	1	0	0	1	0	0	1	1	1	1	0	P1 P0	2	
DGT	EA,rp3 (EA) ← (rp3) - 1		0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	P1 P0	2	No zero
DLT	EA,rp3 (EA) ← (rp3)		0	1	1	1	0	1	0	0	1	0	1	0	1	1	1	0	P1 P0	2	Borrow
DNE	EA,rp3 (EA) ← (rp3)		0	1	1	1	0	1	0	0	1	0	1	0	1	1	1	0	P1 P0	2	Zero
DEO	EA,rp3 (EA) ← (rp3)		0	1	1	1	0	1	0	0	1	1	0	0	1	1	1	0	P1 P0	2	No zero
DON	EA,rp3 (EA) ∧ (rp3)		0	1	1	1	0	1	0	0	1	1	0	0	1	1	1	0	P1 P0	2	Zero
DOFF	EA,rp3 (EA) ∧ (rp3)		0	1	1	1	0	1	0	0	1	1	0	1	1	1	1	0	P1 P0	2	Zero
Multiply/Divide																					
MUL	r2 (EA) ← (A) × (r2)		0	1	0	0	1	0	0	0	0	0	1	0	1	1	1	0	R1 R0	32	2
DIV	r2 (EA) ← (EA) ÷ (r2), (r2) ← Remainder		0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	0	R1 R0	59	2
Increment/Decrement																					
INR	r2 (r2) ← (r2) + 1		0	1	0	0	0	0	0	0	R1 R0								4	1	Carry
INRW	*wa (V)*wa) ← (V)*wa) + 1		0	0	1	0	0	0	0	0									16	2	Carry
INX	rp (rp) ← (rp) + 1		0	0	P1	P0	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	0	0	R1 R0								4	1	Borrow
DCRW	*wa (V)*wa) ← (V)*wa) - 1		0	0	1	1	0	0	0	0									16	2	Borrow
DCX	rp (rp) ← (rp) - 1		0	0	P1	P0	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																Bytes	Skip Condition		
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
Others (cont)																						
NEGA		$(A) \leftarrow (\bar{A}) + 1$	0	1	0	0	1	0	0	0	0	0	1	1	0	1	0	8	2			
Rotate and Shift																						
RLD		Rotate left digit $(A_{3-0}) \leftarrow ((HL)_{7-4}, (HL)_{7-4}, (HL)_{7-4}) \leftarrow (A_{3-0}), ((HL)_{3-0}, (HL)_{3-0}) \leftarrow (A_{3-0})$	0	1	0	0	1	0	0	0	0	0	0	1	1	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	0	0	17	2				
RLL	r2	$(r2_{m+1}) \leftarrow (r2_m), (r2_0) \leftarrow (CY), (CY) \leftarrow (r2_7)$	0	1	0	0	1	0	0	0	0	0	1	1	0	1	8	2				
RRL	r2	$(r2_{m-1}) \leftarrow (r2_m), (r2_7) \leftarrow (CY), (CY) \leftarrow (r2_0)$	0	1	0	0	1	0	0	0	0	0	1	1	0	0	8	2				
SLL	r2	$(r2_{m+1}) \leftarrow (r2_m), (r2_0) \leftarrow 0, (CY) \leftarrow (r2_7)$	0	1	0	0	1	0	0	0	0	0	1	0	0	1	8	2				
SLR	r2	$(r2_{m-1}) \leftarrow (r2_m), (r2_7) \leftarrow 0, (CY) \leftarrow (r2_0)$	0	1	0	0	1	0	0	0	0	0	1	0	0	1	8	2				
SLLC	r2	$(r2_{m+1}) \leftarrow (r2_m), (r2_0) \leftarrow 0, (CY) \leftarrow (r2_7)$	0	1	0	0	1	0	0	0	0	0	0	0	1	8	2					
SLRC	r2	$(r2_{m-1}) \leftarrow (r2_m), (r2_7) \leftarrow 0, (CY) \leftarrow (r2_0)$	0	1	0	0	1	0	0	0	0	0	0	0	0	8	2					
DRLL	EA	$(EA_{n+1}) \leftarrow (EA_n), (EA_0) \leftarrow (CY), (CY) \leftarrow (EA_{15})$	0	1	0	0	1	0	0	0	1	0	1	1	0	1	8	2				
DRLR	EA	$(EA_{n-1}) \leftarrow (EA_n), (EA_{15}) \leftarrow (CY), (CY) \leftarrow (EA_0)$	0	1	0	0	1	0	0	0	1	0	1	1	0	0	8	2				
DSL	EA	$(EA_{n+1}) \leftarrow (EA_n), (EA_0) \leftarrow 0, (CY) \leftarrow (EA_{15})$	0	1	0	0	1	0	0	0	1	0	1	0	0	1	8	2				
DSL	EA	$(EA_{n-1}) \leftarrow (EA_n), (EA_{15}) \leftarrow 0, (CY) \leftarrow (EA_0)$	0	1	0	0	1	0	0	0	1	0	1	0	0	0	8	2				
Jump																						
JMP	*word	PC ← word	0	1	0	1	0	1	0	0	0	0	0	0	0	0	10	3				
High addr																						
JB		$(PC_H) \leftarrow (B), (PC_L) \leftarrow (C)$	0	0	1	0	0	0	0	1	0	0	0	0	0	1	4	1				
JR	word	PC ← PC + 1 + jdisp	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	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	0	0	0	0	0	0	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	0	1	0	1	17	2				

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Bytes	State (Note 1)	Skip Condition
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
Call (cont)																					
CALF	*word	(SP) - 1 ← ((PC) + 2) _H , (SP) - 2 ← ((PC) + 2) _L , (PC ₁₅₋₁₁) ← 0000 ₁ , (PC ₁₀₋₀) ← fa, (SP) ← (SP) - 2	0	1	1	1	1	←	←	←	←	←	←	←	←	←	←	13	2		
CALT	word	(SP) - 1 ← ((PC) + 1) _H , (SP) - 2 ← ((PC) + 1) _L , (PC ₁₅) ← (128 + 2ta), (PC _H) ← (129 + 2ta), (SP) ← (SP) - 2	1	0	0	←	←	←	←	←	←	←	←	←	←	←	16	1			
SOFTI		(SP) - 1 ← (PSW), (SP) - 2 ← ((PC) + 1) _H , (SP) - 3 ← ((PC) + 1) _L , (PC) ← 0600 _H , (SP) ← (SP) - 3	0	1	1	1	0	0	1	0							16	1			
Return																					
RET		(PC _L) ← ((SP), (PC _H)) ← ((SP) + 1), (SP) ← (SP) + 2	1	0	1	1	1	0	0	0							10	1			
RETS		(PC _L) ← ((SP), (PC _H)) ← ((SP) + 1), (SP) ← (SP) + 2, (PC) ← (PC) + n	1	0	1	1	1	0	0	1							10	1	Unconditional Skip		
RETI		(PC _L) ← ((SP), (PC _H)) ← ((SP) + 1), (PSW) ← ((SP) + 2), (SP) ← (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	←	←	←	←	←	←	10	2	Bit Test		
SK	f	Skip if f = 1	0	1	0	0	1	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	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	l ₄	l ₃	l ₂	l ₁	l ₀	8	2	irf = 1
SKWIT	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	l ₄	l ₃	l ₂	l ₁	l 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	0	1	0	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	