

6427525 N E C ELECTRONICS INC 98D 13388

Description

The μ PD78C10, μ PD78C11, and μ PD78C14 single-chip microcomputers integrate sophisticated on-chip peripheral functionality normally provided by external components. The devices' internal 16-bit ALU and data paths, combined with a powerful instruction set and addressing, make them appropriate in data processing as well as control applications. The devices integrate a 16-bit ALU, 4K-byte ROM, 256-byte RAM with an 8-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 μ PD78C11 is a 4K-byte mask ROM high-volume production device embedded with custom customer program. The μ PD78C14 is a 16K-byte mask ROM device. The μ PD78C10 is a ROM-less version for prototyping and small volume production.

Features

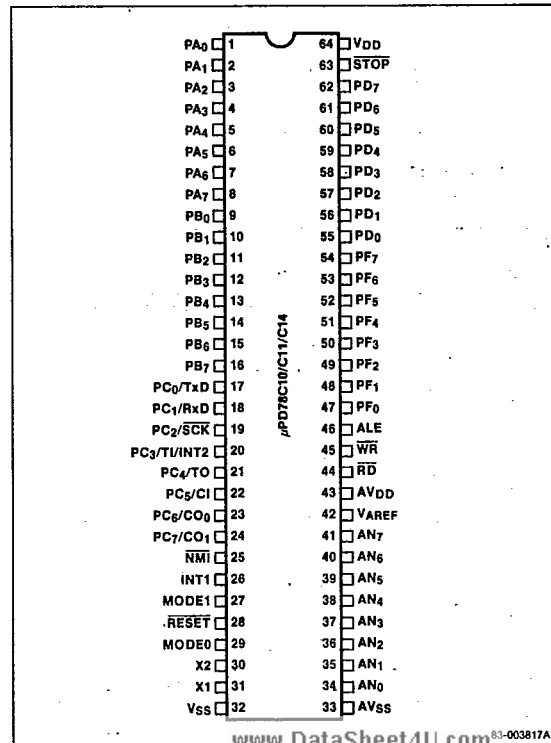
- CMOS technology
 - 2.5 to 6.0 V operating range
 - 30 mA operating current
- Complete single-chip microcomputer
 - 16-bit ALU
 - 4K x 8 ROM (78C11)
 - 16K x 8 ROM (78C14)
 - 256-byte RAM
- 44 I/O lines
- Two zero-cross detect inputs
- Two 8-bit timers
- Expansion capabilities
 - 8085A bus-compatible
 - 60K-byte external memory address range
- 8-channel, 8-bit A/D converter
 - Autoscan mode
 - Channel select mode
- Full duplex USART
 - Synchronous and asynchronous
- 154 instructions
 - 16-bit arithmetic, multiply and divide
 - HALT and STOP instructions
- 1 μ s instruction cycle time (12 MHz operation)
- Prioritized interrupt structure
 - 3 external
 - 8 internal
- Standby function
- On-chip clock generator
- 64-pin plastic QUIP, shrink DIP, or flatpack

Ordering Information

Part Number	Package Type	Max Frequency of Operation
μ PD78C10G-36	64-pin plastic QUIP	12 MHz
μ PD78C11G-36		
μ PD78C14G-36		
μ PD78C10CW	64-pin plastic shrink DIP	12 MHz
μ PD78C11CW		
μ PD78C14CW		
μ PD78C10G-1B	64-pin plastic miniflat	12 MHz
μ PD78C11G-1B		
μ PD78C14G-1B		
μ PD78C10L	68-pin PLCC (available 3Q86)	12 MHz
μ PD78C11L		
μ PD78C14L		

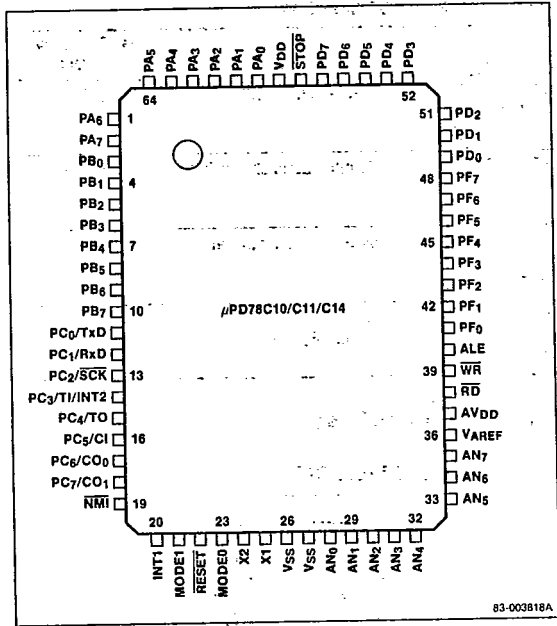
Pin Configurations

64-Pin QUIP or Shrink DIP



Pin Configurations (cont)

64-Pin Miniflat



83-003818A

Pin Identification (cont)

Symbol	Function
V _{SS}	Ground
AV _{SS}	A/D converter power supply ground
AN ₀ -AN ₇	A/D converter analog inputs 0-7
VAREF	A/D converter reference voltage
AV _{DD}	A/D converter power supply voltage
RD	Read strobe output
WR	Write strobe output
ALE	Address latch enable output
PF ₀ -PF ₇	Port F I/O/Expansion memory address bus (bits 8-15)
PD ₀ -PD ₇	Port D I/O/Expansion memory address/data bus
STOP	Stop mode control input
V _{DD}	5 V power supply

Pin Functions

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.

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.

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 and timer. Reset puts all lines of port C in port mode, input.

TxD [Transmit Data]. Serial data output terminal.

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.

TI [Timer Input]. Timer input terminal.

Pin Identification

Symbol	Function
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 /SCK	Port C I/O line 2/Serial clock I/O
PC ₃ /TI/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
NMI	Nonmaskable interrupt input
INT1	Interrupt request 1 input
MODE1	Mode 1 input/Memory cycle 1 output
RESET	Reset input
MODE0	Mode 0 input/I/O/Memory output
X1, X2	Crystal connections 1, 2

INT2 [Interrupt Request 2]. Falling-edge-triggered, maskable interrupt input terminal and AC-input, zero-cross detection terminal.

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

CI [Counter Input]. External pulse input to timer/event counter.

CO₀, CO₁ [Counter Outputs 0, 1]. Programmable rectangular wave outputs based on timer/event counter.

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 an input or output. When external expansion memory is used, port F outputs the high-order address bits.

AN₀-AN₇

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.

AV_{SS} [A/D Converter Power Ground]

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

NMI [Nonmaskable Interrupt]

Falling-edge-triggered nonmaskable interrupt input.

INT1 [Interrupt Request 1]

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

RESET [Reset]

When the RESET input is brought low, it initializes the device.

MODE1, MODE0 [Mode 1, 0]

The MODE1 and MODE0 inputs select the memory expansion mode. MODE1 also outputs the M1 signal during each opcode fetch. MODE0 outputs the $\overline{I/O}$ signal.

VAREF [A/D Converter Reference]

VAREF sets the upper limit for the A/D conversion range.

AV_{DD} [A/D Converter Power]

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

\overline{RD} [Read Strobe]

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

\overline{WR} [Write Strobe]

The \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. Three-state.

ALE [Address Latch Enable]

The ALE output latches the address signal to the output of PD₀-PD₇.

X1, X2 [Crystal Connections 1, 2]

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

V_{SS} [Ground]

Ground potential.

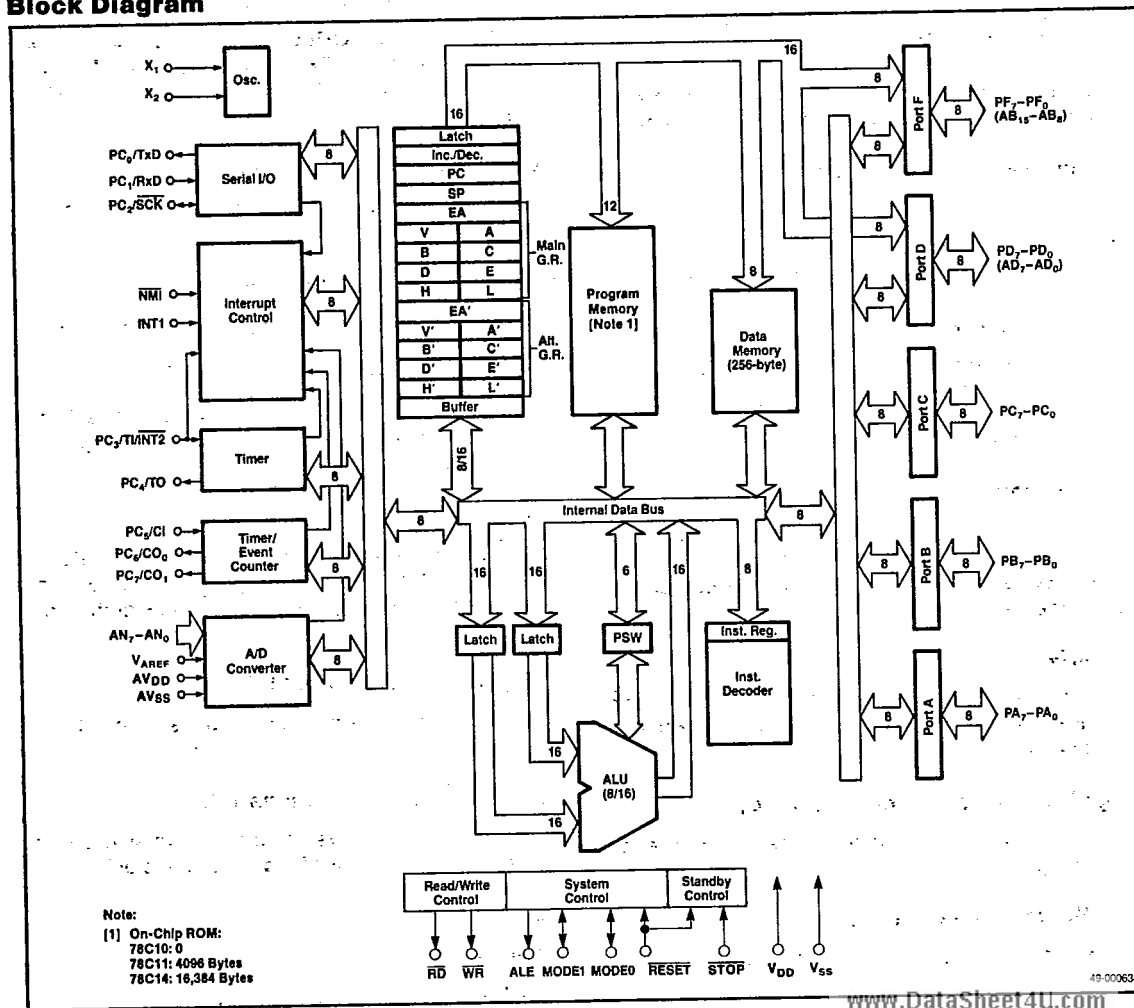
\overline{STOP} [Stop Mode Control Input]

A low-level input on \overline{STOP} stops the system clock oscillator.

V_{DD} [Power Supply]

+5 V power supply.

Block Diagram



Functional Description

Memory Map

The μ PD78C11 can directly address up to 64K bytes of memory. Except for the on-chip ROM (0-4095) and RAM (65,280-65,335), 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 μ PD78C11. On-chip ROM is located from 0-16,383 in the μ PD78C14.

Input/Output

The μ PD78C10/C11/C14 has 8 analog input lines (AN₀-AN₇), 44 digital I/O lines, five 8-bit ports (port A, port B, port C, port D, port F), and 4 input lines (AN₄-AN₇).

Analog Input Lines. AN₀-AN₇ are configured as analog input lines for on-chip A/D converter.

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.

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

AN₄-AN₇. The high-order analog input lines, AN₄-AN₇, can be used as digital input lines for falling-edge detection.

Control Lines. Under software control, each line of port C can be configured individually to provide control lines for the serial interface, timer, and timer/counter.

Memory Expansion. In addition to the single-chip operation mode, the μ PD78C11 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 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 ₀ -F ₃	Address bus
	Port F ₄ -F ₇	I/O port
16K Bytes	Port D	Multiplexed address/data bus
	Port F ₀ -F ₅	Address bus
	Port F ₆ -F ₇	I/O port
60K Bytes	Port D	Multiplexed address/data bus
	Port F	Address bus

Timers

There are two 8-bit timers. The timers may be programmed independently or may be cascaded and used as a 16-bit timer. The timer can be software set to increment at intervals of four machine cycles (1 μ s at 12 MHz operation) or 128 machine cycles (32 μ s at 12 MHz), or to increment on receipt of a pulse at TI. Figure 2 is the block diagram for the timer.

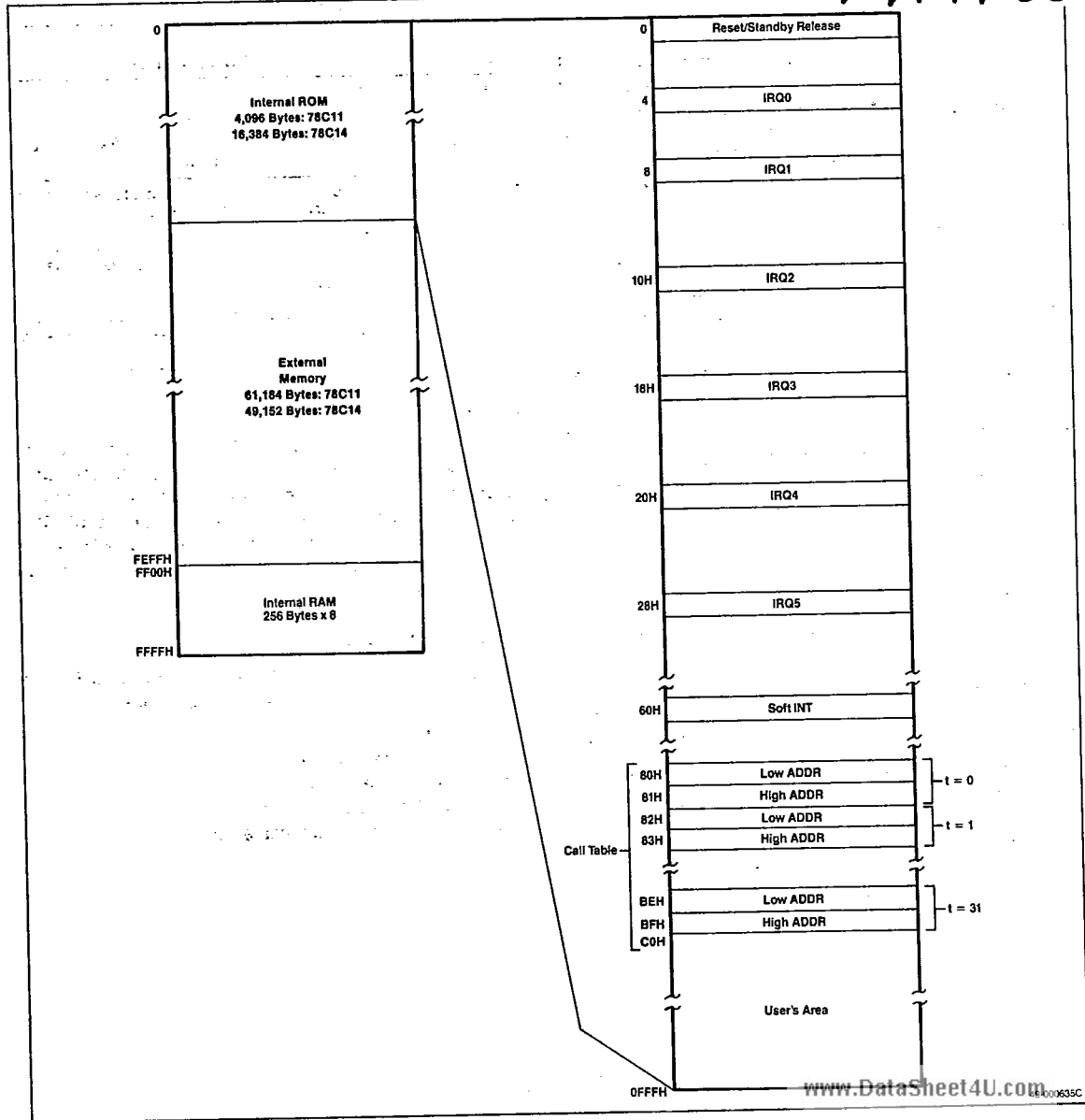
Timer/Event Counter

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 square-wave output

Figure 1. Memory Map

T-49-19-08



6427525 N E C ELECTRONICS INC

98D 13394

Figure 2. Timer Block Diagram

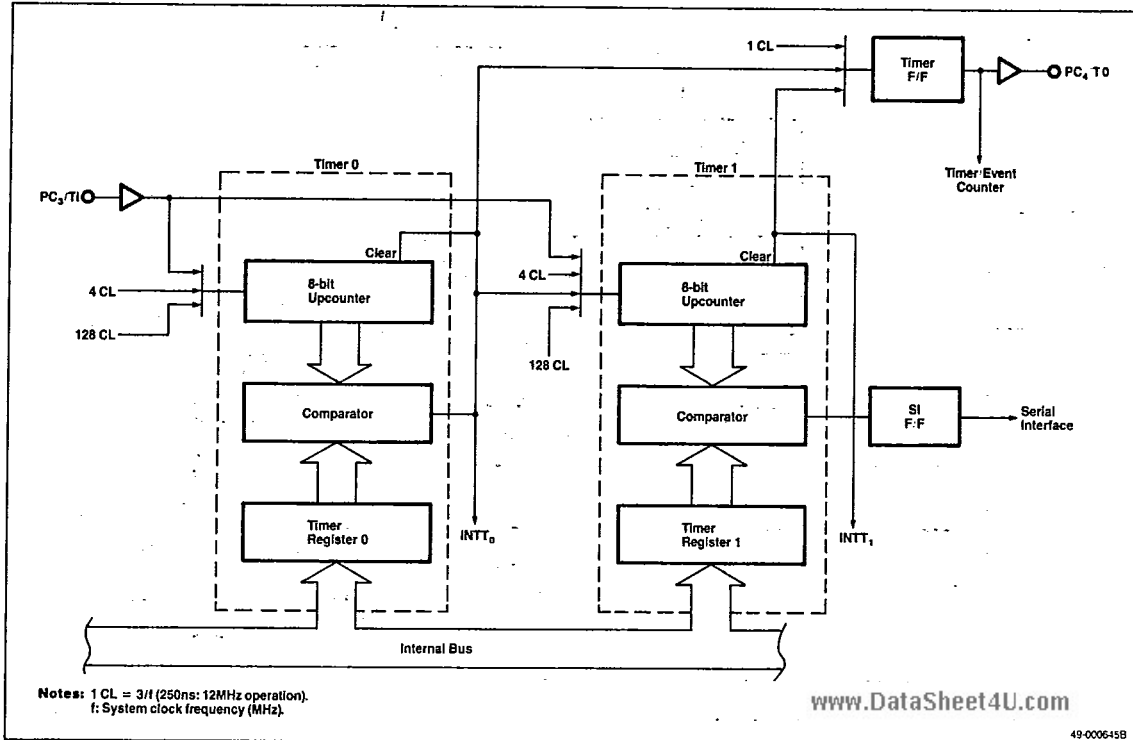
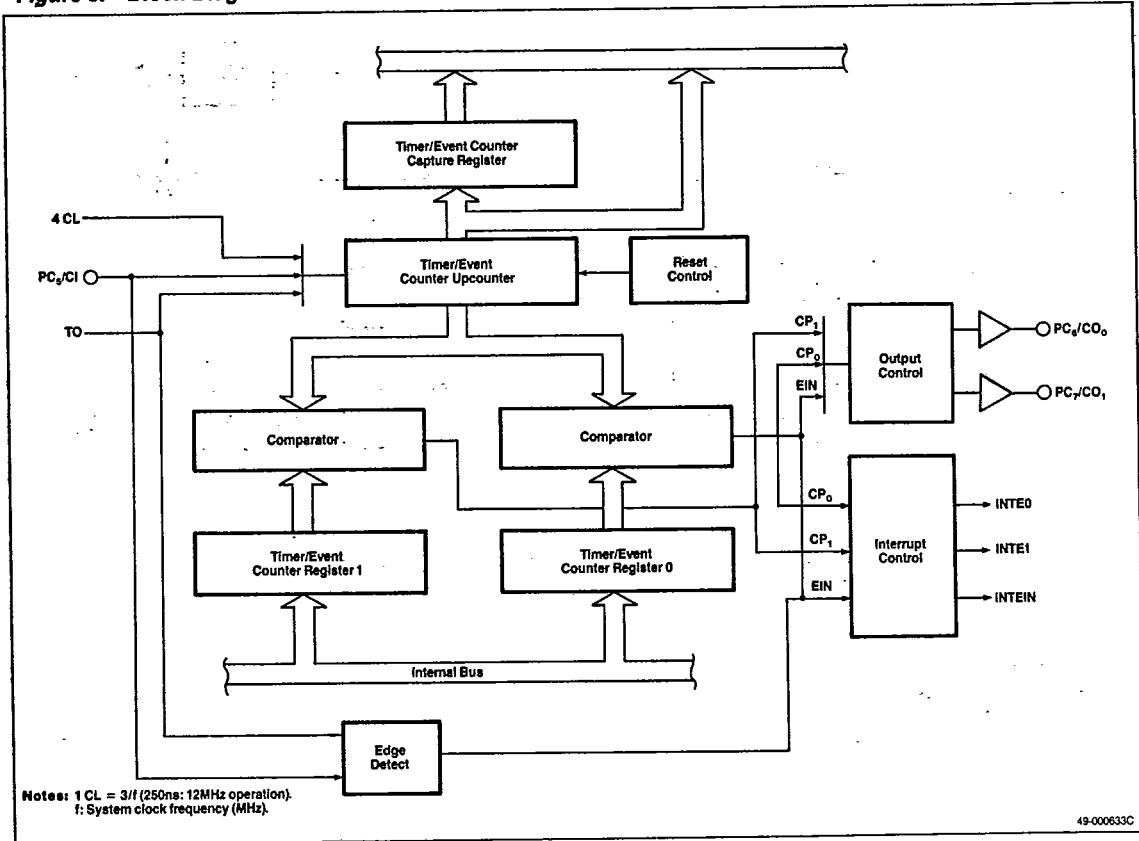


Figure 3. Block Diagram for Timer/Event Counter



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% $\pm 1/2$ LSB
- Conversion range: 0 to 5 V
- Conversion time: 42 μ s
- Interrupt generation

Analog/Digital Converter

The μ PD78C10/C11/C14 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 8-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, 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 shows the block diagram for the A/D converter. To prevent operation of the A/D converter and thus reduce power consumption, set V_{AREF} = 0 V.

Interrupt Structure

There are 11 interrupt sources. Three are external interrupts and eight are internal. Table 2 shows 11 interrupt sources divided into six priority levels. See figure 5.

Standby Function

The μPD78C10/C11/C14 has two standby modes: HALT and STOP. The HALT mode reduces power consumption to less than 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_{CC} is maintained within the operating range (2.5 to 6.0 V), on-board RAM and CPU register contents are saved. If V_{CC} is held above 2.0 V (but less than 2.5 V), only 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 via timer 1. By checking the standby flag (SB), the user can determine whether the processor has been in the standby mode.

Table 2. Interrupt Sources

Interrupt Request	Interrupt Address	Type of Interrupt	Internal/External
IRQ0	4	NMI (Nonmaskable interrupt)	Ext
IRQ1	8	INTT0 (Coincidence signal from timer 0)	Int
		INTT1 (Coincidence signal from timer 1)	
IRQ2	16	INT1 (Maskable interrupt)	Ext
		INT2 (Maskable interrupt)	
IRQ3	24	INTE0 (Coincidence signal from timer/event counter)	Int
		INTE1 (Coincidence signal from timer/event counter)	
IRQ4	32	INTEIN (Falling signal of CI and TO counter)	Int/Ext
		INTAD (A/D converter interrupt)	
IRQ5	40	INTSR (Serial receive interrupt)	Int
		INST (Serial send interrupt)	

Type B is initiated by inputting a low level on the STOP input. Only RAM contents are saved, not the CPU register contents. The oscillator is stopped. The STOP mode is released by raising STOP to a high level. The oscillator stabilization time is fixed at 65 ms; 65 ms after STOP is raised, instruction execution will begin at location 0. You can increase the stabilization time by holding RESET low for the required time period.

Figure 4. A/D Converter Block Diagram

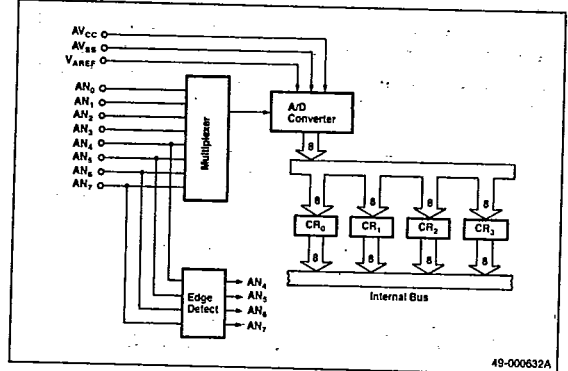
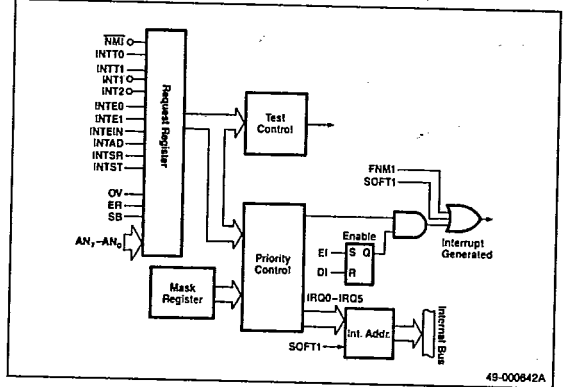


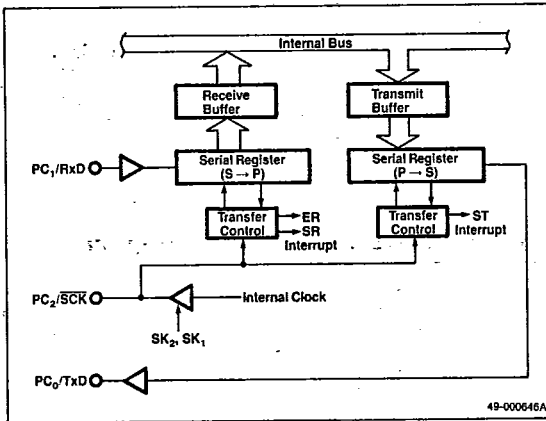
Figure 5. Interrupt Structure Block Diagram



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.

Figure 6. Universal Serial Interface Block Diagram



Zero-Crossing Detector

The INT1 and $\overline{\text{INT2}}$ terminals (used common to T1 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.

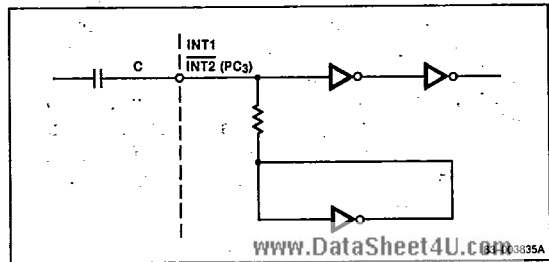
The zero-crossing detection capability allows you to make the 50-60 Hz power signal the basis for system timing and to control voltage phase-sensitive devices.

To use the zero-cross detection mode, an AC signal of approximately 1-3 V AC (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 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 $\overline{\text{INT2}}$ interrupt is generated.

Figure 7. Zero-Crossing Detection Circuit



6427525 N E C ELECTRONICS INC

98D 13398

Absolute Maximum Ratings

Power supply voltages, V_{DD}	-0.5 V to +7.0 V
AV_{DD}	AV_{SS} to $V_{DD} + 0.5$ V
AV_{SS}	-0.5 V to +0.5 V
Input voltage, V_I	-0.5 V to +7.0 V
Output voltage, V_O	-0.5 V to $V_{DD} + 0.5$ V
Output current low, I_{OL}	4.0 mA
Output current low, total for all pins	100 mA
Output current high, I_{OH}	-2.0 mA
Output current high, total for all pins	-50 mA
Reference input voltage, V_{AREF}	-0.5 V to $AV_{DD} + 0.3$ V
Operating temperature, T_{OPR} $f_{XTAL} \leq 12$ MHz	-40°C to +85°C
Storage temperature, T_{STG}	-65°C to +150°C

Comment: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Operating Conditions

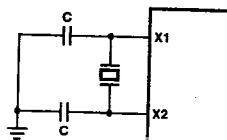
Oscillating Frequency	T_A	V_{DD}, AV_{DD}
$f_{XTAL} \leq 12$ MHz	-40°C to +85°C	+5.0 V \pm 10%

Capacitance

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

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	C_I			10	pF	$Af_c = 1$ MHz. Unmeasured pins returned to 0 V.
Output capacitance	C_O			20	pF	
I/O capacitance	$C_{I/O}$			20	pF	

Recommended XTAL Oscillation Circuit



C = 10 pF

www.DataSheet4U.com

63-003282A

DC Characteristics

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

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input low voltage	V _{IL1}	0		0.8	V	All except Note 1 inputs.
	V _{IL2}	0		0.2 V _{DD}	V	Note 1 inputs.
Input high voltage	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 low voltage	V _{OL}			0.45	V	I _{OL} = 2.0 mA
Output high voltage	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 _I			±200	μA	INT1, T1(PC3); 0 V ≤ V _I ≤ V _{DD}
Input leakage current	I _{LI}			±10	μA	All except INT, T1(PC3) 0 V ≤ V _I ≤ V _{DD}
Output leakage current	I _{LO}			±10	μA	0 V ≤ V _O ≤ V _{DD}
A _{VDD} supply current	A _{DD}		0.3	1.0	mA	
V _{DD} supply current	I _{DD1}		15	30	mA	Operation mode f = 12 MHz
	I _{DD2}		10	20	mA	HALT mode f = 12 MHz
Data retention current	I _{DDDR}		1	15	μA	V _{DDDR} = 2.5 V
			10	50	μA	V _{DDDR} = 5 V ± 10%

Note:

(1) Inputs RESET, STOP, NMI, SCK, INT1, T1, and AN₄-AN₇.

Serial Operation

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
SCK cycle time	t _{CYK}	1		μs	SCK input (1)
		500		ns	(2)
		2		μs	SCK output
SCK width low	t _{KKL}	420		ns	SCK input (1)
		200		ns	SCK input (2)
		900		ns	SCK output
SCK width high	t _{KKH}	420		ns	SCK input (1)
		200		ns	SCK input (2)
		900		ns	SCK output
RxD set-up time to SCK ↑	t _{RXK}	80		ns	(1)
RxD hold time after SCK ↑	t _{KRX}	80		ns	(1)
SCK ↓ TxD delay time	t _{KTX}		210	ns	(1)

Note:

- (1) 1x baud rate in asynchronous, synchronous, or I/O interface mode.
- (2) 16x baud rate or 64x baud rate in asynchronous mode.

Zero-Cross Characteristics

Parameter	Symbol	Limits		Unit	Test Conditions
		Min	Max		
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**Read/Write Operation**
 $T_A = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$; $V_{DD} = +5.0\text{ V} \pm 10\%$; $V_{SS} = 0\text{ V}$

98D 13400

T-49-19-08

Parameter	Symbol	Limits		Unit	Test Conditions (1)
		$f_{XTAL} = 12\text{ MHz}$			
		Min	Max		
RESET pulse width	t_{RP}	10		μs	
Interrupt pulse width (INT1, INT2)	t_{IP}	3.0		μs	
NMI pulse width		10		μs	
Counter input pulse width	t_{CI}	500		ns	Event counter mode
		4.0		μs	Pulse width measurement mode
Timer input pulse width	t_{TI}	500		ns	
X1 Input cycle time	t_{CYC}	83	250	ns	
Address set-up to ALE ↓	t_{AL}	65		ns	
Address hold after ALE ↓	t_{LA}	50		ns	
Address to $\overline{\text{RD}}$ ↓ delay time	t_{AR}	150		ns	
$\overline{\text{RD}}$ ↓ to address floating	t_{AFR}		20	ns	
Address to data input	t_{AD}		360	ns	
ALE ↓ to data input	t_{LDR}		215	ns	
$\overline{\text{RD}}$ ↓ to data input	t_{RD}		180	ns	
ALE ↓ to $\overline{\text{RD}}$ ↓ delay time	t_{LR}	35		ns	
Data hold time to $\overline{\text{RD}}$ ↑	t_{RDH}	0		ns	
$\overline{\text{RD}}$ ↑ to ALE ↑ delay time	t_{RL}	115		ns	
$\overline{\text{RD}}$ width low	t_{RR}	280		ns	Data read
		530		ns	Opcode fetch
ALE width high	t_{LL}	125		ns	
MT Setup time to ALE ↓	t_{ML}	65		ns	
MT Hold time after ALE ↓	t_{LM}	50		ns	
IO/M Setup time to ALE ↓	t_{LI}	65		ns	
IO/M Hold time after ALE ↓	t_{LI}	50		ns	
Address to $\overline{\text{WR}}$ ↓ Delay	t_{AW}	150		ns	
ALE ↓ to data output	t_{LDW}		195	ns	
$\overline{\text{WR}}$ ↓ to data output	t_{WD}		100	ns	
ALE ↓ to $\overline{\text{WR}}$ ↓ delay	t_{LW}	35		ns	
Data set-up time to $\overline{\text{WR}}$ ↑	t_{DW}	230		ns	
Data hold time to $\overline{\text{WR}}$ ↑	t_{WDH}	95		ns	
$\overline{\text{WR}}$ ↑ to ALE ↑ delay time	t_{WL}	115		ns	
$\overline{\text{WR}}$ width low	t_{WW}	280		ns	

Note:(1) Load capacitance: $C_L = 150\text{ pF}$.

A/D Converter Characteristics

$T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$; $V_{SS} = AV_{SS} = 0\text{ V}$; $AV_{DD} - 0.5\text{ V} \leq V_{AREF} \leq AV_{DD}$; $V_{DD} - 0.5 \leq AV_{DD}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Resolution		8			Bits	
Absolute accuracy				$0.4\% \pm 1/2$	LSB	$T_A = -10^\circ\text{C}$ to $+50^\circ\text{C}$
				$0.6\% \pm 1/2$	LSB	
Conversion time	t_{CONV}	567			t_{CYC}	$83\text{ ns} \leq t_{CYC} \leq 110\text{ ns}$
		432			t_{CYC}	$110\text{ ns} \leq t_{CYC} \leq 170\text{ ns}$
Sampling time	t_{SAMP}	96			t_{CYC}	$83\text{ ns} \leq t_{CYC} \leq 110\text{ ns}$
		72			t_{CYC}	$110\text{ ns} \leq t_{CYC} \leq 170\text{ ns}$
Analog input voltage	V_{IA}	0		V_{AREF}	V	
Analog input impedance	R_{AN}		1000		$M\Omega$	
V_{AREF} current	I_{AREF}		1.5	3.0	mA	

Bus Timing Depending on t_{CYC}

Symbol	Calculating Expression	Min/Max
t_{RP}	60T	Min
t_{TI}	6T	Min
t_{CI} (2)	6T	Min
t_{CI} (3)	48T	Min
t_{IP}	36T	Min
t_{AL}	2T - 100	Min
t_{LA}	T - 30	Min
t_{AR}	3T - 100	Min
t_{AD}	7T - 220	Max
t_{LDR}	5T - 200	Max
t_{RD}	4T - 150	Max
t_{LR}	T - 50	Min
t_{RL}	2T - 50	Min
t_{RR}	4T - 50 (Data Read)	Min
	7T - 50 (Opcode Fetch)	Min
t_{LL}	2T - 40	Min
t_{ML}	2T - 100	Min
t_{LM}	T - 30	Min
t_{IL}	2T - 100	Min
t_{LI}	T - 30	Min
t_{AW}	3T - 100	Min
t_{LDW}	T + 110	Max
t_{LW}	T - 50	Min
t_{DW}	4T - 100	Min
t_{WDH}	2T - 70	Min
t_{WL}	2T - 50	Min
t_{WW}	4T - 50	Min
t_{CYK}	12T (SCK input) (1)	Min
	24T (SCK output)	Min
t_{KKL}	5T + 5 (SCK input) (1)	Min
	12T - 100 (SCK output)	Min
t_{KKH}	5T + 5 (SCK input) (1)	Min
	12T - 100 (SCK output)	Min

Note:

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

$$T = t_{CYC} = 1/f_{XTAL}$$

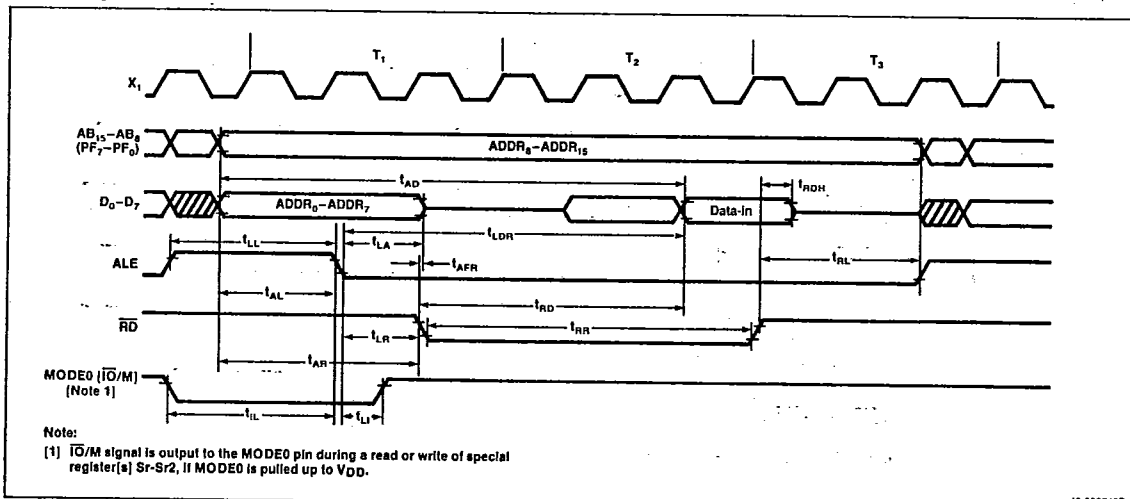
The items not included in this list are independent of oscillator frequency (f_{XTAL}).

(2) Event counter mode.

(3) Pulse width measurement mode.

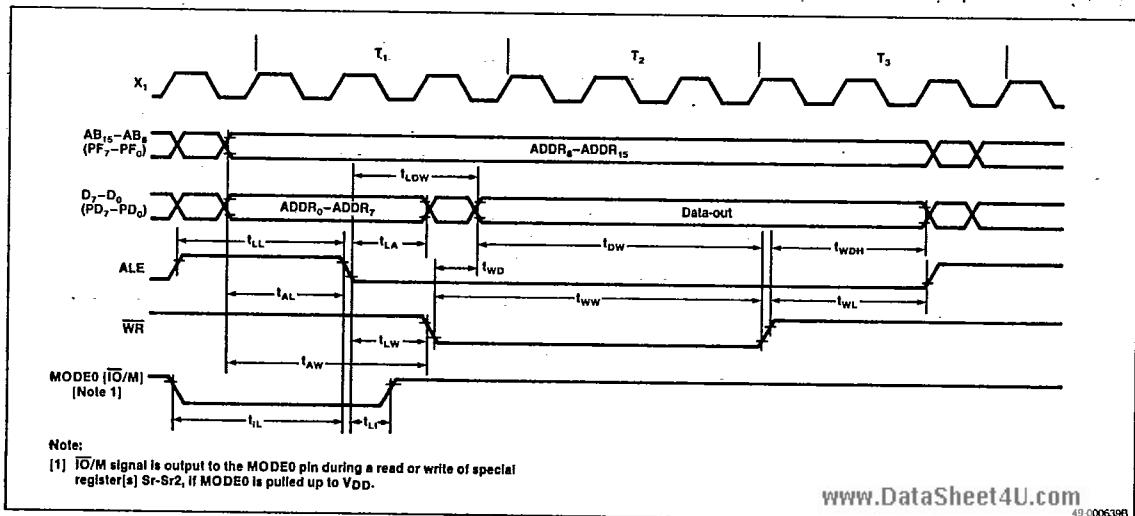
Timing Waveforms

Read Operation [U.com](http://www.DataSheet4U.com)



49-000543B

Write Operation



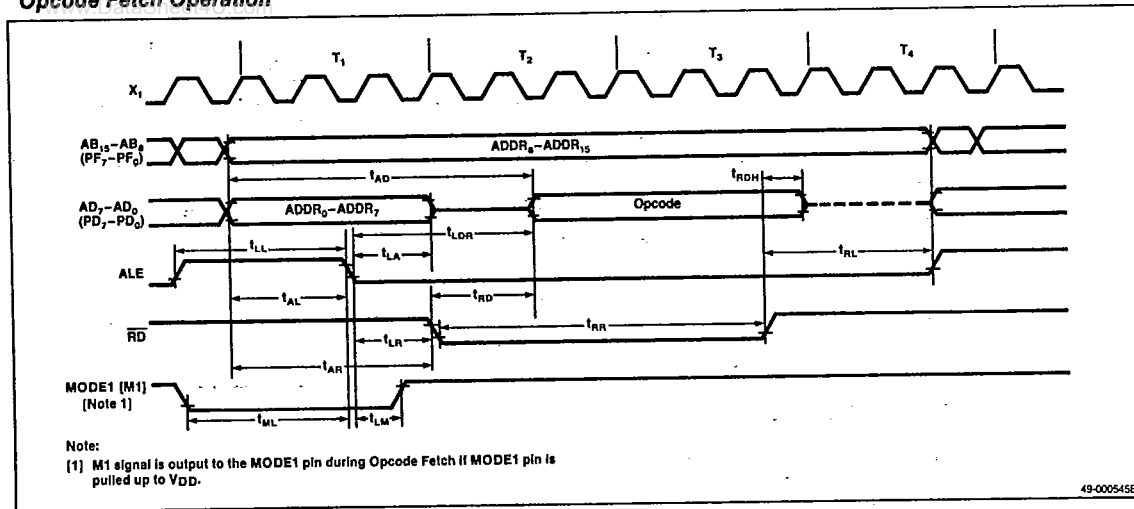
www.DataSheet4U.com

49-000539B

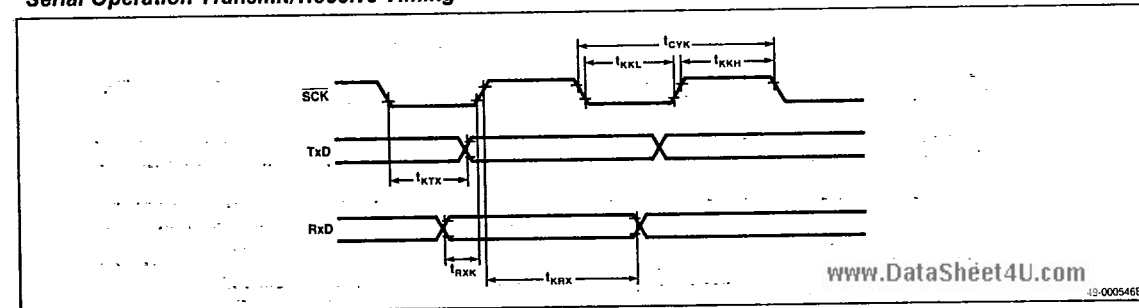
Timing Waveforms (cont)

T-49-19-08

Opcode Fetch Operation



Serial Operation Transmit/Receive Timing



T-49-19-0

Operand Format/Description

Format	Description
r	V, A, B, C, D, E, H, L
r1	EAH, EAL, B, C, D, E, H, L
r2	A, B, C
sr	PA, PB, PC, PD, PF, MKH, MKL, ANM, SMH, SML, EOM, ETMM, TMM, MM, MCC, MA, MB, MC, MF, TxB, TM ₀ , TM ₁ , 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	ETM ₀ , ETM ₁
sr4	ECNT, ECPT
rp	SP, B, D, H
rp1	V, B, D, H, EA
rp2	SP, B, D, H, EA
rp3	B, D, H
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
wa	8-Bit immediate data
word	16-Bit immediate data
byte	8-Bit immediate data
bit	3-Bit immediate data
f	CY, HC, Z
lrf	FNMI, FT0, FT1, F1, F2, FE0, FE1, FEIN, FAD, FSR, FST, ER, OV, AN ₄ , AN ₅ , AN ₆ , AN ₇ , SB

Instruction Set Symbol Definitions

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

Remarks

1. sr-sr4 (special register)	
PA = Port A	ECNT = Timer/Event Counter Upcounter
PB = Port B	PC = Port C
PC = Port C	ECPT = Timer/Event Counter Capture
PD = Port D	PF = Port F
MA = Mode A	ETMM = Timer/Event Counter Mode
MB = Mode B	EOM = Timer/Event Counter Output Mode
MC = Mode C	
MCC = Mode Control C	
MF = Mode F	
MM = Memory Mapping	TxB = Tx Buffer
TM ₀ = Timer Register 0	RxB = Rx Buffer
TM ₁ = Timer Register 1	SMH = Serial Mode High
TMM = Timer Mode	SML = Serial Mode Low
ETM ₀ = Timer/Event Counter Register 0	MKH = Mask High
ETM ₁ = Timer/Event Counter Register 1	MKL = Mask Low
ZCM = Zero-Cross Mode Control Register	ANM = A/D Channel Mode
	CR ₀ = A/D Conversion Result 0-3 to CR ₃
	TxB = Tx Buffer
	RxB = Rx Buffer
	SMH = Serial Mode High
	SML = Serial Mode Low
	MKH = Mask High High
	MKL = Mask Low

2. rp-rp3 (register pair)	
SP = Stack Pointer	H = HL
B = BC	V = VA
D = DE	EA = Extended Accumulator

3. rpa-rpa3 (rp addressing)	
B = (BC)	D++ = (DE)++
D = (DE)	H++ = (HL)++
H = (HL)	D+ byte = (DE) + byte
D+ = (DE) +	H+A = (HL) + (A)
H- = (HL) +	H+B = (HL) + (B)
D- = (DE) -	H+EA = (HL) + (EA)
H- = (HL) -	H+ byte = (HL) + byte

4. f (flag)	
CY = Carry	HC = Half Carry
	Z = Zero

5. lrf (interrupt flag)	
FNMI = NMI* Input	FEIN = INTFEIN
	FAD = INTFAD
FT0 = INTFT0	FSR = INTFSR
FT1 = INTFT1	FST = INTFST
F1 = INTF1	ER = Error
F2 = INTF2	OV = Overflow
FE0 = INTFE0	AN ₄ to AN ₇ = Analog Input 4-7
FE1 = INTFE1	SB = Standby

Instruction Set

Operation Code

Operation Code										B1		B2		B3		B4		Skip Condition		
Mnemonic	Operand	Operation	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1		0	State(1)

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	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	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

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

*r1,byte (r) ← byte			0	1	1	0	1	R ₂	R ₁	R ₀									7	2
set L1 if r = A																				
set L0 if r = L																				

L1 = 1 and r = A
L0 = 1 and r = L

sr2,byte (sr2) ← byte			0	1	1	0	0	1	0	0	0	S ₃	0	0	0	S ₂	S ₁	S ₀	14	3
-----------------------	--	--	---	---	---	---	---	---	---	---	---	----------------	---	---	---	----------------	----------------	----------------	----	---

Data

*wa,byte (V) ← (wa) ← byte			0	1	1	1	0	0	0	1									13	3
----------------------------	--	--	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	----	---

Offset

MVW	*rpa1,byte (rpa1) ← byte		0	1	0	0	1	0	A ₁	A ₀									10	2
-----	--------------------------	--	---	---	---	---	---	---	----------------	----------------	--	--	--	--	--	--	--	--	----	---

Data

STAW	*wa (V) ← (wa) ← A		0	1	1	0	0	0	1	1									10	2
LDAW	*wa (A) ← (V) ← (wa)		0	0	0	0	0	0	0	1									10	2

Offset

STAX	*rpa2 (rpa2) ← (A)		A ₃	0	1	1	1	A ₂	A ₁	A ₀									7/13(3)	2
------	--------------------	--	----------------	---	---	---	---	----------------	----------------	----------------	--	--	--	--	--	--	--	--	---------	---

Data (2)

LDAX	*rpa2 (A) ← ((rpa2))		A ₃	0	1	0	1	A ₂	A ₁	A ₀									7/13(3)	2
------	----------------------	--	----------------	---	---	---	---	----------------	----------------	----------------	--	--	--	--	--	--	--	--	---------	---

Data (2)

EXX	(B) ↔ (B'), (C) ↔ (C'), (D) ↔ (D')		0	0	0	1	0	0	0	1									4	1
	(E) ↔ (E'), (H) ↔ (H'), (L) ↔ (L')																			

Data

EXAL	(V) ↔ (V'), (A) ↔ (A'), (EA) ↔ (EA')		0	0	0	1	0	0	0	0									4	1
------	--------------------------------------	--	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	---	---

Data

EXHL	(H) ↔ (H'), (L) ↔ (L')		0	1	0	1	0	0	0	0									4	1
------	------------------------	--	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	---	---

Data

16-bit Data Transfer																				
BLOCK	D (DE) ↔ (HL), (DE) ↔ (DE + 1), (HL) ↔ (HL) + 1, (C) ↔ (C) - 1		0	0	1	1	0	0	0	1									13 x (C + 1)	1

Data

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

7-49

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Bytes	State(1)	Ship Condition			
			B1	B2	B3	B4	7	6	5	4	3	2	1	0	7	6	5	4				3	2	1
16-Bit Data Transfer (cont)																								
DMOV	sr3, EA (sr3) ← (EA) EA, sr4 (EA) ← (sr4)		0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	U ₀	14	2	
SBCD	word (word) ← (C), (word + 1) ← (B)		0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	V ₀	14	2	
SDED	word (word) ← (E), (word + 1) ← (D)		Low addr																4					
	word (word) ← (L), (word + 1) ← (H)		High addr																4					
SSPD	word (word) ← (SP _L), (word + 1) ← (SP _H)		Low addr																4					
	rpa3 (rpa3) ← (EAL), ((rpa3) + 1) ← (EAH)		High addr																4					
LBOD	word (C) ← (word), (B) ← (word + 1)		Data(4)																3					
	word (E) ← (word), (D) ← (word + 1)		Low addr																4					
LHLD	word (L) ← (word), (H) ← (word + 1)		Low addr																4					
	word (SP _L) ← (word), (SP _H) ← ((word) + 1)		High addr																4					
LSPD	rpa3 (EAL) ← ((rpa3), (EAH)) ← ((rpa3) + 1)		Low addr																4					
	rp1 ((SP) - 1) ← (rp1 _H) ((SP) - 2) ← (rp1 _L) (SP) ← (SP) - 2		High addr																3					
POP	rp1 (rp1 _L) ← (SP), (rp1 _H) ← ((SP) + 1) (SP) ← (SP) + 2		Data(4)																3					
	*rp2, word (rp2) ← (word) set LO if rp2 = H		0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	C ₃	14/20(3)	3	
LXI	rp1 ((SP) - 1) ← (rp1 _H) ((SP) - 2) ← (rp1 _L) (SP) ← (SP) - 2		Low byte																10					
	rp1 (rp1 _L) ← (SP), (rp1 _H) ← ((SP) + 1) (SP) ← (SP) + 2		High byte																10					
TABLE 1 (C) ← ((PC)+3+(A)), B ← ((PC)+3+(A)+1)																		17	2					
8-Bit Arithmetic Register																								
ADD	r, A (r) ← (A) + (r)		0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	R ₂	R ₀	8	2	
ADC	r, A (r) ← (A) + (r) + (CY)		0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	R ₂	R ₁	R ₀	8	2
	r, A (r) ← (r) + (A) + (CY)		0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0	0	R ₂	R ₁	R ₀	8	2

Instruction Set (cont)

Operation Code

Mnemonic	Operand	Operation																Bytes	State(1)	Skip Condition	
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
		B1								B2											
		B3	B3	B3	B3	B3	B3	B3	B3	B4	B4	B4	B4	B4	B4	B4	B4				
8-Bit Arithmetic Register [cont]																					
ADDNC	$A_i, (A) \leftarrow (A) + (r)$	0	1	1	0	0	0	0	0	1	0	1	0	0	R ₂	R ₁	R ₀	8	2	No carry	
	$r, A (r) \leftarrow (r) + (A)$	0	1	1	0	0	0	0	0	0	0	1	0	0	R ₂	R ₁	R ₀	8	2	No carry	
SUB	$A_i, (A) \leftarrow (A) - (r)$	0	1	1	0	0	0	0	0	1	1	0	0	0	R ₂	R ₁	R ₀	8	2		
	$r, A (r) \leftarrow (r) - (A)$	0	1	1	0	0	0	0	0	1	1	0	0	0	R ₂	R ₁	R ₀	8	2		
SBB	$A_i, (A) \leftarrow (A) - (r) - (CY)$	0	1	1	0	0	0	0	0	1	1	1	0	0	R ₂	R ₁	R ₀	8	2		
	$r, A (r) \leftarrow (r) - (A) - (CY)$	0	1	1	0	0	0	0	0	1	1	1	0	0	R ₂	R ₁	R ₀	8	2		
SUBNB	$A_i, (A) \leftarrow (A) - (r)$	0	1	1	0	0	0	0	0	1	0	1	1	0	R ₂	R ₁	R ₀	8	2	No borrow	
	$r, A (r) \leftarrow (r) - (A)$	0	1	1	0	0	0	0	0	1	0	1	1	0	R ₂	R ₁	R ₀	8	2	No borrow	
ANA	$A_i, (A) \leftarrow (A) \wedge (r)$	0	1	1	0	0	0	0	0	1	0	0	1	0	R ₂	R ₁	R ₀	8	2		
	$r, A (r) \leftarrow (r) \wedge (A)$	0	1	1	0	0	0	0	0	1	0	0	1	0	R ₂	R ₁	R ₀	8	2		
ORA	$A_i, (A) \leftarrow (A) \vee (r)$	0	1	1	0	0	0	0	0	1	0	0	1	0	R ₂	R ₁	R ₀	8	2		
	$r, A (r) \leftarrow (r) \vee (A)$	0	1	1	0	0	0	0	0	1	0	0	1	0	R ₂	R ₁	R ₀	8	2		
XRA	$A_i, (A) \leftarrow (A) \vee (r)$	0	1	1	0	0	0	0	0	1	0	0	1	0	R ₂	R ₁	R ₀	8	2	No borrow	
	$r, A (r) \leftarrow (r) \vee (A)$	0	1	1	0	0	0	0	0	1	0	0	1	0	R ₂	R ₁	R ₀	8	2	No borrow	
GTA	$A_i, (A) - (r) - 1$	0	1	1	0	0	0	0	0	1	0	1	0	1	R ₂	R ₁	R ₀	8	2	Borrow	
	$r, A (r) - (r) - 1$	0	1	1	0	0	0	0	0	1	0	1	0	1	R ₂	R ₁	R ₀	8	2	Borrow	
LTA	$A_i, (A) - (r)$	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	$A_i, (A) - (r)$	0	1	1	0	0	0	0	0	1	1	0	1	0	R ₂	R ₁	R ₀	8	2	No zero	
	$r, A (r) - (A)$	0	1	1	0	0	0	0	0	1	1	0	1	0	R ₂	R ₁	R ₀	8	2	No zero	
EOA	$A_i, (A) - (r)$	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	1	1	1	1	1	R ₂	R ₁	R ₀	8	2	Zero	
ONA	$A_i, (A) \wedge (r)$	0	1	1	0	0	0	0	0	1	1	0	0	1	R ₂	R ₁	R ₀	8	2	No zero	
	$r, A (r) \wedge (A)$	0	1	1	0	0	0	0	0	1	1	0	0	1	R ₂	R ₁	R ₀	8	2	Zero	
8-Bit Arithmetic (Memory)																					
ADIX	$r, A (A) \leftarrow (A) + ((r, A))$	0	1	1	1	0	0	0	0	1	1	0	0	0	A ₂	A ₁	A ₀	11	2		
ADCX	$r, A (A) \leftarrow (A) + ((r, A)) + (CY)$	0	1	1	1	0	0	0	0	1	1	0	1	0	A ₂	A ₁	A ₀	11	2		
ADIBX	$r, A (A) \leftarrow (A) + ((r, A))$	0	1	1	1	0	0	0	0	1	0	1	0	0	A ₂	A ₁	A ₀	11	2	No carry	
SUBX	$r, A (A) \leftarrow (A) - ((r, A))$	0	1	1	1	0	0	0	0	1	1	0	0	0	A ₂	A ₁	A ₀	11	2		
SBBX	$r, A (A) \leftarrow (A) - ((r, A)) - (CY)$	0	1	1	1	0	0	0	0	1	1	1	0	0	A ₂	A ₁	A ₀	11	2		
SUBNBX	$r, A (A) \leftarrow (A) - ((r, A))$	0	1	1	1	0	0	0	0	1	0	1	0	0	A ₂	A ₁	A ₀	11	2	No borrow	
ANAX	$r, A (A) \leftarrow (A) \wedge ((r, A))$	0	1	1	1	0	0	0	0	1	0	0	0	1	A ₂	A ₁	A ₀	11	2		
ORAX	$r, A (A) \leftarrow (A) \vee ((r, A))$	0	1	1	1	0	0	0	0	1	0	0	1	0	A ₂	A ₁	A ₀	11	2		

Instruction Set (cont)

Operation Code													SS/Op Condition						
Mnemonic	Operand	Operation											Bytes						
		7	6	5	4	3	2	1	0	7	6	5			4	3	2	1	0
B-Bit Arithmetic (Memory) (cont)																			
		B1	B2	B3	B4														
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
XRAX	rpa (A) ← (A) V ((rpa))	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	1	2	No borrow
GTAX	rpa (A) ← ((rpa)) - 1	0	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	2	Borrow
LTAX	rpa (A) ← ((rpa))	0	1	1	1	0	0	0	0	1	0	1	1	1	1	1	0	2	No zero
NEAX	rpa (A) ← ((rpa))	0	1	1	1	0	0	0	0	1	1	1	0	1	1	0	1	2	Zero
EOAX	rpa (A) ← ((rpa))	0	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	2	No zero
ONAX	rpa (A) ∧ ((rpa))	0	1	1	1	0	0	0	0	1	1	0	1	1	0	1	0	2	Zero
OFFAX	rpa (A) ∨ ((rpa))	0	1	1	1	0	0	0	0	1	1	0	1	1	0	1	1	2	Zero
Immediate Data																			
ADI	*A,byte (A) ← (A) + byte	Data																7	2
	r,byte (r) ← (r) + byte	Data																11	3
	sr2,byte (sr2) ← (sr2) + byte	Data																20	3
ACI	*A,byte (A) ← (A) + byte + (CY)	Data																7	2
	r,byte (r) ← (r) + byte + (CY)	Data																11	3
	sr2,byte (sr2) ← (sr2) + byte + (CY)	Data																20	3
ADINC	*A,byte (A) ← (A) + byte	Data																7	2
	r,byte (r) ← (r) + byte	Data																11	3
	sr2,byte (sr2) ← (sr2) + byte	Data																20	3
SUI	*A,byte (A) ← (A) - byte	Data																7	2
	r,byte (r) ← (r) - byte	Data																11	3
	sr2,byte (sr2) ← (sr2) - byte	Data																20	3
SBI	*A,byte (A) ← (A) - byte - (CY)	Data																7	2
	r,byte (r) ← (r) - byte - (CY)	Data																11	3
	sr2,byte (sr2) ← (sr2) - byte - (CY)	Data																20	3

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Bytes	State(I)	Skip Condition		
			B1	B2	B3	B4	7	6	5	4	3	2	1	0	7	6	5	4				3	2
Immediate Data (cont)																							
SUIB	*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	Data										11	3	No borrow
	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		
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	Data										11	3	
	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			
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	Data										11	3	
	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			
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	Data										11	3	
	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			
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	Data										11	3	No borrow
	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		
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	Data										11	3	Borrow
	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		
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	Data										11	3	No zero

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Bytes	Skip Condition	
			B1	B2	B3	B4	7 6 5 4 3 2 1 0				S3 1 1 0 1 S2 S1 S0				Start[]						
NEI	Immediate Data (cont)		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	14	3	No zero
	sr2.byte (sr2) - byte		0	1	1	0	0	1	0	0	S3	1	1	0	1	S2	S1	S0	14	3	No zero
	Data																				
EOR	*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	R2	R1	R0	11	3	Zero
	Data																				
	sr2.byte (sr2) - byte		0	1	1	0	0	1	0	0	S3	1	1	1	1	S2	S1	S0	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	R2	R1	R0	11	3	No zero
	Data																				
	sr2.byte (sr2) ^ byte		0	1	1	0	0	1	0	0	S3	1	0	0	1	S2	S1	S0	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	R2	R1	R0	11	3	Zero
	Data																				
	sr2.byte (sr2) ^ byte		0	1	1	0	0	1	0	0	S3	1	0	1	1	S2	S1	S0	14	3	Zero
	Data																				
Working Register																					
ADDW	wa	$(A) \leftarrow (A) + ((V) \cdot (wa))$	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	14	3	
	Offset																				
ADDCW	wa	$(A) \leftarrow (A) + ((V) \cdot (wa)) + (CY)$	0	1	1	1	0	1	0	0	1	1	0	1	0	0	0	0	14	3	
	Offset																				
ADDCW	wa	$(A) \leftarrow (A) + ((V) \cdot (wa))$	0	1	1	1	0	1	0	0	1	0	1	0	0	0	0	0	14	3	No carry
	Offset																				
SUBW	wa	$(A) \leftarrow (A) - ((V) \cdot (wa))$	0	1	1	1	0	1	0	0	1	1	1	0	0	0	0	0	14	3	
	Offset																				
SEB#4	wa	$(A) \leftarrow (A) - ((V) \cdot (wa)) - (CY)$	0	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	14	3	
	Offset																				
SUBW	wa	$(A) \leftarrow (A) - ((V) \cdot (wa))$	0	1	1	1	0	1	0	0	1	0	1	1	0	0	0	0	14	3	No borrow
	Offset																				
ANAW	wa	$(A) \leftarrow (A) \wedge ((V) \cdot (wa))$	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										Bytes	Skip Condition							
			B1	B2	B3	B4	7	6	5	4	3	2			1	0					
Working Register (cont)			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	State(1)		
ORAW	wa	$(A) \leftarrow (A) \vee (V) \bullet (wa)$	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	14	3	No borrow
		Offset																			
XRAW	wa	$(A) \leftarrow (A) \vee (V) \bullet (wa)$	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	14	3	No borrow
		Offset																			
GTAW	wa	$(A) \leftarrow (V) \bullet (wa) - 1$	0	1	1	0	1	0	0	1	0	1	0	1	0	0	0	0	14	3	No borrow
		Offset																			
LTAW	wa	$(A) \leftarrow (V) \bullet (wa)$	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	0	14	3	Borrow
		Offset																			
NEAW	wa	$(A) \leftarrow (V) \bullet (wa)$	0	1	1	0	1	0	0	1	1	1	0	1	0	0	0	0	14	3	No zero
		Offset																			
EDAW	wa	$(A) \leftarrow (V) \bullet (wa)$	0	1	1	0	1	0	0	1	1	1	0	1	0	0	0	0	14	3	Zero
		Offset																			
ONAW	wa	$(A) \wedge (V) \bullet (wa)$	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	14	3	No zero
		Offset																			
OFFAW	wa	$(A) \wedge (V) \bullet (wa)$	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	14	3	Zero
		Offset																			
ANIW	*wa,byte	$((V) \bullet (wa)) \leftarrow ((V) \bullet (wa)) \wedge \text{byte}$	0	0	0	0	0	1	0	1								19	3		
		Data																			
ORIW	*wa,byte	$((V) \bullet (wa)) \leftarrow ((V) \bullet (wa)) \vee \text{byte}$	0	0	0	1	0	1	0	1								19	3		
		Data																			
GTIW	*wa,byte	$((V) \bullet (wa)) - \text{byte} - 1$	0	0	1	0	1	0	1									13	3	No borrow	
		Data																			
LTIW	*wa,byte	$((V) \bullet (wa)) - \text{byte}$	0	0	1	1	0	1	0	1								13	3	Borrow	
		Data																			
NEIW	*wa,byte	$((V) \bullet (wa)) - \text{byte}$	0	1	1	0	0	1	0	1								13	3	No zero	
		Data																			
EQIW	*wa,byte	$((V) \bullet (wa)) - \text{byte}$	0	1	1	1	0	1	0	1								13	3	Zero	
		Data																			
ONIW	*wa,byte	$((V) \bullet (wa)) \wedge \text{byte}$	0	1	0	0	1	0	1									13	3	No zero	
		Data																			
OFFIW	*wa,byte	$((V) \bullet (wa)) \wedge \text{byte}$	0	1	0	1	0	1	0	1								13	3	Zero	
		Data																			

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Bytes	State(1)	Ship Condition															
			B1				B2				B3				B4																					
			7	6	5	4	3	2	1	0	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 0	0 1 0 0 0 0 0 0	0 1 0 0 0 0 0 0	R ₁	R ₀	11	2																											
DADD	EA,rp3 (EA) ← (EA) + (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 1 0 0 0 0 1 0	1 1 0 0 0 0 1 0	P ₁	P ₀	11	2																											
DADC	EA,rp3 (EA) ← (EA) + (rp3) + (CY)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 1 0 1 0 1 0 0	1 1 0 1 0 1 0 0	P ₁	P ₀	11	2																											
DADDNC	EA,rp3 (EA) ← (EA) + (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 0 1 0 0 0 0	1 0 0 1 0 0 0 0	P ₁	P ₀	11	2																											
ESUB	EA,r2 (EA) ← (EA) - (r2)	0 1 1 1 0 0 0 0	0 1 0 0 0 0 0 0	0 1 1 0 0 0 0 0	0 1 1 0 0 0 0 0	R ₁	R ₀	11	2																											
DSUB	EA,rp3 (EA) ← (EA) - (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 1 1 0 0 0 1 0	1 1 1 0 0 0 1 0	P ₁	P ₀	11	2																											
DSBB	EA,rp3 (EA) ← (EA) - (rp3) - (CY)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 1 1 0 0 0 1 0	1 1 1 0 0 0 1 0	P ₁	P ₀	11	2																											
DSUNB	EA,rp3 (EA) ← (EA) - (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 0 1 0 0 0 0	1 0 0 1 0 0 0 0	P ₁	P ₀	11	2																											
DAN	EA,rp3 (EA) ← (EA) ∧ (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	P ₁	P ₀	11	2																											
DOR	EA,rp3 (EA) ← (EA) ∨ (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 0 1 0 0 0 0	1 0 0 1 0 0 0 0	P ₁	P ₀	11	2																											
DXR	EA,rp3 (EA) ← (EA) ∨ (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 0 1 0 0 0 0	1 0 0 1 0 0 0 0	P ₁	P ₀	11	2																											
DGT	EA,rp3 (EA) ← (rp3) - 1	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 0 1 0 0 0 0	1 0 0 1 0 0 0 0	P ₁	P ₀	11	2																											
DLI	EA,rp3 (EA) ← (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 1 0 0 0 1 0	1 0 1 0 0 0 1 0	P ₁	P ₀	11	2																											
DNE	EA,rp3 (EA) ← (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 0 1 0 0 0 1 0	1 0 1 0 0 0 1 0	P ₁	P ₀	11	2																											
DEQ	EA,rp3 (EA) ← (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 1 1 0 0 0 1 0	1 1 1 0 0 0 1 0	P ₁	P ₀	11	2																											
DON	EA,rp3 (EA) ∧ (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 1 1 0 0 0 1 0	1 1 1 0 0 0 1 0	P ₁	P ₀	11	2																											
DOFF	EA,rp3 (EA) ∧ (rp3)	0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0	1 1 0 0 0 0 1 0	1 1 0 0 0 0 1 0	P ₁	P ₀	11	2																											
Multiply/Divide																																				
MUL	r2 (EA) ← (A) × (r2)	0 1 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	R ₁	R ₀	32	2																											
DIV	r2 (EA) ← (EA) + (r2), (r2) ← Remainder	0 1 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	R ₁	R ₀	59	2																											
Increment/Decrement																																				
INR	r2 (r2) ← (r2) + 1	0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	R ₁	R ₀	4	1																											
INRIW	*wa ((V)*wa) ← ((V)*wa) + 1	0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Offset																														
INX	rp (rp) ← (rp) + 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Offset																														
EA	(EA) ← (EA) + 1	1 0 1 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0			7	1																											
DCR	r2 (r2) ← (r2) - 1	0 1 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	R ₁	R ₀	4	1																											
DCRIW	*wa ((V)*wa) ← ((V)*wa) - 1	0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Offset																														
DCX	rp (rp) ← (rp) - 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Offset																														
EA	(EA) ← (EA) - 1	1 0 1 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0			7	1																											
Other																																				
DAA	Decimal Adjust Accumulator	0 1 1 0 0 0 0 0	0 1 1 0 0 0 0 0	0 1 1 0 0 0 0 0	0 1 1 0 0 0 0 0			4	1																											
STC	(CY) ← 1	0 1 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0			8	2																											
CLC	(CY) ← 0	0 1 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0			8	2																											

Instruction Set (cont)

Operation Code

Mnemonic	Operand	Operation	Operation Code																Bytes	State(s)	Ship Condition						
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0									
Others (cont)																											
NEGAI	(A) ← (A) + 1		0	1	0	0	1	0	0	0	0	0	0	1	1	0	1	0	8						2		
Rotate and Shift																											
RLD	Rotate left digit		0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	17						2		
RRD	Rotate right digit		0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	1	17						2		
RL	(R _{2m} +1) ← (R _{2m}), (R ₂₀) ← (CY), (CY) ← (R ₂₇)		0	1	0	0	1	0	0	0	0	0	1	1	0	1	R ₁	R ₀	8					2		Carry	
RLR	(R _{2m} -1) ← (R _{2m}), (R ₂₇) ← (CY), (CY) ← (R ₂₀)		0	1	0	0	1	0	0	0	0	1	1	0	0	R ₁	R ₀	8					2		Carry		
SLL	(R _{2m} +1) ← (R _{2m}), (R ₂₀) ← 0, (CY) ← (R ₂₇)		0	1	0	0	1	0	0	0	0	0	1	0	0	1	R ₁	R ₀	8				2				
SLR	(R _{2m} -1) ← (R _{2m}), (R ₂₇) ← 0, (CY) ← (R ₂₀)		0	1	0	0	1	0	0	0	0	0	1	0	0	0	R ₁	R ₀	8				2				
SLLC	(R _{2m} +1) ← (R _{2m}), (R ₂₀) ← 0, (CY) ← (R ₂₇)		0	1	0	0	1	0	0	0	0	0	0	0	0	1	R ₁	R ₀	8				2		Carry		
SLRC	(R _{2m} -1) ← (R _{2m}), (R ₂₇) ← 0, (CY) ← (R ₂₀)		0	1	0	0	1	0	0	0	0	0	0	0	0	0	R ₁	R ₀	8				2		Carry		
DRLL	(EA _n +1) ← (EA _n), (EA ₀) ← (CY), (CY) ← (EA ₁₅)		0	1	0	0	1	0	0	0	1	0	1	0	1	0	0	0	8					2			
DLRL	(EA _n -1) ← (EA _n), (EA ₁₅) ← (CY), (CY) ← (EA ₀)		0	1	0	0	1	0	0	0	1	0	1	1	0	0	0	0	8					2			
DSLL	(EA _n +1) ← (EA _n), (EA ₀) ← 0, (CY) ← (EA ₁₅)		0	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0	8					2			
DSLRL	(EA _n -1) ← (EA _n), (EA ₁₅) ← 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	High addr		Low addr									10		3			
JB	(PC _H) ← (B), (PC _L) ← (C)		0	0	1	0	0	0	0	1	High addr		Low addr									4		1			
JR	word (PC) ← (PC) + 1 + jdispl →		1	1	← jdispl →																10		1				
JRE	*word (PC) ← (PC) + 2 + jdispl		0	1	0	0	1	1	1	← jdispl →												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 → ((PC) + 3) _H , (SP) ← 2 → ((PC) + 3) _L , (PC) ← word, (SP) ← (SP) - 2		0	1	0	0	0	0	0	0	High addr		Low addr									16		3			
CALLB	(SP) ← 1 → ((PC) + 2) _H , (SP) ← 2 → ((PC) + 2) _L , (PC _H) ← (B), (PC _L) ← (C), (SP) ← (SP) - 2		0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	17				2		
CALLF	*word (SP) ← 1 → ((PC) + 2) _H , (SP) ← 2 → ((PC) + 2) _L , (PC ₁₅₋₁₁) ← 00001, (PC ₁₀₋₀) ← fa, (SP) ← (SP) - 2		0	1	1	1	1	1	← fa →													13		2			

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code																Bytes	State(1)	Skip Condition		
			B1	B2				B3				B4											
Call (cont)			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0					
CALL	word	$(SP) - 1 \leftarrow (PC) + 1)_H$ $(SP) - 2 \leftarrow (PC) + 1)_L$ $(PC) \leftarrow (128 + 21a), (PC_H) \leftarrow (29 + 21a), (SP) \leftarrow (SP) - 2$	1	0	0				1a									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		0	1	0	1	1	B ₂	B ₁	B ₀	Offset						10	2	2	Bit Test			
CPU Control																							
SK	f	Skip if f = 1	0	1	0	0	1	0	0	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	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	I ₄	I ₃	I ₂	I ₁	I ₀	8	2	irf = 1		
SKNT	irf	Skip if irf = 0 Reset irf if irf = 1	0	1	0	0	1	0	0	0	0	0	1	I ₄	I ₃	I ₂	I ₁	I ₀	8	2	irf = 0		
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	1								4	1				
HLT		Halt CPU operation	0	1	0	0	1	0	0	0	0	0	0	1	1	0	1	1	12	2			
STOP		Stop system clock	0	1	0	0	1	0	0	0	1	0	1	1	0	1	1	12	2				

Notes

- In the case of skip condition, the idle states are as follows:
 1-byte instruction: 4 states
 2-byte instruction (with *): 7 states
 3-byte instruction (with *): 10 states
 4-byte instruction: 14 states
- B2 (Data): rpa2 = D + byte, H + byte.
- Right side of slash (/) in states indicates case rpa2, rpa3 = D + byte, H + A, H + B, H + EA, H + byte.
- B3 (Data): rpa3 = D + byte, H + byte