

**Description**

The μPD8748H and μPD8749H are part of the μPD8048 family of single-chip 8-bit microcomputers. They are high-speed NMOS processors that function efficiently in control and arithmetic applications. The flexible instruction set allows you to directly set and reset individual data bits within the accumulator and the I/O ports. The variety of branch and table look-up instructions simplifies the implementation of standard logic functions.

The instruction set is made up of one- and two-byte instructions. Over 70% are single-byte instructions. Instruction execution requires only one or two cycles. Over 50% require a single cycle.

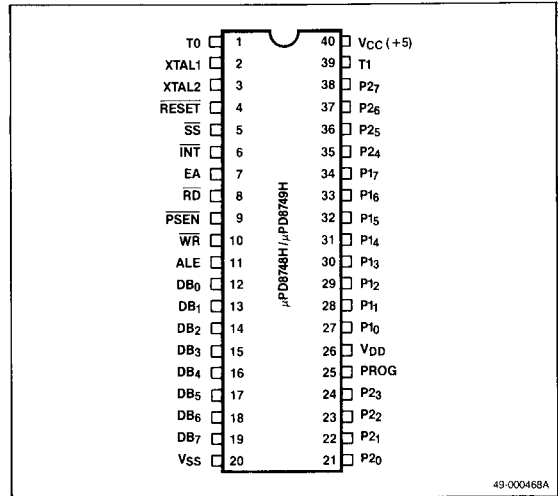
The μPD8748H/49H function as stand-alone microcomputers. You can expand their function with standard 8080A/8085A peripherals and memories. They each contain 1024 × 8 bits (μPD8748H), or 2048 × 8 bits (μPD8749H) of ROM program memory, 64 × 8 bits (μPD8748H), or 128 × 8 bits (μPD8749H) of RAM data memory, 27 I/O lines, an 8-bit internal timer/event counter, oscillator, and clock circuitry.

The μPD8748H/49H differs from the μPD8048/49 in that they have 1K (μPD8748H) or 2K (μPD8749H) of on-board EPROM. This is useful in preproduction or prototype applications where the software is not complete or in system designs in quantities that do not require a mask ROM. See the μPD8048H/8035HL or μPD8049H/8039HL data sheets for more information.

**Features**

- Low programming voltage (21 V)
- μPD8748H is fully compatible with 8048/8748/8035
- μPD8749H is fully compatible with 8049/8749/8039
- NMOS silicon gate technology
- Single +5V supply
- 1.36μs instruction execution time
- 96 instructions; 70% single byte
- Internal timer/event counter
- 1024 × 8 EPROM program memory (μPD8748H only)
- 2048 × 8 EPROM program memory (μPD8749H only)
- 64 × 8 byte RAM data memory
- Single interrupt level
- 27 I/O lines
- Internal clock generator
- 8-level stack
- Compatible with 8080A/8085A peripherals
- Available in one-time-programmable plastic package

**Pin Configuration**



**Ordering Information**

Part Number	Package Type	Max Frequency of Operation
μPD8748HC	40-Pin plastic DIP	11 MHz
μPD8748HD	40-Pin cerdip with quartz window	11 MHz
μPD8749HC	40-Pin plastic DIP	11 MHz
μPD8749HD	40-Pin cerdip with quartz window	11 MHz

**Pin Identification**

No.	Symbol	Function
1, 39	T0, T1	Testable inputs 0 and 1
2, 3	XTAL1, XTAL2	Crystal inputs
4	RESET	System reset input
5	SS	Single step input
6	INT	Interrupt input
7	EA	External access input
8	RD	Read strobe output
9	PSEN	Program store enable output
10	WR	Write strobe output
11	ALE	Address latch enable output
12-19	D <sub>0</sub> -D <sub>7</sub>	8-bit bidirectional port
20	V <sub>SS</sub>	Ground
21-24, 35-38	P <sub>20</sub> -P <sub>27</sub>	8-bit quasibidirectional port 2
25	PROG	Program pulse input during EPROM programming output when interfacing to 8243
26	V <sub>DD</sub>	Programming power supply
27-34	P <sub>10</sub> -P <sub>17</sub>	8-bit quasibidirectional port 1
40	V <sub>CC</sub>	Primary power supply

**Pin Functions****T0, T1 (Testable inputs 0 and 1)**

T0 uses the conditional transfer functions JT0 and JNT0; T1 uses JT1 and JNT1 to branch on condition of the external pin level. The ENT0 CLK instruction allows T0 to output, the internal state clock (CLK). Use the STRT CNT instruction to use T1 as the timer/counter input source.

**XTAL1, XTAL2 (Crystal inputs)**

XTAL1 and XTAL2 are two sides of the crystal input for an external oscillator or frequency (non-TTL compatible  $V_{IH}$ ).

**RESET (Reset)**

Active low input for processor initialization.  $\overline{\text{RESET}}$  is also used for PROM programming verification and power down (non-TTL compatible  $V_{IH}$ ).

**SS (Single step)**

Active low single step input.  $\overline{\text{SS}}$  and ALE allow the processor to single step through each instruction in program memory.

**INT (Interrupt)**

Active low interrupt input.  $\overline{\text{INT}}$  starts an interrupt if an enable interrupt instruction has been executed.  $\overline{\text{RESET}}$  disables the interrupt. You can test  $\overline{\text{INT}}$  with a conditional jump instruction.

**EA (External access)**

A logic 1 at the EA input tells the processor to perform all program memory fetches from external memory.

 **$\overline{\text{RD}}$  (Read strobe)**

Active low read strobe output.  $\overline{\text{RD}}$  pulses low when the processor performs a bus read.  $\overline{\text{RD}}$  also enables data onto the processor bus from a peripheral device and functions as a read strobe for external data memory.

 **$\overline{\text{PSEN}}$  (Program store enable)**

Active low program store enable output.  $\overline{\text{PSEN}}$  becomes active only during external memory fetches.

 **$\overline{\text{WR}}$  (Write strobe)**

Active low write strobe output.  $\overline{\text{WR}}$  pulses low when the processor performs a bus write.  $\overline{\text{WR}}$  also functions as a write strobe for external data memory.

**ALE (Address latch enable)**

Once each cycle, the falling edge of ALE latches the address for external memory or peripherals. You can also use ALE as a clock output.

**D<sub>0</sub>-D<sub>7</sub> (8-bit bidirectional bus)**

The  $\overline{\text{RD}}$  and  $\overline{\text{WR}}$  strobes allow you to perform synchronous reads and writes on this port. The contents of D<sub>0</sub>-D<sub>7</sub> can be latched in static mode. During an external memory fetch, D<sub>0</sub>-D<sub>7</sub> holds the LSBs of the program counter.  $\overline{\text{PSEN}}$  controls the incoming addressed instruction. D<sub>0</sub>-D<sub>7</sub> also holds address and data information for external RAM data store instruction (controlled by ALE,  $\overline{\text{RD}}$ , and  $\overline{\text{WR}}$ ).

**V<sub>SS</sub> (Ground)**

Ground.

**P<sub>20</sub>-P<sub>27</sub> (Port 2)**

Port 2 is one of two 8-bit quasibidirectional ports. P<sub>20</sub>-P<sub>23</sub> hold the four MSBs of the program counter for external data memory fetches; P<sub>24</sub>-P<sub>27</sub> hold data. P<sub>20</sub>-P<sub>23</sub> are also used as a 4-bit I/O bus for the μPD8243 I/O expander.

**PROG (Program pulse)**

Apply a +18 V pulse to the PROG input to program the μPD8748H. You can also use PROG as an output strobe for the μPD8243.

**V<sub>DD</sub> (Programming power supply)**

V<sub>DD</sub> must be +21 V to program the μPD8748H or +5 V for the ROM and PROM versions for normal operation.

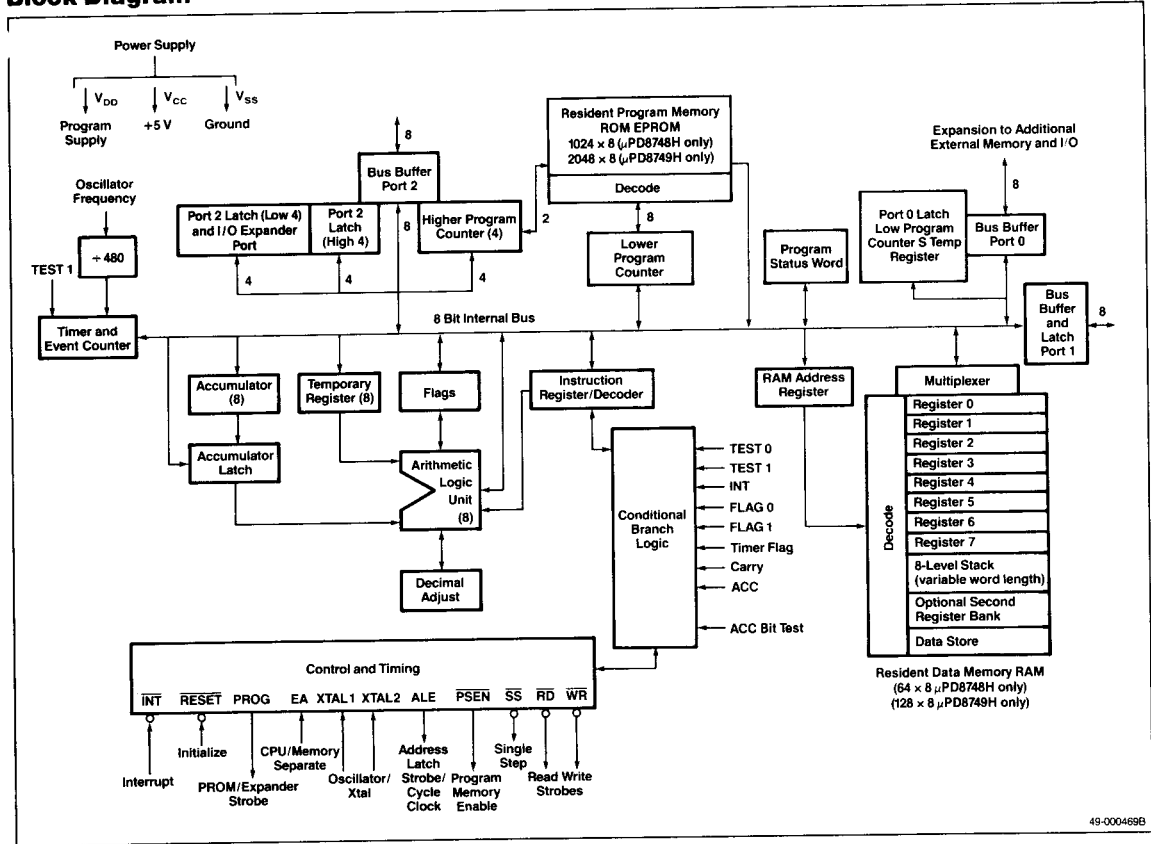
**P<sub>10</sub>-P<sub>17</sub> (Port 1)**

Port 1 is one of two 8-bit quasibidirectional ports used for external data memory fetches.

**V<sub>CC</sub> (Power supply)**

V<sub>CC</sub> must be +5 V to program and operate the μPD8748H.

**Block Diagram**



49-000469B

**Absolute Maximum Ratings**

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

Operating temperature, $T_{OP}$	$0^\circ\text{C}$ to $+70^\circ\text{C}$
Storage temperature, $T_{ST}$	$-65^\circ\text{C}$ to $+150^\circ\text{C}$
Output voltage, $V_O$	$-0.5\text{V}$ to $+7.0\text{V}$
Input voltage, $V_I$	$-0.5\text{V}$ to $+7.0\text{V}$
Power supply voltages, $V_{CC}, V_{DD}$	$-0.5\text{V}$ to $+7.0\text{V}$

**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 the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**DC Characteristics**

$T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = V_{DD} = +5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input low voltage (except XTAL1, XTAL2, RESET)	$V_{IL}$	-0.5		0.8	V	
Input low voltage (XTAL1, XTAL2, RESET)	$V_{IL1}$	-0.5		0.6	V	
Input high voltage (except XTAL1, XTAL2, RESET)	$V_{IH}$	2.0		$V_{CC}$	V	
Input high voltage (XTAL1, XTAL2, RESET)	$V_{IH1}$	3.8		$V_{CC}$	V	
Output low voltage (Bus)	$V_{OL}$			0.45	V	$I_{OL} = 2.0\text{mA}$
Output low voltage (RD, WR, PSEN, ALE)	$V_{OL1}$			0.45	V	$I_{OL} = 1.8\text{mA}$

**DC Characteristics (cont)** $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = V_{DD} = +5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ 

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Output low voltage (PROG)	$V_{OL2}$			0.45	V	$I_{OL} = 1.0\text{ mA}$
Output low voltage (all other outputs)	$V_{OL3}$			0.45	V	$I_{OL} = 1.6\text{ mA}$
Output high voltage (Bus)	$V_{OH}$	2.4			V	$I_{OH} = -400\ \mu\text{A}$
Output high voltage (RD, WR, PSEN, ALE)	$V_{OH1}$	2.4			V	$I_{OH} = -100\ \mu\text{A}$
Output high voltage (all other outputs)	$V_{OH2}$	2.4			V	$I_{OH} = -40\ \mu\text{A}$
Input leakage current (T1, INT)	$I_{L1}$			$\pm 10$	$\mu\text{A}$	$V_{SS} \leq V_i \leq V_{CC}$
Input leakage current (P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , EA, SS)	$I_{L11}$			-500	$\mu\text{A}$	$V_{SS} + 0.45\text{ V} \leq V_i \leq V_{CC}$
Output leakage current (Bus, T0, high impedance)	$I_{L0}$			$\pm 10$	$\mu\text{A}$	$V_{SS} + 0.45\text{ V} \leq V_i \leq V_{CC}$
Supply current ( $V_{DD}$ )	$I_{DD}$		2	5	mA	
Total supply current	$I_{DD} + I_{CC}$		85	110	mA	

**Programming DC Characteristics** $T_A = 25^\circ\text{C} \pm 5^\circ\text{C}$ ,  $V_{CC} = +5\text{V} \pm 5\%$ ,  $V_{DD} = +21\text{V} \pm 0.5\text{V}$ 

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
$V_{DD}$ voltage high level	$V_{DDH}$	20.5		21.5	V	
$V_{DD}$ voltage low level	$V_{DDL}$	4.75		5.25	V	
PROG voltage high level	$V_{PH}$	17.5		18.5	V	
PROG voltage low level	$V_{PL}$	4.0		$V_{CC}$	V	
EA program / verify voltage high level	$V_{EAH}$	17.5		18.5	V	
$V_{DD}$ high voltage supply current	$I_{DD}$			20.0	mA	
PROG high voltage supply current	$I_{PROG}$			1.0	mA	
EA high voltage supply current	$I_{EA}$			1.0	mA	

**AC Characteristics** $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = V_{DD} = +5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ 

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Read, Write, and Instruction Fetch — External Data and Program Memory</b>						
ALE pulse width	$t_{LL}$	150			ns	(1, 3)
Address setup before ALE	$t_{AL}$	70			ns	(1, 3)
Address hold after ALE	$t_{LA}$	50			ns	(1, 3)
Control pulse width (RD, WR)	$t_{CC1}$	480			ns	(1, 3)
Control pulse width (PSEN)	$t_{CC2}$	350			ns	(1, 3)
Data setup before WR	$t_{DW}$	390			ns	(1, 3)
Data hold after WR	$t_{WD}$	40			ns	(1, 2, 3)
Cycle time	$t_{CY}$	1.36		15.0	$\mu\text{s}$	
Data hold after RD, PSEN	$t_{DR}$	0		110	ns	(1, 3)
RD to data in	$t_{RD1}$			330	ns	(1, 3)
PSEN to data in	$t_{RD2}$			190	ns	(1, 3)
Address setup before WR	$t_{AW}$	300			ns	(1, 3)
Address setup before data in (RD)	$t_{AD1}$			730	ns	(1, 3)
Address setup before data in (PSEN)	$t_{AD2}$			460	ns	(1, 3)
Address float to RD, WR	$t_{AFC1}$	140			ns	(1, 3)
Address float to PSEN	$t_{AFC2}$	10			ns	(1, 3)
ALE to RD, WR delay time	$t_{LAFC1}$	200			ns	(1, 3)
ALE to PSEN delay time	$t_{LAFC2}$	60			ns	(1, 3)
RD, WR, PROG to ALE delay time	$t_{CA1}$	50			ns	(1, 3)
PSEN to ALE delay time	$t_{CA2}$	320			ns	(1, 3)

**Note:**

- (1) Control Output:  $C_L = 80\text{ pF}$ , Bus Output:  $C_L = 150\text{ pF}$
- (2) Bus high impedance, load =  $20\text{ pF}$
- (3) Clock oscillation frequency,  $f_{OSC} = 11\text{ MHz}$

**AC Characteristics (cont)**

$T_A = 0^\circ\text{C to } +70^\circ\text{C}, V_{CC} = V_{DD} = +5\text{ V} \pm 10\%, V_{SS} = 0\text{ V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
<b>Port 2 Timing</b>						
Port control setup before PROG	$t_{CP}$	100			ns	(1, 3)
Port control hold after PROG	$t_{PC}$	160			ns	(1, 3)
Input data setup before PROG	$t_{PR}$			650	ns	(1, 3)
Input data hold after PROG	$t_{PF}$	0		140	ns	(1, 3)
Output data setup before PROG	$t_{DP}$	400			ns	(1, 3)
Output data hold after PROG	$t_{PD}$	90			ns	(1, 3)
PROG pulse width	$t_{PP}$	700			ns	(1, 3)
Port 2 I/O data setup before ALE	$t_{PL}$	160			ns	(1, 3)
Port 2 I/O data setup after ALE	$t_{LP}$	15			ns	(1, 3)
ALE to port output time	$t_{PV}$			510	ns	(1, 3)
TO output cycle time	$t_{OPRR}$	270			ns	(1, 3)

**Note:**

- (1) Control output:  $C_L = 80\text{ pF}$ , bus output:  $C_L = 150\text{ pF}$
- (2) Bus high impedance, load =  $20\text{ pF}$
- (3) Clock oscillation frequency,  $f_{OSC} = 11\text{ MHz}$

**Programming AC Characteristics**

$T_A = 25^\circ\text{C} \pm 5^\circ, V_{DD} = +21\text{ V} \pm 0.5\text{ V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Address setup before RESET↑	$t_{AW}$	$4t_{CY}$				
Address hold after RESET↑	$t_{WA}$	$4t_{CY}$				
Data Input setup before PROG↓	$t_{DW}$	$4t_{CY}$				
Data input hold after PROG↓	$t_{WD}$	$4t_{CY}$				
RESET hold after verify	$t_{PH}$	$4t_{CY}$				
$V_{DD}$ setup before PROG↑	$t_{VDDW}$	0		1.0	ms	
$V_{DD}$ hold after PROG↓	$t_{VDDH}$	0		1.0	ms	
PROG pulse width	$t_{PW}$	50		60	ms	
TEST0 setup before program mode	$t_{TW}$	$4t_{CY}$				
TEST0 hold after program mode	$t_{WT}$	$4t_{CY}$				

**Programming AC Characteristics (cont)**

$T_A = 25^\circ\text{C} \pm 5^\circ, V_{DD} = +21\text{ V} \pm 0.5\text{ V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
TEST0 to data output delay(1)	$t_{DO}$			$4t_{CY}$		
RESET pulse width to latch address	$t_{WW}$	$4t_{CY}$				
$V_{DD}$ and PROG rise and fall times	$t_r, t_f$	0.5		100	$\mu\text{s}$	
CPU cycle time	$t_{CY}$	4.0		15	$\mu\text{s}$	$4.0\mu\text{s} / 3.7\text{ MHz}$
RESET setup before EA↑	$t_{RE}$	$4t_{CY}$				

**Note:**

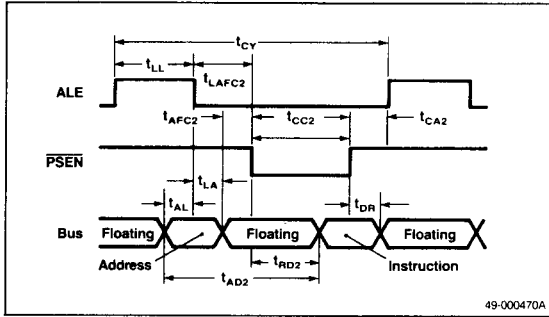
- (1) If TEST0 is high,  $t_{DO}$  is triggered by RESET↑.

**Bus Timing Requirements**

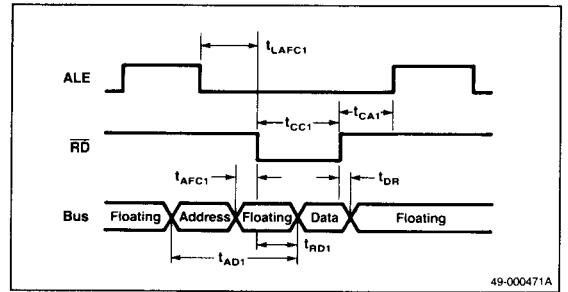
Symbol	Timing Formula	Min/Max	Unit
$t_{LL}$	$(7/30)t_{CY} - 170$	Min	ns
$t_{AL}$	$(2/15)t_{CY} - 110$	Min	ns
$t_{LA}$	$(1/15)t_{CY} - 40$	Min	ns
$t_{CC1}$	$(1/2)t_{CY} - 200$	Min	ns
$t_{CC2}$	$(2/5)t_{CY} - 200$	Min	ns
$t_{DW}$	$(13/30)t_{CY} - 200$	Min	ns
$t_{WD}$	$(1/15)t_{CY} - 50$	Min	ns
$t_{DR}$	$(1/10)t_{CY} - 30$	Max	ns
$t_{RD1}$	$(11/30)t_{CY} - 170$	Max	ns
$t_{RD2}$	$(4/15)t_{CY} - 170$	Max	ns
$t_{AW}$	$(1/3)t_{CY} - 150$	Min	ns
$t_{AD1}$	$(7/10)t_{CY} - 220$	Max	ns
$t_{AD2}$	$(1/2)t_{CY} - 220$	Max	ns
$t_{AFC1}$	$(2/15)t_{CY} - 40$	Min	ns
$t_{AFC2}$	$(1/30)t_{CY} - 40$	Min	ns
$t_{LAFC1}$	$(1/5)t_{CY} - 75$	Min	ns
$t_{LAFC2}$	$(1/10)t_{CY} - 75$	Min	ns
$t_{CA1}$	$(1/15)t_{CY} - 40$	Min	ns
$t_{CA2}$	$(4/15)t_{CY} - 40$	Min	ns
$t_{CP}$	$(2/15)t_{CY} - 80$	Min	ns
$t_{PC}$	$(4/15)t_{CY} - 200$	Min	ns
$t_{PR}$	$(17/30)t_{CY} - 120$	Max	ns
$t_{PF}$	$(1/10)t_{CY}$	Max	ns
$t_{DP}$	$(2/5)t_{CY} - 150$	Min	ns
$t_{PD}$	$(1/10)t_{CY} - 50$	Min	ns
$t_{PP}$	$(7/10)t_{CY} - 250$	Min	ns
$t_{PL}$	$(4/15)t_{CY} - 200$	Min	ns
$t_{LP}$	$(1/30)t_{CY} - 30$	Min	ns
$t_{PV}$	$(3/10)t_{CY} + 100$	Max	ns
$t_{OPRR}$	$(1/5)t_{CY}$	Min	ns
$t_{CY}$	$(1/f_{OSC}) \times 15$		$\mu\text{s}$

### Timing Waveforms

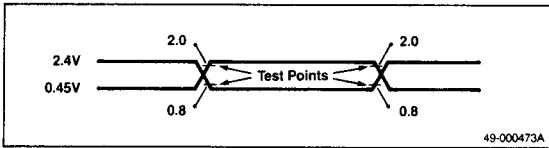
Instruction Fetch (External Program Memory)



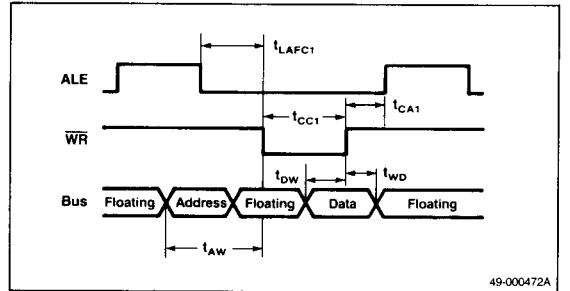
Read (External Data Memory)



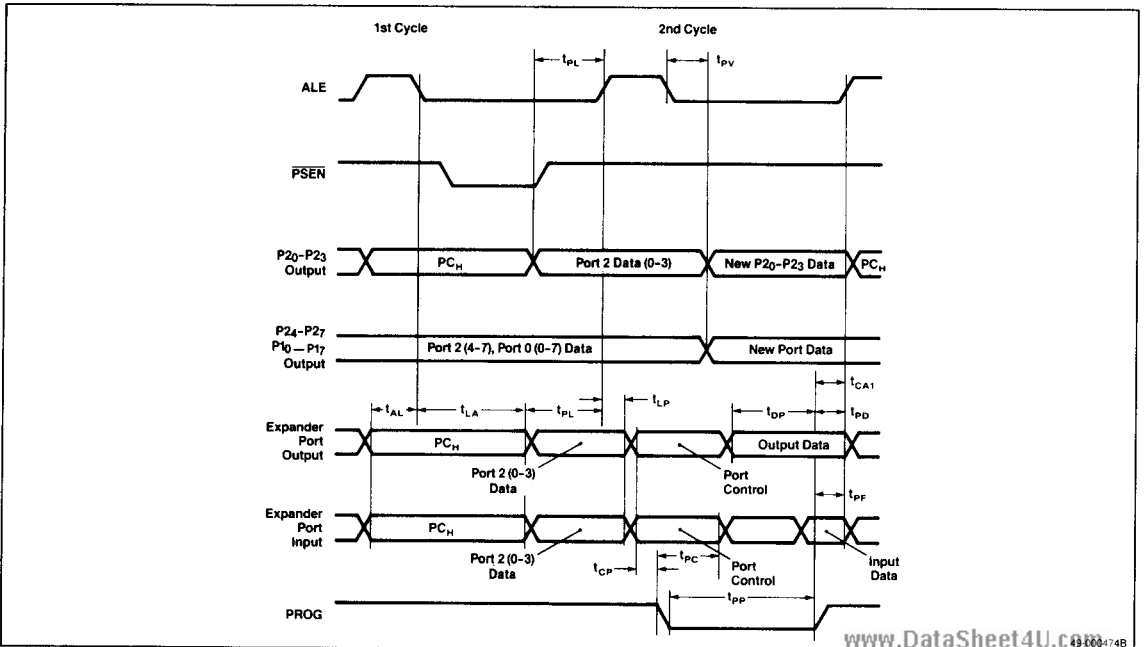
AC Test I/O Waveform



Write (External Data Memory)

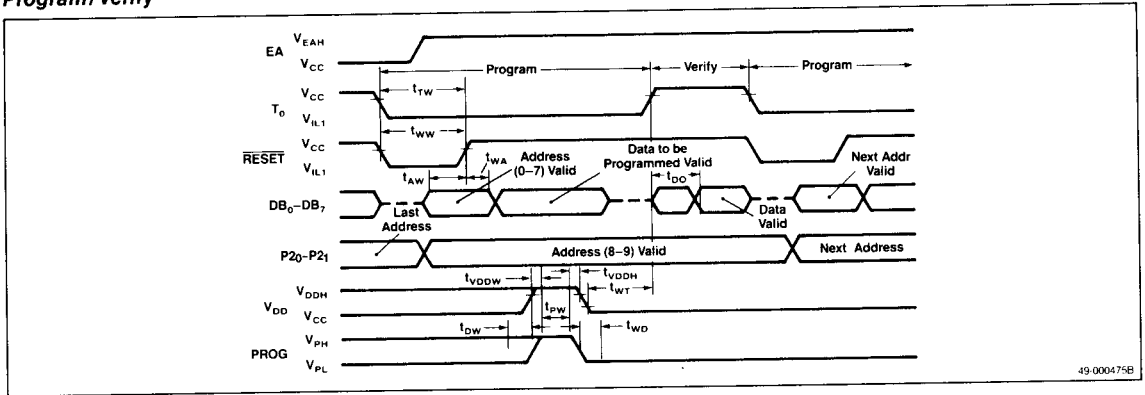


Port 1/Port 2



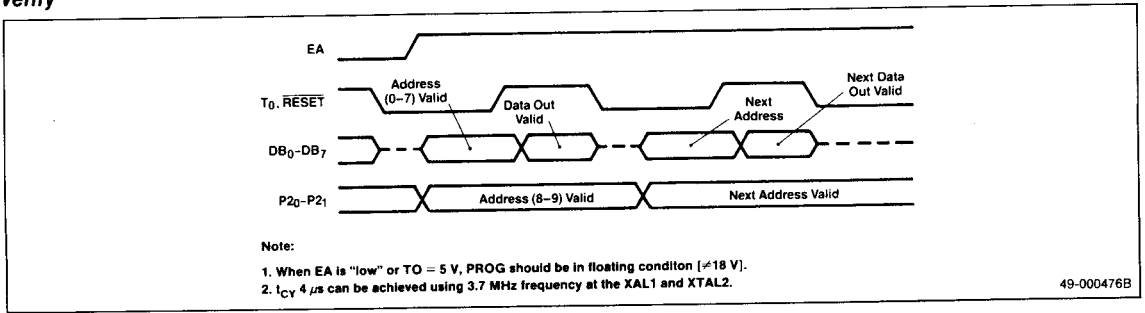
### Timing Waveforms (cont)

#### Program/Verify



49-000475B

#### Verify



**Note:**

1. When EA is "low" or T<sub>0</sub> = 5 V, PROG should be in floating condition ( $\approx 18$  V).
2.  $t_{CY}$  4  $\mu$ s can be achieved using 3.7 MHz frequency at the XAL1 and XTAL2.

49-000476B

## Instruction Set

Mnemonic Accumulator	Operation	Description	Operation Code																Flags		
			D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0	F1					
ADD A, # data	(A) ← (A) + data	Add immediate the specified data to the accumulator.	0	0	0	0	0	0	0	1	1	2	2	•							
ADD A, Rr	(A) ← (A) + (Rr) for r = 0-7	Add contents of designated register to the accumulator.	0	1	1	0	1	r	r	r	r	1	1	•							
ADD A, @ Rr	(A) ← (A) + ((Rr)) for r = 0-1	Add indirect the contents of the data memory location to the accumulator.	0	1	1	0	0	0	0	0	r	1	1	•							
ADDC A, # data	(A) ← (A) + (C) + data	Add immediate with carry the specified data to the accumulator.	0	0	0	1	0	0	1	1	2	2	•								
ADDC A, Rr	(A) ← (A) + (C) + (Rr) for r = 0-7	Add with carry the contents of the designated register to the accumulator.	0	1	1	1	1	r	r	r	1	1	•								
ADDC A, @ Rr	(A) ← (A) + (C) + ((Rr)) for r = 0-1	Add indirect with carry the contents of data memory location to the accumulator.	0	1	1	1	0	0	0	0	r	1	•								
ANL A, # data	(A) ← (A) AND data	Logical AND specified immediate data with accumulator.	0	1	0	1	0	0	1	1	2	2									
ANL A, Rr	(A) ← (A) AND (Rr) for r = 0-7	Logical AND contents of designated register with accumulator.	0	1	0	1	1	r	r	r	1	1									
ANL A, @ Rr	(A) ← (A) AND ((Rr)) for r = 0-1	Logical AND indirect the contents of data memory with accumulator.	0	1	0	1	0	0	0	0	r	1									
CPL A	(A) ← NOT (A)	Complement the contents of the accumulator.	0	0	1	1	0	1	1	1	1	1									
CLR A	(A) ← 0	Clear the contents of the accumulator.	0	0	1	0	0	1	1	1	1	1									
DA A	(A) ← (A) - 1	Decimal adjust the contents of the accumulator.	0	1	0	1	0	1	1	1	1	1	•								
DEC A	(A) ← (A) - 1	Decrement by 1 the accumulator's contents.	0	0	0	0	0	1	1	1	1	1									
INC A	(A) ← (A) + 1	Increment by 1 the accumulator's contents.	0	0	0	1	0	1	1	1	1	1									
ORL A, # data	(A) ← (A) OR data	Logical OR specified immediate data with accumulator.	0	1	0	0	0	1	1	1	2	2									
ORL A, Rr	(A) ← (A) OR (Rr) for r = 0-7	Logical OR contents of designated register with accumulator.	0	1	0	0	1	r	r	r	1	1									
ORL A, @ Rr	(A) ← (A) OR ((Rr)) for r = 0-1	Logical OR indirect the contents of data memory location with accumulator.	0	1	0	0	0	0	0	0	r	1									
RL A	(AN+1) ← (AN); N = 0-6 (A0) ← (A7)	Rotate accumulator left by 1 bit without carry.	1	1	1	0	0	1	1	1	1	1									
RLC A	(AN+1) ← (AN); N = 0-6 (A0) ← (C) (C) ← (A7)	Rotate accumulator left by 1 bit through carry.	1	1	1	1	0	1	1	1	1	1	•								
RR A	(AN) ← (AN+1); N = 0-6 (A7) ← (A0)	Rotate accumulator right by 1 bit without carry.	0	1	1	1	0	1	1	1	1	1									
RRC A	(AN) ← (AN+1); N = 0-6 (A7) ← (C) (C) ← (A0)	Rotate accumulator right by 1 bit through carry.	0	1	1	1	0	0	1	1	1	1	•								



**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code										Flags		
			D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0
<b>Accumulator (cont)</b>															
SWAP A	$(A_4-A_7) \leftrightarrow (A_0-A_3)$	Swap the 2 4-bit nibbles in the accumulator.	0	1	0	0	0	1	1	1	1	1	1	1	1
XRL A, # data	$(A) \leftarrow (A) \text{ XOR data}$	Logical XOR specified immediate data with accumulator.	1	1	0	1	0	0	1	1	2	2			
			d7	d6	d5	d4	d3	d2	d1	d0					
XRL A, Rr	$(A) \leftarrow (A) \text{ XOR (Rr)}$ for $r = 0-7$	Logical XOR contents of designated register with accumulator.	1	1	0	1	1	r	r	r	1	1			
XRL A, @ Rr	$(A) \leftarrow (A) \text{ XOR ((Rr))}$ for $r = 0-1$	Logical XOR Indirect the contents of data memory location with accumulator.	1	1	0	1	0	0	0	r	1	1			
<b>Branch</b>															
DJNZ Rr, addr	$(Rr) \leftarrow (Rr) - 1$ ; $r = 0-7$ if $(Rr) \neq 0$ ; $(PC_0-PC_7) \leftarrow \text{addr}$	Decrement the specified register and test contents.	1	1	1	0	1	r	r	r	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JBB addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $B_0 = 1$ $(PC) \leftarrow (PC) + 2$ if $B_0 = 0$	Jump to specified address if accumulator bit is set.	b2	b1	b0	1	0	0	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JC addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $C = 1$ $(PC) \leftarrow (PC) + 2$ if $C = 0$	Jump to specified address if carry flag is set.	1	1	1	1	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JFO addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $F0 = 1$ $(PC) \leftarrow (PC) + 2$ if $F0 = 0$	Jump to specified address if flag F0 is set.	1	0	1	1	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JF1 addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $F1 = 1$ $(PC) \leftarrow (PC) + 2$ if $F1 = 0$	Jump to specified address if flag F1 is set.	0	1	1	1	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JMP addr	$(PC_0-PC_{10}) \leftarrow (\text{addr}_0-\text{addr}_{10})$ $(PC_0-PC_7) \leftarrow (\text{addr}_0-\text{addr}_7)$ $(PC_{11}) \leftarrow \text{DBF}$	Direct jump to specified address within the 2K address block.	a10	a9	a8	0	0	1	0	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JMPP @ A	$(PC_0-PC_7) \leftarrow (A)$	Jump indirect to specified address with address page.	1	0	1	1	0	0	1	1	2	1			
JNC addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $C = 0$ $(PC) \leftarrow (PC) + 2$ if $C = 1$	Jump to specified address if carry flag is low.	1	1	1	0	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JNI addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $I = 0$ $(PC) \leftarrow (PC) + 2$ if $I = 1$	Jump to specified address if interrupt is low.	1	0	0	0	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JNTO addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $TO = 0$ $(PC) \leftarrow (PC) + 2$ if $TO = 1$	Jump to specified address if test 0 is low.	0	0	1	0	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JNFI addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $T1 = 0$ $(PC) \leftarrow (PC) + 2$ if $T1 = 1$	Jump to specified address if test 1 is low.	0	1	0	0	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JNZ addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $A = 0$ $(PC) \leftarrow (PC) + 2$ if $A = 1$	Jump to specified address if accumulator is non-zero.	1	0	0	1	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JTF addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $TF = 1$ $(PC) \leftarrow (PC) + 2$ if $TF = 0$	Jump to specified address if timer flag is set to 1.	0	0	0	1	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JTO addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $TO = 1$ $(PC) \leftarrow (PC) + 2$ if $TO = 0$	Jump to specified address if test 0 is a 1.	0	0	1	1	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
JTF1 addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if $T1 = 1$ $(PC) \leftarrow (PC) + 2$ if $T1 = 0$	Jump to specified address if test 1 is a 1.	0	1	0	1	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					

## Instruction Set (cont)

Mnemonic	Operation	Description	Operation Code										Flags		
			D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0
<b>Branch (cont)</b>															
JZ addr	$(PC_0-PC_7) \leftarrow \text{addr}$ if A = 0 $(PC) \leftarrow (PC) + 2$ if A = 1	Jump to specified address if accumulator is 0.	1	1	0	0	0	1	1	0	2	2			
			a7	a6	a5	a4	a3	a2	a1	a0					
<b>Control</b>															
EMI		Enable the external interrupt input.	0	0	0	0	0	1	0	1	1	1			
DIS I		Disable the external interrupt input.	0	0	0	1	0	1	0	1	1	1			
ENI/O CLK		Enable the clock output pin I/O.	0	1	1	1	0	1	0	1	1	1			
SEL MB0	(DBF) $\leftarrow 0$	Select bank 0 (locations 0-2047) of program memory.	1	1	1	0	0	1	0	1	1	1			
SEL MB1	(DBF) $\leftarrow 1$	Select bank 1 (locations 2048-4095) of program memory.	1	1	1	1	0	1	0	1	1	1			
SEL RB0	(BS) $\leftarrow 0$	Select bank 0 (locations 0-7) of data memory.	1	1	0	0	0	1	0	1	1	1			
SEL RB1	(BS) $\leftarrow 1$	Select bank 1 (locations 24-31) of data memory.	1	1	0	1	0	1	0	1	1	1			
<b>Data Moves</b>															
MOV A, # data	(A) $\leftarrow$ data	Move immediate the specified data into the accumulator.	0	0	1	0	0	0	1	1	2	2			
			d7	d6	d5	d4	d3	d2	d1	d0					
MOV A, Rr	(A) $\leftarrow$ (Rr); r = 0-7	Move the contents of the designated registers into the accumulator.	1	1	1	1	1	1	r	r	1	1			
MOV A, @ Rr	(A) $\leftarrow$ ((Rr)); r = 0-1	Move indirect the contents of data memory location into the accumulator.	1	1	1	1	0	0	0	r	1	1			
MOV A, PSW	(A) $\leftarrow$ (PSW)	Move contents of the program status word into the accumulator.	1	1	0	0	0	1	1	1	1	1			
MOV Rr, # data	(Rr) $\leftarrow$ data; r = 0-7	Move immediate the specified data into the designated register.	1	0	1	1	1	r	r	r	2	2			
			d7	d6	d5	d4	d3	d2	d1	d0					
MOV Rr, A	(Rr) $\leftarrow$ (A); r = 0-7	Move accumulator contents into the designated register.	1	0	1	0	1	r	r	r	1	1			
MOV @ Rr, A	((Rr)) $\leftarrow$ (A); r = 0-1	Move indirect accumulator contents into data memory location.	1	0	1	0	0	0	0	r	1	1			
MOV @ Rr, # data	((Rr)) $\leftarrow$ data; r = 0-1	Move immediate the specified data into data memory.	1	0	1	1	0	0	0	1	2	2			
			d7	d6	d5	d4	d3	d2	d1	d0					
MOV PSW, A	(PSW) $\leftarrow$ (A)	Move contents of accumulator into the program status word.	1	1	0	1	0	1	1	1	1	1			
MOV P A, @ A	$(PC_0-PC_7) \leftarrow$ (A) (A) $\leftarrow$ ((PC))	Move data in the current page into the accumulator.	1	0	1	0	0	0	1	1	2	1			
MOV P A, @ A	$(PC_0-PC_7) \leftarrow$ (A) $(PC_8-PC_{10}) \leftarrow 011$ (A) $\leftarrow$ ((PC))	Move program data in page 3 into the accumulator.	1	1	1	0	0	0	1	1	2	1			
MOV A, @ R	(A) $\leftarrow$ ((Rr)); r = 0-1	Move indirect the contents of external data memory into the accumulator.	1	0	0	0	0	0	0	r	2	1			
MOV @ R, A	((Rr)) $\leftarrow$ (A); r = 0-1	Move indirect the contents of the accumulator into external data memory.	1	0	0	1	0	0	0	r	2	1			
XCH A, Rr	(A) $\leftrightarrow$ (Rr); r = 0-7	Exchange the accumulator and designated register's contents.	0	0	1	0	1	r	r	r	1	1			

**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code										Flags													
			D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0	F1										
<b>Data Moves (cont)</b>																										
XCH A, @ Rr	(A) ↔ ((Rr)); r = 0-1	Exchange indirect contents of accumulator and location in data memory.	0	0	1	0	0	0	0	0	0	0	0	0	1	1										
XCHD A, @ Rr	(A <sub>0</sub> -A <sub>3</sub> ) ↔ ((Rr)) <sub>p</sub> -((Rr)) <sub>s</sub> ; r = 0-1	Exchange indirect 4-bit contents of accumulator and data memory.	0	0	1	1	0	0	0	0	0	0	0	0	1	1										
<b>Flags</b>																										
CPL C	(C) ← NOT (C)	Complement contents of carry bit.	1	0	1	0	0	1	1	1	1	1	1	1	1	1	•									
CPL F0	(F0) ← NOT (F0)	Complement contents of flag F0.	1	0	0	1	0	1	0	1	0	1	1	1	1	1	•									
CPL F1	(F1) ← NOT (F1)	Complement contents of flag F1.	1	0	1	1	0	1	0	1	0	1	1	1	1	1	•									
CLR C	(C) ← 0	Clear contents of carry bit to 0.	1	0	0	1	0	1	1	1	1	1	1	1	1	1	•									
CLR F0	(F0) ← 0	Clear contents of flag 0 to 0.	1	0	0	0	0	1	0	1	0	1	1	1	1	1	•									
CLR F1	(F1) ← 0	Clear contents of flag 1 to 0.	1	0	1	0	0	1	0	1	0	1	1	1	1	1	•									
<b>Input / Output</b>																										
ANL BUS, # data	(bus) ← (bus) AND data	Logical AND immediate specified data with contents of bus.	1	0	0	1	1	0	0	0	0	0	0	0	2	2		d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>	
ANL Pp, # data	(Pp) ← (Pp) AND data p = 1-2	Logical AND immediate specified data with designated port (1 or 2).	1	0	0	1	1	0	0	0	0	0	0	0	2	2		d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>	
ANLD Pp, A	(Pp) ← (Pp) AND (A <sub>0</sub> -A <sub>3</sub> ); p = 4-7	Logical AND contents of accumulator with designated port (4-7).	1	0	0	1	1	1	1	1	1	1	1	1	2	1										
IN A, Pp	(A) ← (Pp); p = 1-2	Input data from designated port (1-2) into accumulator.	0	0	0	0	1	0	0	0	0	0	0	0	2	1										
INS A, BUS	(A) ← (bus)	Input strobed bus data into accumulator.	0	0	0	0	1	0	0	0	0	0	0	0	2	1										
MOVD A, Pp	(A <sub>0</sub> -A <sub>3</sub> ) ← (Pp); p = 4-7 (A <sub>4</sub> -A <sub>7</sub> ) ← 0	Move contents of designated port (4-7) into accumulator.	0	0	0	0	1	1	1	1	1	1	1	1	2	1										
MOVD Pp, A	(Pp) ← (A <sub>0</sub> -A <sub>3</sub> ); p = 4-7	Move contents of accumulator to designated port (4-7).	0	0	1	1	1	1	1	1	1	1	1	1	2	1										
ORL BUS, # data	(bus) ← (bus) OR data	Logical OR immediate specified data with contents of bus.	1	0	0	0	1	0	0	0	0	0	0	0	2	2		d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>	
ORLD Pp, A	(Pp) ← (Pp) OR (A <sub>0</sub> -A <sub>3</sub> ); p = 4-7	Logical OR contents of accumulator with designated port (4-7).	1	0	0	0	1	1	1	1	1	1	1	1	2	1										
ORL Pp, # data	(Pp) ← (Pp) OR data p = 1-2	Logical OR immediate specified data with designated port (1-2).	1	0	0	0	1	0	0	0	0	0	0	0	2	2		d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>	
OUT BUS, A	(bus) ← (A)	Output contents of accumulator onto bus.	0	0	0	0	0	0	0	0	0	1	0	0	2	1										
OUT Pp, A	(Pp) ← (A); p = 1-2	Output contents of accumulator to designated port (1-2).	0	0	1	1	1	1	1	1	1	1	1	1	2	1										
<b>Registers</b>																										
DEC Pp, (Rr)	(Rr) ← (Rr) - 1; r = 0-7	Decrement by 1 contents of designated register.	1	1	0	0	1	0	0	1	0	0	0	0	1	1		r	r	r	r	r	r	r	r	r
INC Rr	(Rr) ← (Rr) + 1; r = 0-7	Increment by 1 contents of designated register.	0	0	0	0	1	1	1	1	1	1	1	1	1	1		r	r	r	r	r	r	r	r	
INC @ Rr	((Rr)) ← ((Rr)) + 1; r = 0-1	Increment indirect by 1 the contents of data memory location.	0	0	0	0	1	0	0	0	0	0	0	0	1	1										

## Instruction Set (cont)

Mnemonic Subroutine	Operation	Description	Operation Code										Flags							
			D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0	F1				
CALL addr	$(SP) \leftarrow (PC)$ $(PSW_4-PSW_7)$ $(SP) \leftarrow (SP) + 1$ $(PC_8-PC_{10}) \leftarrow (addr_8-addr_{10})$ $(PC_0-PC_7) \leftarrow (addr_0-addr_7)$ $(PC_{11}) \leftarrow DBF$	Call designated subroutine.	a <sub>10</sub> a <sub>7</sub>	a <sub>9</sub> a <sub>6</sub>	a <sub>8</sub> a <sub>5</sub>	1	0	1	0	1	0	0	0	0	2	2				
RET	$(SP) \leftarrow (SP) = 1$ $(PC) \leftarrow (SP)$	Return from subroutine without restoring program status word.	1	0	0	0	0	0	0	0	1	1	2	1						
RETR	$(SP) \leftarrow (SP) = 1$ $(PC) \leftarrow (SP)$ $(PSW_4-PSW_7) \leftarrow ((SP))$	Return from subroutine restoring program status word.	1	0	0	1	0	0	1	1	2	1								
<b>Timer / Counter</b>																				
EN TCNTI		Enable internal interrupt flag for timer / counter output.	0	0	1	0	0	1	0	1	0	1	1	1						
DIS TCNTI		Disable internal interrupt flag for timer / counter output.	0	0	1	1	0	1	0	1	0	1	1	1						
MOV A, T	$(A) \leftarrow (T)$	Move contents of timer / counter into accumulator.	0	1	0	0	0	0	1	0	1	0	1	1						
MOV T, A	$(T) \leftarrow (A)$	Move contents of accumulator into timer / counter.	0	1	1	0	0	0	1	0	1	0	1	1						
STOP TCNT		Stop count for event counter.	0	1	1	0	0	1	0	1	0	1	1	1						
STR T CNT		Start count for event counter.	0	1	0	0	0	1	0	1	0	1	1	1						
STR T		Start count for timer.	0	1	0	1	0	1	0	1	0	1	1	1						
<b>Miscellaneous</b>																				
NOP		No operation performed.	0	0	0	0	0	0	0	0	0	0	0	0	1	1				

**Note:**

- Instruction code designations r and p form the binary representation of the registers and ports involved.
- The dot under the appropriate flag bit indicates that its content is subject to change by the instruction it appears in.
- References to the address and data are specified in bytes 2 and/or 1 of the instruction.
- Numerical subscripts appearing in the function column reference the specific bits affected.

**Instruction Set Symbol Definitions**

Symbol	Description
A	Accumulator
AC	Auxiliary carry flag
addr	Program memory address (12 bits)
B <sub>b</sub>	Bit designator (b = 0-7)
BS	Bank switch
BUS	Bus port
C	Carry flag
CLK	Clock signal
CNT	Event counter
D	Nibble designator (4 bits)
data	Number or expression (8 bits)
DBF	Memory bank flip-flop
F0, F1	Flags 0, 1
I	Interrupt
P	"In-page" operation designator

Symbol	Description
Pp	Port designator (p = 1, 2 or 4-7)
PSW	Program status word
Rr	Register designator (r = 0, 1 or 0-7)
SP	Stack pointer
T	Timer
TF	Timer flag
T0, T1	Testable flags 0, 1
X	External RAM
#	Prefix for immediate data
@	Prefix for indirect address
\$	Program counter's current value
(x)	Contents of external RAM location
((x))	Contents of memory location addressed by the contents of external RAM location
←	Replaced by

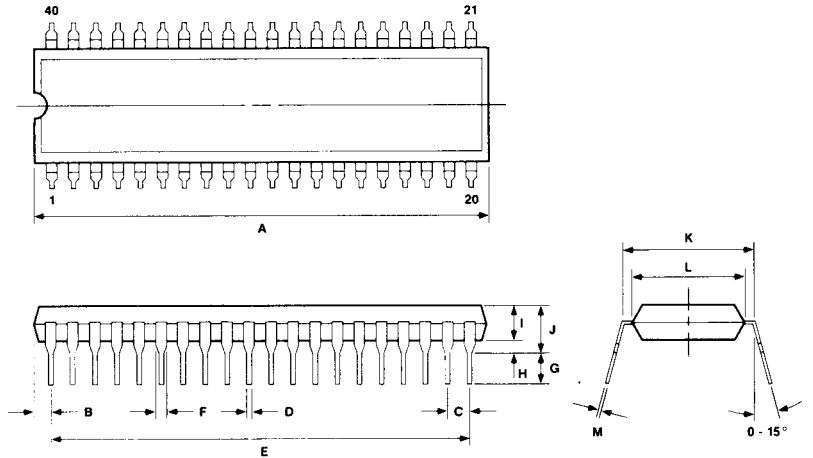
**Packaging Information**

**40-Pin Plastic DIP (600 mil)**

Item	Millimeters	Inches
A	53.34 max	2.100 max
B	2.54 max	.100 max
C	2.54 [TP]	.100 [TP]
D	.50 ± .10	.020 <sup>+.004</sup> -.005
E	48.26	1.900
F	1.2 min	.047 min
G	3.6 ± .3	.142 ± .012
H	.51 min	.020 min
I	4.31 max	.170 max
J	5.72 max	.226 max
K	15.24 [TP]	.600 [TP]
L	13.2	.520
M	.25 <sup>+.10</sup> -.05	.010 <sup>+.004</sup> -.003

**Notes:**

- [1] Each lead centerline is located within .25 mm [.010 inch] of its true position [TP] at maximum material condition.
- [2] Item "K" to center of leads when formed parallel.



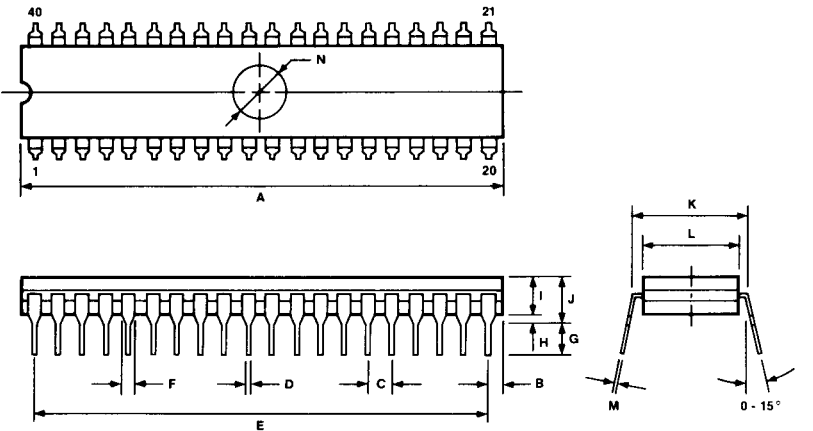
83-001399B

**40-Pin Cerdip with Window (600 mil)**

Item	Millimeters	Inches
A	53.34 max	2.100 max
B	2.54 max	.100 max
C	2.54 [TP]	.100 [TP]
D	.50 ± .10	.020 <sup>+.004</sup> -.005
E	48.26	1.900
F	1.2 min	.047 min
G	3.5 ± .3	.138 ± .012
H	.51 min	.020 min
I	3.80	.150
J	5.08 max	.200 max
K	15.24 [TP]	.600 [TP]
L	13.21	.520
M	.25 ± .05	.010 <sup>+.002</sup> -.003
N	φ 7.62	φ .300

**Note:**

- [1] Each lead centerline is located within .25 mm [.01 inch] of its true position [TP] at maximum material condition.
- [2] Item "K" to center of leads when formed parallel.



83-003785B

**Notes:**

**NEC**

**NEC Electronics Inc.**

CORPORATE HEADQUARTERS

401 Ellis Street  
P.O. Box 7241  
Mountain View, CA 94039  
TEL 415-960-6000  
TWX 910-379-6985

**For Literature Call Toll Free: 1-800-632-3531**  
**1-800-632-3532 (In California)**

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics Inc. The information in this document is subject to change without notice. Devices sold by NEC Electronics Inc. are covered by the warranty and patent indemnification provisions appearing in NEC Electronics Inc. Terms and Conditions of Sale only. NEC Electronics Inc. makes no warranty, express, statutory, implied, or by description, regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. NEC Electronics Inc. makes no warranty of merchantability or fitness for any purpose. NEC Electronics Inc. assumes no responsibility for any errors that may appear in this document. NEC Electronics Inc. makes no commitment to update or to keep current the information contained in this document.