NEC Electronics Inc.

μPD8748H/49H HIGH-SPEED, 8-BIT, SINGLE-CHIP NMOS MICROCOMPUTER WITH UV EPROM

August 1987

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.

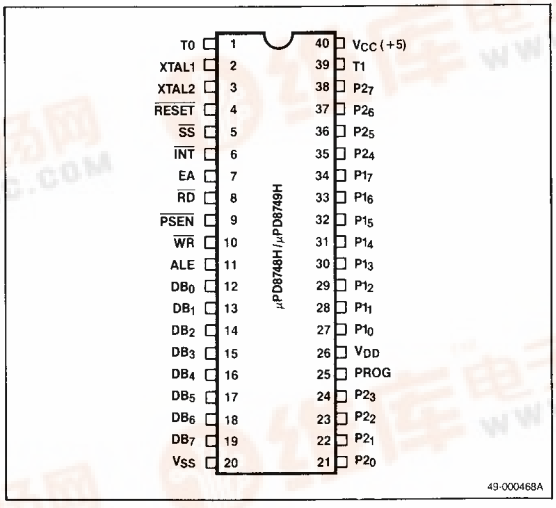
The μPD8748H/49H function as stand-alone microcomputers. You can expand their function with standard 8080A/8085A peripherals and memories. They each contain 1024 × 8 bits (μPD8748H), or 2048 × 8 bits (μPD8749H) of ROM program memory, 64 × 8 bits (μPD8748H), or 128 × 8 bits (μPD8749H) of RAM data memory, 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 (21V)
- ☐ 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	$\overline{\text{RESET}}$	System reset input
5	SS	Single step input
6	$\overline{\text{INT}}$	Interrupt input
7	EA	External access input
8	$\overline{\text{RD}}$	Read strobe output
9	$\overline{\text{PSEN}}$	Program store enable output
10	$\overline{\text{WR}}$	Write strobe output
11	ALE	Address latch enable output
12–19	D_0–D_7	8-bit bidirectional port
20	V_{SS}	Ground
21–24, 35–38	$P2_0$–$P2_7$	8-bit quasibidirectional port 2
25	PROG	Program pulse input during EPROM programming output when interfacing to 8243
26	V_{DD}	Programming 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}).

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

$\overline{\text{SS}}$ (Single step)

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

$\overline{\text{INT}}$ (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_0-D_7 (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_0-D_7 can be latched in static mode. During an external memory fetch, D_0-D_7 holds the LSBs of the program counter. $\overline{\text{PSEN}}$ controls the incoming addressed instruction. D_0-D_7 also holds address and data information for external RAM data store instruction (controlled by ALE, $\overline{\text{RD}}$, and $\overline{\text{WR}}$).

V_{SS} (Ground)

Ground.

$P2_0$-$P2_7$ (Port 2)

Port 2 is one of two 8-bit quasibidirectional ports. $P2_0$-$P2_3$ hold the four MSBs of the program counter for external data memory fetches; $P2_4$-$P2_7$ hold data. $P2_0$-$P2_3$ 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_{DD} (Programming power supply)

V_{DD} must be +21 V to program the μPD8748H or +5 V for the ROM and PROM versions for normal operation.

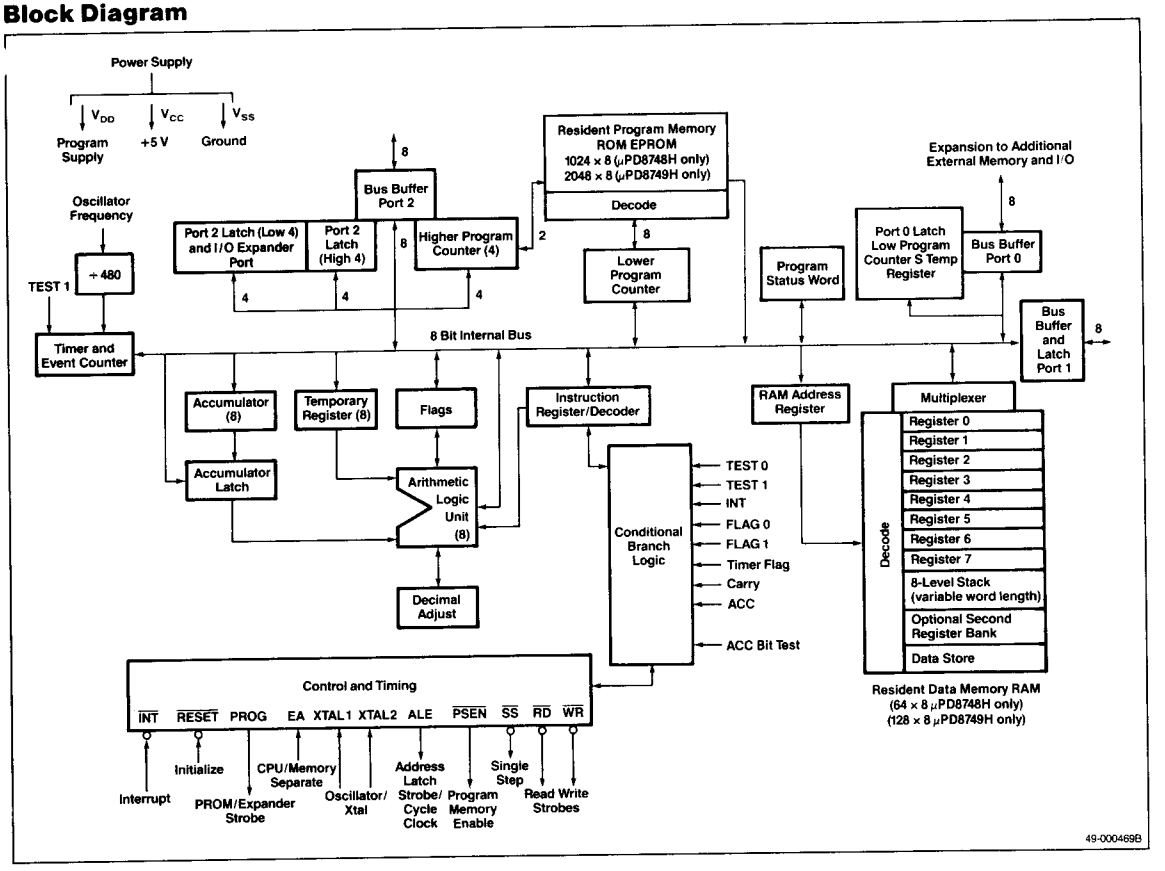
$P1_0$-$P1_7$ (Port 1)

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

V_{CC} (Power supply)

V_{CC} must be +5 V to program and operate the μPD8748H.

Block Diagram



Absolute Maximum Ratings

$T_A = 25°C$

Operating temperature, T_{OP}	0°C to +70°C
Storage temperature, T_{ST}	−65°C to +150°C
Output voltage, V_O	−0.5 V to +7.0 V
Input voltage, V_I	−0.5 V to +7.0 V
Power supply voltages, V_{CC}, V_{DD}	−0.5 V to +7.0 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

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

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

DC Characteristics (cont)

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

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Output low voltage (PROG)	V_{OL2}			0.45	V	$I_{OL} = 1.0$ mA
Output low voltage (all other outputs)	V_{OL3}			0.45	V	$I_{OL} = 1.6$ mA
Output high voltage (Bus)	V_{OH}	2.4			V	$I_{OH} = -400\,\mu A$
Output high voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	V_{OH1}	2.4			V	$I_{OH} = -100\,\mu A$
Output high voltage (all other outputs)	V_{OH2}	2.4			V	$I_{OH} = -40\,\mu A$
Input leakage current (T1, $\overline{INT}$)	I_{LI}			±10	μA	$V_{SS} \leq V_I \leq V_{CC}$
Input leakage current ($P1_0$–$P1_7$, $P2_0$–$P2_7$, EA, $\overline{SS}$)	I_{LI1}			−500	μA	$V_{SS} + 0.45V \leq V_I \leq V_{CC}$
Output leakage current (Bus, T0, high impedance)	I_{LO}			±10	μA	$V_{SS} + 0.45V \leq V_I \leq V_{CC}$
Supply current (V_{DD})	I_{DD}		2	5	mA	
Total supply current	$I_{DD} + I_{CC}$		85	110	mA	

Programming DC Characteristics

$T_A = 25°C \pm 5°C$, $V_{CC} = +5V \pm 5\%$, $V_{DD} = +21V \pm 0.5V$

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
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

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

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

Note:

(1) Control Output: $C_L = 80$ pF, Bus Output: $C_L = 150$ pF

(2) Bus high impedance, load = 20 pF

(3) Clock oscillation frequency, $f_{OSC} = 11$ MHz

AC Characteristics (cont)

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

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Port 2 Timing						
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)
T0 output cycle time	t_{OPRR}	270			ns	(1, 3)

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

Programming AC Characteristics

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

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Address setup before $\overline{RESET}\uparrow$	t_{AW}	$4\,t_{CY}$				
Address hold after $\overline{RESET}\uparrow$	t_{WA}	$4\,t_{CY}$				
Data input setup before PROG↓	t_{DW}	$4\,t_{CY}$				
Data input hold after PROG↓	t_{WD}	$4\,t_{CY}$				
$\overline{RESET}$ hold after verify	t_{PH}	$4\,t_{CY}$				
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}	$4\,t_{CY}$				

Programming AC Characteristics (cont)

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

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

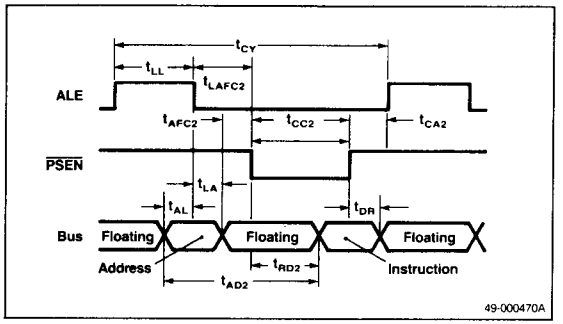
Note:
(1) If TEST0 is high, t_{DO} is triggered by $\overline{RESET}\uparrow$.

Bus Timing Requirements

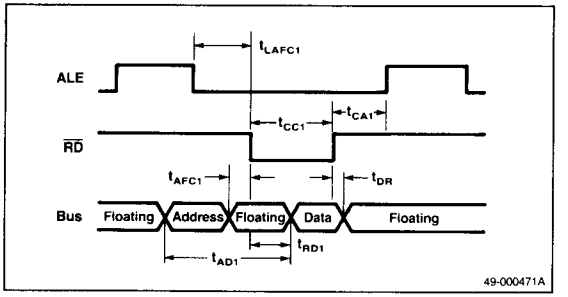
Symbol	Timing Formula	Min/Max	Unit
t_{LL}	$(7/30)\,t_{CY} - 170$	Min	ns
t_{AL}	$(2/15)\,t_{CY} - 110$	Min	ns
t_{LA}	$(1/15)\,t_{CY} - 40$	Min	ns
t_{CC1}	$(1/2)\,t_{CY} - 200$	Min	ns
t_{CC2}	$(2/5)\,t_{CY} - 200$	Min	ns
t_{DW}	$(13/30)\,t_{CY} - 200$	Min	ns
t_{WD}	$(1/15)\,t_{CY} - 50$	Min	ns
t_{DR}	$(1/10)\,t_{CY} - 30$	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

Timing Waveforms

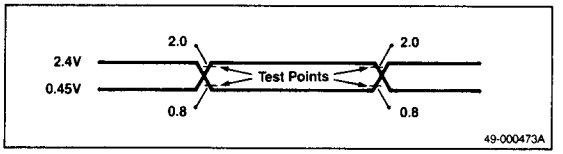
Instruction Fetch (External Program Memory)



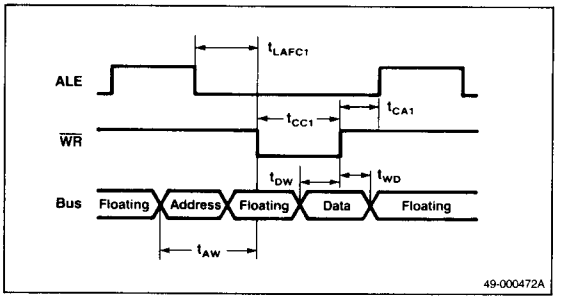
Read (External Data Memory)



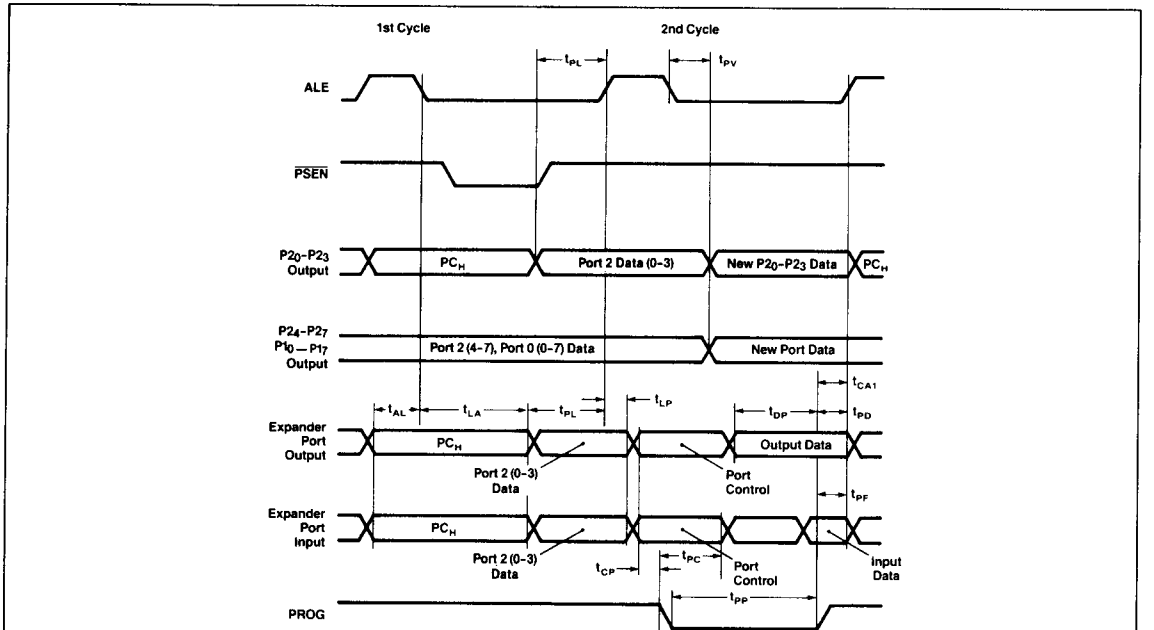
AC Test I/O Waveform



Write (External Data Memory)

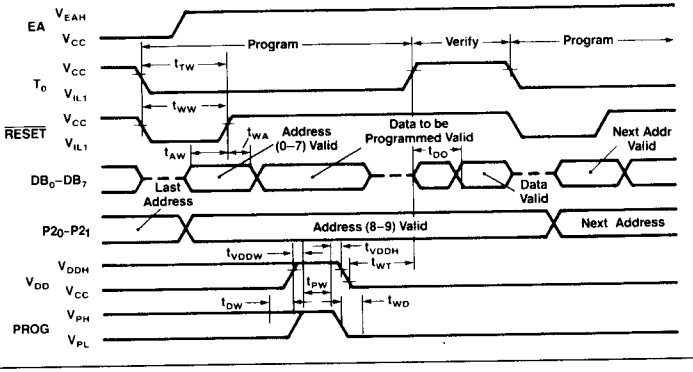


Port 1/Port 2



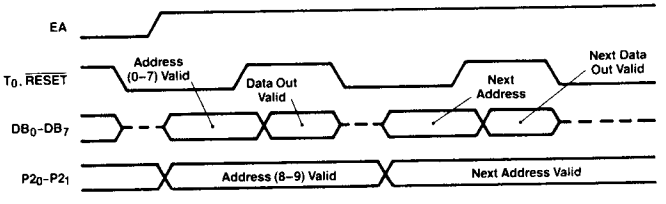
Timing Waveforms (cont)

Program/Verify



49-000475B

Verify



Note:

1. When EA is "low" or TO = 5 V, PROG should be in floating conditon [≠18 V].
2. t_{CY} 4 μs can be achieved using 3.7 MHz frequency at the XAL1 and XTAL2.

49-000476B

truction Set

nonic	Operation	Description	Operation Code								Cycles	Bytes	Flags			
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1
mulator																
A, # data	(A) ← (A) + data	Add immediate the specified data to the accumulator.	0 d_7	0 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	•			
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	1	1	•			
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	r	1	1	•			
A, # data	(A) ← (A) + (C) + data	Add immediate with carry the specified data to the accumulator.	0 d_7	0 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	•			
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	•			
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	r	1	1	•			
A, # data	(A) ← (A) AND data	Logical AND specified immediate data with accumulator.	0 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2				
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				
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	r	1	1				
A	(A) ← NOT (A)	Complement the contents of the accumulator.	0	0	1	1	0	1	1	1	1	1				
A	(A) ← 0	Clear the contents of the accumulator.	0	0	1	0	0	1	1	1	1	1				
		Decimal adjust the contents of the accumulator.	0	1	0	1	0	1	1	1	1	1	•			
A	(A) ← (A) – 1	Decrement by 1 the accumulator's contents.	0	0	0	0	0	1	1	1	1	1				
	(A) ← (A) + 1	Increment by 1 the accumulator's contents.	0	0	0	1	0	1	1	1	1	1				
A, # data	(A) ← (A) OR data	Logical OR specified immediate data with accumulator.	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2				
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				
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	r	1	1				
	$(A_{N+1}) \leftarrow (A_N)$; N = 0–6 $(A_0) \leftarrow (A_7)$	Rotate accumulator left by 1 bit without carry.	1	1	1	0	0	1	1	1	1	1				
A	$(A_{N+1}) \leftarrow (A_N)$; N = 0–6 $(A_0) \leftarrow$ (C) (C) ← (A_7)	Rotate accumulator left by 1 bit through carry.	1	1	1	1	0	1	1	1	1	1	•			
	$(A_N) \leftarrow (A_{N+1})$; N = 0–6 $(A_7) \leftarrow (A_0)$	Rotate accumulator right by 1 bit without carry.	0	1	1	1	0	1	1	1	1	1				
A	$(A_N) \leftarrow (A_{N+1})$; N = 0–6 $(A_7) \leftarrow$ (C) (C) ← (A_0)	Rotate accumulator right by 1 bit through carry.	0	1	1	0	0	1	1	1	1	1	•			

truction Set (cont)

onic	Operation	Description	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes	Flags C	AC	F0	F1
ulator (cont)																
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				
, # data	(A) ← (A) XOR data	Logical XOR specified immediate data with accumulator.	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2				
, Rr	(A) ← (A) XOR (Rr) for r = 0–7	Logical XOR contents of designated register with accumulator.	1	1	0	1	1	r	r	r	1	1				
, @ Rr	(A) ← (A) 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				
n																
Rr, addr	(Rr) ← (Rr) – 1; r = 0–7 If (Rr) ≠ 0; $(PC_0–PC_7)$ ← addr	Decrement the specified register and test contents.	1 a_7	1 a_6	1 a_5	0 a_4	1 a_3	r a_2	r a_1	r a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if B_b = 1 (PC) ← (PC) + 2 if B_b = 0	Jump to specified address if accumulator bit is set.	b_2 a_7	b_1 a_6	b_0 a_5	1 a_4	0 a_3	0 a_2	1 a_1	0 a_0	2	2				
dr	$(PC_0–PC_7)$ ← addr if C = 1 (PC) ← (PC) + 2 if C = 0	Jump to specified address if carry flag is set.	1 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if F0 = 1 (PC) ← (PC) + 2 if F0 = 0	Jump to specified address if flag F0 is set.	1 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	Jump to specified address if flag F1 is set.	0 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
addr	$(PC_8–PC_{10})$ ← $(addr_8–addr_{10})$ $(PC_0–PC_7)$ ← $(addr_0–addr_7)$ (PC_{11}) ← DBF	Direct jump to specified address within the 2K address block.	a_{10} a_7	a_9 a_6	a_8 a_5	0 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2				
@ A	$(PC_0–PC_7)$ ← ((A))	Jump indirect to specified address with address page.	1	0	1	1	0	0	1	1	2	1				
ddr	$(PC_0–PC_7)$ ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump to specified address if carry flag is low.	1 a_7	1 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if I = 0 (PC) ← (PC) + 2 if I = 1	Jump to specified address if interrupt is low.	1 a_7	0 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
addr	$(PC_0–PC_7)$ ← addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	Jump to specified address if test 0 is low.	0 a_7	0 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
addr	$(PC_0–PC_7)$ ← addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	Jump to specified address if test 1 is low.	0 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if A = 0 (PC) ← (PC) + 2 if A = 1	Jump to specified address if accumulator is non-zero.	1 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if TF = 1 (PC) ← (PC) + 2 if TF = 0	Jump to specified address if timer flag is set to 1.	0 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if T0 = 1 (PC) ← (PC) + 2 if T0 = 0	Jump to specified address if test 0 is a 1.	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
ddr	$(PC_0–PC_7)$ ← addr if T1 = 1 (PC) ← (PC) + 2 if T1 = 0	Jump to specified address if test 1 is a 1.	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				

truction Set (cont)

onic	Operation	Description	Operation Code								Cycles	Bytes	Flags			
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1
(cont)																
r	$(PC_0\text{-}PC_7) \leftarrow$ addr if A = 0 $(PC) \leftarrow (PC) + 2$ if A = 1	Jump to specified address if accumulator is 0.	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2				
		Enable the external interrupt input.	0	0	0	0	0	1	0	1	1	1				
		Disable the external interrupt input.	0	0	0	1	0	1	0	1	1	1				
CLK		Enable the clock output pin T0.	0	1	1	1	0	1	0	1	1	1				
B0	(DBF) ← 0	Select bank 0 (locations 0–2047) of program memory.	1	1	1	0	0	1	0	1	1	1				
B1	(DBF) ← 1	Select bank 1 (locations 2048–4095) of program memory.	1	1	1	1	0	1	0	1	1	1				
30	(BS) ← 0	Select bank 0 (locations 0–7) of data memory.	1	1	0	0	0	1	0	1	1	1				
31	(BS) ← 1	Select bank 1 (locations 24–31) of data memory.	1	1	0	1	0	1	0	1	1	1				
loves																
, # data	(A) ← data	Move immediate the specified data into the accumulator.	0 d_7	0 d_6	1 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2				
, Rr	(A) ← (Rr); r = 0–7	Move the contents of the designated registers into the accumulator.	1	1	1	1	1	r	r	r	1	1				
, @ Rr	(A) ← ((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				
, PSW	(A) ← (PSW)	Move contents of the program status word into the accumulator.	1	1	0	0	0	1	1	1	1	1				
r, # data	(Rr) ← data; r = 0–7	Move immediate the specified data into the designated register.	1 d_7	0 d_6	1 d_5	1 d_4	1 d_3	r d_2	r d_1	r d_0	2	2				
r, A	(Rr) ← (A); r = 0–7	Move accumulator contents into the designated register.	1	0	1	0	1	r	r	r	1	1				
@ Rr, A	((Rr)) ← (A); r = 0–1	Move indirect accumulator contents into data memory location.	1	0	1	0	0	0	0	r	1	1				
@ Rr,	((Rr)) ← data; r = 0–1	Move immediate the specified data into data memory.	1 d_7	0 d_6	1 d_5	1 d_4	0 d_3	0 d_2	0 d_1	1 d_0	2	2				
SW, A	(PSW) ← (A)	Move contents of accumulator into the program status word.	1	1	0	1	0	1	1	1	1	1				
A, @ A	$(PC_0\text{-}PC_7) \leftarrow$ (A) (A) ← ((PC))	Move data in the current page into the accumulator.	1	0	1	0	0	0	1	1	2	1				
3 A, @ A	$(PC_0\text{-}PC_7) \leftarrow$ (A) $(PC_8\text{-}PC_{10}) \leftarrow$ 011 (A) ← ((PC))	Move program data in page 3 into the accumulator.	1	1	1	0	0	0	1	1	2	1				
A, @ R	(A) ← ((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				
@ R, A	((Rr)) ← (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				
Rr	(A) ↔ (Rr); r = 0–7	Exchange the accumulator and designated register's contents.	0	0	1	0	1	r	r	r	1	1				

ruction Set (cont)

nic	Operation	Description	Operation Code								Cycles	Bytes	Flags			
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1
ves (cont)																
@ Rr	(A) ↔ ((Rr)); r = 0–1	Exchange indirect contents of accumulator and location in data memory.	0	0	1	0	0	0	0	r	1	1				
, @ Rr	$(A_0–A_3)$ ↔ $((Rr))_0$–$((Rr))_3$; r = 0–1	Exchange indirect 4-bit contents of accumulator and data memory.	0	0	1	1	0	0	0	r	1	1				
	(C) ← NOT (C)	Complement contents of carry bit.	1	0	1	0	0	1	1	1	1	1	•			
	(F0) ← NOT (F0)	Complement contents of flag F0.	1	0	0	1	0	1	0	1	1	1			•	
	(F1) ← NOT (F1)	Complement contents of flag F1.	1	0	1	1	0	1	0	1	1	1				•
	(C) ← 0	Clear contents of carry bit to 0.	1	0	0	1	0	1	1	1	1	1	•			
	(F0) ← 0	Clear contents of flag 0 to 0.	1	0	0	0	0	1	0	1	1	1			•	
	(F1) ← 0	Clear contents of flag 1 to 0.	1	0	1	0	0	1	0	1	1	1				•
Output																
IS,	(bus) ← (bus) AND data	Logical AND immediate specified data with contents of bus.	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2				
,	(Pp) ← (Pp) AND data p = 1–2	Logical AND immediate specified data with designated port (1 or 2).	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2				
Pp, A	(Pp) ← (Pp) AND $(A_0–A_3)$; p = 4–7	Logical AND contents of accumulator with designated port (4–7).	1	0	0	1	1	1	p	p	2	1				
p	(A) ← (Pp); p = 1–2	Input data from designated port (1–2) into accumulator.	0	0	0	0	1	0	p	p	2	1				
BUS	(A) ← (bus)	Input strobed bus data into accumulator.	0	0	0	0	1	0	0	0	2	1				
A, Pp	$(A_0–A_3)$ ← (Pp); p = 4–7 $(A_4–A_7)$ ← 0	Move contents of designated port (4–7) into accumulator.	0	0	0	0	1	1	p	p	2	1				
Pp, A	(Pp) ← $(A_0–A_3)$; p = 4–7	Move contents of accumulator to designated port (4–7).	0	0	1	1	1	1	p	p	2	1				
JS,	(bus) ← (bus) OR data	Logical OR immediate specified data with contents of bus.	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2				
Pp, A	(Pp) ← (Pp) OR $(A_0–A_3)$; p = 4–7	Logical OR contents of accumulator with designated port (4–7).	1	0	0	0	1	1	p	p	2	1				
,	(Pp) ← (Pp) OR data p = 1–2	Logical OR immediate specified data with designated port (1–2).	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2				
BUS, A	(bus) ← (A)	Output contents of accumulator onto bus.	0	0	0	0	0	0	1	0	2	1				
Pp,A	(Pp) ← (A); p = 1–2	Output contents of accumulator to designated port (1–2).	0	0	1	1	1	0	p	p	2	1				
ers																
r (Rr)	(Rr) ← (Rr) + 1; r = 0–7	Decrement by 1 contents of designated register.	1	1	0	0	1	r	r	r	1	1				
	(Rr) ← (Rr) = 1; r = 0–7	Increment by 1 contents of designated register.	0	0	0	1	1	r	r	r	1	1				
Rr	((Rr)) ← ((Rr)) = 1; r = 0–1	Increment indirect by 1 the contents of data memory location.	0	0	0	1	0	0	0	r	1	1				

truction Set (cont)

onic	Operation	Description	Operation Code								Cycles	Bytes	Flags			
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1
utine																
ddr	((SP)) ← (PC) (PSW_4–PSW_7), (SP) ← (SP) + 1 (PC_8–PC_{10}) ← ($addr_8$–$addr_{10}$) (PC_0–PC_7) ← ($addr_0$–$addr_7$) (PC_{11}) ← DBF	Call designated subroutine.	a_{10} a_7	a_9 a_6	a_8 a_5	1 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2				
	(SP) ← (SP) = 1 (PC) ← ((SP))	Return from subroutine without restoring program status word.	1	0	0	0	0	0	1	1	2	1				
	(SP) ← (SP) = 1 (PC) ← ((SP)) (PSW_4–PSW_7) ← ((SP))	Return from subroutine restoring program status word.	1	0	0	1	0	0	1	1	2	1				
/ Counter																
NTI		Enable internal interrupt flag for timer / counter output.	0	0	1	0	0	1	0	1	1	1				
NTI		Disable internal interrupt flag for timer / counter output.	0	0	1	1	0	1	0	1	1	1				
, T	(A) ← (T)	Move contents of timer / counter into accumulator.	0	1	0	0	0	0	1	0	1	1				
, A	(T) ← (A)	Move contents of accumulator into timer / counter.	0	1	1	0	0	0	1	0	1	1				
CNT		Stop count for event counter.	0	1	1	0	0	1	0	1	1	1				
NT		Start count for event counter.	0	1	0	0	0	1	0	1	1	1				
		Start count for timer.	0	1	0	1	0	1	0	1	1	1				
laneous																
		No operation performed.	0	0	0	0	0	0	0	0	1	1				

truction code designations r and p form the binary representation of the registers and ports involved.
e dot under the appropriate flag bit indicates that its content is subject to change by the instruction it appears in.
ferences to the address and data are specified in bytes 2 and/or 1 of the instruction.
merical 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_b	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

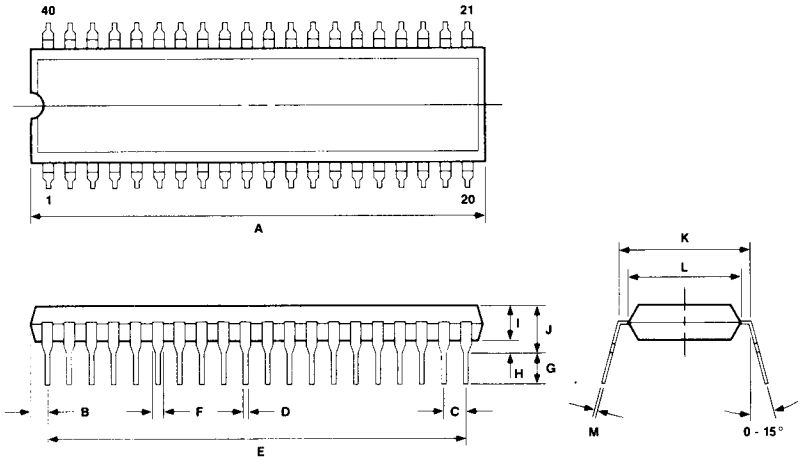
40-Pin Plastic DIP (600 mil)

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	$.020^{+.004}_{-.005}$
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	$.25^{+.10}_{-.05}$	$.010^{+.004}_{-.003}$

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

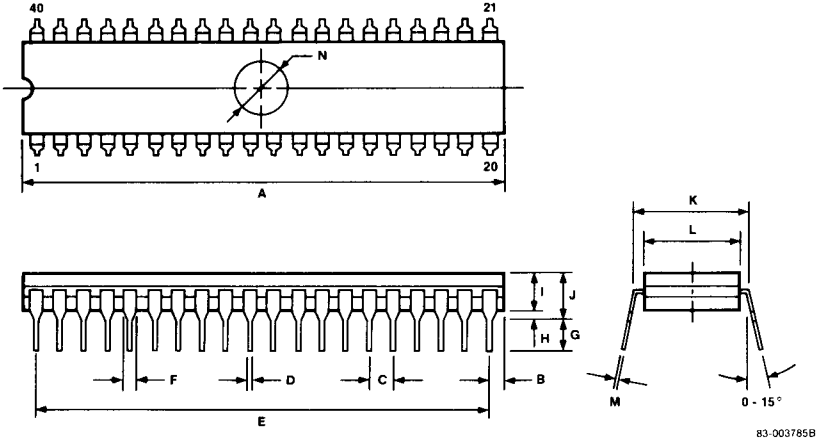
40-Pin Cerdip with Window (600 mil)

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	$.020^{+.004}_{-.005}$
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	$.010^{+.002}_{-.003}$
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

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