NEC

NEC Electronics Inc.

μPD8039HL/49H, μPD8749H HIGH-SPEED, 8-BIT, SINGLE-CHIP HMOS MICROCOMPUTERS

Description

The NEC μPD8039HL, μPD8049H and the μPD8749H are high performance, single component, 8-bit parallel microcomputers using n-channel silicon gate MOS technology. The processors differ only in their internal program memory options: the μPD8049H has 2K × 8 bytes of mask ROM, the μPD8749H has 2K × 8 of UV erasable EPROM and the μPD8039HL has external program memory.

The μPD8049H family functions efficiently in control as well as arithmetic applications. The powerful instruction set eases bit handling applications and provides facilities for binary and BCD arithmetic. Standard logic functions implementation is facilitated by the large variety of branch and table look-up instructions. The instruction set is comprised of 1 and 2 byte instructions, most of which are single-byte. The instruction set requires only 1 or 2 cycles per instruction with over 50 percent of the instructions single-cycle.

The μPD8049H family of microprocessors will function as stand-alone microcomputers. Their functions can easily be expanded using standard 8080A/8085A peripherals and memories. The μPD8039HL is intended for applications using external program memory only. It contains all the features of the μPD8049H except for the internal ROM. The external program memory can be implemented using standard 8080A/8085A memory products. The μPD8049H contains the following functions usually found in external peripheral devices: 2048 × 8 bits of mask ROM program memory; 128 × 8 bits of RAM data memory; 27 I/O lines; an 8-bit interval timer/event counter; and oscillator and clock circuitry. The μPD8749H differs from the μPD8049H in its 2048 × 8-bit UV erasable EPROM program memory instead of the mask ROM memory. It is useful in preproduction or prototype applications where the software design has not yet been finalized or in system designs whose quantities do not require a mask ROM.

Features

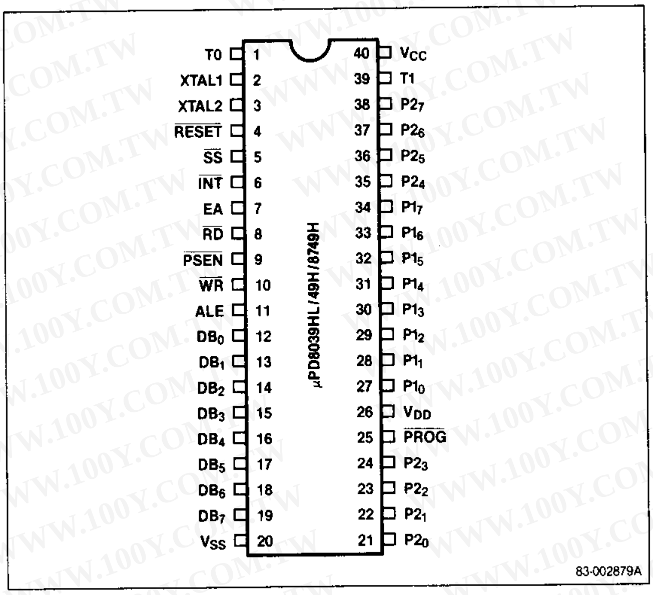
- □ High performance 11 MHz operation
- □ Fully compatible with industry standard 8039/8049/8749
- □ Pin compatible with the μPD8048/8748
- □ 1.36 μs cycle time. All instructions 1 or 2 bytes
- □ Programmable interval timer/event counter
- □ 2K × 8 bytes of ROM, 128 × 8 bytes of RAM
- □ External and internal interrupts
- □ 96 instructions: 70 percent single byte
- □ 27 I/O lines
- □ Internal clock generator
- □ Expandable with 8080A/8085A peripherals
- □ HMOS silicon gate technology
- □ Single +5 V ± 10 percent power supply

Ordering Information

Part Number	Package Type	Max Frequency of Operation
μPD8039HLC	40-pin plastic DIP	11 MHz
μPD8049HC	40-pin plastic DIP	11 MHz
μPD8749HC	40-pin plastic DIP	11 MHz
μPD8749HD	40-pin cerdip (Note 1)	11 MHz

Note:
(1) With quartz window.

Pin Configuration



Pin Identification

No.	Symbol	Function
1	T0	Test 0 input/output
2	XTAL1	Crystal 1 input
3	XTAL2	Crystal 2 input
4	$\overline{\text{RESET}}$	Reset input
5	$\overline{\text{SS}}$	Single step input
6	$\overline{\text{INT}}$	Interrupt input
7	EA	External access input
8	$\overline{\text{RD}}$	Read output
9	$\overline{\text{PSEN}}$	Program store enable output
10	$\overline{\text{WR}}$	Write output
11	ALE	Address latch enable output
12-19	DB_0-DB_7	Bidirectional data bus
20	V_{SS}	Ground
21-24	$P2_0$-$P2_7$	Quasi-bidirectional Port 2
25, 35-38	PROG	Program output
26	V_{DD}	RAM power supply
27-34	$P1_0$-$P1_7$	Quasi-bidirectional Port 1
39	T1	Test 1 input
40	V_{CC}	Primary power supply

Pin Functions

XTAL 1 (Crystal 1)

XTAL1 is one side of the crystal, LC, or external frequency source (non-TTL-compatible V_{IH}).

XTAL 2 (Crystal 2)

XTAL2 is the other side of the crystal or frequency source. For external sources, XTAL2 must be driven with the logical complement of the XTAL1 input.

T0 (Test 0)

T0 is the testable input using conditional transfer functions JT0 and JNT0. The internal state clock (CLK) is available to T0 using the ENT0 CLK instruction. T0 can also be used during programming as a testable flag.

T1 (Test 1)

T1 is the testable input using conditional transfer functions JT1 and JNT1. T1 can be made the counter/timer input using the STRT CNT instruction.

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

An active low on $\overline{\text{RESET}}$ initializes the processor. $\overline{\text{RESET}}$ is also used for PROM programming verification and power-down (non-TTL compatible V_{IH}).

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

An active low on $\overline{\text{SS}}$, together with ALE, causes the processor to execute the program one step at a time.

$\overline{\text{INT}}$ (Interrupt)

An active low on $\overline{\text{INT}}$ starts an interrupt if interrupts are enabled. A reset disables an interrupt. $\overline{\text{INT}}$ can be tested with the JNI instruction and, depending on the results, a jump to the specified address can occur.

EA (External Access)

An active high on EA disables internal program memory and fetches and accesses external program memory. EA is used for system testing and debugging.

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

$\overline{\text{RD}}$ will pulse low when the processor performs a bus read. An active low on $\overline{\text{RD}}$ enables data onto the processor bus from a peripheral device and functions as a read strobe for external data memory.

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

$\overline{\text{WR}}$ will pulse low when the processor performs a bus write. $\overline{\text{WR}}$ can also function as a write strobe for external data memory.

$\overline{\text{PSEN}}$ (Program Store Enable)

$\overline{\text{PSEN}}$ becomes active only during an external memory fetch. (Active low).

ALE (Address Latch Enable)

ALE occurs at each cycle. ALE can also be used as a clock output. The falling edge of ALE addresses external data memory or external program memory.

DB_0-DB_7 (Data Bus)

DB_0-DB_7 is a bidirectional port. Synchronous reads and writes can be performed on this port using $\overline{\text{RD}}$ and $\overline{\text{WR}}$ strobes. The contents of the DB_0-DB_7 bus can be latched in a static mode.

During an external memory fetch, DB_0-DB_7 output the low order eight bits of the memory address. $\overline{\text{PSEN}}$ fetches the instruction. DB_0-DB_7 also output the address of an external data memory fetch. The addressed data is controlled by ALE, $\overline{\text{RD}}$, and $\overline{\text{WR}}$.

$P1_0$-$P1_7$ (Port 1)

$P1_0$-$P1_7$ is an 8-bit quasi-bidirectional port.

$P2_0$-$P2_7$ (Port 2)

$P2_0$-$P2_7$ is an 8-bit quasi-bidirectional port. $P2_0$-$P2_3$ output the high order four bits of the address during an external program memory fetch. $P2_0$-$P2_3$ also function as a 4-bit I/O bus for the μPD82C43 I/O port expander.

PROG (Program Pulse)

PROG is used as an output pulse during a fetch when interfacing with the μPD82C43 I/O port expander. When the μPD8049H is used in a stand-alone mode, PROG can be allowed to float.

V_{CC} (Primary Power Supply)

V_{CC} is the primary power supply. V_{CC} is +5 V during normal operation.

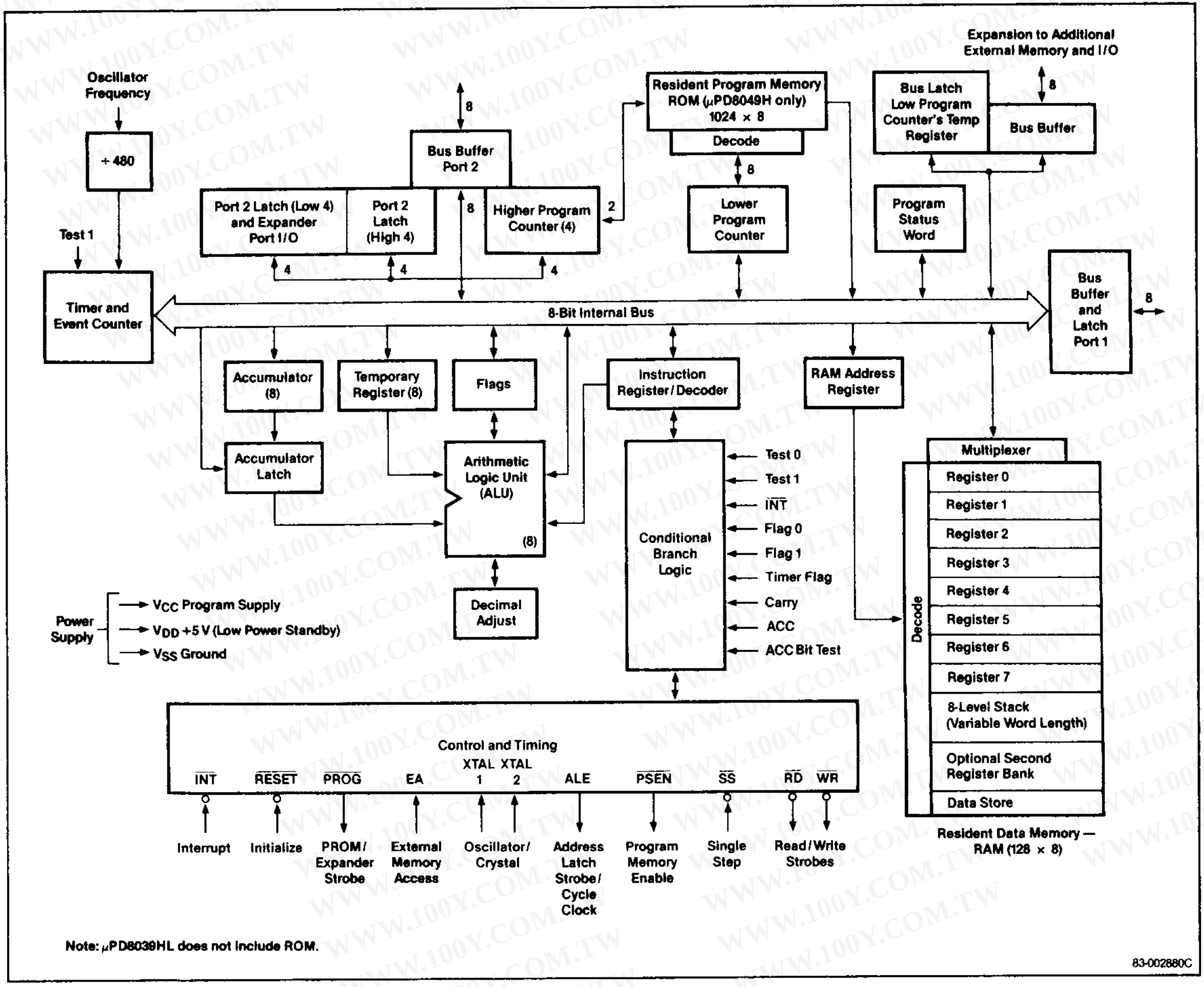
V_{DD} (RAM Power Supply)

V_{DD} provides +5 V to the 128 x 8-bit RAM section. During normal operation, V_{CC} must also be +5 V to provide power to the other functions in the device. During standby operation, V_{DD} must remain at +5 V while V_{CC} is at ground potential.

V_{SS} (Ground)

V_{SS} is ground potential.

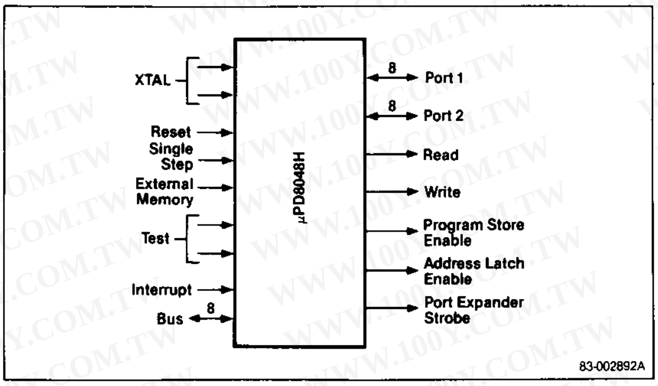
Block Diagram



Note: μPD8039HL does not include ROM.

83-002880C

Logic Symbol



Absolute Maximum Ratings

$T_A = 25°C$

Operating temperature, T_{OPT}	0°C to +70°C
Storage temperature, T_{STG}	−65°C to +150°C
Voltage on any pin	−0.5 V to +7.0 V (Note 1)
Power dissipation, P_D	1.5 W

Note:

(1) With respect to ground.

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

		Limits				
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Input low voltage (All except XTAL1, XTAL2)	V_{IL}	−0.5		0.8	V	
Input high voltage (All 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, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	V_{OL}			0.45	V	$I_{OL} = 2.0\,mA$
Output low voltage (All others except PROG)	V_{OL1}			0.45	V	$I_{OL} = 2.0\,mA$
Output low voltage (PROG)	V_{OL2}			0.45	V	$I_{OL} = 2.0\,mA$
Output high voltage (•••)	V_{OH}	2.4			V	$I_{OH} = -400\,\mu A$
Output high voltage (RD, WR, PSEN, ALE)	V_{OH1}	2.4			V	$I_{OH} = -400\,\mu A$
Output high voltage (all other outputs)	V_{OH2}	2.4			V	$I_{OH} = -40\,\mu A$
Input leakage current (T1, EA, INT)	I_{IL}			±10	µA	$V_{SS} \leq V_{IN} \leq V_{CC}$
Input leakage current ($P1_0$–$P1_7$, $P2_0$–$P2_7$, EA, SS)	I_{IL1}			−500	µA	$V_{SS} + 0.45\,V \leq V_{IN} \leq V_{CC}$
Output leakage current (BUS, T0, high impedance state)	I_{LO}			±10	µA	$V_{CC} \geq V_{IN} \geq V_{SS} + 0.45\,V$
Power down supply current	I_{DD}		5	10	mA	$T_A = 25°C$
			2	5		8749H only
Total supply current	$I_{DD} + I_{CC}$		80	110	mA	$T_A = 25°C$
			85	110		8749H only

DC Programming Characteristics

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

		Limits				
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
V_{DD} program voltage high level	V_{DDH}	20.5		21.5	V	
V_{DD} program voltage low level	V_{DDL}	4.75		5.25	V	
PROG program 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 $+70°C$, $V_{CC} = V_{DD} = +5\,V \pm 10\%$, $V_{SS} = 0\,V$

		Limits				
Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
ALE pulse width	t_{LL}	150			ns	
Address setup to ALE	t_{AL}	70			ns	
Address hold from ALE	t_{LA}	50			ns	
Control pulse width ($\overline{\text{RD}}$, $\overline{\text{WR}}$)	t_{CC1}	480			ns	
Control pulse width ($\overline{\text{PSEN}}$)	t_{CC2}	350			ns	
Data setup before $\overline{\text{WR}}$	t_{DW}	390			ns	
Data hold after $\overline{\text{WR}}$	t_{WD}	40			ns	(Note 2)
Data hold ($\overline{\text{RD}}$, $\overline{\text{PSEN}}$)	t_{DR}	0		110	ns	
$\overline{\text{RD}}$ to data in	t_{RD1}			350	ns	
$\overline{\text{PSEN}}$ to data in	t_{RD2}			210	ns	
Address setup to $\overline{\text{WR}}$	t_{AW}	300			ns	
Address setup to data ($\overline{\text{RD}}$)	t_{AD1}			750	ns	
Address setup to data ($\overline{\text{PSEN}}$)	t_{AD2}			480	ns	
Address float to $\overline{\text{RD}}$, $\overline{\text{WR}}$	t_{AFC1}	140			ns	
Address float to $\overline{\text{PSEN}}$	t_{AFC2}	10			ns	
ALE to control ($\overline{\text{RD}}$, $\overline{\text{WR}}$)	t_{LAFC1}	200			ns	
ALE to control ($\overline{\text{PSEN}}$)	t_{LAFC2}	60			ns	
Control to ALE ($\overline{\text{RD}}$, $\overline{\text{WR}}$, PROG)	t_{CA1}	50			ns	
Control to ALE ($\overline{\text{PSEN}}$)	t_{CA2}	320			ns	
Port control setup to $\overline{\text{PROG}}$	t_{CP}	100			ns	
Port control hold to $\overline{\text{PROG}}$	t_{PC}	160			ns	
$\overline{\text{PROG}}$ to P2 input valid	t_{PR}			650	ns	
Input data hold from $\overline{\text{PROG}}$	t_{PF}	0		140	ns	
Output data setup	t_{DP}	400			ns	
Output data hold	t_{PD}	90			ns	
$\overline{\text{PROG}}$ pulse width	t_{PP}	700			ns	
Port 2 I/O data setup to ALE	t_{PL}	160			ns	
Port 2 I/O data hold to ALE	t_{LP}	40			ns	
Port output from ALE	t_{PV}			510	ns	
Cycle time	t_{CY}	1.36		15	μs	
I/O rep rate	t_{OPRR}	270			ns	

Note:

(1) Control outputs: $C_L = 60\,pF$, bus outputs: $C_L = 150\,pF$

(2) Bus high impedance, load = 20 pF

(3) Calculated values will be equal to or better than published 8049 values.

AC Programming Characteristics

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

		Limits				
Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Address setup time to $\overline{\text{RESET}}$ ↑	t_{AW}	$4\,t_{CY}$				
Address hold time after $\overline{\text{RESET}}$ ↑	t_{WA}	$4\,t_{CY}$				
Data in setup time to PROG ↑	t_{DW}	$4\,t_{CY}$				
Data in hold time after PROG ↓	t_{WD}	$4\,t_{CY}$				
$\overline{\text{RESET}}$ hold time to verify	t_{PH}	$4\,t_{CY}$				
V_{DD}	t_{VDDW}	0		1.0	ms	
V_{DD} hold time after PROG ↓	t_{VDDH}	0		1.0	ms	
PROG pulse width	t_{PW}	50		60	ms	
TEST0 setup time for program mode	t_{TW}	$4\,t_{CY}$				
TEST0 hold time after program mode	t_{WT}	$4\,t_{CY}$				
TEST0 to data out delay(1)	t_{DO}			$4\,t_{CY}$		
$\overline{\text{RESET}}$ 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	

AC Programming Characteristics (cont)

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

		Limits				Test
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Conditions**
CPU operation cycle time	t_{CY}	4.0		15	μs	
RESET setup time before EA↑	t_{RE}	4 t_{CY}				

Note:

(1) Control outputs: $C_L = 60$ pF, bus outputs: $C_L = 150$ pF

(2) Bus high impedance, load = 20 pF

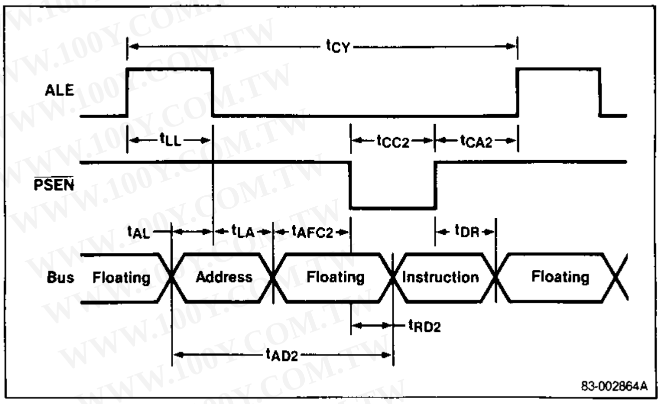
(3) Calculated values will be equal to or better than published 8049 values.

Bus Timing Requirements

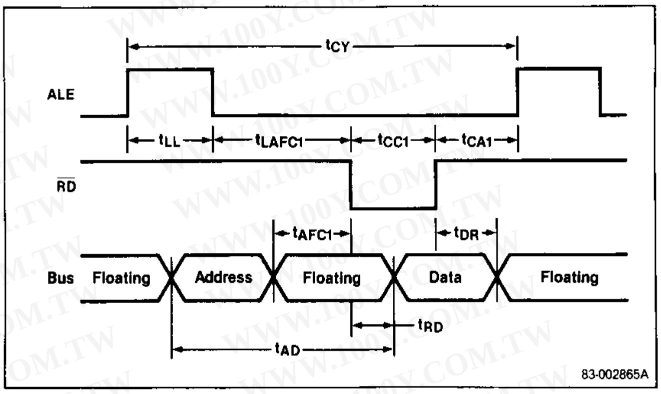
Symbol	Timing Formula	Min/Max	Unit
t_{LL}	(7/30) t_{CY} − 170	Min	ns
t_{AL}	(2/15) t_{CY} − 110	Min	ns
t_{LA}	(1/15) t_{CY} − 40	Min	ns
t_{CC1}	(1/2) t_{CY} − 200	Min	ns
t_{CC2}	(2/5) t_{CY} − 200	Min	ns
t_{DW}	(13/30) t_{CY} − 200	Min	ns
t_{WD}	(1/15) t_{CY} − 50	Min	ns
t_{DR}	(1/10) t_{CY} − 30	Max	ns
t_{RD1}	(2/5) t_{CY} − 200	Max	ns
t_{RD2}	(3/10) t_{CY} − 200	Max	ns
t_{AW}	(1/3) t_{CY} − 150	Min	ns
t_{AD1}	(11/15) t_{CY} − 250	Max	ns
t_{AD2}	(8/15) t_{CY} − 250	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}	(1/10) t_{CY} − 100	Min	ns
t_{PV}	(3/10) t_{CY} − 100	Max	ns
t_{OPRR}	(3/15) t_{CY}	Min	ns
t_{CY}	11 MHz		μs

Timing Waveforms

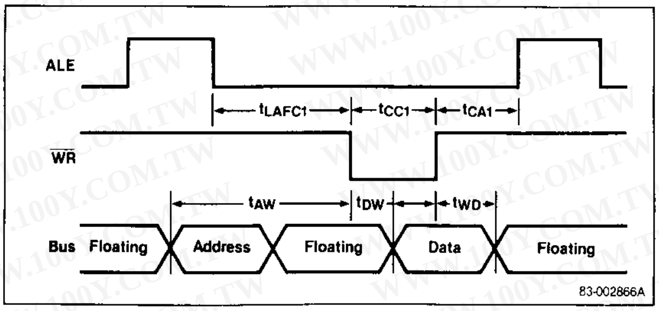
Instruction Fetch from External Memory



Read from External Data Memory

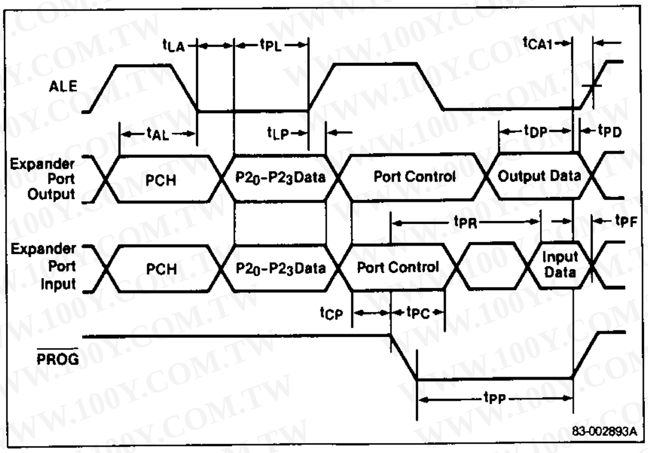


Write to External Memory

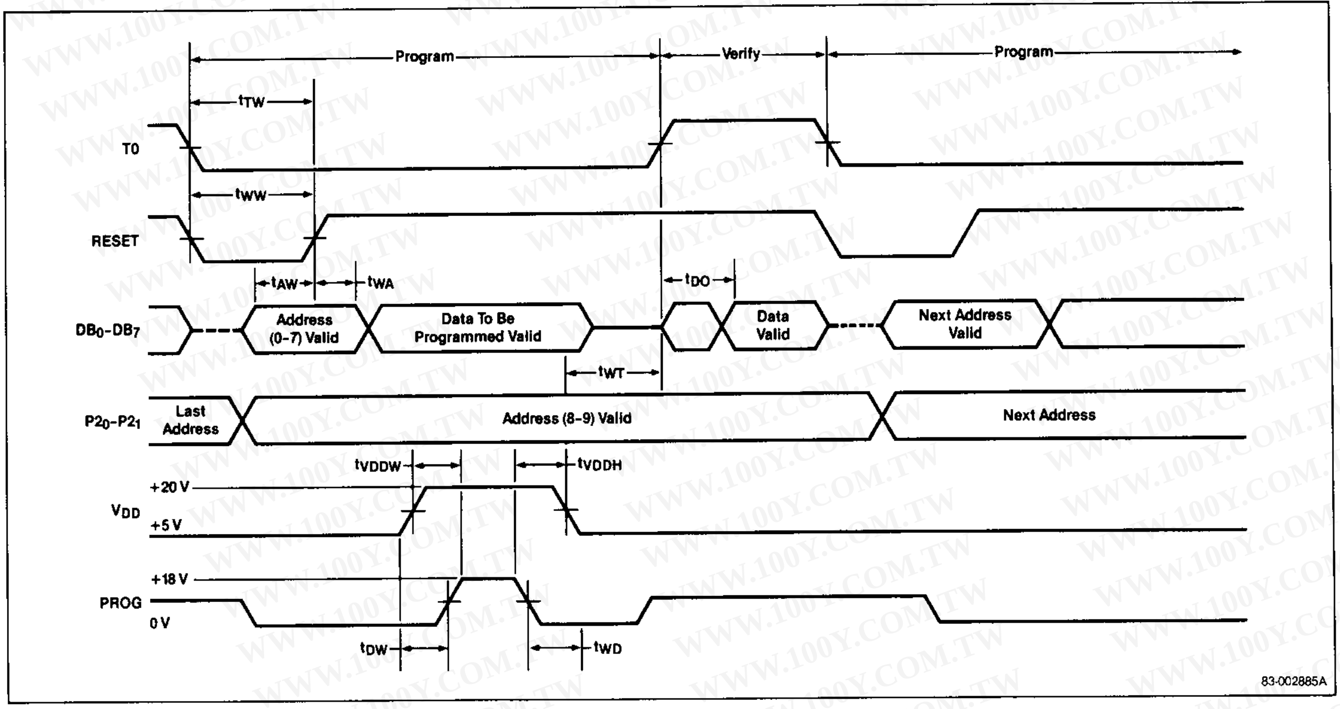


Timing Waveforms (cont)

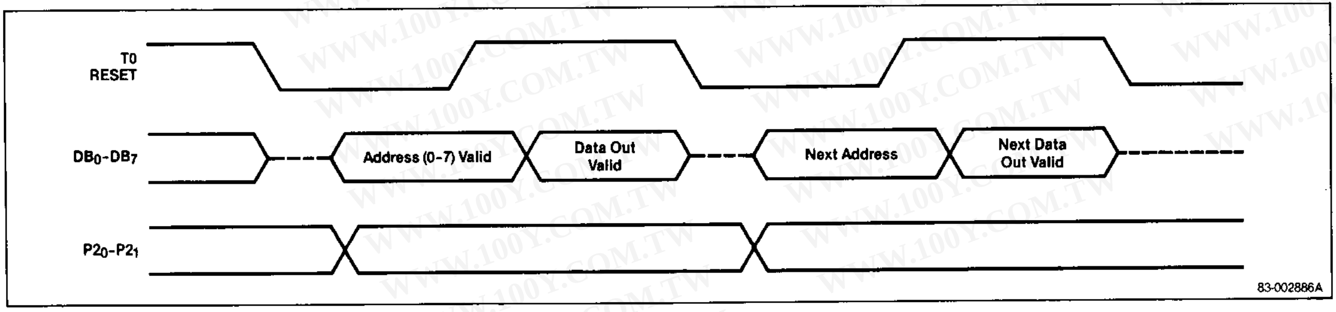
Port 2 Timing



Waveforms for Programming the µPD8749H



Program/Verify Timing (ROM/EPROM)



Instruction Set

			Operation Code										Flags			
Mnemonic	Funciton	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes	C	AC	F0	F1
Accumulator																
ADD 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	●			
ADD A, Rr	(A) ← (A) + (Rr) r = 0–7	Add contents of designated register to the accumulator.	0	1	1	0	1	r	r	r	1	1	●			
ADD A, @ Rr	(A) ← (A) + ((Rr)) 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	●			
ADDC 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	●			
ADDC A, Rr	(A) ← (A) + (C) + (Rr) 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)) 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	●			
ANL 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				
ANL A, Rr	(A) ← (A) AND (Rr) 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)) r = 0–1	Logical AND indirect the contents of data memory with accumulator.	0	1	0	1	0	0	0	r	1	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		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 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				
ORL A, Rr	(A) ← (A) OR (Rr) 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)) r = 0–1	Logical OR indirect the contents of data memory location with accumulator.	0	1	0	0	0	0	0	r	1	1				
RL A	(AN + 1) ← (AN); N = 0–6 (A_0) ← (A_7)	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 (A_0) ← (C) (C) ← (A_7)	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 (A_7) ← (A_0)	Rotate accumulator right by 1 bit without carry.	0	1	1	1	0	1	1	1	1	1				

Instruction Set (cont)

Mnemonic	Function	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
Accumulator (cont)																
RRC A	(AN) ← (AN + 1); N = 0–6 (A_7) ← (C) (C) ← (A_0)	Rotate accumulator right by 1 bit through carry.	0	1	1	0	0	1	1	1	1	1	•			
SWAP A	$(A_4–A_7)$ ← $(A_0–A_3)$	Swap the 2 4-bit nibbles in the accumulator.	0	1	0	0	0	1	1	1	1	1				
XRL A, # 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				
XRL A, Rr	(A) ← (A) XOR (Rr) 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) ← (A) XOR ((Rr)) r = 0–1	Logical XOR indirect the contents of data memory location with accumulator.	1	1	0	1	0	0	0	r	1	1				
Branch																
DJNZ 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				
JBb addr	$(PC_0–PC_7)$ ← 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				
JC addr	$(PC_0–PC_7)$ ← 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				
JF0 addr	$(PC_0–PC_7)$ ← 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				
JF1 addr	$(PC_0–PC_7)$ ← 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				
JMP 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				
JMPP @ A	$(PC_0–PC_7)$ ← ((A))	Jump indirect to specified address with address page.	1	0	1	1	0	0	1	1	2	1				
JNC addr	$(PC_0–PC_7)$ ← 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				
JNI addr	$(PC_0–PC_7)$ ← 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				
JNT0 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				
JNT1 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				
JNZ addr	$(PC_0–PC_7)$ ← 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				
JTF addr	$(PC_0–PC_7)$ ← 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				

Instruction Set (cont)

			Operation Code										Flags			
Mnemonic	Function	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes	C	AC	F0	F1
Branch (cont)																
JT0 addr	$(PC_0–PC_7)$ ← 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				
JT1 addr	$(PC_0–PC_7)$ ← 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				
JZ addr	$(PC_0–PC_7)$ ← addr if A = 0 (PC) ← (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				
Control																
EN I		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				
ENT0 CLK		Enable the clock output pin T0.	0	1	1	1	0	1	0	1	1	1				
SEL MB0	(DBF) ← 0	Select bank 0 (locations 0–2047) of program memory.	1	1	1	0	0	1	0	1	1	1				
SEL MB1	(DBF) ← 1	Select bank 1 (locations 2048–4095) of program memory.	1	1	1	1	0	1	0	1	1	1				
SEL RB0	(BS) ← 0	Select bank 0 (locations 0–7) of data memory.	1	1	0	0	0	1	0	1	1	1				
SEL RB1	(BS) ← 1	Select bank 1 (locations 24–31) of data memory.	1	1	0	1	0	1	0	1	1	1				
Data Moves																
MOV A, # 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				
MOV A, 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				
MOV A, @ 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				
MOV A, PSW	(A) ← (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) ← 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				
MOV Rr, A	(Rr) ← (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)) ← (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)) ← 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				
MOV PSW, A	(PSW) ← (A)	Move contents of accumulator into the program status word.	1	1	0	1	0	1	1	1	1	1				
MOVP A, @ A	$(PC_0–PC_7)$ ← (A) (A) ← ((PC))	Move data in the current page into the accumulator.	1	0	1	0	0	0	1	1	2	1				
MOVP3 A, @ A	$(PC_0–PC_7)$ ← (A) $(PC_8–PC_{10})$ ← 011 (A) ← ((PC))	Move program data in page 3 into the accumulator.	1	1	1	0	0	0	1	1	2	1				

Instruction Set (cont)

			Operation Code										Flags			
Mnemonic	Function	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes	C	AC	F0	F1
Data Moves (cont)																
MOVX 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				
MOVX @ 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				
XCH A, Rr	(A) ↔ (Rr); r = 0–7	Exchange the accumulator and designated register's contents.	0	0	1	0	1	r	r	r	1	1				
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	r	1	1				
XCHD A, @ 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				
Flags																
CPL C	(C) ← NOT (C)	Complement contents of carry bit.	1	0	1	0	0	1	1	1	1	1	•			
CPL F0	(F0) ← NOT (F0)	Complement contents of flag F0.	1	0	0	1	0	1	0	1	1	1			•	
CPL F1	(F1) ← NOT (F1)	Complement contents of flag F1.	1	0	1	1	0	1	0	1	1	1				•
CLR C	(C) ← 0	Clear contents of carry bit to 0.	1	0	0	1	0	1	1	1	1	1	•			
CLR F0	(F0) ← 0	Clear contents of flag 0 to 0.	1	0	0	0	0	1	0	1	1	1			•	
CLR F1	(F1) ← 0	Clear contents of flag 1 to 0.	1	0	1	0	0	1	0	1	1	1				•
Input / Output																
ANL BUS, # data	(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				
ANL Pp, # data	(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				
ANLD 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				
IN A, Pp	(A) ← (Pp); p = 1–2	Input data from designated port (1–2) into accumulator.	0	0	0	0	1	0	p	p	2	1				
INS A, BUS	(A) ← (bus)	Input strobed bus data into accumulator.	0	0	0	0	1	0	0	0	2	1				
MOVD 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				
MOVD 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	1	1				
ORL BUS, # data	(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				
ORLD 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	1	1				
ORL Pp, # data	(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				
OUTL BUS, A	(bus) ← (A)	Output contents of accumulator onto bus.	0	0	0	0	0	0	1	0	1	1				
OUTL Pp, A	(Pp) ← (A); p = 1–2	Output contents of accumulator to designated port (1–2).	0	0	1	1	1	0	p	p	1	1				

Instruction Set (cont)

Mnemonic	Function	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
Registers																
DEC Rr (Rr)	(Rr) ← (Rr) − 1; r = 0–7	Decrement by 1 contents of designated register.	1	1	0	0	1	r	r	r	1	1				
INC Rr	(Rr) ← (Rr) + 1; r = 0–7	Increment by 1 contents of designated register.	0	0	0	1	1	r	r	r	1	1				
INC @ 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				
Subroutine																
CALL addr	((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				
RET	(SP) ← (SP) = 1 (PC) ← ((SP))	Return from subroutine without restoring program status word.	1	0	0	0	0	0	1	1	2	1				
RETR	(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				
Timer / Counter																
EN TCNTI		Enable internal interrupt flag for timer / counter output.	0	0	1	0	0	1	0	1	1	1				
DIS TCNTI		Disable internal interrupt flag for timer / counter output.	0	0	1	1	0	1	0	1	1	1				
MOV A, T	(A) ← (T)	Move contents of timer / counter into accumulator.	0	1	0	0	0	0	1	0	1	1				
MOV T, A	(T) ← (A)	Move contents of accumulator into timer / counter.	0	1	1	0	0	0	1	0	1	1				
STOP TCNT		Stop count for event counter.	0	1	1	0	0	1	0	1	1	1				
STRT CNT		Start count for event counter.	0	1	0	0	0	1	0	1	1	1				
STRT T		Start count for timer.	0	1	0	1	0	1	0	1	1	1				
Miscellaneous																
NOP		No operation performed.	0	0	0	0	0	0	0	0	1	1				

Note:

(1) Operation code designations r and p form the binary representation of the registers and ports involved.

(2) The dot under the appropriate flag bit indicates that its contents are subject to change by the instruction it appears in.

(3) References to the address and data are specified in bytes 2 and/or 1 of the instruction.

(4) Numerical subscripts appearing in the function column reference the specific bits affected.

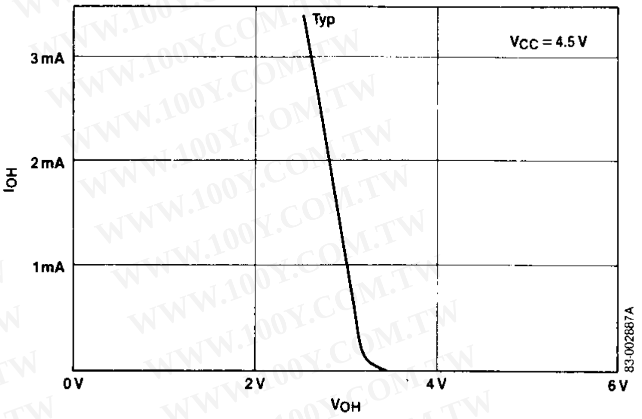
(5) When the bus is written to with an OUTL instruction, the bus remains an output port until either the device is reset or a MOVX instruction is executed.

Instruction Set Symbol Definitions

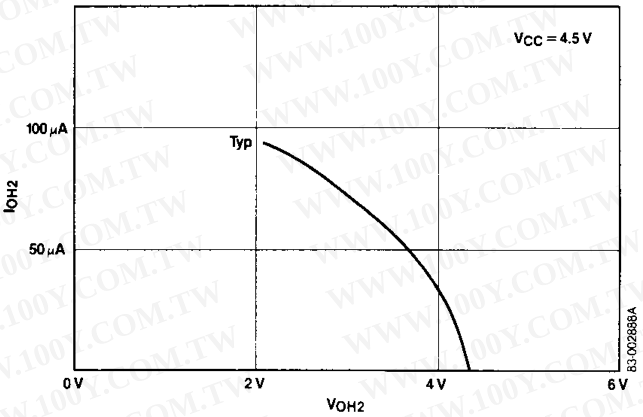
Symbol	Description
A	Accumulator
AC	Auxiliary carry flag
addr	Program memory address (12 bits)
B_b	Bit designator (b=0–7)
BS	Bank switch
BUS	Bus port
C	Carry flag
CLK	Clock signal
CNT	Event counter
D	Nibble designator (4 bits)
data	Number of expression (8 bits)
DBF	Memory bank flip-flop
F0, F1	Flags 0, 1
I	Interrupt
P	"In-page" operation designator
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
AND	Logical product (logical AND)
OR	Logical sum (logical OR)
EXOR	Exclusive-OR

Operating Characteristics

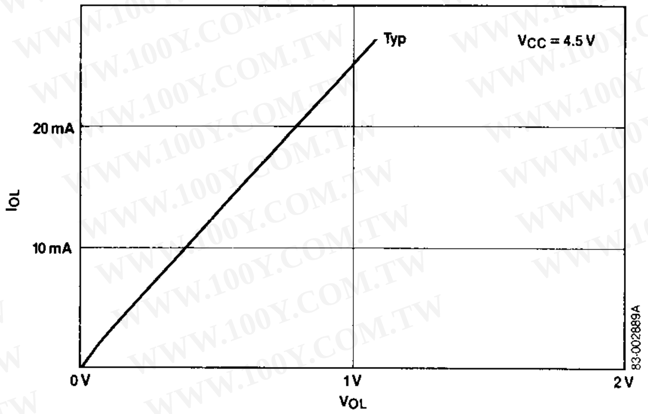
Bus Output High Voltage vs. Source Current



Port P1 & P2 Output High Voltage vs. Source Current



Bus Output Low Voltage vs. Sink Current


Typ
VCC = 4.5 V
20 mA
10 mA
IOL
0 V
1 V
2 V
VOL
83-002889A