NEC

NEC Electronics Inc.

μPD80C39H/49H, μPD49H
HIGH-SPEED, 8-BIT, SINGLE-CHIP CMOS MICROCOMPUTERS

Description

The μPD80C39H, μPD80C49H, and μPD49H are single-chip, 8-bit microcomputers containing an 8-bit CPU, ROM (80C49H and 49H), RAM, I/O ports, and control circuitry. Through CMOS technology, the devices can retain data with low power consumption. In addition, the processor uses two standby modes (HALT and STOP) to further minimize power drain.

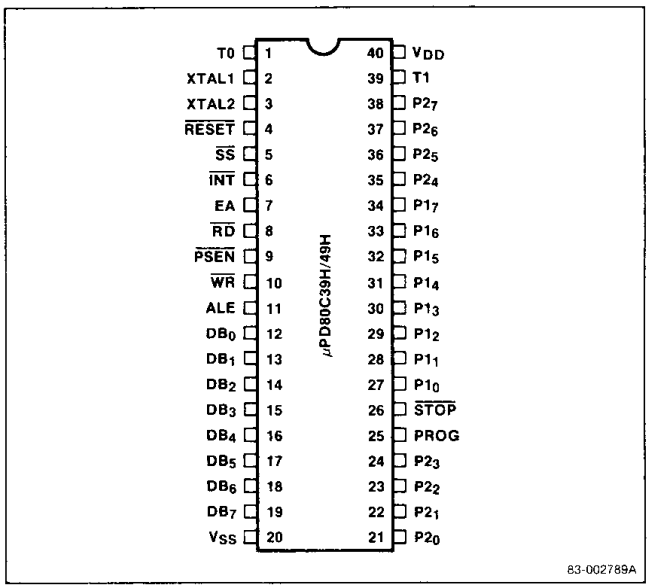
Features

- ☐ 98 instructions
- ☐ 1.25 μs instruction cycle time (12 MHz crystal)
- ☐ Addition, logic, and decimal adjust functions
- ☐ 2K × 8-bit ROM (μPD80C49H and μPD49H)
- ☐ 256 × 8-bit RAM
- ☐ Standby function
- ☐ 8-level stack
- ☐ Two sets of working registers
- ☐ Interrupt capability
- ☐ Two test inputs
- ☐ Internal timer/event counter
- ☐ Input/output ports (8 bits × 2)
  - — Data bus alternative to I/O ports (8 bits × 1)
- ☐ Expandable memory and I/O ports
- ☐ Single-step function
- ☐ Internal clock generator
- ☐ CMOS technology
- ☐ Single power supply of +2.5 to +6.0 V
- ☐ Intel 8049H, 8039H pin compatible

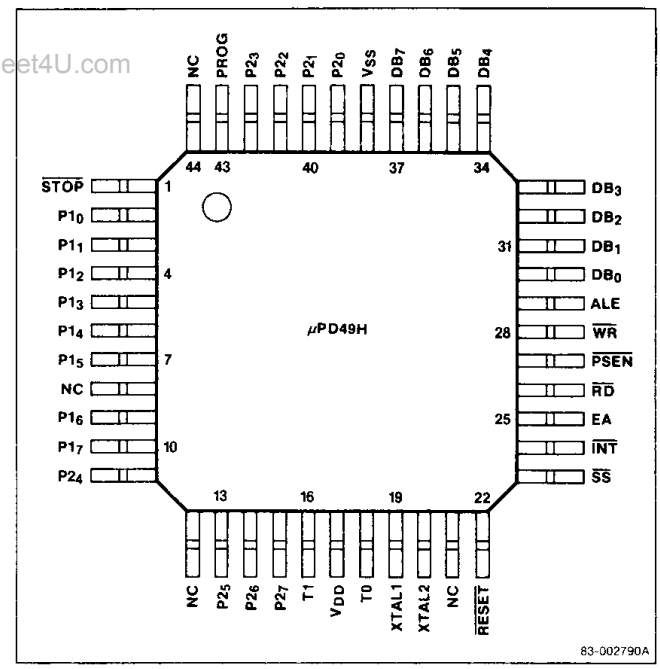
Item	μPD80C49H	μPD80C49
Instructions	98 (STOP instruction added)	97
Instruction Cycle	1.25 μs (12 MHz crystal)	1.875 μs (8 MHz crystal)
Standby Modes	3 (HALT, hardware $\overline{STOP}$, software STOP)	2 (STOP and HALT)
Standby Functions	All standby modes stop at the same timing. The control signal (ALE) stops in the inactive state whether or not internal or external ROM is accessed.	HALT and STOP modes stop at different timing.
Port Options	Type 0: $I_{OH} = -5\ \mu A$; $V_{DD} = 5\ V \pm 10\%$ Type 1: $I_{OH} = -50\ \mu A$; $V_{DD} = 5\ V \pm 10\%$ Type 2: no pullup resistor	Type 0: $I_{OH} = -5\ \mu A$; $V_{DD} = 5\ V \pm 10\%$ Type 1: $I_{OH} = -50\ \mu A$; $V_{DD} = 5\ V \pm 10\%$

Pin Configurations

40-Pin Plastic DIP



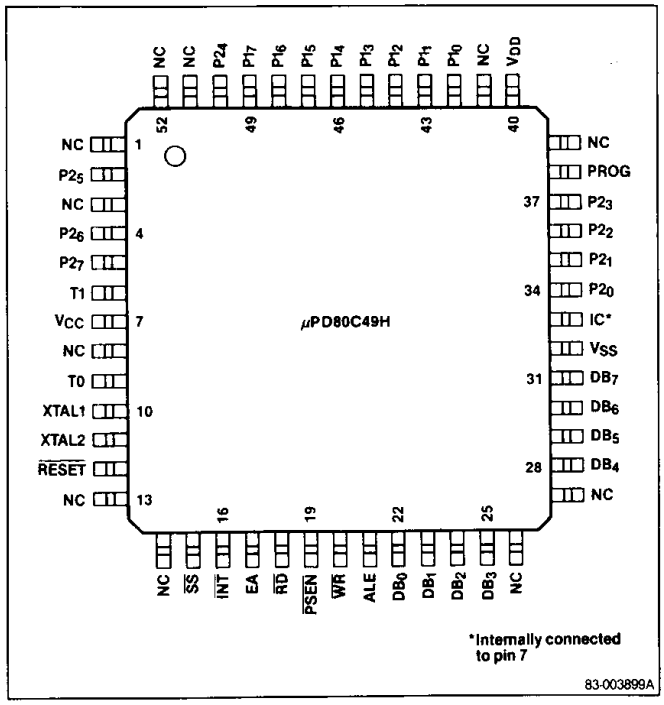
44-Pin Plastic Miniflat



4

Pin Configurations (cont)

52-Pin Plastic Miniflat



Ordering Information

Part Number	Package Type	Max Frequency of Operation	ROM
μPD80C39HC	40-pin plastic DIP	12 MHz	None
μPD80C49HG-00	52-pin plastic miniflat	12 MHz	2K × 8 bits
μPD80C49HC	40-pin plastic DIP	12 MHz	2K × 8 bits
μPD49HG-22	44-pin plastic miniflat	12 MHz	2K × 8 bits

Pin Identification

Symbol	Function
T0	Test 0 input / clock output
XTAL1	Crystal 1 input
XTAL2	Crystal 2 input
$\overline{RESET}$	Reset input
$\overline{SS}$	Single step input
$\overline{INT}$	Interrupt input
EA	External access input
$\overline{RD}$	Read output
$\overline{PSEN}$	Program store enable output
$\overline{WR}$	Write output
ALE	Address latch enable output
DB_0–DB_7	Bidirectional data bus
V_{SS}	Ground
$P2_0$–$P2_7$	Quasi-bidirectional port 2
PROG	Program output
$\overline{STOP}$	Stop input
$P1_0$–$P1_7$	Quasi-bidirectional port 1
T1	Test 1 input
V_{DD}	Power supply
NC	Not connected
IC	Internal connection

Pin Functions

XTAL1, XTAL2 (Crystals 1, 2)

XTAL1 and XTAL2 are the crystal inputs for the internal clock oscillator. XTAL1 is also used as an input for external clock signals.

T0 (Test 0)

The JT0 and JNT0 instructions test the level of T0 and, if it is high, the program address jumps to the specified address. T0 becomes a clock output when the ENT0 CLK instruction is executed.

T1 (Test 1)

The JT1 and JNT1 instructions test the level of T1 and, if it is high, the program address jumps to the specified address. T1 becomes an internal counter input when the STRT CNT instruction is executed.

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

$\overline{\text{RESET}}$ initializes the processor and is also used to verify the internal ROM. $\overline{\text{RESET}}$ determines the oscillation stabilizing time during the release of STOP mode. The $\overline{\text{RESET}}$ pulse width requires at least 5 machine cycles when the supply voltage is within specifications and the oscillation frequency is stable. (Active low).

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

$\overline{\text{SS}}$ causes the processor to execute the program one step at a time. $\overline{\text{SS}}$ also determines the oscillation stabilizing time during the release of the software STOP mode.

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

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

EA disables internal program memory and fetches and accesses external program memory. EA is used for system testing and debugging. (Active high).

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

$\overline{\text{RD}}$ enables a data read from external memory. (Active low).

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

$\overline{\text{WR}}$ enables a data write to external memory.

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

$\overline{\text{PSEN}}$ fetches instructions only from external program memory. (Active low).

ALE (Address Latch Enable)

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

DB_0-DB_7 (Data Bus)

DB_0-DB_7 is a bidirectional port. DB_0-DB_7 reads and writes data using $\overline{\text{RD}}$ and $\overline{\text{WR}}$ for latching. During an external program 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 read and written by $\overline{\text{RD}}$ 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.

$\overline{\text{STOP}}$ (Stop)

$\overline{\text{STOP}}$ controls the hardware STOP mode. $\overline{\text{STOP}}$ stops the oscillator when active low.

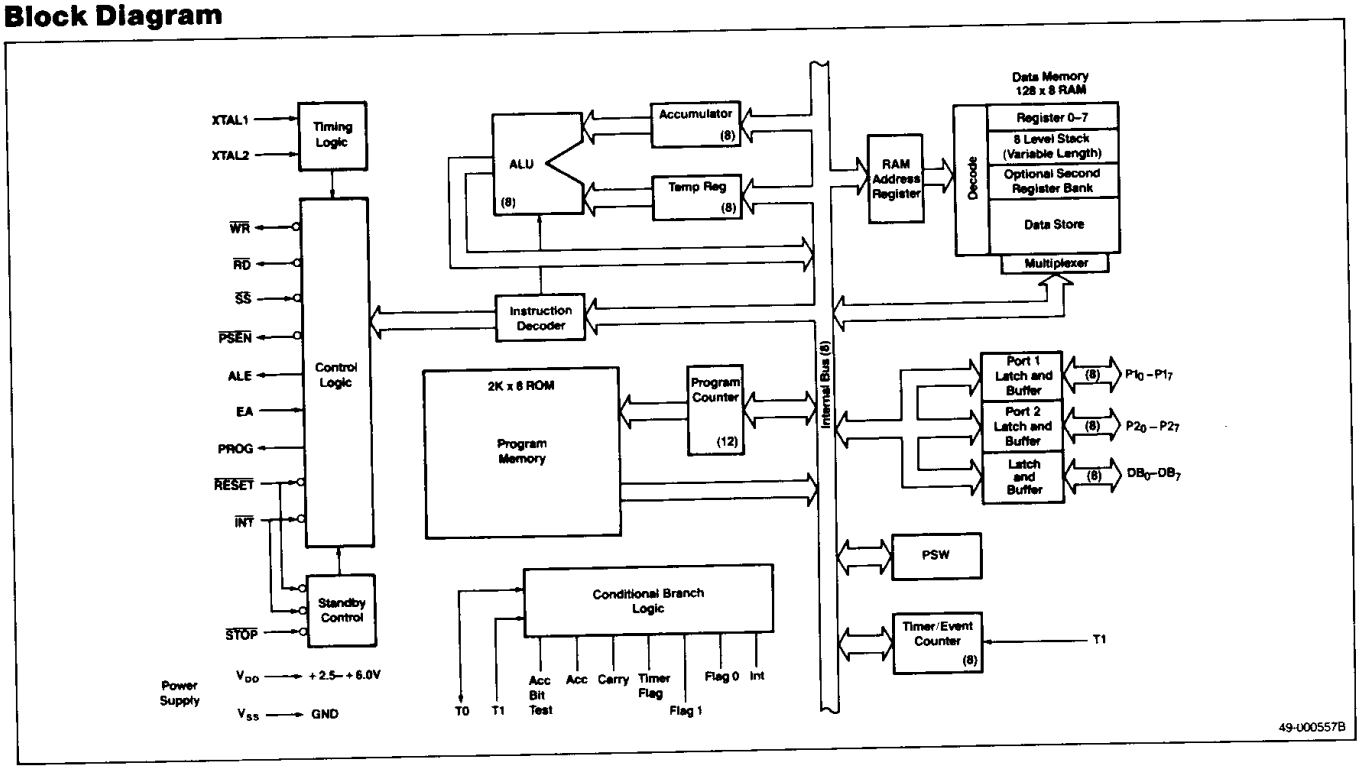
V_{DD} (Power Supply)

V_{DD} is the positive power supply (+2.5 V to +6.0 V).

V_{SS} (Ground)

V_{SS} is ground potential.

Block Diagram



Absolute Maximum Ratings

$T_A = 25°C$

Power supply voltage, V_{DD}	V_{SS} −0.3 V to +7 V
Input voltage, V_I	V_{SS} −0.3 V to V_{DD} +0.3 V
Output voltage, V_O	V_{SS} −0.3 V to V_{DD} +0.3 V
Operating temperature, T_{OPT}	−40°C to +85°C
Storage temperature, T_{STG}	−65°C to +150°C

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

DC Characteristics

Standard Voltage Range

$T_A = -40°C$ to +85°C, $V_{DD} = +5\,V \pm 10\%$, $V_{SS} = 0\,V$

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Input voltage low	V_{IL}	−0.3		+0.8	V	
Input voltage high	V_{IH}	$V_{DD}-2$		V_{DD}	V	Except XTAL1, XTAL2, RESET, SS
	V_{IH1}	$V_{DD}-1$		V_{DD}	V	RESET, XTAL1, XTAL2, SS
Output voltage low	V_{OL}			+0.45	V	$I_{OL} = 2.0$ mA
Output voltage high	V_{OH}	2.4			V	Bus, RD, WR, PSEN, ALE, PROG, T0; $I_{OH} = -400\,\mu A$
	V_{OH1}(1)	2.4			V	$I_{OH} = -5\,\mu A$ (type 0) port 1, port 2
		2.4			V	$I_{OH} = -50\,\mu A$ (type 1) port 1, port 2
	V_{OH2}	$V_{DD}-0.5$			V	All outputs, $I_{OH} = -0.2\,\mu A$
Input current	I_{ILP}(1)		−15	−40	μA	Port 1, port 2; $V_I \leq V_{IL}$ (type 0)
				−500	μA	Port 1, port 2; $V_I \leq V_{IL}$ (type 1)
	I_{ILC}			−40	μA	SS, RESET; $V_I \leq V_{IL}$

DC Characteristics (cont)

Standard Voltage Range (cont)

$T_A = -40°C$ to $+85°C$, $V_{DD} = +5\,V \pm 10\%$, $V_{SS} = 0\,V$

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Input leakage current	I_{LI1}			±1	μA	T1, $\overline{INT}$, $\overline{STOP}$; $V_{SS} \leq V_I \leq V_{DD}$
	I_{LI2}			±3	μA	EA; $V_{SS} \leq V_I \leq V_{DD}$
Output leakage current	I_{LO}			±1	μA	$V_{SS} \leq V_O \leq V_{DD}$ High impedance, bus, T0(3)
Standby current	I_{DD1}(4)		1.5	3.0	mA	$t_{CY} = 1.25\,\mu s$
	I_{DD2}(5)		1	20	μA	(2)
Supply current (total)	I_{DD}		6	18	mA	$t_{CY} = 1.25\,\mu s$
Data retention voltage	V_{DDDR}	2.0			V	At hardware STOP mode ($\overline{STOP}$, $\overline{RESET} \leq 0.4\,V$) or $\overline{RESET}$ ($\overline{RESET} \leq 0.4\,V$)

Extended Voltage Range

$T_A = -40°C$ to $+85°C$, $V_{DD} = +2.5\,V$ to $+6.0\,V$, $V_{SS} = 0\,V$

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Input voltage low	V_{IL}	−0.3		+0.18 V_{DD}	V	
Input voltage high	V_{IH}	0.7 V_{DD}		V_{DD}	V	Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{SS}$
	V_{IH1}	0.8 V_{DD}		V_{DD}	V	$\overline{RESET}$, XTAL1, XTAL2, $\overline{SS}$
Output voltage low	V_{OL}			+0.45	V	$I_{OL} = 1.0\,mA$
Output voltage high	V_{OH}	0.75 V_{DD}			V	Bus, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE, PROG, T0; $I_{OH} = -100\,\mu A$
	V_{OH1}(1)	0.7 V_{DD}			V	$I_{OH} = -1\,\mu A$ (type 0) port 1, port 2
		0.7 V_{DD}			V	$I_{OH} = -10\,\mu A$ (type 1) port 1, port 2
Input current	I_{ILP}(1)		−15	−40	μA	Port 1, port 2; $V_I \leq V_{IL}$ (type 0)
				−500	μA	Port 1, port 2; $V_I \leq V_{IL}$ (type 1)
	I_{ILC}			−40	μA	$\overline{SS}$, $\overline{RESET}$; $V_I \leq V_{IL}$
Input leakage current	I_{LI1}			±1	μA	T1, $\overline{INT}$, $\overline{STOP}$; $V_{SS} \leq V_I \leq V_{DD}$
	I_{LI2}			±5	μA	EA; $V_{SS} \leq V_I \leq V_{DD}$
Output leakage current	I_{LO}			±1	μA	$V_{SS} \leq V_O \leq V_{DD}$ High impedance, bus, T0 (3)

Extended Voltage Range (cont)

$T_A = -40°C$ to $+85°C$, $V_{DD} = +2.5\,V$ to $+6.0\,V$, $V_{SS} = 0\,V$

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Standby current	I_{DD1}(4)		0.3	0.6	mA	$V_{DD} = 3\,V$; $t_{CY} = 5\,\mu s$
			2.0	4.0	mA	$V_{DD} = 6\,V$ $t_{CY} = 1.25\,\mu s$
	I_{DD2}(5)		1	20	μA	(2) ; $V_{DD} = 3\,V$
			1	50	μA	$V_{DD} = 6\,V$
Supply current	I_{DD}		2.0	4.0	mA	$V_{DD} = 3\,V$; $t_{CY} = 5\,\mu s$
			10	20	mA	$V_{DD} = 6\,V$; $t_{CY} = 1.25\,\mu s$

Note:

(1) Types 0, 1, and 2 options can be specified for μPD80C49H. Type 0 for μPD80C39H only.

(2) Input pin voltage is $V_I \leq V_{IL}$ or $V_I \geq V_{IH}$.

(3) Includes port 1 and port 2 pins optionally specified with type 2.

(4) HALT mode.

(5) STOP mode.

AC Characteristics

$T_A = -40°C$ to $+85°C$, $V_{SS} = 0\,V$

Parameter	Symbol	Limits V_{DD} = +5 V ±10% Min	Max	V_{DD} = 2.5 V to 6.0 V Min	Max	Unit	Test Conditions
Cycle time	t_{CY}	1.25	150	5	150	μs	
ALE pulse width	t_{LL}	125		995		ns	(1)
Address setup before ALE	t_{AL}	140		890		ns	(1)
Address hold from ALE	t_{LA}	45		295		ns	(1)
Control pulse width ($\overline{RD}$, $\overline{WR}$)	t_{CC1}	425		2300		ns	(1)
Control pulse width ($\overline{PSEN}$)	t_{CC2}	300		1400		ns	(1)
Data setup before $\overline{WR}$	t_{DW}	340		1965		ns	(1)
Data hold after $\overline{WR}$	t_{WD}	45		295		ns	(2)
Data hold after $\overline{RD}$, $\overline{PSEN}$	t_{DR}	0	95	0	470	ns	(1)
$\overline{RD}$ to data in	t_{RD1}		300		1800	ns	(1)
$\overline{PSEN}$ to data in	t_{RD2}		175		1300	ns	(1)
Address setup before $\overline{WR}$	t_{AW}	350		1850		ns	(1)
Address setup before data in ($\overline{RD}$)	t_{AD1}		700		3585		(1)

4

AC Characteristics (cont)

$T_A = -40°C$ to $+85°C$, $V_{SS} = 0\,V$

Parameter	Symbol	Limits V_{DD} = +5 V ±10% Min	Limits V_{DD} = +5 V ±10% Max	Limits V_{DD} = 2.5 V to 6.0 V Min	Limits V_{DD} = 2.5 V to 6.0 V Max	Unit	Test Conditions
Address setup before data in ($\overline{PSEN}$)	t_{AD2}		500		2750	ns	(1)
Address float to $\overline{RD}$, $\overline{WR}$	t_{AFC1}	105		600		ns	(1)
Address float to $\overline{PSEN}$	t_{AFC2}	5		125		ns	(1)
ALE to control signal ($\overline{RD}$, $\overline{WR}$)	t_{LAFC1}	175		925		ns	(1)
ALE to control signal ($\overline{PSEN}$)	t_{LAFC2}	50		425		ns	(1)
Control signal ($\overline{RD}$, $\overline{WR}$, PROG) to ALE	t_{CA1}	35		285		ns	(1)
Control signal ($\overline{PSEN}$) to ALE	t_{CA2}	280		1285		ns	(1)
Port control setup before falling edge of PROG	t_{CP}	85		460		ns	(3)
Port control hold	t_{PC1}	0	80	0	200	ns	(3, 4)
Port control hold after falling edge of PROG	t_{PC2}	135		1135		ns	(3, 5)
PROG to time P2 input must be valid	t_{PR}		585		2715	ns	(3)
Input data hold time	t_{PF}	0	125	0	500	ns	(3)
Output data setup time	t_{DP}	350		1850		ns	(3)
Output data hold time	t_{PD}	75		450		ns	(3)
PROG pulse width	t_{PP}	625		3250		ns	(3)
Port 2 I/O data setup time	t_{PL}	135		1135		ns	(3)
Port 2 I/O data hold time	t_{LP}	5		125		ns	(3)
ALE to port output	t_{PV}		475		1600	ns	(3)
T0 clock period	t_{OPRR}	250		1000		ns	(3)

Note:

(1) Control output: $C_L = 80\,pF$, bus output: $C_L = 150\,pF$

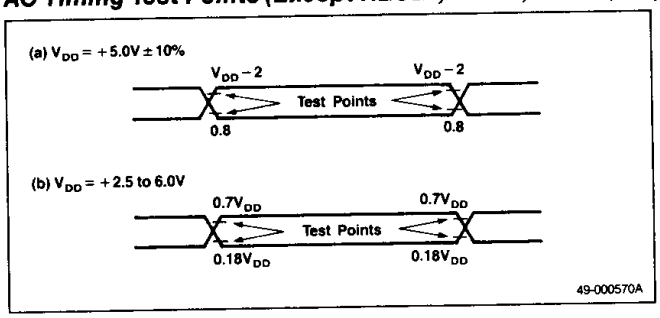
(2) $C_L = 20\,pF$

(3) Control output: $C_L = 80\,pF$

(4) At execution of MOVD A, Pp instruction

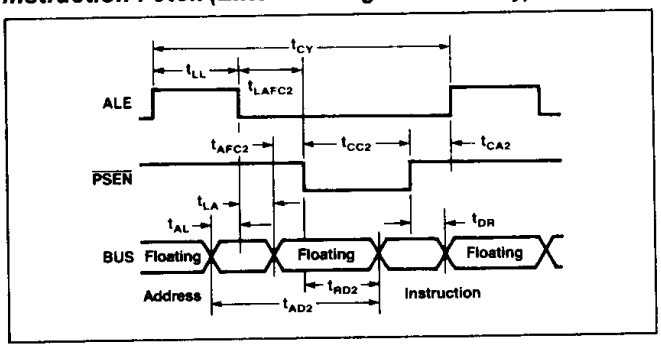
(5) At execution of MOVD Pp, A; ANLD Pp, A; ORLD Pp, A instructions

AC Timing Test Points (Except RESET, XTAL1, XTAL2, SS)

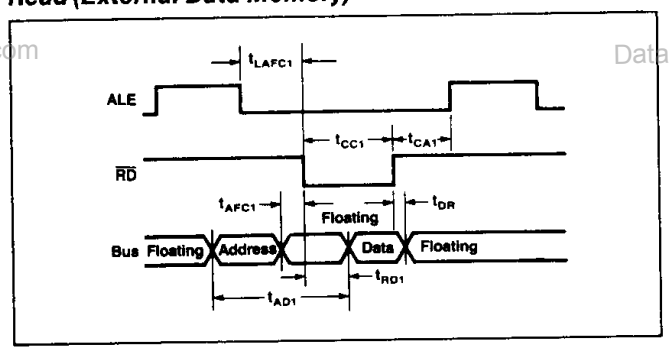


Timing Waveforms

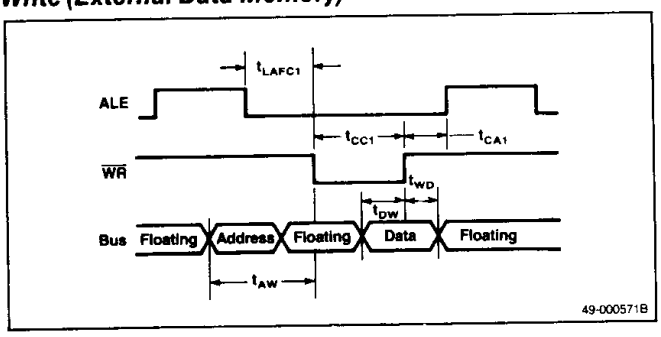
Instruction Fetch (External Program Memory)



Read (External Data Memory)

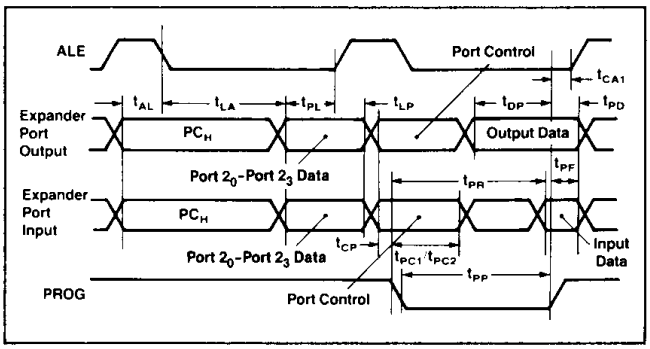


Write (External Data Memory)

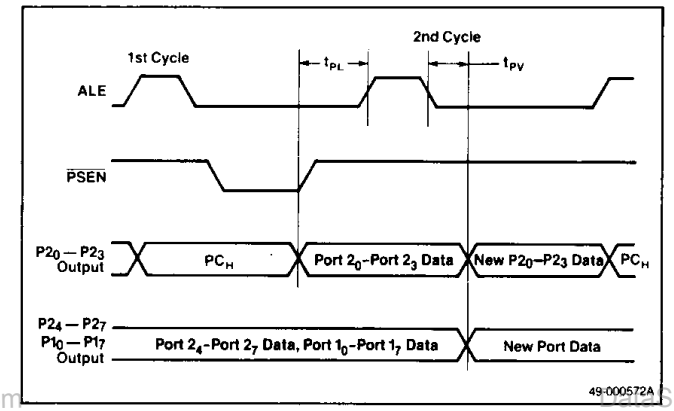


Timing Waveforms (cont)

Port 2 Expansion Timing



I/O Port Timing

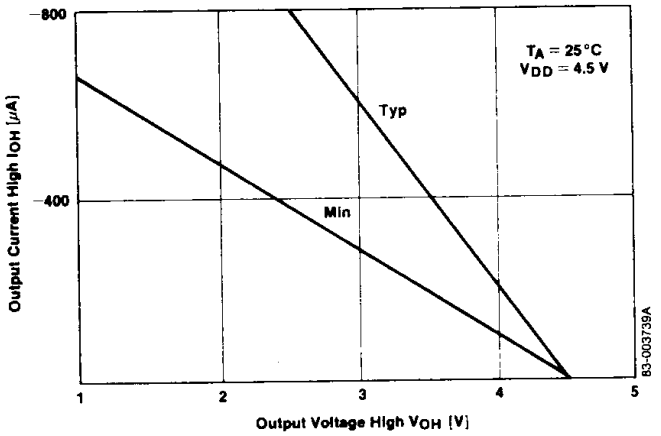


Bus Timing Requirements

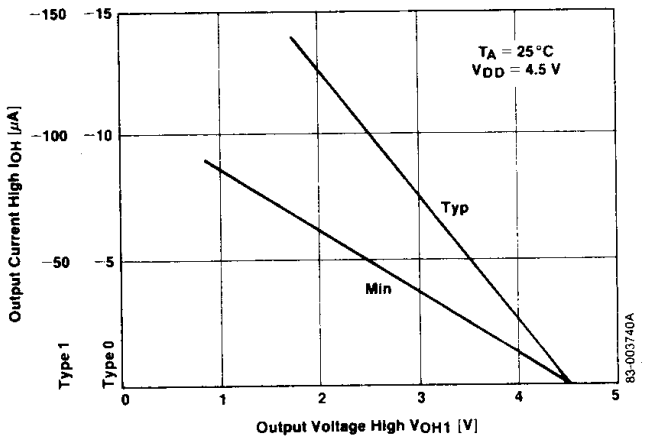
Symbol	Timing Formula	Min/Max	Unit
t_{LL}	(7/30) t_{CY} – 170	Min	ns
t_{AL}	(1/5) t_{CY} – 110	Min	ns
t_{LA}	(1/15) t_{CY} – 40	Min	ns
t_{CC1}	(1/2) t_{CY} – 200	Min	ns
t_{CC2}	(2/5) t_{CY} – 200	Min	ns
t_{DW}	(13/30) t_{CY} – 200	Min	ns
t_{WD}	(1/15) t_{CY} – 40	Min	ns
t_{DR}	(1/10) t_{CY} – 30	Max	ns
t_{RD1}	(2/5) t_{CY} – 200	Max	ns
t_{RD2}	(3/10) t_{CY} – 200	Max	ns
t_{AW}	(2/5) t_{CY} – 150	Min	ns
t_{AD1}	(23/30) t_{CY} – 250	Max	ns
t_{AD2}	(3/5) t_{CY} – 250	Max	ns
t_{AFC1}	(2/15) t_{CY} – 65	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} – 50	Min	ns
t_{CA2}	(4/15) t_{CY} – 50	Min	ns
t_{CP}	(1/10) t_{CY} – 40	Min	ns
t_{PC2}	(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} – 40	Min	ns
t_{PV}	(3/10) t_{CY} + 100	Max	ns
t_{OPRR}	(1/5) t_{CY}	Min	ns
t_{CY}	(1/f_{XTAL}) × 15		μs

Operating Characteristics

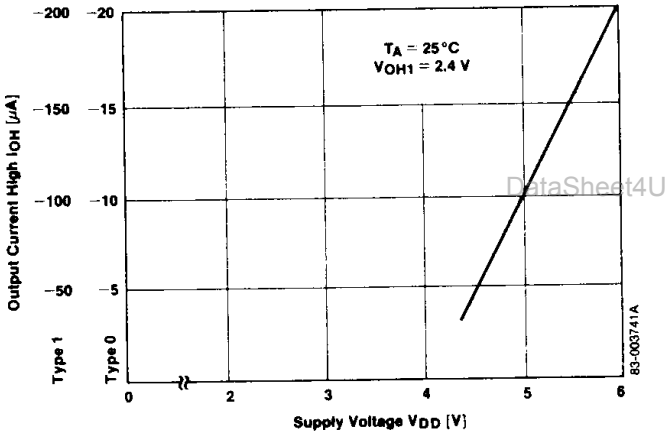
I_{OH} vs V_{OH}



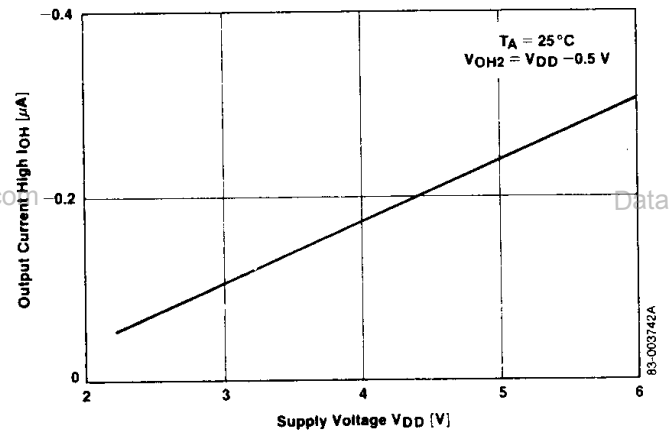
I_{OH} vs V_{OH}



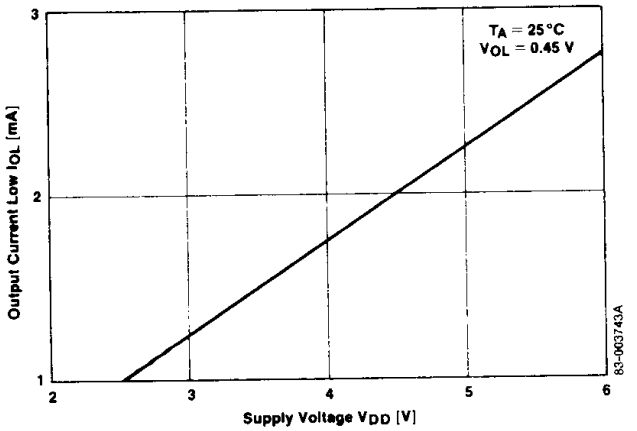
I_{OH} vs V_{DD}



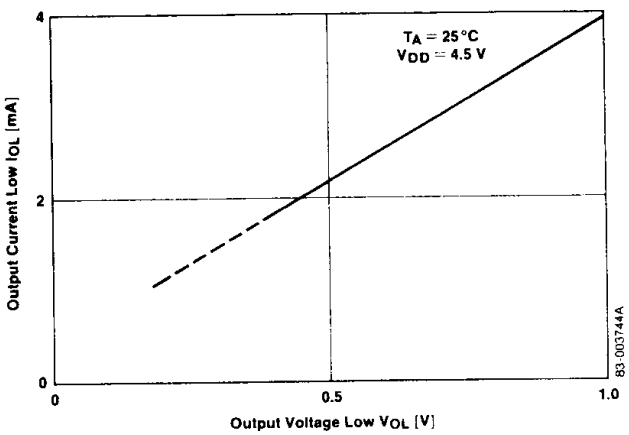
I_{OH} vs V_{DD}



I_{OL} vs V_{DD}

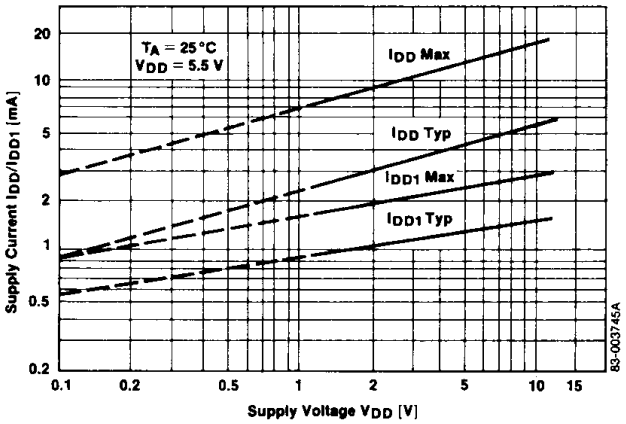


I_{OL} vs V_{OL}



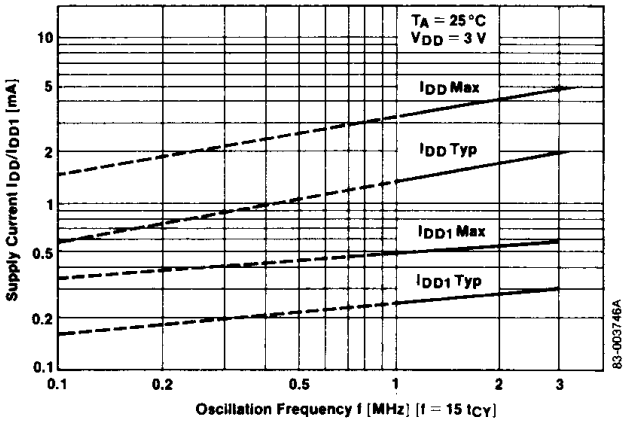
Operating Characteristics (cont)

I_{DD}/I_{DD1} vs f



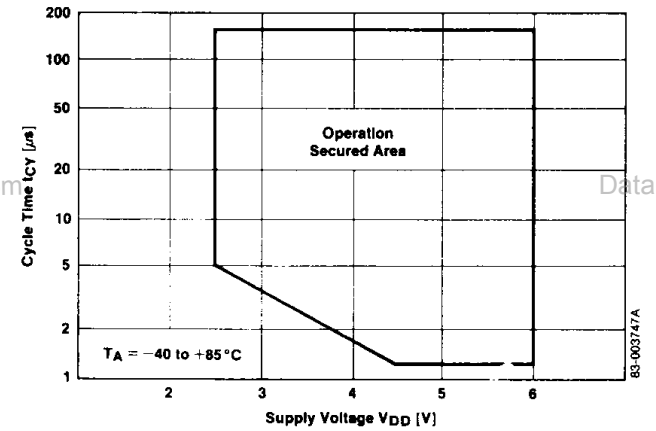
Curves below 1 MHz show characteristics for external oscillation

I_{DD}/I_{DD1} vs f

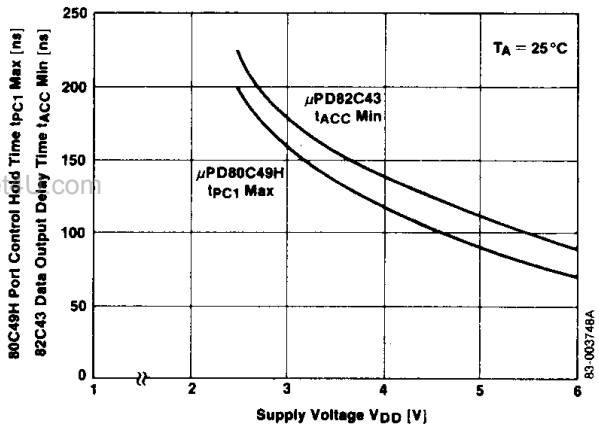


Curves below 1 MHz show characteristics for external oscillation.

t_{CY} vs V_{DD}



t_{PC1} Max [80C49H] and t_{ACC} Min [82C43] vs V_{DD}



Note: Curves without "operation secured area" show reference data.

4

Functional Description

The μPD80C39H/μPD80C49H has the following functional blocks:

Instruction Decoder

The instruction decoder stores the operation code of each instruction and converts it into outputs that control the functions of each block. These outputs control the functions executed by the ALU, data source, and specified registers.

Arithmetic Logic Unit (ALU)

The ALU receives 8-bit data from the accumulator or temporary register and computes an 8-bit result under control of the instruction decoder.

The ALU executes the following functions:

- Add with carry or add without carry
- Logical AND, OR, XOR
- Increment and decrement
- Bit complement
- Rotate left and right
- Swap nibbles
- BCD decimal correction

When a carry results from ALU overflows, the carry bit of the program word is set.

Accumulator

The accumulator is an 8-bit register that stores ALU input data and arithmetic results. It can also be used for transferring data between I/O ports and memory.

Temporary Register

The temporary register is an 8-bit register used for the internal processing necessary with arithmetic operations. The contents of the temporary register are input to the ALU.

Program Counter

The program counter is a 12-bit register that addresses on-chip program memory. The program counter specifies the address of the next instruction to be executed.

Program Memory

The μPD80C49H contains a mask-programmable ROM of 2048 × 8 bits that can be addressed by a program counter. The μPD80C39H has no internal ROM, so it uses external program memory. You can expand internal program memory to 4096 bytes by connecting external program memory. When the contents of the program counter exceed the built-in ROM area, the external program memory will be automatically accessed by DB_0–DB_7, $P2_0$–$P2_3$, and $\overline{PSEN}$.

Data Memory

The μPD80C39H/μPD80C49H has 128 words × 8 bits of data memory that can be externally expanded 256 words maximum when needed.

RAM Address Register

The RAM address register specifies the next address to be accessed in data memory.

Program Status Word

The PSW (figure 1) is an 8-bit status word containing the information shown in table 1.

Figure 1. Program Status Word

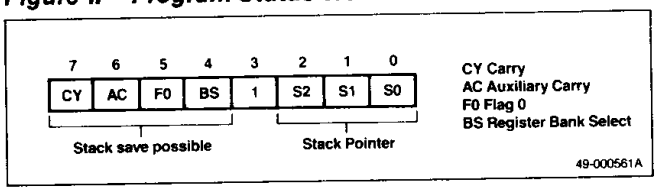


Table 1. PSW Bit Functions

Bits 0–2	Stack pointer bits (S0–S2) A $\overline{RESET}$ clears the stack pointer to 0.
Bit 3	Not used (1).
Bit 4	Working register bank switch bit (BS) 0 = Bank 0 1 = Bank 1
Bit 5	Flag bit (F0). User-controlled bit that can be complemented, cleared, or tested by conditional jump instruction JF0.
Bit 6	Auxiliary Carry (AC) Generated by an auxiliary carry, ADD instruction. Can by used by decimal adjust instruction DA A.
Bit 7	Carry flag (CY) Indicates that an accumulator overflow has taken place with the previously executed instruction.

Conditional Branch Logic

The conditional branch logic is used to test processor conditions. Use a conditional jump instruction to test the conditions shown in table 2.

Control Logic

The control logic generates or receives the signals that control various functions including memory reads and writes, interrupts, software STOP mode, resets, and external memory fetches.

Table 2. Branching Conditions

Test Device	Conditional Jump	
Accumulator	All 0	Not all 0
Accumulator bit	—	1
Carry flag	0	1
User flags (F0, F1)	—	1
Timer overflow flag	—	1
Test inputs (T0, T1)	0	1
Interrupt input (INT)	0	—

Reset Functions

A reset performs the following functions:

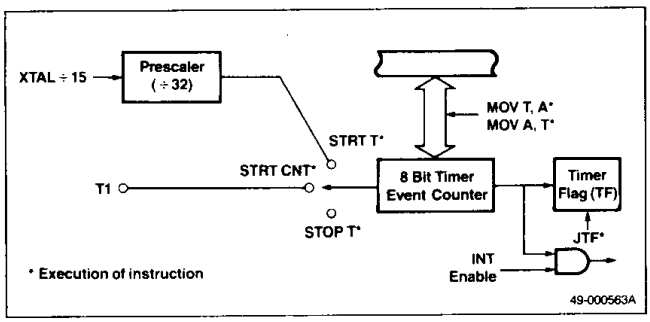
- Clears the program counter and the stack pointer to 0
- Selects register bank and memory bank 0
- Sets the data bus in a high impedance state (except when EA is high)
- Sets ports 1, 2 in input mode
- Disables interrupts (timer and external)
- Stops the timer
- Clears the timer flag, F0, and F1
- Disables the clock output from T0
- Releases HALT and STOP modes

Timer/Event Counter

The timer/event counter can count external events in order to generate a precise time delay. The counter operation is the same in both modes, the only difference is the input source.

The counter is an 8-bit binary up counter (figure 2) that can be reset. It is possible to transfer the contents of the timer to the accumulator and vice-versa by using the MOV A, T and MOV T, A instructions, respectively. The contents of the counter can be independently initialized by the MOV T, A instruction. Use the STRT T instruction to use the counter as a timer and the STRT CNT instruction to use the counter as an event counter.

Figure 2. Timer/Event Counter



Once the counter starts, it continues counting until the program executes a STOP TCNT instruction or $\overline{\text{RESET}}$ becomes active. The counter is incremented up to the maximum count (FFH) and overflows when the count goes from FFH to 00H.

Event Counter. When the T1 pin and counter input are connected by the execution of a STRT CNT instruction, the counter starts counting as an event counter. A change in T1 from high to low causes a count signal which increments the counter by +1. The maximum speed of a count increment is one count per 3 machine cycles. When a 12 MHz crystal is used, the maximum speed is 1 count per 3.75 μs. There is no mimimum speed. After a count signal the T1 input must be held low at least 250 ns (at 12 MHz).

Timer. When an internal clock is connected with the counter input by the execution of the STRT T instruction, the counter starts counting as a timer. When used as a machine cycle clock, ALE is passed through a prescaler which generates an internal clock that increments the timer every 32 machine cycles. The prescaler is reset during the execution of a STRT T instruction. With a 12 MHz crystal, the counter is incremented by +1 at each 25 kHz clock every 40 μs.

You can obtain a delay from 40 μs to 10 ms (256 counts) by presetting the counter and detecting the overflow. To obtain time, through software control, in excess of 10 ms, count overflows in a separate register. To count in steps of 40 μs or less, an external clock can be supplied to the T1 input which causes the counter to operate in the event counter mode. Use the ALE frequency divided by 3 or more for the external clock. Use a software delay loop for fine adjustment of an extremely small or large delay.

Ports 1 and 2 Latch and Buffer

Ports 1 and 2 are 8-bit input/output ports. The data written to the port by an output instruction is latched and output and the data is maintained unless a new output instruction is executed. Input data is not latched, so it is necessary to stabilize input data when reading data by an input instruction.

Several port-loading options are available. At the time you order a mask ROM, (μPD80C49H), you can designate the pullup resistors for port lines $P1_0$-$P1_7$, $P2_0$-$P2_3$, and $P2_4$-$P2_7$.

Three types of pullup resistors are available:

Type 0	(I_{OH} = −5 μA: V_{DD} = +5 V ±10%)
Type 1	(I_{OH} = −50 μA: V_{DD} = +5 V ±10%)
Type 2	No pullup resistor

4

Only type 0 pullup resistors are available with the μPD80C39H.

Timing Logic

The oscillator generates a clock signal that controls all system timing operations. Oscillation is generated by either an external self-oscillating element or external clock input. The oscillator acts as an internal high-gain amplifier for serial resonance. To obtain the oscillation frequency, an external LC network or a crystal or ceramic external resonator may be connected.

As the crystal frequency is lowered, there is an equivalent reduction in series resistance (R). As the temperature of the crystal is lowered, R is increased. Due to this relationship, it becomes difficult to stabilize oscillation where there is low power supply voltage. When V_{CC} is less than 2.7 V and the oscillator frequency is 3 MHz or less, T_A (ambient temperature) should not be less than −10°C.

Standby Control

The standby control circuitry allows low power consumption operation. The standby function operates in 2 modes: HALT and STOP.

HALT Mode

In HALT mode, the oscillation circuit continues to operate but the internal clock stops. The CPU holds all the status of the internal circuits just prior to execution of the HALT instruction. In HALT mode, power consumption is much less than normal.

Setting HALT Mode. HALT mode is set by execution of the HALT instruction and released by either $\overline{INT}$ or $\overline{RESET}$. If interrupts are disabled and $\overline{INT}$ becomes low at a machine cycle right before the HALT instruction and remains low during 2 machine cycles, the HALT instruction byte will be fetched and decoded, but the HALT mode will not be set. Program operation resumes from the instruction following the HALT instruction.

If interrupts are enabled under the same conditions as above, the HALT instruction byte will be fetched and decoded but the HALT mode will not be set and the program will jump to the interrupt start address. After returning from the interrupt routine, the program will continue from the instruction following the HALT instruction.

Releasing HALT Mode. Release HALT mode by activating $\overline{INT}$ or $\overline{RESET}$. When using $\overline{INT}$ to release HALT mode, a low level is present at the $\overline{INT}$ pin and the internal clock is restarted. If interrupts are enabled, the interrupt is executed after the first instruction following the HALT instruction.

In the interrupt enable state, hold the $\overline{INT}$ pin low until the interrupt procedure is started to ensure the interrupt.

When using $\overline{RESET}$ to release HALT mode, a low level is present at the $\overline{RESET}$ pin and the HALT mode is reset and a normal reset operation is executed. When $\overline{RESET}$ goes to a high level, the program starts from address 0.

STOP Mode

In STOP mode, the oscillator stops and only the contents of RAM are maintained. Power consumption is lower than that of the HALT mode. You can set the STOP mode with hardware, by controlling the $\overline{RESET}$ and $\overline{STOP}$ pins; and by software, by executing the corresponding instruction.

Hardware STOP Mode

In hardware STOP mode, the contents of RAM can be held at a voltage as low as +2.0 V.

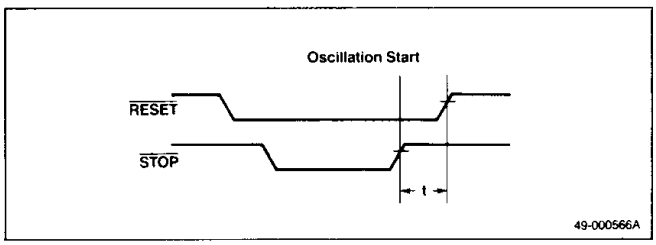
To set hardware STOP mode, set the $\overline{RESET}$ pin to a low level to protect the contents of RAM. Set the $\overline{STOP}$ pin to a low level to stop operation of the oscillation circuit.

To release hardware STOP mode, apply the normal operating level (+2.5 V to +6.0 V) to the power supply at the V_{DD} pin. As figure 3 shows, set the $\overline{STOP}$ pin to a high level while holding the $\overline{RESET}$ pin at a low level. This will restart the oscillation circuit. When $\overline{RESET}$ is set high after oscillation circuit operation is stabilized, the program is started from address 000H. Because the $\overline{STOP}$ pin controls oscillator operation, be careful to protect the $\overline{STOP}$ pin from noise.

When power is turned on, or when STOP mode is released, the oscillation circuit restarts. Because the crystal or ceramic resonator utilizes mechanical vibration, a certain time is required for the oscillation to stabilize. The "t" represents the oscillation stabilizing wait time in the timing waveform.

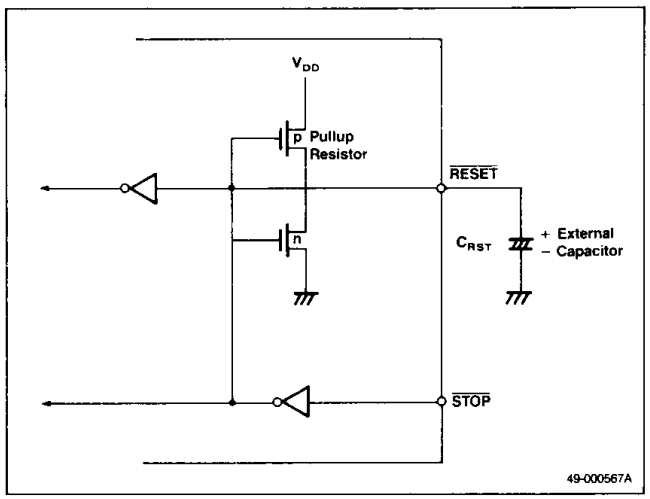
During this wait time, it is necessary to stop instruction execution in order to prevent CPU errors, Therefore, "t" must be longer than the oscillator's stabilizing time.

Figure 3. Oscillator Stop and Start



Oscillation stabilizing time differs somewhat by the type of oscillator used. With a 6 MHz oscillation frequency, a crystal resonator needs several milliseconds to stabilize, while a ceramic resonator needs several hundred microseconds. Figure 4 shows how to easily control the hardware STOP mode by externally connecting a capacitor to the RESET pin. This allows control of the oscillation stabilizing time.

Figure 4. Hardware STOP Mode Control Circuit



Software STOP Mode

In software STOP mode, the oscillation circuits stop, but the CPU maintains all status of internal circuits and data existing just before the STOP instruction. Software STOP mode is the same as when the oscillation circuit stops in HALT mode.

In software STOP mode, if a capacitor (C_{SS}) is connected to the SS pin as shown in figure 5, you can obtain the oscillation stabilizing wait time when releasing STOP mode.

Setting Software STOP Mode. To set software STOP mode, execute the STOP instruction. This sets the internal software STOP mode flip-flop which stops the oscillator and turns transistors A and B off and on, respectively. Capacitor C_{SS} discharges through transistor B causing the SS pin to go low.

Releasing Software STOP Mode. To release software STOP mode, apply an INT or RESET input.

When using the INT input (figure 6), a low at the INT pin resets the software STOP flip-flop and turns transistors A and B on and off, respectively. Then the oscillator restarts, but since SS is still low, program execution remains stopped. With transistor A on, C_{SS} charges and causes SS to go to a high level. Then, program execution restarts. The time it takes for SS to reach the threshold of a logic 1 determines the oscillation stabilizing wait time.

After software STOP mode is released, if interrupts are disabled as in the HALT mode, program execution is resumed from the instruction following the STOP instruction. If interrupts are enabled, the interrupt procedure is initiated (address 003H) after the execution of 1 instruction following the STOP instruction. To assure the interrupt, hold INT at a low level until the interrupt procedure is initiated. Even with short low level timing, the interrupt procedure will be assured if you place a 1-machine cycle instruction after the STOP instruction. However, it is recommended that you hold INT low for at least 2 machine cycles.

When using the RESET input, a low level at the RESET pin resets the software STOP flip-flop. The oscillator starts and the SS pin goes to a high level as C_{SS} is charged. The program starts from address 000H when RESET goes high. Also, since the oscillation stabilizing wait time is generated when SS is low, the RESET pin should be held low longer than the SS pin. When the oscillation stabilizing wait time is obtained by the externally connected capacitor, the value of the capacitor (C_{RST}) connected to the RESET pin (figure 4) should be set at least 3 times larger than that of capacitor C_{SS} connected to the SS pin. For example, if C_{SS} is set to 0.33 μF, C_{RST} should be 1 μF.

When no capacitor is connected to the SS pin, the low level time of the RESET pin should be set to a value larger than the oscillation stabilizing time and SS should be open or pulled up with a 1 k or more resistor.

4

Figure 5. Software STOP Mode Control Circuit

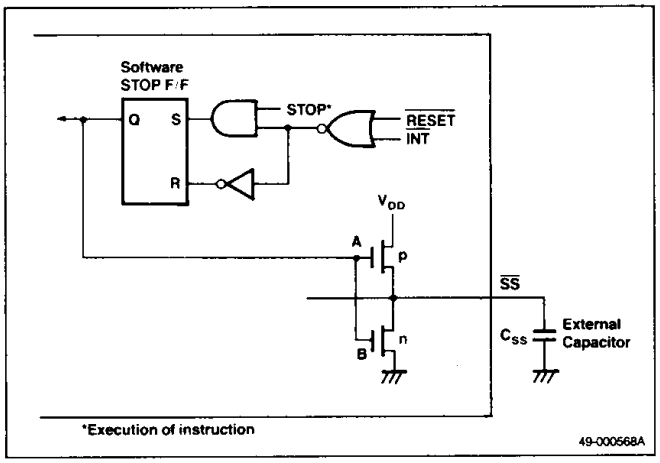
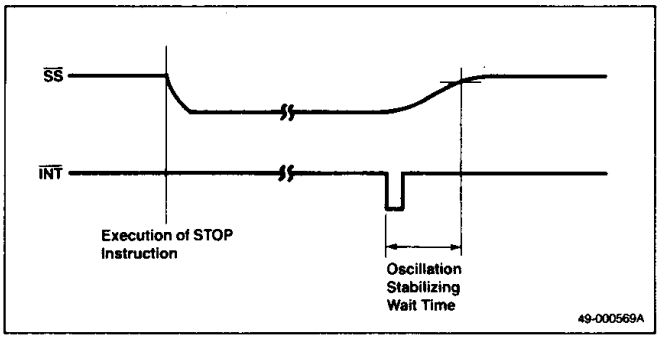


Figure 6. Software STOP Mode Timing



Instruction Set

Mnemonic	Operation	Description	Hex Code	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes
Accumulator													
ADD A, # data	(A) ← (A) + data	Add immediate the specified data to the accumulator.(2)	03	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) for r = 0–7	Add contents of designated register to the accumulator.(2)	6n(4)	0	1	1	0	1	r	r	r	1	1
ADD A, @ Rr	(A) ← (A) + ((Rr)) for r = 0–1	Add indirect the contents the data memory location to the accumulator.(2)	6n(4)	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.(2)	13	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) for r = 0–7	Add with carry the contents of the designated register to the accumulator.(2)	7n(4)	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.(2)	7n(4)	0	1	1	1	0	0	0	r	1	1
ANL A, # data	(A) ← (A) AND data	Logical AND specified immediate data with accumulator.	53	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) for r = 0–7	Logical AND contents of designated register with accumulator.	5n(4)	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.	5n(4)	0	1	0	1	0	0	0	r	1	1
CPL A	(A) ← NOT (A)	Complement the contents of the accumulator.	37	0	0	1	1	0	1	1	1	1	1
CLR A	(A) ← 0	Clear the contents of the accumulator.	27	0	0	1	0	0	1	1	1	1	1
DA A		Decimal adjust the contents of the accumulator.(2)	57	0	1	0	1	0	1	1	1	1	1
DEC A	(A) ← (A) – 1	Decrement by 1 the accumulator's contents.	07	0	0	0	0	0	1	1	1	1	1
INC A	(A) ← (A) + 1	Increment by 1 the accumulator's contents.	17	0	0	0	1	0	1	1	1	1	1
ORL A, # data	(A) ← (A) OR data	Logical OR specified immediate data with accumulator.	43	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) for r = 0–7	Logical OR contents of designated register with accumulator.	4n(4)	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.	4n(4)	0	1	0	0	0	0	0	r	1	1
RL A	$(A_{N+1}) \leftarrow (A_N)$ $(A_0) \leftarrow (A_7)$ for N = 0–6	Rotate accumulator left by 1 bit without carry.	E7	1	1	1	0	0	1	1	1	1	1
RLC A	$(A_{N+1}) \leftarrow (A_N)$; N = 0–6 $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7)$	Rotate accumulator left by 1 bit through carry.	F7	1	1	1	1	0	1	1	1	1	1
RR A	$(A_N) \leftarrow (A_{N+1})$; N = 0–6 $(A_7) \leftarrow (A_0)$	Rotate accumulator right by 1 bit without carry.	77	0	1	1	1	0	1	1	1	1	1

Instruction Set (cont)

Mnemonic	Operation	Description	Hex Code	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes
Accumulator (cont)													
RRC A	$(A_N) \leftarrow (A_{N+1})$; N = 0–6 $(A_7) \leftarrow (C)$ $(C) \leftarrow (A_0)$	Rotate accumulator right by 1 bit through carry.	67	0	1	1	0	0	1	1	1	1	1
SWAP A	$(A_4–A_7) \longleftrightarrow (A_0–A_3)$	Swap the 2 4-bit nibbles in the accumulator.	47	0	1	0	0	0	1	1	1	1	1
XRL A, # data	(A) ← (A) XOR data	Logical XOR specified immediate data with accumulator.	D3	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) for r = 0–7	Logical XOR contents of designated register with accumulator.	Dn(4)	1	1	0	1	1	r	r	r	1	1
XRL A, @ Rr	(A) ← (A) XOR ((Rr)) for r = 0–1	Logical XOR indirect the contents of data memory location with accumulator.	Dn(4)	1	1	0	1	0	0	0	r	1	1
Branch													
DJNZ Rr, addr	(Rr) ← (Rr) – 1; r = 0–7 If (Rr) ≠ 0; $(PC_0–PC_7) \leftarrow$ addr	Decrement the specified register and test contents.	En	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	$(PC_0–PC_7) \leftarrow$ addr if Bb = 1 (PC) ← (PC) + 2 if Bb = 0	Jump to specified address if accumulator bit is set.	x2(6)	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	$(PC_0–PC_7) \leftarrow$ addr if C = 1 (PC) ← (PC) + 2 if C = 0	Jump to specified address if carry flag is set.	F6	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	$(PC_0–PC_7) \leftarrow$ addr if F0 = 1 (PC) ← (PC) + 2 if F0 = 0	Jump to specified address if flag F0 is set.	B6	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	$(PC_0–PC_7) \leftarrow$ addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	Jump to specified address if flag F1 is set.	76	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	$(PC_8–PC_{10}) \leftarrow (addr_8–addr_{10})$ $(PC_0–PC_7) \leftarrow (addr_0–addr_7)$ $(PC_{11}) \leftarrow$ DBF	Direct jump to specified address within the 2K address block.	x4(6)	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	$(PC_0–PC_7) \leftarrow ((A))$	Jump indirect to specified address with address page.	B3	1	0	1	1	0	0	1	1	2	1
JNC addr	$(PC_0–PC_7) \leftarrow$ addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump to specified address if carry flag is low.	E6	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	$(PC_0–PC_7) \leftarrow$ addr if I = 0 (PC) ← (PC) + 2 if I = 1	Jump to specified address if interrupt is low.	86	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	$(PC_0–PC_7) \leftarrow$ addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	Jump to specified address if test 0 is low.	26	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	$(PC_0–PC_7) \leftarrow$ addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	Jump to specified address if test 1 is low.	46	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	$(PC_0–PC_7) \leftarrow$ addr if A ≠ 0 (PC) ← (PC) + 2 if A = 0	Jump to specified address if accumulator is non-zero.	96	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

Instruction Set (cont)

Mnemonic	Operation	Description	Hex Code	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes
Branch (cont)													
JTF addr	$(PC_0\text{–}PC_7) \leftarrow$ addr if TF = 1 $(PC) \leftarrow (PC) + 2$ if TF = 0	Jump to specified address if timer flag is set to 1.	16	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
JT0 addr	$(PC_0\text{–}PC_7) \leftarrow$ addr if T0 = 1 $(PC) \leftarrow (PC) + 2$ if T0 = 0	Jump to specified address if test 0 is a 1.	36	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	$(PC_0\text{–}PC_7) \leftarrow$ addr if T1 = 1 $(PC) \leftarrow (PC) + 2$ if T1 = 0	Jump to specified address if test 1 is a 1.	56	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	$(PC_0\text{–}PC_7) \leftarrow$ addr if A = 0 $(PC) \leftarrow (PC) + 2$ if A = 1	Jump to specified address if accumulator is 0.	C6	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 interrput input.	05	0	0	0	0	0	1	0	1	1	1
DIS I		Disable the external interrupt input.	15	0	0	0	1	0	1	0	1	1	1
ENT0 CLK		Enable the clock output pin T0.	75	0	1	1	1	0	1	0	1	1	1
SEL MB0	(DBF) ← 0	Select bank 0 (locations 0–2047) of program memory.	E5	1	1	1	0	0	1	0	1	1	1
SEL MB1	(DBF) ← 1	Select bank 1 (locations 2048–4095) of program memory.	F5	1	1	1	1	0	1	0	1	1	1
SEL RB0	(BS) ← 0	Select bank 0 (locations 0–7) of data memory.	C5	1	1	0	0	0	1	0	1	1	1
SEL RB1	(BS) ← 1	Select bank 1 (locations 24–31) of data memory.	D5	1	1	0	1	0	1	0	1	1	1
HALT		Initiates halt mode.	01	0	0	0	0	0	0	0	1	1	1
STOP		Sets CPU to software stop mode.	82	1	0	0	0	0	0	1	0	1	1
Data Moves													
MOV A, # data	(A) ← data	Move immediate the specified data into the accumulator.	23	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.	Fn(4)	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.	Fn(4)	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.	C7	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.	Bn(4)	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.	An(4)	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.	An(4)	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.	Bn(4)	1 d_7	0 d_6	1 d_5	1 d_4	0 d_3	0 d_2	0 d_1	r d_0	2	2
MOV PSW, A	(PSW) ← (A)	Move contents of accumulator into the program status word.	D7	1	1	0	1	0	1	1	1	1	1

Instruction Set (cont)

Mnemonic	Operation	Description	Hex Code	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes
Data Moves (cont)													
MOVP A, @ A	$(PC_0\text{–}PC_7) \leftarrow (A)$ $(A) \leftarrow ((PC))$	Move data in the current page into the accumulator.	A3	1	0	1	0	0	0	1	1	2	1
MOVP3 A, @ A	$(PC_0\text{–}PC_7) \leftarrow (A)$ $(PC_8\text{–}PC_{11}) \leftarrow 0011$ $(A) \leftarrow ((PC))$	Move program data in page 3 into the accumulator.	E3	1	1	1	0	0	0	1	1	2	1
MOVX A, @ R	$(A) \leftarrow ((Rr))$; r = 0–1	Move indirect the contents of external data memory into the accumulator.	8n(4)	1	0	0	0	0	0	0	r	2	1
MOVX @ R, A	$((Rr)) \leftarrow (A)$; r = 0–1	Move indirect the contents of the accumulator into external data memory.	9n(4)	1	0	0	1	0	0	0	r	2	1
XCH A, Rr	$(A) \longleftrightarrow (Rr)$; r = 0–7	Exchange the accumulator and designated register's contents.	2n(4)	0	0	1	0	1	r	r	r	1	1
XCH A, @ Rr	$(A) \longleftrightarrow ((Rr))$; r = 0–1	Exchange indirect contents of accumulator and location in data memory.	2n(4)	0	0	1	0	0	0	0	r	1	1
XCHD A, @ Rr	$(A_0\text{–}A_3) \longleftrightarrow ((Rr))_0\text{–}((Rr))_3$; r = 0–1	Exchange indirect 4-bit contents of accumulator and data memory.	3n(4)	0	0	1	1	0	0	0	r	1	1
Flags													
CPL C	$(C) \leftarrow$ NOT (C)	Complement contents of carry bit.	A7	1	0	1	0	0	1	1	1	1	1
CPL F0	$(F0) \leftarrow$ NOT (F0)	Complement contents of flag F0.	95	1	0	0	1	0	1	0	1	1	1
CPL F1	$(F1) \leftarrow$ NOT (F1)	Complement contents of flag F1.	B5	1	0	1	1	0	1	0	1	1	1
CLR C	$(C) \leftarrow 0$	Clear contents of carry bit to 0.	97	1	0	0	1	0	1	1	1	1	1
CLR F0	$(F0) \leftarrow 0$	Clear contents of flag 0 to 0.	85	1	0	0	0	0	1	0	1	1	1
CLR F1	$(F1) \leftarrow 0$	Clear contents of flag 1 to 0.	A5	1	0	1	0	0	1	0	1	1	1
Input / Output													
ANL BUS, # data	$(bus) \leftarrow (bus)$ AND data	Logical AND immediate specified data with contents of bus.	98	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	$(Pp) \leftarrow (Pp)$ AND data p = 1–2	Logical AND immediate specified data with designated port (1 or 2).	9n(5)	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	$(Pp) \leftarrow (Pp)$ AND $(A_0\text{–}A_3)$; p = 4–7	Logical AND contents of accumulator with designated port (4–7).	9n(5)	1	0	0	1	1	1	p	p	2	1
IN A, Pp	$(A) \leftarrow (Pp)$; p = 1–2	Input data from designated port (1–2) into accumulator.	0n(5)	0	0	0	0	1	0	p	p	2	1
INS A, BUS	$(A) \leftarrow (bus)$	Input strobed bus data into accumulator.	08	0	0	0	0	1	0	0	0	2	1
MOVD A, Pp	$(A_0\text{–}A_3) \leftarrow (Pp)$; p = 4–7 $(A_4\text{–}A_7) \leftarrow 0$	Move contents of designated port (4–7) into accumulator.	0n(5)	0	0	0	0	1	1	p	p	2	1

Instruction Set (cont)

Mnemonic	Operation	Description	Hex Code	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes
Input / Output (cont)													
MOVD Pp, A	(Pp) ← (A_0–A_3); p = 4–7	Move contents of accumulator to designated port (4–7).	3n(5)	0	0	1	1	1	1	p	p	2	1
ORL BUS, # data	(bus) ← (bus) OR data	Logical OR immediate specified data with contents of bus.	88	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).	8n(5)	1	0	0	0	1	1	p	p	2	1
ORL Pp, # data	(Pp) ← (Pp) OR data p = 1–2	Logical OR immediate specified data with designated port (1–2).	9n(5)	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.	02	0	0	0	0	0	0	1	0	2	1
OUTL Pp,A	(Pp) ← (A); p = 1–2	Output contents of accumulator to designated port (1–2).	3n(5)	0	0	1	1	1	0	p	p	2	1
Registers													
DEC Rr	(Rr) ← (Rr) – 1; r = 0–7	Decrement by 1 contents of designated register.	Cn(4)	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.	1n(4)	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.	1n(4)	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.	x4(6)	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.	83	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.	93	1	0	0	1	0	0	1	1	2	1
Timer / Counter													
EN TCNTI		Enable internal interrupt flag for timer / counter output.	25	0	0	1	0	0	1	0	1	1	1
DIS TCNTI		Disable internal interrupt flag for timer / counter output.	35	0	0	1	1	0	1	0	1	1	1
MOV A, T	(A) ← (T)	Move contents of timer / counter into accumulator.	42	0	1	0	0	0	0	1	0	1	1
MOV T, A	(T) ← (A)	Move contents of accumulator into timer / counter.	62	0	1	1	0	0	0	1	0	1	1
STOP TCNT		Stop count for event counter.	65	0	1	1	0	0	1	0	1	1	1
STRT CNT		Start count for event counter.	45	0	1	0	0	0	1	0	1	1	1
STRT T		Start count for timer.	55	0	1	0	1	0	1	0	1	1	1

Instruction Set (cont)

Mnemonic	Operation	Description	Hex Code	Operation Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Cycles	Bytes
Miscellaneous													
NOP		No operation performed.	00	0	0	0	0	0	0	0	0	1	1

Note:

(1) Binary instruction code designations r and p represent encoded values or the lowest-order bit value of specified registers and ports, respectively.

(2) Execution of the ADD, ADDC, and DA instructions affect the carry flags, which are not shown in the respective function equations. These instructions set the carry flags when there is an overflow in the accumulator (the auxiliary carry flag is set when there is an overflow of bit 3 of the accumulator) and clear the carry flags when there is no overflow. Flags that are specifically addressed by flag instructions are shown in the function equations for those instructions.

(3) References to addresses and data are specified in byte 1 and/or 2 in the opcode of the corresponding instruction.

(4) The hex value of n for specific registers is as follows:

a) Direct addressing

R0: n = 8 R2: n = A R4: n = C R6: n = E
R1: n = 9 R3: n = B R5: n = D R7: n = F

b) Indirect addressing

@ R0: n = 0 @ R1: n = 1

(5) The hex value of n for specific ports is as follows:

P1: n = 9 P4: n = C P6: n = E
P2: n = A P5: n = D P7: n = E

(6) The hex value of x for specific accumulator or address bits is as follows:

a) JBb instruction

B_0: x = 1 B_2: x = 5 B_4: x = 9 B_6: x = D
B_1: x = 3 B_3: x = 7 B_5: x = B B_7: x = F

b) JMP instruction

Page 0: x = 0 Page 2: x = 4 Page 4: x = 8 Page 6: x = C
Page 1: x = 2 Page 3: x = 6 Page 5: x = A Page 7: x = E

c) CALL instruction

Page 0: x = 1 Page 2: x = 5 Page 4: x = 9 Page 6: x = D
Page 1: x = 3 Page 3: x = 7 Page 5: x = B Page 7: x = F

Symbol Definitions

Symbol	Description
A	Accumulator
AC	Auxiliary carry flag
addr	Program memory address (a_0–a_7) or (a_0–a_{10})
b	Accumulator bit (b = 0–7)
BS	Bank switch
BUS	Bus port
C	Carry flag
CLK	Clock signal
CNT	Event counter
data	Number or expression (8 bits)
DBF	Memory bank flip-flop
F0, F1	Flags 0, 1
$\overline{INT}$	Interrupt
n	Indicates the hex number of the specified register or port
PC	Program counter
Pp	Port designator (p = 1, 2 or 4–7)
PSW	Program status word
Rr	Register designator (r = 0–7)

Symbol	Description
SP	Stack pointer
T	Timer
TF	Timer flag
T0, T1	Testable flags 0, 1
#	Prefix for immediate data
@	Prefix for indirect address
x	Indicates the hex number corresponding to the accumulator bit or page number specified in the operand
(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)
XOR	Exclusive-OR
———	Complement

4