OKI semiconductor

MSM80C35/48
MSM80C39/49
MSM80C40/50

CMOS 8-BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

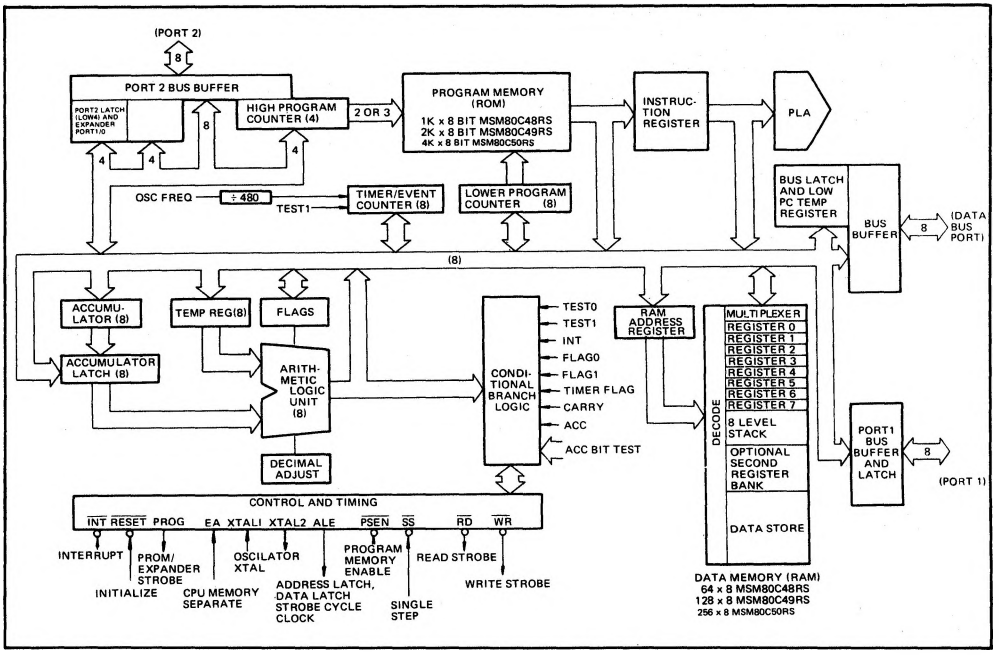
The OKI MSM80C48/MSM80C49/MSM80C50 microcontroller is a low-power, high-performance 8-bit single chip device implemented in silicon gate complementary metal oxide semiconductor technology. Integrated within these chips are 8K/16K/32K bits of mask program ROM, 512/1024/2048 bits of data RAM, 27 I/O lines, built-in 8 bit timer/counter, and oscillator. Program memory and data paths are byte wide. Eleven new instructions have been added to the NMOS version's instruction set, thereby optimizing power down, port data transfer, decrement and port float functions.

Available in 40-pin plastic DIP (RS) or 44-pin plastic flat packages (GSK).

FEATURES

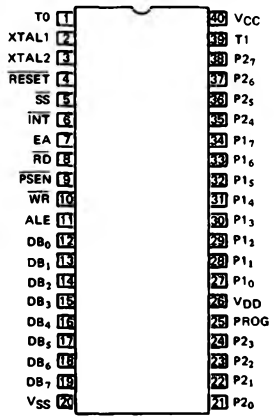
- Lower power consumption enabled by CMOS silicon gate process
- Completely static operation
- Improved power-down feature
- Minimum instruction cycle 1.36 μs (11MHz) @ $V_{CC} = +5V \pm 10\%$
  11 MHz version of MSM80C40/50 (6 MHz < XTAL1.2 < 11 MHz) is under development.
- Every signal input terminal is provided with a Schmitt circuit, except XTAL1 Pin.
- Every signal output terminal is capable of driving a standard TTL, except XTAL2 Pin.
- 111 instructions
- All instructions are usable even during execution of external ROM instructions.
- Operation facility
  Addition, logical operations, and decimal adjust
- Program memory (ROM) : 1K × 8 bits (MSM80C48)
  : 2K × 8 bits (MSM80C49)
  : 4K × 8 bits (MSM80C50)
- Data memory (RAM) : 64 × 8 bits (MSM80C48)
  : 128 × 8 bits (MSM80C49)
  : 256 × 8 bits (MSM80C50)
- Two sets of working registers
- External and timer interrupts
- Two test inputs
- Built-in 8-bit timer counter
- Extendable external memory and I/O ports
- Input/output ports : Input/output ports – 8 bits × 2
  : Data bus input/output – 8 bits × 1
- Single-step execution function
- Every signal input terminal is provided with a Schmitt circuit, except XTAL 2 Pin
- Every signal output terminal is capable of driving a standard TTL, except X'tal 2 Pin.
- Wide range of operating voltage, from +2.5V to +6V of V_{CC}.
- High noise margin action
- Two kinds of package; 40-pin plastic DIP and 44-pin plastic flat package
- Compatible with Intel's 8048, 8049 and 8050

FUNCTIONAL BLOCK DIAGRAM

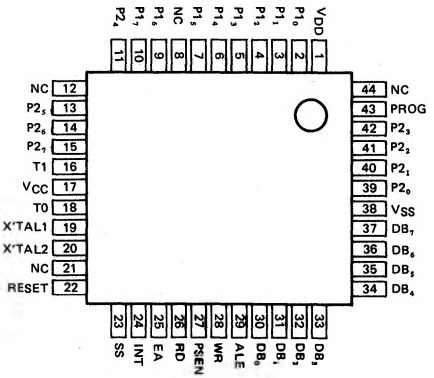


PIN CONFIGURATION

(Top View) 40 Lead Plastic DIP



(Top View) 44 Lead Plastic FLAT



Pin Name

$P1_0 \sim P1_7$	: Input/output port (PORT1)	ALE	: Address Latch Enable
$P2_0 \sim P2_7$	: Input/output port (PORT2)	$\overline{PSEN}$	: Program Store Enable
$DB_0 \sim DB_7$	: Data bus port (BUS PORT)	$\overline{RESET}$	: Reset
T0, T1	: Test	$\overline{SS}$	: Single Step
$\overline{INT}$	: Interrupt	EA	: ROM Mode
$\overline{RD}$	: Read	XTAL 1, 2	: Crystal Controlled Oscillator
$\overline{WR}$	: Write		

PIN DESCRIPTION

Designation	Input/Output	Function
$P1_0 \sim P1_7$ (PORT 1)	Input/Output	8-bit quasi-bidirectional port
$P2_0 \sim P2_7$ (PORT 2)	Input/Output	8-bit quasi-bidirectional port The high-order four bits of external program memory addresses can be output from P20-P23, to which the I/O expander MSM82C43RS may also be connected.
$DB_0 \sim DB_7$ (BUS)	Input/Output	Bidirectional port The low-order eight bits of external program memory address can be output from this port, and the addressed instruction is fetched under the control of $\overline{PSEN}$ signal. Also, the external data memory address is output, and data is read and written synchronously using $\overline{RD}$ and $\overline{WR}$ signals. The port can also serve as either a statically latched output port or a non-latching input port.
T0 (Test 0)	Input/Output	The input can be tested with the conditional jump instructions JT0 and JNT0. The execution of the ENT0 CLK instruction causes a clock output to be generated.
T1	Input	The input can be tested with the conditional jump instructions JT1 and JNT1. The execution of a STRT CNT instruction causes an internal counter input to be activated.
$\overline{INT}$ (Interrupt)	Input	Interrupt input. If interrupt is enabled, $\overline{INT}$ input initiates an interrupt. Interrupt is disabled after a reset. Also testable with a JNI instruction. Can be used to terminate the power-down mode. (Active "0" level)
$\overline{RD}$ (Read)	Output	A signal to read data from external data memory. (Active "0" level)
$\overline{WR}$ (Write)		A signal to write data to external data memory. (Active "0" level)
ALE Address & Data Latch Clock		This signal is generated in each cycle. It may be used as a clock output. External data memory or external program memory is addressed upon the falling edge. For the external ROM, this signal is used to latch the bus port data upon the ALE signal rise-up after the execution of the OUTL BUS, A instruction.
$\overline{PSEN}$ Program Store Enable	Output	A signal to fetch an instruction from external program memory (Active "0" level)
$\overline{RESET}$	Output	($\overline{RESET}$) input initialize the processor. (Active "0" level) Used to terminate the power-down mode.
$\overline{SS}$ (Single Step)	Output	A program is executed step by step. This pin can also be used to control internal oscillation when the power-down mode is reset. (Active "0" level)
EA (External Access)	Input	When held at high level, all instructions are fetched from external memory. (Active "1" level)
$\overline{PROG}$ (Expander Strobe)	Output	This output strobes the MSM82C43RS I/O expander.

PIN DESCRIPTION (CONT.)

Designation	Input/Output	Function
XTAL 1 (Crystal 1)	Input	One side of the crystal input for the internal oscillator. An external source can also be input.
XTAL 2 (Crystal 2)	Output	Other side of Crystal input for internal oscillator.
V_{CC}	–	Power supply terminal
V_{DD}	–	Standby control input. Normally, "1" level. When set to "0" level, oscillation is stopped and processor goes into standby mode.
V_{SS}	–	GND

Note: The required $\overline{\text{RESET}}$ pulse duration is at least two machine cycles under the condition that the power supply and the oscillator have been stabilized.

ADDED FUNCTIONS OF MSM80C48, MSM80C49 AND MSM80C50

The MSM80C48, MSM80C49 and MSM80C50 basically incorporate the capabilities of Intel's 8048, 8049, and 8050 plus the following new functions:

1. Power-Down Mode Enhancements

1.1 Power-down by software

(1) Clock (See item 4, "Power-down mode", for details.)
   a. Crystal-controlled oscillator halt (HLTS instruction)
      Power requirements can be minimized.
   b. Clock supply halt (HALT instruction)
      Restart is accomplished without oscillator wait.

(2) I/O ports (See Table 4-1 and 4-2 for details.)
   I/O port floating instructions
   Power consumption resulting from inputs/outputs can be minimized with FLT and FLTT instructions.
   Port floating is cancelled by executing FRES instruction, "0" level at $\overline{INT}$ pin or "0" level at $\overline{RESET}$ pin.

(3) Six types of power-down can be done by a combination of HLTS/HALT and FLT/FLTT instructions.

1.2 Power-down by hardware (See 4.3, Power-down mode by V_{DD} pin utilization for details.)

Crystal-controlled oscillators can be halted by controlling the V_{DD} terminal, thereby floating all I/O ports for minimum power consumption.

2. Additional Instructions (11)

HLTS	MOV A, P2
HALT	MOV P1, @ R3
FLT	MOVP1 P, @ R3
FLTT	DEC @ Rr
FRES	DJNZ @ R, addr
MOV A, P1	

3. Improved Uses of BUS $P_{0 \sim 7}$, $P1_{0 \sim 7}$, $P2_{0 \sim 7}$, and $\overline{SS}$ terminals

3.1 BUS $P_{0 \sim 7}$

The MSM80C48, MSM80C49, and MSM80C50 remove the limitation on the use of OUTL BUS, A instructions during the external ROM access mode by having an independent data latch and external ROM mode address latch in BUS $P_{0 \sim 7}$.

Consequently, there is no need to relocate bus port instructions when in the external ROM access mode.

3.2 $P1_{0 \sim 7}$ and $P2_{0 \sim 7}$

The MSM80C48, MSM80C49 and MSM80C50 are designed to minimize power consumption when $P1_{0 \sim 7}$ and $P2_{0 \sim 7}$ are used as input/output ports, to maximize the performance of CMOS.

When these ports are used as output ports, the acceleration circuit is actuated only when output data changes from "0" to "1", thus speeding up the rise time of the output signals.

When these ports are used as input ports, the internal pullup resistance becomes approximately 9 kΩ when input data is "1".

The internal pullup resistance rises to approximately 100 kΩ when input data is "0". Thus, a high noise margin can be obtained by selecting the impedance and thus the outflow of current is minimized whenever these ports are used as output or input ports.

3.3 Clock generation control via the $\overline{SS}$ terminal

When the crystal-controlled oscillator is halted in the HLTS or hardware power-down mode, the $\overline{SS}$ terminal is pulled down by a resistor of 20 – 50 kΩ, while its internal pullup resistor of 200 – 500 kΩ is isolated from V_{CC}. When the power-down mode is cancelled, the internal resistor of the $\overline{SS}$ terminal is changed from pulldown to pullup. Consequently, the CPU can be halted for any period of time until the crystal-controlled oscillator resumes normal oscillation when a capacitor is connected to the SS terminal.

4. Power-Down Mode

The MSM80C48, MSM80C49, and MSM80C50 power-down mode can be enabled in 2 different ways-through software by a combination of clock control and port floating instructions, and through hardware by control of the V_{DD} pin.

4.1 Software power-down mode

Power-down mode can be done by a combination of the following instructions.

(1) HALT (clock supply halt to control circuit)

Instruction code:

0	0	0	0	0	0	0	1

Description: Although crystal-controlled oscillator operation is continued, the clock supply to the CPU control circuit is halted and CPU operations suspended. When cancelling this software mode, restart is accomplished without oscillator wait. Timing charts are outlined in Figs. 4-1 and 4-2.

(2) HLTS (oscillation stop)

Instruction code:

1	0	0	0	0	0	1	0

Description: The oscillator operation is halted and CPU operations suspended. in cancelling this power down mode, connecting a capacitor to the $\overline{SS}$ pin enables a reasonable

wait period to be accomplished before normal operation is resumed. [Except in the case of using the $\overline{\text{RESET}}$ pin]
Timing charts are outlined in Figs. 4-3 and 4-4.

(3) FLT (floating $P1_0 \sim 7$, $P2_0 \sim 7$, and $BP_0 \sim 7$)

Instruction code:

1	0	1	0	0	0	1	0

Description:

	Internal ROM mode	External ROM mode
P1	Floating	Floating
P2	Floating	$P2_0 \sim 3$ operation $P2_4 \sim 7$ floating
BP	Floating	Operation

Details of IC pin status as a result of executing the FLT instruction are shown in Table 4-1.

(4) FLTT (floating of all output pins)

Instruction code:

1	1	0	0	0	0	1	0

Description:

	Internal ROM mode	External ROM mode
ALE	Floating	Operation
$\overline{\text{PSEN}}$	Floating	Operation
PROG	Floating	Floating
$\overline{\text{WR}}$	Floating	Floating
$\overline{\text{RD}}$	Floating	Floating
TO OUT	Floating	Floating
P1	Floating	Floating
P2	Floating	$P2_0 \sim 3$ operation $P2_4 \sim 7$ floating
BP	Floating	Operation
XTAL	Operation	Operation

Details of IC pin status as a result of executing the FLTT instruction are shown in Table 4-2.

Example 1: Power-down mode accomplished by stopping oscillation.
- ○ Setting by execution of HLTS [82H] instruction.

Example 2: Power-down mode accomplished by stopping the clock supply to the CPU control circuit.
- ○ Setting by execution of HALT [01H] instruction.

Example 3: Power-down mode by floating of $P1_0 \sim 7$, $P2_0 \sim 7$ and $BP_0 \sim 7$, and subsequent stopping of CPU oscillation.
- ○ Setting by first executing the FLT[A2H] instruction and then the HLTS[82H] instruction.

Example 4: Power-down mode by floating $P1_0 \sim 7$, $P2_0 \sim 7$ and $BP_0 \sim 7$, and then stopping the clock supply to the CPU control circuit.
- ○ Setting by first executing the FLT[A2H] instruction, and then the HALT[01H] instruction.

Example 5: Power-down mode by floating all output pins, followed by stopping oscillation.
- ○ Setting by first executing the FLTT[C2H] instruction followed by execution of the HLTS[82H] instruction.

Example 6: Power-down mode by floating all output pins, followed by stopping of the clock supply to the CPU control circuit.
- ○ Setting by first executing the FLTT[C2H] instruction, followed by execution of the HALT[01H] instruction.

4.2 Cancellation of software power-down mode

The power-down mode status outlined above in examples 1 to 6 can be cancelled by using either the interrupt pin or the $\overline{\text{RESET}}$ pin.

(1) Use of the $\overline{\text{INT}}$ pin during external interrupt enabled mode (i.e. following execution of EN I instruction).
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{INT}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, an external interrupt is generated, and execution proceeds from address 3. If, however, the power-down mode has been done during the interrupt processing routine, execution is resumed just after the power-down instruction.

(2) Use of the $\overline{\text{INT}}$ pin during external interrupt disabled mode (i.e. following execution of DIS I instruction or hardware reset)
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{INT}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, execution is resumed just after the power-down instruction.

(3) Use of the $\overline{\text{RESET}}$ pin
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{RESET}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, the CPU is reset and execution proceeds from address 0. In case cancellation is done in oscillation stop mode, the "0" level must be input to the $\overline{\text{RESET}}$ PIN until oscillation is stabilized.

Table 4-1 Details of Pin Status Following Execution of FLT Instruction

Pin No.	Pin Name	Internal ROM	External ROM
1P	T0	Active	Active
2P	XTAL1	Active	Active
3P	XTAL2	Active	Active
4P	$\overline{\text{RESET}}$	Active	Active
5P	$\overline{\text{SS}}$	200 ~ 500 kΩ pullup	200 ~ 500 kΩ pullup
6P	$\overline{\text{INT}}$	Active	Active
7P	EA	Active	Active
8P	$\overline{\text{RD}}$	Active	Active
9P	$\overline{\text{PSEN}}$	Active	Active
10P	$\overline{\text{WR}}$	Active	Active
11P	ALE	Active	Active
12P	DB0	Floating	Active
13P	DB1	Floating	Active
14P	DB2	Floating	Active
15P	DB3	Floating	Active
16P	DB4	Floating	Active
17P	DB5	Floating	Active
18P	DB6	Floating	Active
19P	DB7	Floating	Active
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Floating	Active
22P	P21	Floating	Active
23P	P22	Floating	Active
24P	P23	Floating	Active
25P	PROG	Active	Active
26P	V_{DD}	"1" level	"1" level
27P	P10	Floating	Floating
28P	P11	Floating	Floating
29P	P12	Floating	Floating
30P	P13	Floating	Floating
31P	P14	Floating	Floating
32P	P15	Floating	Floating
33P	P16	Floating	Floating
34P	P17	Floating	Floating
35P	P24	Floating	Floating
36P	P25	Floating	Floating
37P	P26	Floating	Floating
38P	P27	Floating	Floating
39P	T1	Active	Active
40P	V_{CC}	+2 to +6 [V]	+2 to +6 [V]

Note: The FLT mode itself is reset by executing the FRES instruction, or supplying "0" level to $\overline{\text{INT}}$ or $\overline{\text{RESET}}$ pin.

Table 4-2 Details of Pin Status Following Execution of FLTT Instruction

Pin No.	Pin Name	Internal ROM	External ROM
1P	T0	Floating if output enabled	Floating if output enabled
2P	XTAL1	Active	Active
3P	XTAL2	Active	Active
4P	$\overline{\text{RESET}}$	Active	Active
5P	SS	200 to 500 kΩ pullup	200 to 500 kΩ pullup
6P	$\overline{\text{INT}}$	Active	Active
7P	EA	Active	Active
8P	$\overline{\text{RD}}$	Floating	Floating
9P	$\overline{\text{PSEN}}$	Floating	Active
10P	$\overline{\text{WR}}$	Floating	Floating
11P	ALE	Floating	Active
12P	DB0	Floating	Active
13P	DB1	Floating	Active
14P	DB2	Floating	Active
15P	DB3	Floating	Active
16P	DB4	Floating	Active
17P	DB5	Floating	Active
18P	DB6	Floating	Active
19P	DB7	Floating	Active
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Floating	Active
22P	P21	Floating	Active
23P	P22	Floating	Active
24P	P23	Floating	Active
25P	PROG	Floating	Floating
26P	V_{DD}	"1" level	"1" level
27P	P10	Floating	Floating
28P	P11	Floating	Floating
29P	P12	Floating	Floating
30P	P13	Floating	Floating
31P	P14	Floating	Floating
32P	P15	Floating	Floating
33P	P16	Floating	Floating
34P	P17	Floating	Floating
35P	P24	Floating	Floating
36P	P25	Floating	Floating
37P	P26	Floating	Floating
38P	P27	Floating	Floating
39P	T1	Active	Active
40P	V_{CC}	+2.5 to +6 [V]	+2.5 to +6 [V]

Note: The FLTT mode itself is reset by executing the FRES instruction, or supplying "0" level to $\overline{\text{INT}}$ or $\overline{\text{RESET}}$ pin.

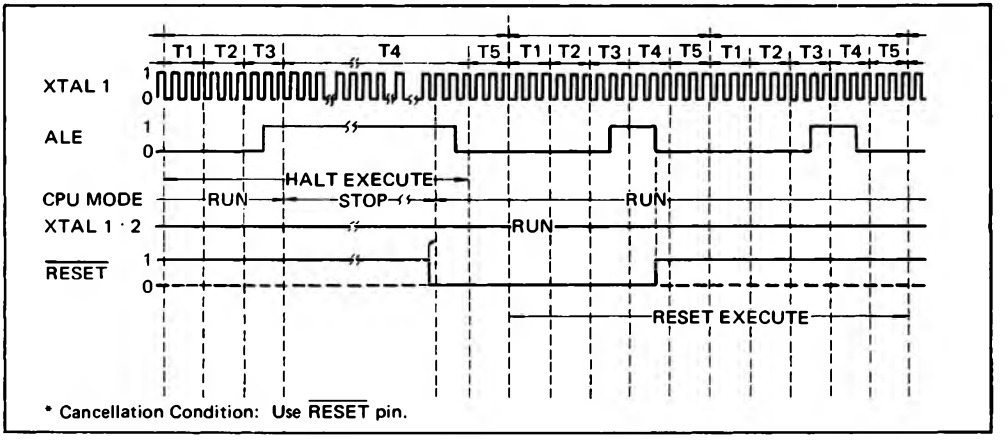


Fig. 4-1 HALT [01H] Instruction Execution Timing Chart

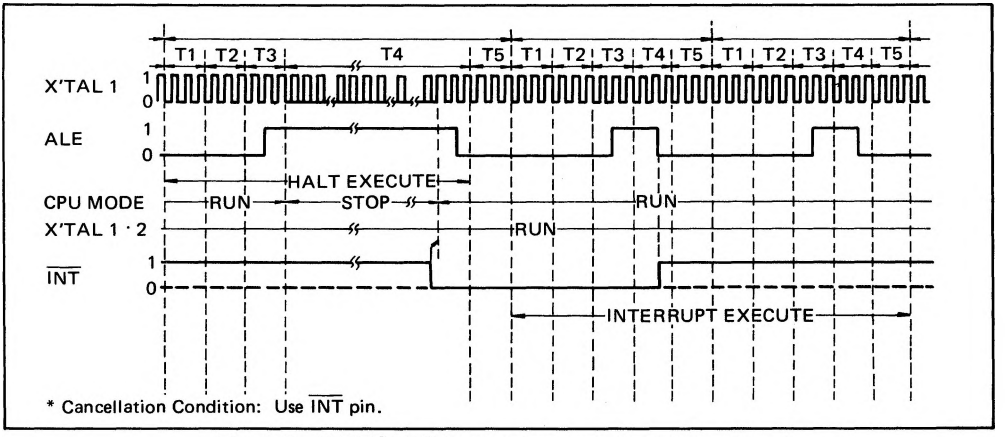


Fig. 4-2 HALT [01H] Instruction Execution Timing Chart

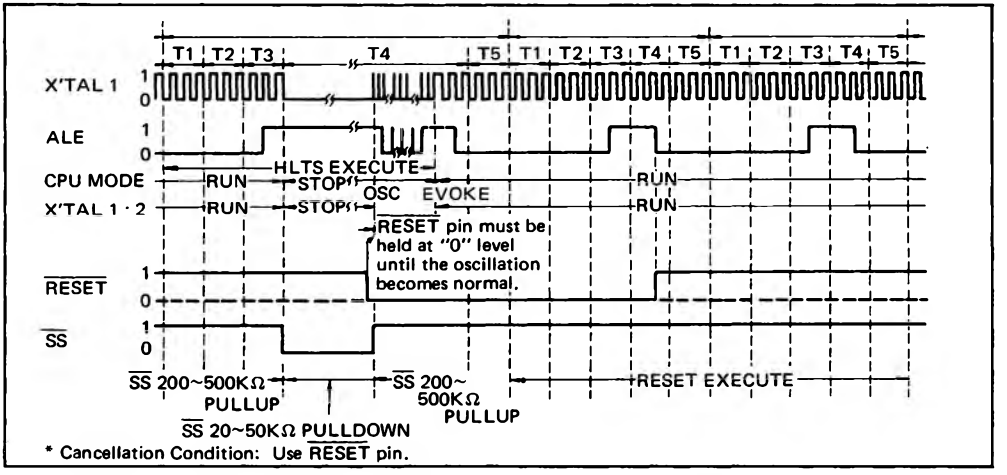


Fig. 4-3 HLTS [82H] Instruction Execution Timing Chart

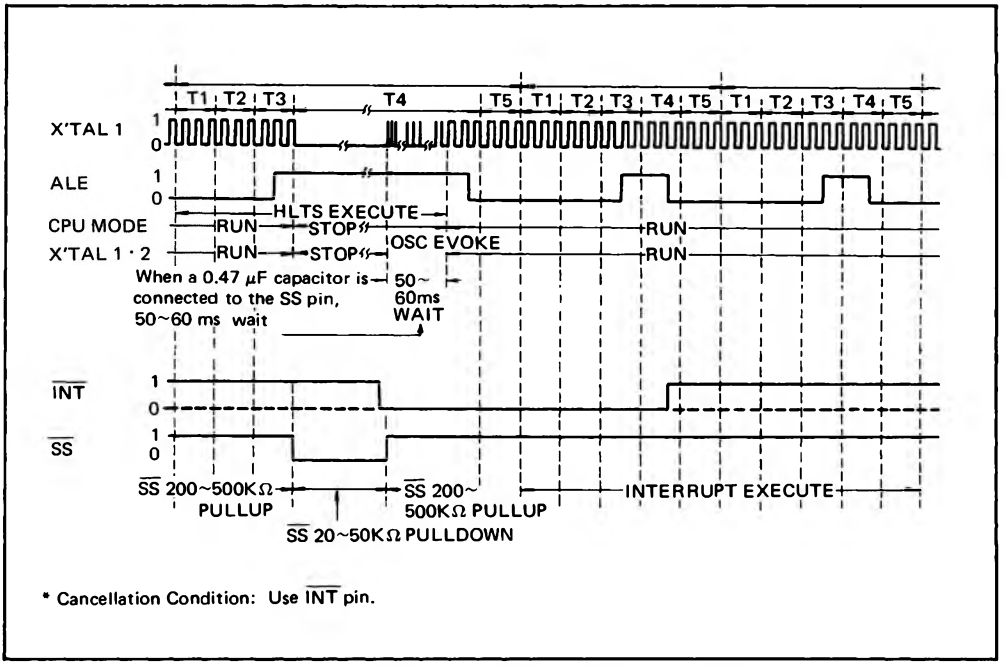


Fig. 4-4 HLTS [82H] Instruction Execution Timing Chart

4.3 Hardware power-down mode

In the MSM80C48, MSM80C49 and MSM80C50, forcing the level at the V_{DD} pin [pin 26] to a "0" during either external ROM or internal ROM mode results in suspension of the oscillator function and subsequent floating (high impedance) of all the I/O pins except the $\overline{RESET}$, $\overline{SS}$ and XTAL 1/2 pins. The CPU is thereby stopped while maintaining internal status. Details of the IC pin status at this time are outlined in Table 4-3.

4.4 Cancellation of hardware power-down mode

(1) Use of $\overline{RESET}$ pin

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is input to the $\overline{RESET}$ pin. If this "0" level is kept applied to the $\overline{RESET}$ pin until oscillation become stable, the CPU will be reset and will start executing from address 0. The timing chart is outlined in Fig. 4-5.

(2) Use of the $\overline{INT}$ pin during external interrupt enabled status (i.e. following execution of EN I instruction)

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is applied to the $\overline{INT}$ pin.

If this "0" level is maintained until at least 2 ALE output signals occur, an external interrupt is generated, and execution starts from address 3.

However, if the power-down mode is started during an interrupt processing routine, execution will be continued on the next instruction after the present instruction. The timing chart is outlined in Fig. 4-6.

(3) Use of the $\overline{INT}$ pin during external interrupt disabled mode (i.e. following execution of DIS I instruction or hardware reset)

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is applied to the $\overline{INT}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, execution is continued on the next instruction after the present instruction. The timing chart is outlined in Fig. 4-6.

(4) Use of V_{DD} pin only

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "1" level is also applied to both the $\overline{RESET}$ and $\overline{INT}$ pins. In this case, execution is resumed from the stopped position. The timing chart is outlined in Fig. 4-7.

Table 4-3 Details of Pin Status during Hardware Power-Down Mode

Pin No.	Pin Name	Normal Operation (V_{DD} = "1" level)	Power Down Mode (V_{DD} = "0" level)
1P	T0	Active	Floating if output enabled
2P	XTAL1	Active	Active
3P	XTAL2	Active	Active
4P	$\overline{RESET}$	Active	Active
5P	$\overline{SS}$	200 to 500 kΩ pullup	20 to 50 kΩ pulldown
6P	$\overline{INT}$	Active	Active
7P	EA	Active	Active
8P	$\overline{RD}$	Active	Floating
9P	$\overline{PSEN}$	Active	Floating
10P	$\overline{WR}$	Active	Floating
11P	ALE	Active	Floating
12P	DB0	Active	Floating
13P	DB1	Active	Floating
14P	DB2	Active	Floating
15P	DB3	Active	Floating
16P	DB4	Active	Floating
17P	DB5	Active	Floating
18P	DB6	Active	Floating
19P	DB7	Active	Floating
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Active	Floating
22P	P21	Active	Floating
23P	P22	Active	Floating
24P	P23	Active	Floating
25P	PROG	Active	Floating
26P	V_{DD}	"1" level	"0" level
27P	P10	Active	Floating
28P	P11	Active	Floating
29P	P12	Active	Floating
30P	P13	Active	Floating
31P	P14	Active	Floating
32P	P15	Active	Floating
33P	P16	Active	Floating
34P	P17	Active	Floating
35P	P24	Active	Floating
36P	P25	Active	Floating
37P	P26	Active	Floating
38P	P27	Active	Floating
39P	T1	Active	Active
40P	V_{CC}	+2 to +6 [V]	+2 to +6 [V]

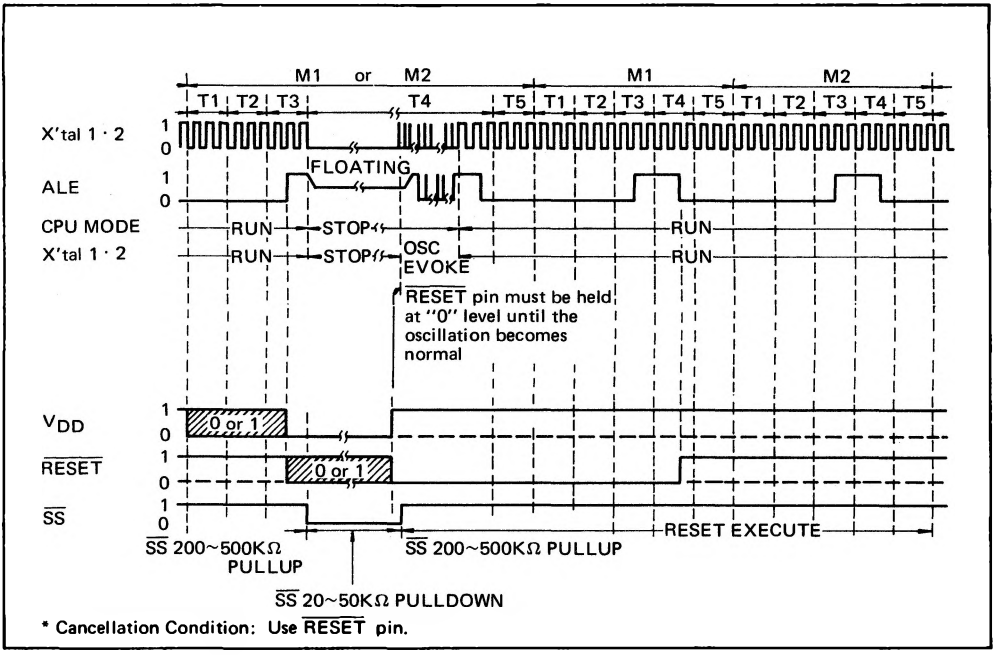


Fig. 4-5 Hardware Power-Down Mode Timing Chart

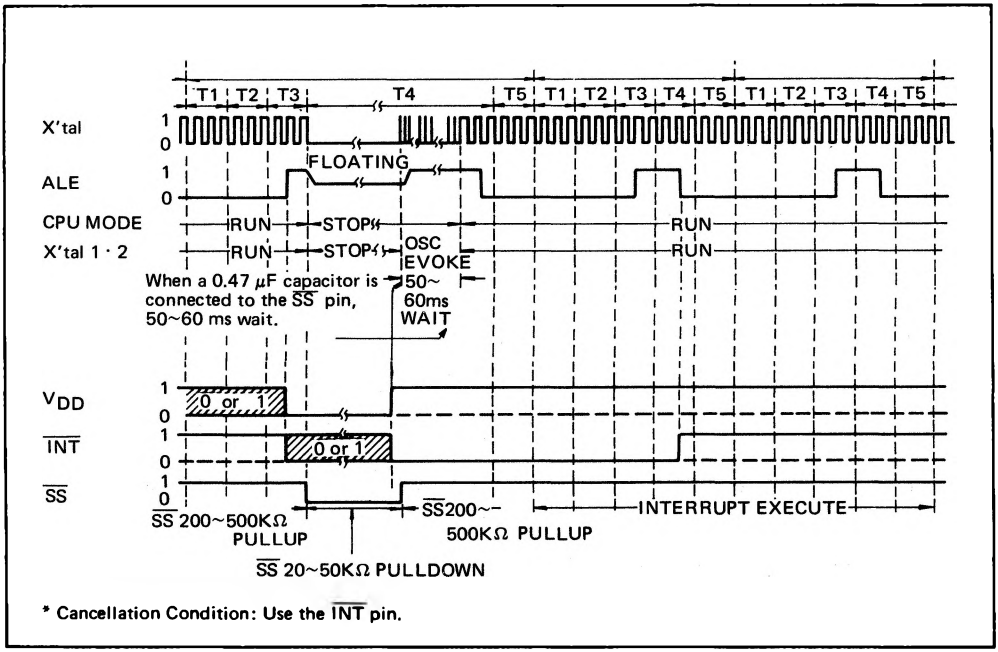


Fig. 4-6 Hardware Power-Down Mode Timing Chart

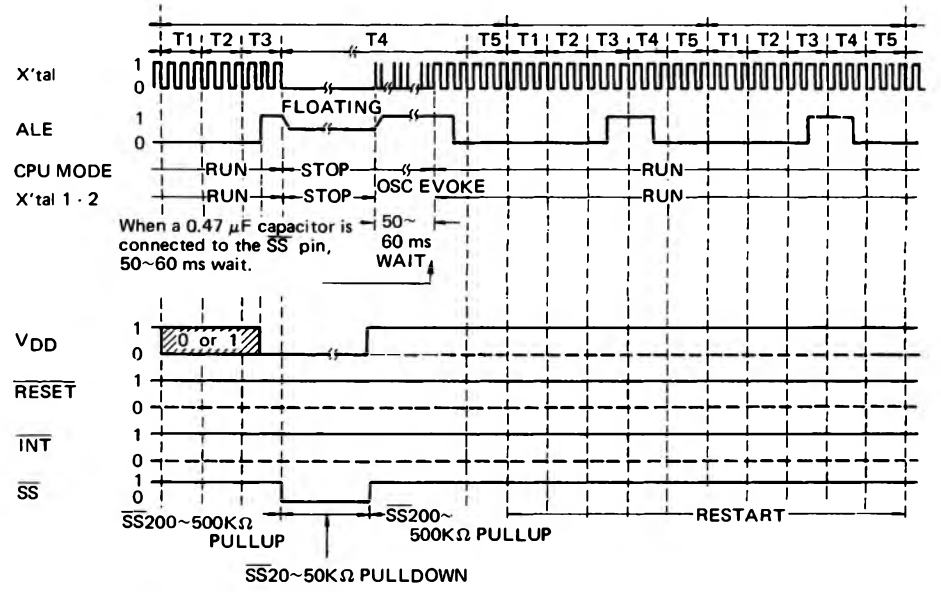


Fig. 4-7 Hardware Power-Down Mode Timing Chart

MSM80C48/MSM80C49/MSM80C50 INSTRUCTION TABLE

H \ L	0 0 0 0 0	1 0 0 0 1	2 0 0 1 0	3 0 0 1 1	4 0 1 0 0	5 0 1 0 1	6 0 1 1 0	7 0 1 1 1	8 1 0 0 0	9 1 0 0 1	A 1 0 1 0	B 1 0 1 1	C 1 1 0 0	D 1 1 0 1	E 1 1 1 0	F 1 1 1 1
0 0 0 0 0	NOP	HALT Added	OUTL BUS A	ADD A, # data	JMP	EN I		DEC A	INS A, BUS	IN A, P1	INA, P2		MOVD A, Pp			
1 0 0 0 1	INC @, R0	INC @, R1	JB0 addr	ADDC A, # data	CALL	DIS I	JTF addr	INC A	INC Rr							
2 0 0 1 0	XCHA @, R0	XCHA @, R1		MOV A, # data	JMP	EN TCNTI	JNT0 addr	CLR A	XCH A, Rr							
3 0 0 1 1	XCHD A, @R0	XCHD A, @R1	JB1 addr		CALL	DIS TCNTI	JT0 addr	CPL A		OUTL P1, A	OUTL P2, A		MOVD Pp, A			
4 0 1 0 0	ORL A, @R0	ORL A, @R1	MOV A, T	ORL A, # data	JMP	STRT CNT	JNT1 addr	SWAP A	ORLA, Rr							
5 0 1 0 1	ANL A, @R0	ANL A, @R1	JB2 addr	ANL A, # data	CALL	STRT T	JT1 addr	DA A	ANL A, Rr							
6 0 1 1 0	ADD A, @ R0	ADD A, @ R1	MOV T, A	MOV A, P1 Added	JMP	STOP TCNT		RRC A	ADD A, Rr							
7 0 1 1 1	ADDC A, @ R0	ADDC A, @ R1	JB3 addr	MOV A, P2 Added	CALL	ENT0 CLK	JF1 addr	RR A	ADDC A, Rr							
8 1 0 0 0	MOVX A, @ R0	MOVX A, @ R1	HLTS Added	RET	JMP	CLR F0	JN1 addr		ORL BUS, # data	ORL P1, # data	ORL P2, # data		ORLD Pp, A			
9 1 0 0 1	MOVX @ R0, A	MOVX @ R1, A	JB4 addr	RETR	CALL	CLR F0	JNZ addr	CLR C	ANL BUS, # data	ANL P1, # data	ANL P2, # data		ANL D Pp, A			
A 1 0 1 0	MOV @ R0, A	MOV @ R1, A	FLT Added	MOVP A, @A	JMP	CLR F1		CPL C	MOV Rr, A							
B 1 0 1 1	MOV @ R0, #data	MOV @R1, #data	JB5 addr	JMPP @A	CALL	CPL F1	JF0 addr		MOV R, #data							
C 1 1 0 0	DEC @R0 Added	DEC @R1 Added	FLTT Added	MOVP1 P,@ R3 Added	JMP	SEL RB0	JZ addr	MOV A, PSW	DEC Rr							
D 1 1 0 1	XRL A, @R0	XRL A, @R1	JB6 addr	XRLA, #data	CALL	SEL RB1		MOV PSW, A	XRLA, Rr							
E 1 1 1 0	DJNZ @R0 Added	DJNZ @R1 Added	FRES Added	MOVP3 A, @A	JMP	SEL MB0	JNC addr	RL A	DJNZ Rr							
F 1 1 1 1	MOV A, @R0	MOV A, @R1	JB7 addr	MOVP1, @ R3 Added	CALL	SEL MB1	JC addr	RLC A	MOV A, Rr							

EXPLANATION OF INSTRUCTION SYMBOLS

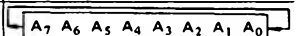
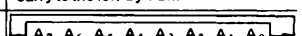
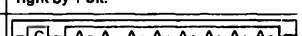
Symbols are listed below.

A	: Accumulator	PC	: Program counter
AC	: Auxiliary carry	Pp	: Port indicator (p = 4 ~ 7)
addr	: 12-bit program memory address or its part	PSW	: Program status word
Bb	: Bit indicator (b = 0 ~ 7)	Rr	: Resister indicator (r = 0 ~ 7)
BS	: Bank switch	SP	: Stack pointer
BUS	: BUS PORT	T	: Timer
C	: Carry	TF	: Timer flag
CLK	: Clock	T0, T1	: Test pins T0 and T1
CNT	: Counter	X	: External RAM
D	: 4-bit data	#	: Symbol denoting immediate data
data	: 8-bit numerical value	@	: Symbol denoting indirect address
DBF	: Memory data bank flip-flop	(X)	: Denotes contents of X
F0, F1	: F0 flag and F1 flag	((X))	: Denotes contents addressed by X
I	: Interrupt	←	: Transference

LIST OF INSTRUCTIONS

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Accumulator operation instructions	ADD A, Rr	0	1	1	0	1	r_2	r_1	r_0	68 ~ 6F	1	1	(AC), (C), (A) ← (A) + (Rr)
	ADD A, @Rr	0	1	1	0	0	0	0	r_0	60~61	1	1	(AC), (C), (A) ← (A) + ((Rr))
	ADD A, #data	0 d_7	0 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	03 Byte 2	2	2	(AC), (C), (A) ← (A) + data
	ADDC A, Rr	0	1	1	1	1	r_2	r_1	r_0	78~7F	1	1	(AC), (C), (A) ← (A) + (Rr) + (C)
	ADDC A, @Rr	0	1	1	1	0	0	0	r_0	70~71	1	1	(AC), (C), (A) ← (A) + ((Rr)) + (C)
	ADDC A, #data	0 d_7	0 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	13 Byte 2	2	2	(AC), (C), (A) ← (A) + data + (C)
	ANL A, Rr	0	1	0	1	1	r_2	r_1	r_0	58~5F	1	1	(A) ← (A) AND (Rr)
	ANL A, @Rr	0	1	0	1	0	0	0	r_0	50~51	1	1	(A) ← (A) AND ((Rr))
	ANL A, #data	0 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	53 Byte 2	2	2	(A) ← (A) AND data
	ORL A, Rr	0	1	0	0	1	r_2	r_1	r_0	48~4F	1	1	(A) ← (A) OR (Rr)
	ORL A, @Rr	0	1	0	0	0	0	0	r_0	40~41	1	1	(A) ← (A) OR ((Rr))
	ORL A, #data	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	43 Byte 2	2	2	(A) ← (A) OR data
	XRLA, Rr	1	1	0	1	1	r_2	r_1	r_0	D8~DF	1	1	(A) ← (A) XOR (Rr)
	XRLA, @Rr	1	1	0	1	0	0	0	r_0	D0~D1	1	1	(A) ← (A) XOR ((Rr))
	XRL A, #data	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	D3 Byte 2	2	2	(A) ← (A) XOR data
	INC A	0	0	0	1	0	1	1	1	17	1	1	(A) ← (A) + 1
	DEC A	0	0	0	0	0	1	1	1	07	1	1	(A) ← (A) − 1
	CLR A	0	0	1	0	0	1	1	1	27	1	1	(A) ← 0
	CPL A	0	0	1	1	0	1	1	1	37	1	1	(A) ← $(\bar{A})$
	DA A	0	1	0	1	0	1	1	1	57	1	1	Add 6 to bits 0 ~ 3 when contents of accumulator bits 0 ~ 3 exceed 9 or when auxiliary carry (AC) is 1. Then add 6 to bits 4 ~ 7 when the result of adding the carry from the lower 0 ~ 3 exceeds 9, or when carry (C) is 1. Set 1 in the carry flag if an overflow is generated in the end result, or when the carry prior to adjustment is 1.

LIST OF INSTRUCTIONS (CONT.)

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Accumulator operation instructions	SWAP A	0	1	0	0	0	1	1	1	47	1	1	$(A_{4\sim7}) \rightleftarrows (A_{0\sim3})$
	RL A	1	1	1	0	0	1	1	1	E7	1	1	 Rotate accumulator contents to the left by 1 bit.
	RLC A	1	1	1	1	0	1	1	1	F7	1	1	 Rotate accumulator contents with carry to the left by 1 bit.
	RR A	0	1	1	1	0	1	1	1	77	1	1	 Rotate accumulator contents to the right by 1 bit.
	RRC A	0	1	1	0	0	1	1	1	67	1	1	 Rotate accumulator contents with carry to the right by 1 bit.
Input/output instructions	IN A, P1	0	0	0	0	1	0	0	1	09	1	2	(A) ← (P1)
	IN A, P2	0	0	0	0	1	0	1	0	0A	1	2	(A) ← (P2)
	OUTL P1, A	0	0	1	1	1	0	0	1	39	1	2	(P1) ← (A)
	OUTL P2, A	0	0	1	1	1	0	1	0	3A	1	2	(P2) ← (A)
	ANL P1, #data	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	1 d_0	99 Byte 2	2	2	(P1) ← (P1) AND data
	ANL P2, #data	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	1 d_1	0 d_0	9A Byte 2	2	2	(P2) ← (P2) AND data
	ORL P1, #data	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	1 d_0	89 Byte 2	2	2	(P1) ← (P1) OR data
	ORL P2, #data	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	1 d_1	0 d_0	8A Byte 2	2	2	(P2) ← (P2) OR data
	INS A, BUS	0	0	0	0	1	0	0	0	08	1	2	(A) ← (BUS)
	OUTL BUS, A	0	0	0	0	0	0	1	0	02	1	2	(BUS) ← (A)
	ANL BUS, #data	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	98 Byte 2	2	2	(BUS) ← (BUS) AND data
	ORL BUS, #data	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	88 Byte 2	2	2	(BUS) ← (BUS) OR data
	MOVD A, Pp	0	0	0	0	1	1	P_1	P_0	0C~0F	1	2	$(A_{0\sim3}) \leftarrow$ (Pp) p=4~7 $(A_{4\sim7}) \leftarrow 0$
	MOVD Pp, A	0	0	1	1	1	1	P_1	P_0	3C~3F	1	2	(Pp) $\leftarrow (A_{0\sim3})$ p=4~7
	ANLD Pp, A	1	0	0	1	1	1	P_1	P_0	9C~9F	1	2	(Pp) ← (Pp) AND $(A_{0\sim3})$ p=4~7
	ORLD Pp, A	1	0	0	0	1	1	P_1	P_0	8C~8F	1	2	(Pp) ← (Pp) OR $(A_{0\sim3})$ p=4~7
Register operation instructions	INC Rr	0	0	0	1	1	r_2	r_1	r_0	18~1F	1	1	(Rr) ← (Rr) + 1
	INC @Rr	0	0	0	1	0	0	0	r_0	10~11	1	1	((Rr)) ← ((Rr)) + 1
	DEC Rr	1	1	0	0	1	r_2	r_1	r_0	C8~CF	1	1	(Rr) ← (Rr) − 1
	DEC @Rr	1	1	0	0	0	0	0	r_0	C0~C1	1	1	((Rr)) ← ((Rr)) − 1
Branching instructions	JMP addr	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	ϕ4 ~ E4 Byte 2	2	2	$(PC_{8\sim10}) \leftarrow$ addr 8~10 $(PC_{0\sim7}) \leftarrow$ addr 0~7 $(PC_{11}) \leftarrow$ DBF
	JMPP @A	1	0	1	1	0	0	1	1	B3	1	2	$(PC_{0\sim7}) \leftarrow$ ((A))

LIST OF INSTRUCTIONS (CONT.)

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Branching instructions	DJNZ Rr, addr	1	1	1	0	1	r_2	r_1	r_0	E8~EF	2	2	(Rr) ←(Rr) − 1 ($PC_{0\sim7}$) ←addr if (Rr) = 0 (PC) ←(PC) + 2 if (Rr) = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	DJNZ @Rr, addr	1	1	1	0	0	0	0	r_0	E0~E1	2	2	((Rr)) ←((Rr)) − 1 ($PC_{0\sim7}$) ←addr if ((Rr)) = 0 (PC) ←(PC) + 2 if ((Rr)) = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JC addr	1	1	1	1	0	1	1	0	F6	2	2	($PC_{0\sim7}$) ←addr if C = 1 (PC) ←(PC) + 2 if C = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNC addr	1	1	1	0	0	1	1	0	E6	2	2	($PC_{0\sim7}$) ←addr if C = 0 (PC) ←(PC) + 2 if C = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JZ addr	1	1	0	0	0	1	1	0	C6	2	2	($PC_{0\sim7}$) ←addr if A = 0 (PC) ←(PC) + 2 if A ≠ 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNZ addr	1	0	0	1	0	1	1	0	96	2	2	($PC_{0\sim7}$) ←addr if A ≠ 0 (PC) ←(PC) + 2 if A = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JT0 addr	0	0	1	1	0	1	1	0	36	2	2	($PC_{0\sim7}$) ←addr if T0 = 1 (PC) ←(PC) + 2 if T0 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNT0 addr	0	0	1	0	0	1	1	0	26	2	2	($PC_{0\sim7}$) ←addr if T0 = 0 (PC) ←(PC) + 2 if T0 = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JT1 addr	0	1	0	1	0	1	1	0	56	2	2	($PC_{0\sim7}$) ←addr if T1 = 1 (PC) ←(PC) + 2 if T1 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNT1 addr	0	1	0	0	0	1	1	0	46	2	2	($PC_{0\sim7}$) ←addr if T1 = 0 (PC) ←(PC) + 2 if T1 = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JF0 addr	1	0	1	1	0	1	1	0	B6	2	2	($PC_{0\sim7}$) ←addr if F0 = 1 (PC) ←(PC) + 2 if F0 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JF1 addr	0	1	1	1	0	1	1	0	76	2	2	($PC_{0\sim7}$) ←addr if F1 = 1 (PC) ←(PC) + 2 if F1 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JTF addr	0	0	0	1	0	1	1	0	16	2	2	($PC_{0\sim7}$) ←addr TF ←0 } if TF = 1 (PC) ←(PC) + 2 if TF = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNI addr	1	0	0	0	0	1	1	0	86	2	2	($PC_{0\sim7}$) ←addr if $\overline{INT}$ = 0 (PC) ←(PC)+2 if $\overline{INT}$ = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JBb addr	b_2	b_1	b_0	1	0	0	1	0	12 ~ F2	2	2	($PC_{0\sim7}$) ←addr if Bb = 1 (PC) ←(PC)+2 if Bb = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
Sub-routine instructions	CALL addr	a_{10}	a_9	a_8	1	0	1	0	0	14 ~ F4	2	2	((SP)) ←(PC) + 2, ($PSW_{4\sim7}$) ($PC_{8\sim10}$) ←addr 8 ~ 10 ($PC_{0\sim7}$) ←addr 0 ~ 7 (PC_{11}) ←DBF (SP) ←(SP) + 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	RET	1	0	0	0	0	0	1	1	83	1	2	(SP) ←(SP) − 1 (PC) ←((SP))
	RETR	1	0	0	1	0	0	1	1	93	1	2	(SP) ←(SP) − 1 (PC) ←((SP)) ($PSW_{4\sim7}$) ←((SP)) INT END
Flag operation instructions	CLR C	1	0	0	1	0	1	1	1	97	1	1	(C) ←0
	CPL C	1	0	1	0	0	1	1	1	A7	1	1	(C) ←$\overline{(C)}$
	CLR F0	1	0	0	0	0	1	0	1	85	1	1	(F0) ←0
	CPL F0	1	0	0	1	0	1	0	1	95	1	1	(F0) ←$\overline{(F0)}$
	CLR F1	1	0	1	0	0	1	0	1	A5	1	1	(F1) ←0
	CPL F1	1	0	1	1	0	1	0	1	B5	1	1	(F1) ←$\overline{(F1)}$
Data transfer instructions	MOV A, Rr	1	1	1	1	1	r_2	r_1	r_0	F8~FF	1	1	(A) ←(Rr)
	MOV A, @Rr	1	1	1	1	0	0	0	r_0	F0~F1	1	1	(A) ←((Rr))
	MOV A, #data	0	0	1	0	0	0	1	1	23	2	2	(A) ←data
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 2			

LIST OF INSTRUCTIONS (CONT.)

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Data transfer instructions	MOV Rr, A	1	0	1	0	1	r_2	r_1	r_0	A8~AF	1	1	(Rr) ←(A)
	MOV @Rr, A	1	0	1	0	0	0	0	r_0	A0~A1	1	1	((Rr)) ←(A)
	MOV Rr, #data	1 d_7	0 d_6	1 d_5	1 d_4	1 d_3	r_2 d_2	r_1 d_1	r_0 d_0	B8~BF Byte 2	2	2	(Rr) ←data
	MOV @Rr, #data	1 d_7	0 d_6	1 d_5	1 d_4	0 d_3	0 d_2	0 d_1	r_0 d_0	B0~B1 Byte 2	2	2	((Rr)) ←data
	MOV A, PSW	1	1	0	0	0	1	1	1	C7	1	1	(A) ←(PSW)
	MOV PSW, A	1	1	0	1	0	1	1	1	D7	1	1	(PSW) ←(A)
	XCH A, Rr	0	0	1	0	1	r_2	r_1	r_0	28~2F	1	1	(A) ⇄(Rr)
	XCH A, @Rr	0	0	1	0	0	0	0	r_0	20~21	1	1	(A) ⇄((Rr))
	XCHD A, @Rr	0	0	1	1	0	0	0	r_0	30~31	1	1	$(A_{0\sim3})$⇄$((Rr_{0\sim3}))$
	MOVX A, @Rr	1	0	0	0	0	0	0	r_0	80~81	1	2	(A) ←((Rr)) External RAM
	MOVX @Rr, A	1	0	0	1	0	0	0	r_0	90~91	1	2	((Rr)) ←(A) External RAM
	MOVP A, @A	1	0	1	0	0	0	1	1	A3	1	2	(A) ←$((PC_{8\sim10}, A))$
	MOVP3 A, @A	1	1	1	0	0	0	1	1	E3	1	2	(A) ←$((PC_{011}, A))$
	MOVP1 P, @R3	1	1	0	0	0	0	1	1	C3	1	2	(P1) ←$(((PC_{0\sim7})$←((R3)))
	MOV P1, @R3	1	1	1	1	0	0	1	1	F3	1	2	(P1) ←((R3))
	MOV A, P1	0	1	1	0	0	0	1	1	63	1	1	(A) ←(P1) Latch data
	MOV A, P2	0	1	1	1	0	0	1	1	73	1	1	(A) ←(P2) Latch data
Control instructions	EN TCNTI	0	0	1	0	0	1	0	1	25	1	1	TINT Enable F/F ←1
	DIS TCNTI	0	0	1	1	0	1	0	1	35	1	1	TINT Enable F/F ←0
	EN I	0	0	0	0	0	1	0	1	05	1	1	EXINT Enable F/F ←1
	DIS I	0	0	0	1	0	1	0	1	15	1	1	EXINT Enable F/F ←0
	SEL RB0	1	1	0	0	0	1	0	1	C5	1	1	(BS) ←0
	SEL RB1	1	1	0	1	0	1	0	1	D5	1	1	(BS) ←1
	SEL MB0	1	1	1	0	0	1	0	1	E5	1	1	(DBF) ←0
	SEL MB1	1	1	1	1	0	1	0	1	F5	1	1	(DBF) ←1
	$\overline{\text{ENT0 CLK}}$	0	1	1	1	0	1	0	1	75	1	1	T0 ←1/3 XTAL 1
	FLT	1	0	1	0	0	0	1	0	A2	1	1	P1, P2, BUS Floating
	FLTT	1	1	0	0	0	0	1	0	C2	1	1	CPU Output Signal Floating
	FRES	1	1	1	0	0	0	1	0	E2	1	1	FLT, FLTT RESET
	HLT	0	0	0	0	0	0	0	1	01	1	1	CPU Control Clock Stop
	HALTS	1	0	0	0	0	0	1	0	82	1	1	XTAL 1·2 Stop
Timer/counter instructions	MOV A, T	0	1	0	0	0	0	1	0	42	1	1	(A) ←(T)
	MOV T, A	0	1	1	0	0	0	1	0	62	1	1	(T) ←(A)
	STRT T	0	1	0	1	0	1	0	1	55	1	1	(T) ←÷32←÷15←XTAL
	STRT CNT	0	1	0	0	0	1	0	1	45	1	1	(T) ←T1 Clock
	STOP TCNT	0	1	1	0	0	1	0	1	65	1	1	(T) Count Stop
Other instruction	NOP	0	0	0	0	0	0	0	0	00	1	1	(PC) ←(PC) + 1

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{CC}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{CC}	V
Storage Temperature	T_{stg}		−55 to +150	°C

OPERATING RANGE

- MSM80C35/48, 80C39/49 ... DC to 11 MHz, VCC = 5V ±20%
- MSM80C40/50 ... DC to 6 MHz, VCC = 50 ±20%

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{CC}	fosc = DC~11 MHz*	+2.5 to +6	V
RAM Retention Voltage	V_{CC}		+2 to +6	V
Ambient Temperature	T_A		−40 to +85	°C
Fan Out	N	MOS load	10	
		TTL load	1	

* 11 MHz version of MSM80C40/50 (6 MHz < XTAL1.2 < 11 MHz) is under development.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V±10%, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit	Measuring Circuit
"L" Input Voltage	V_{IL}		−0.3		0.8	V	1
"H" Input Voltage (1)	V_{IH}		2.2		V_{CC}	V	
"H" Input Voltage (2)	V_{IH}		3.8		V_{CC}	V	
"L" Output Voltage (3)	V_{OL}	I_{OL} = 2 mA			0.45	V	
"L" Output Voltage (4)	V_{OL}	I_{OL} = 1.6 mA			0.45	V	
"H" Output Voltage (3)	V_{OH}	I_{OH} = 400 μA	2.4			V	
"H" Output Voltage (4)	V_{OH}	I_{OH} = 50 μA	2.4			V	
"H" Output Voltage (3)	V_{OH}	I_{OH} = 20 μA	4.2			V	
"H" Output Voltage (4)	V_{OH}	I_{OH} = 10 μA	4.2			V	
Input Leak Current	I_{IL}	$V_{SS} \leqq V_{IN} \leqq V_{CC}$			±10	μA	2
Output Leak Current (5)	I_{OL}	$V_{SS} \leqq V_O \leqq V_{CC}$			±10	μA	3
$\overline{\text{RESET}}$ Pull up Resistance	R_R	$V_{IN} \geqq V_{IH} \leqq$ / $V_{IN} \leqq V_{IL}$	20/500		50/750	kΩ	2
$\overline{\text{SS}}$ Pull up Resistance (6)	R_{SS}	Oscillation stop/oscillation	20/200		50/500	kΩ	
P1, P2 Pull up Resistance	$R_{P1, P2}$	$V_{IN} \geqq V_{IH}$/ $V_{IN} \leqq V_{iL}$	5/75		15/150	kΩ	3
Power Supply Current	I_{CC}	At hardware power down V_{CC}=2V (TA = +25°C) (7)		1	10	μA	4
		At HLTS execution V_{CC}=2V (TA = +25°C) (7)		1	10	μA	
		At HALT (6 MHz)		1.5	3	mA	
		At HALT (11 MHz) (8)		2.5	5	mA	
		Atexecution (6 MHz)		5	10	mA	
		At execution (11 MHz) (8)		10	20	mA	

Notes: (1) This does not apply to $\overline{\text{RESET}}$, XTAL1, XTAL2, and V_{DD}.
(2) $\overline{\text{RESET}}$, XTAL1, XTAL2, V_{DD}
(3) BUS, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE
(4) Other outputs
(5) High-impedance state
(6) This operates as a pull-down resistor when the oscillation is stopped in the HLTS or hardware power-down mode and as a pull-up resistor in other states.
(7) This does not contain flow out current from I/O Ports and Signal pins.
(8) MSM80C35/48, 80C39/49

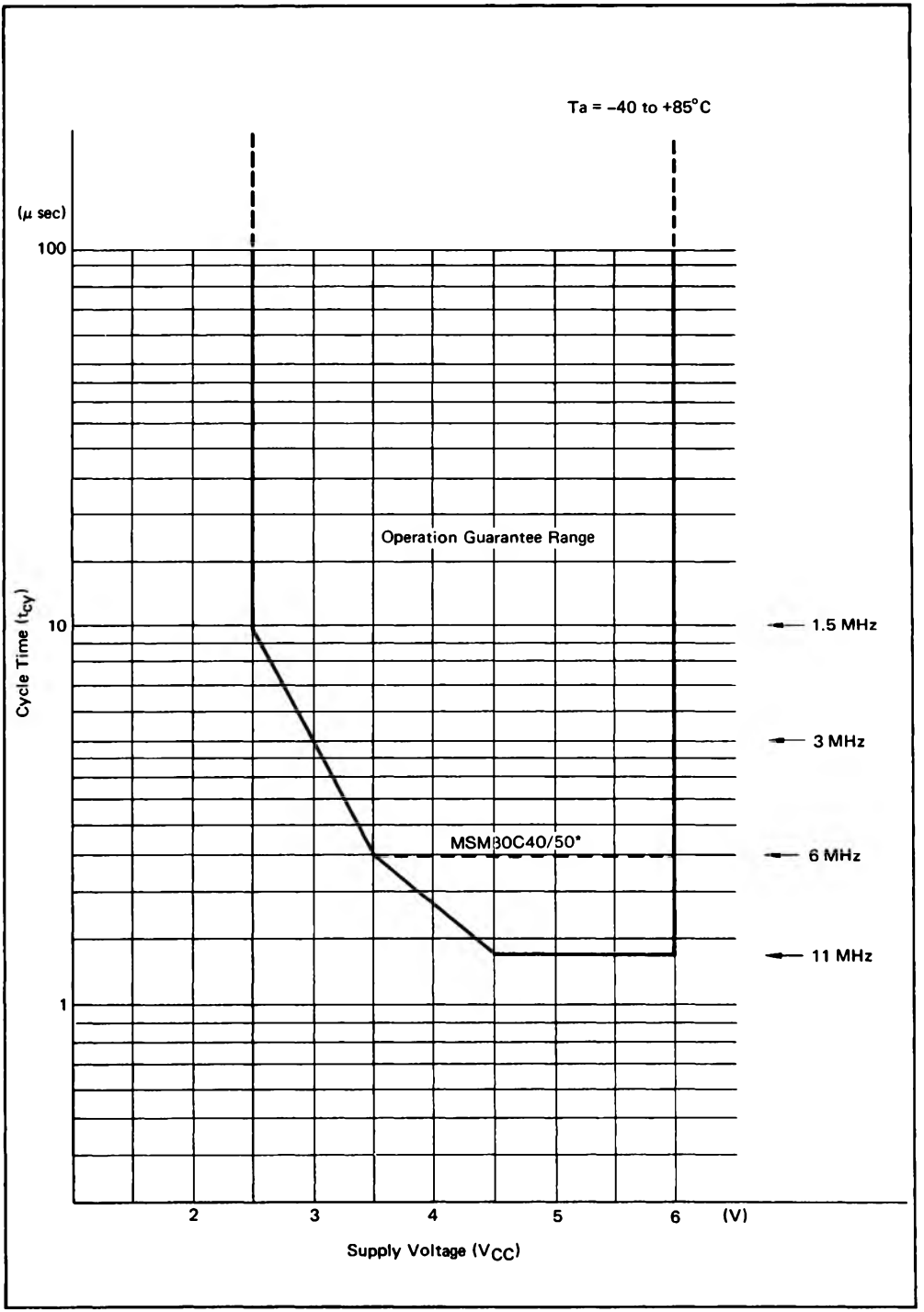
AC CHARACTERISTICS

(V_{CC} = 5V±10%, TA = −40°C to +85°C)

Parameter	Symbol	Limits				Unit
		11 MHz Clock		Variable Clock (0 – 11 MHz)		
		Min.	Max.	Min.	Max.	
Cycle Time	t_{CY}	1.36		1.36		μS
ALE Pulse Width	t_{LL}	150		$7/30t_{CY}-165$		ns
Address Set up ALE	t_{AL}	70		$2/15t_{CY}-110$		ns
Address Hold from ALE	t_{LA}	50		$1/15t_{CY}-40$		ns
Bus Port Latch Data Setup to ALE	t_{BL}	110		$5/30t_{CY}-115$		ns
Bus Port Latch Data Hold from ALE	t_{LB}	90		$3/30t_{CY}-45$		ns
Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$)	t_{CC}	300		$6/15t_{CY}-245$		ns
Data Setup before $\overline{WR}$	t_{DW}	250		$6/15t_{CY}-295$		ns
Data Hold after $\overline{WR}$	t_{WD}	40		$2/15t_{CY}-140$		ns
Data Hold after $\overline{RD}$	t_{DR}	0	100	0	100	ns
$\overline{PSEN}$, $\overline{RD}$ to Data-in	t_{RD}		200		$5/15t_{CY}-250$	ns
Address Setup to $\overline{WR}$	t_{AW}	200		$6/15t_{CY}-345$		ns
Address Setup to Data-in	t_{AD}		400		$8/15t_{CY}-325$	ns
Address Float to $\overline{RD}$, $\overline{PSEN}$	t_{AFC}	0		0		ns
Port Control Setup to $\overline{PROG}$	t_{CP}	100		$2/15t_{CY}-80$		ns
Port Control Hold from $\overline{PROG}$	t_{PC}	60		$4/15t_{CY}-300$		ns
$\overline{PROG}$ to P2 Input Valid	t_{PR}	–	650		$9/15t_{CY}-165$	ns
Output Data Setup	t_{DP}	200		$6/15t_{CY}-345$		ns
Output Data Hold	t_{PD}	20		$3/15t_{CY}-250$		ns
Input Data Hold from $\overline{PROG}$	t_{PF}	0	150	0	150	ns
$\overline{PROG}$ Pulse Width	t_{PP}	700		$10/15t_{CY}-205$		ns
Port 2 I/O Setup to ALE	t_{PL}	150		$9/30t_{CY}-255$		ns
Port 2 I/O Hold from ALE	t_{LP}	20		$3/30t_{CY}-115$		ns

Note: Control output: C_L = 80 pF
Bus output: C_L = 150 pF [for 20 pF (t_{WD})]

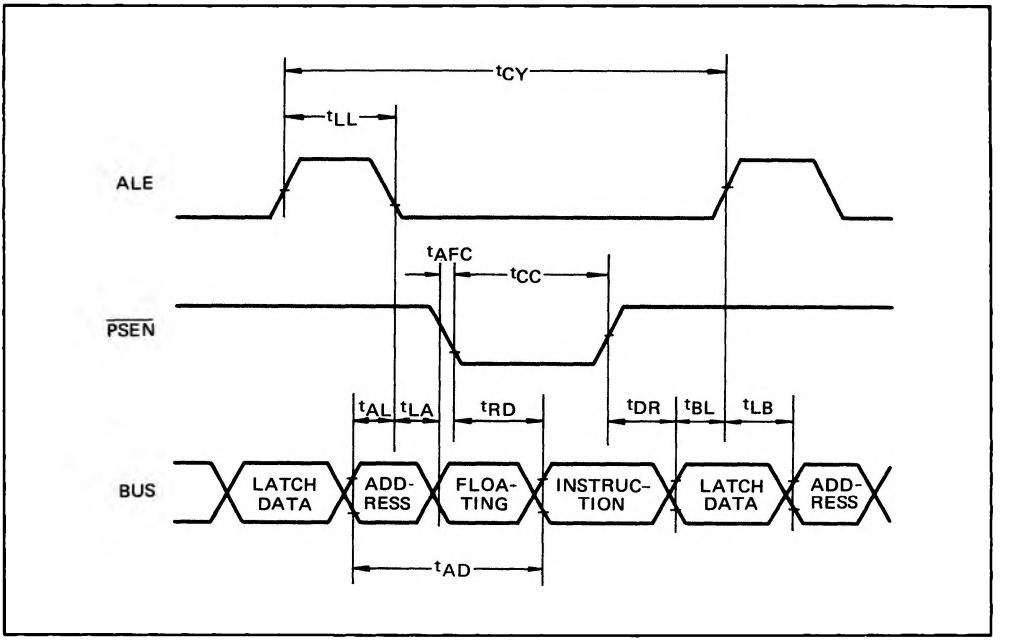
MSM80C49 OPERATION GUARANTEE RANGE



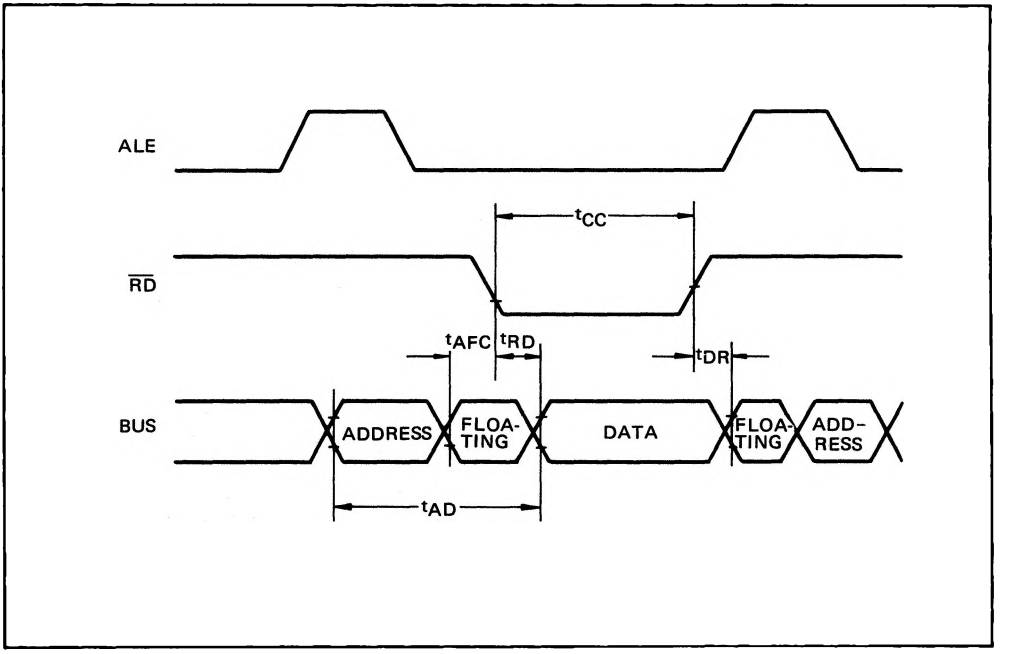
* 11 MHz version of MSM80C40/50 is under development

TIMING CHART

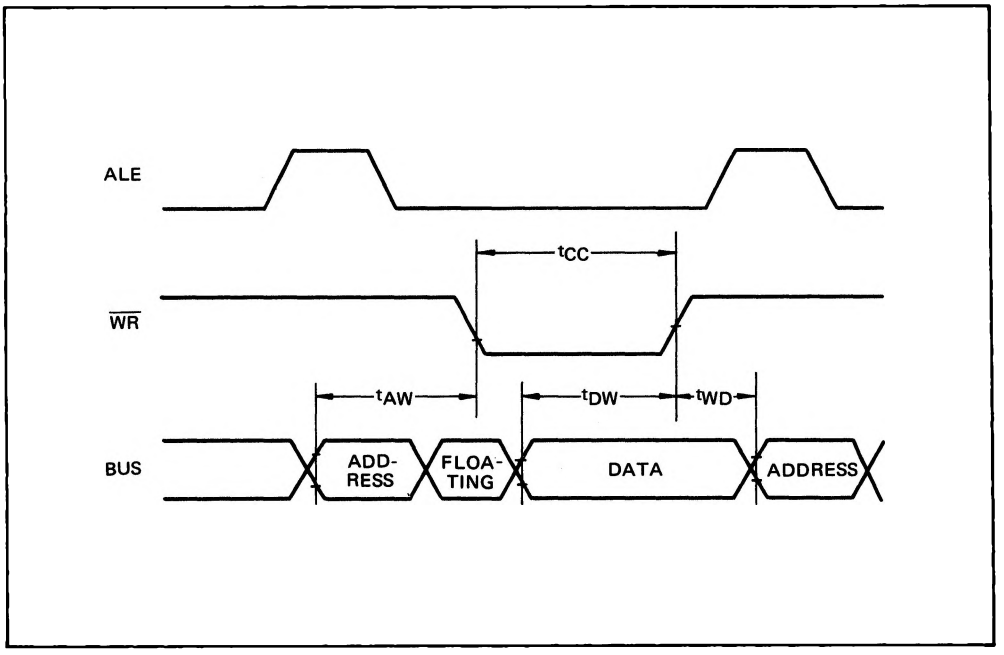
Instruction Fetch (from external program memory)



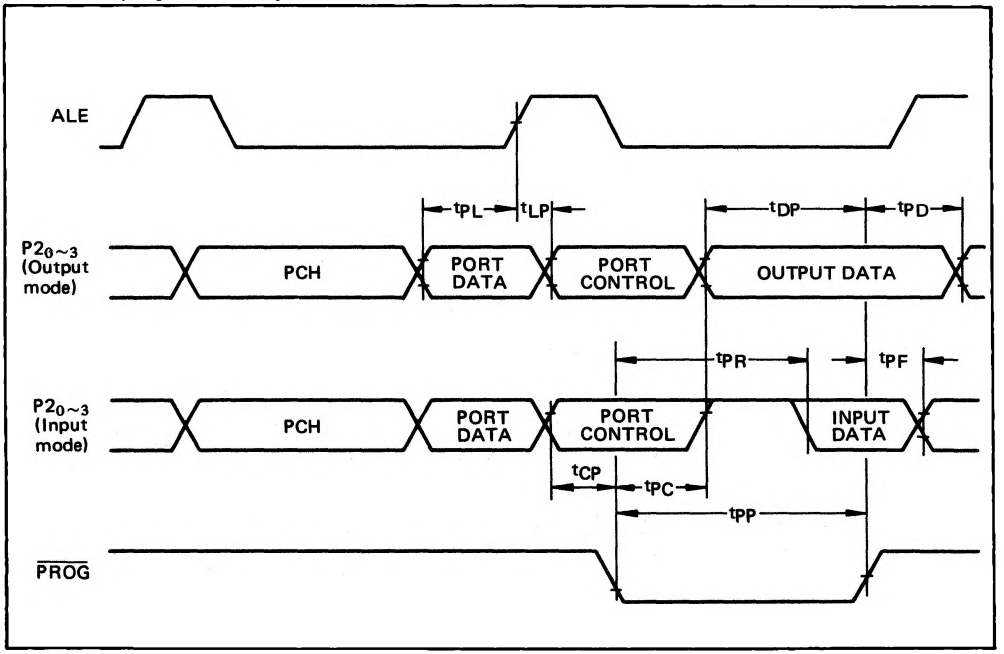
Read (from external data memory)



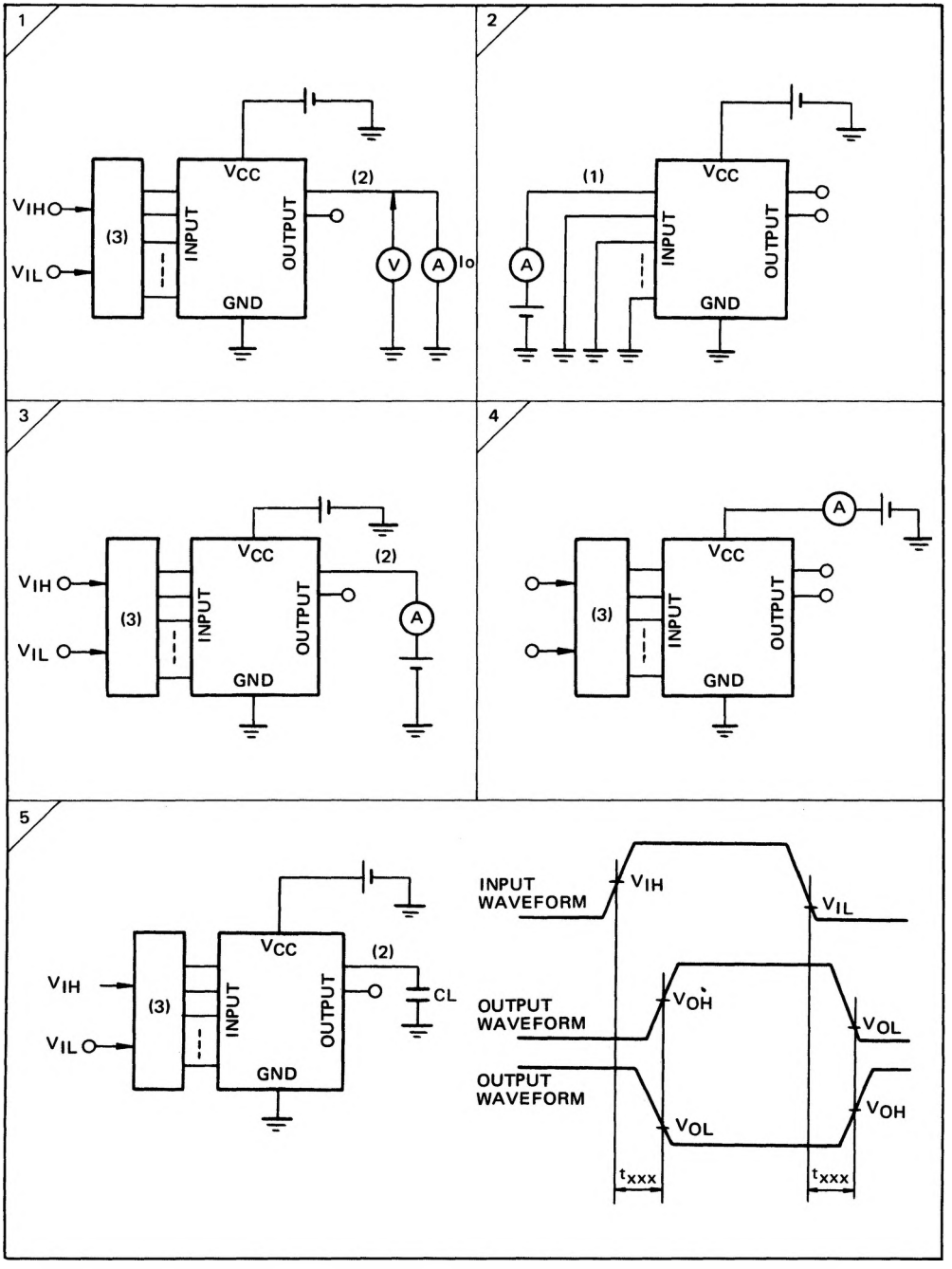
Write (to external memory)



4 low-order bits input/output of port 2 when expanded I/O is used
(in external program memory access mode)



MEASUREMENT CIRCUIT



Notes: (1) This is repeated for each specified input pin.
(2) This is repeated for each specified output pin.
(3) Input logic for setting the specified state.