MBL80C49H/N
MBL80C39H/N
CMOS Single-Chip 8-Bit Microcomputer

November 1984
Edition 1.0

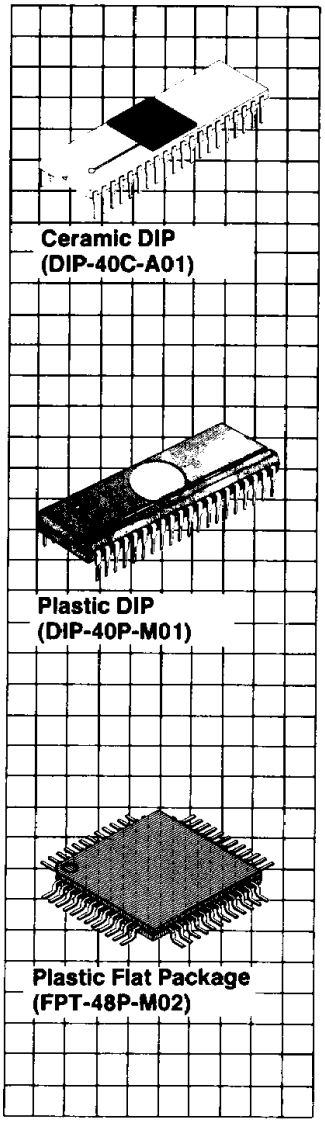
Description

The Fujitsu MBL80C49/MBL80C39 is a totally self-contained 8-bit single-chip microcomputer fabricated with silicon-gate CMOS technology. The MBL80C49 has an 8-bit MPU, a 2K × 8 ROM program memory, and 128 × 8 RAM data memory, 27 I/O ports, an 8-bit timer/counter, and clock generator on chip. The MBL80C39 is identical to the MBL80C49 except without internal program memory. It can be used with external memory for system prototyping and preproduction systems.

The design is optimized for low cost and high performance applications because the MBL80C49/MBL80C39 is fabricated on a single silicon chip and can be used for applications that require additional expansion of ROM, RAM, I/O ports, and so on.

This microcomputer permits external program operation and single-step operation. Low power applications are possible by using the standby-mode feature. The software is upward compatible with the MBL8049/MBL8039 and Intel 8049/8039.

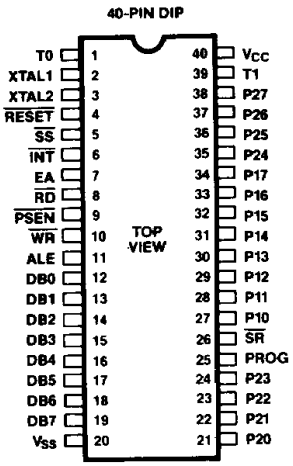
The MBL80C49/MBL80C39 uses a single power supply of +5V. It is packaged in a 40-pin DIP or a 48-pin Flat Package. Operation of N version (6 MHz) is guaranteed over the range of −40°C to +85°C, and H version (11 MHz), 0°C to +70°C.

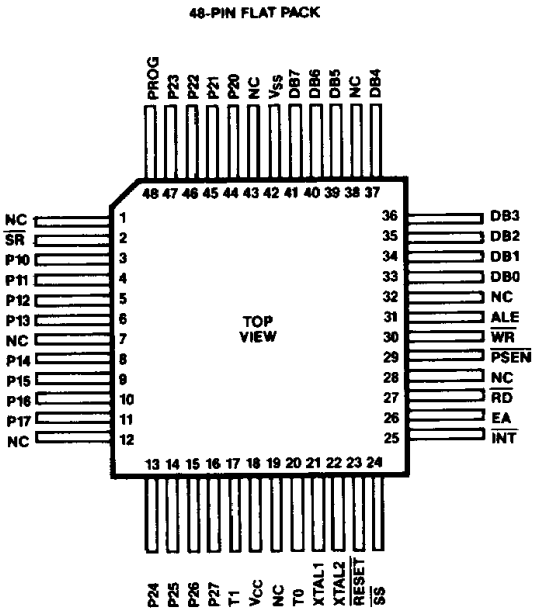


Features

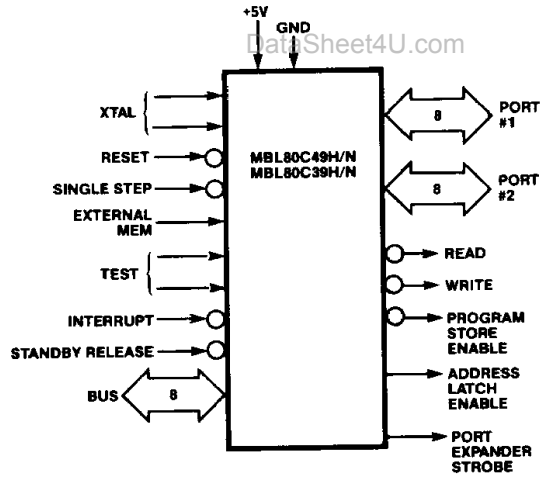
- 8-bit Single-chip Microcomputer
- 12-bit Addressing
- 98 Instructions (232 Instruction Codes): 70% of Instructions are Single Byte.
- 2.5 μs Instruction Cycle for N version, 1.36 μs instruction cycle for H version: All instructions are 1 or 2 cycle.
- ALU Functions: Addition, Decimal Adjust Addition, and Logic Operations
- 2K x 8-bit ROM
- 128 x 8-bit RAM
- 8-level Stack
- 8 pairs of Working Registers
- 8-bit Interval Timer/Event Counter
- 27 I/O Lines: Two 8-bit I/O Ports, One Data Bus, Two Test Pins and One Interrupt
- Easily Expandable Memory and I/O
- On-chip Clock Generator (or External Clock)
- Single-level Interrupt Capability
- Single-step Operation Capability
- External Program Mode Capability
- Low-power Standby Mode Capability by HALT and STOP Instructions
- Single +5V Power Supply
- Silicon-gate CMOS Technology
- Standard 40-pin DIP (Ceramic/Plastic)
- Standard 48-pin Flat Package
- Compatible with Intel 8049/8039 and Fujitsu MBL8049/MBL8039

Pin Assignment

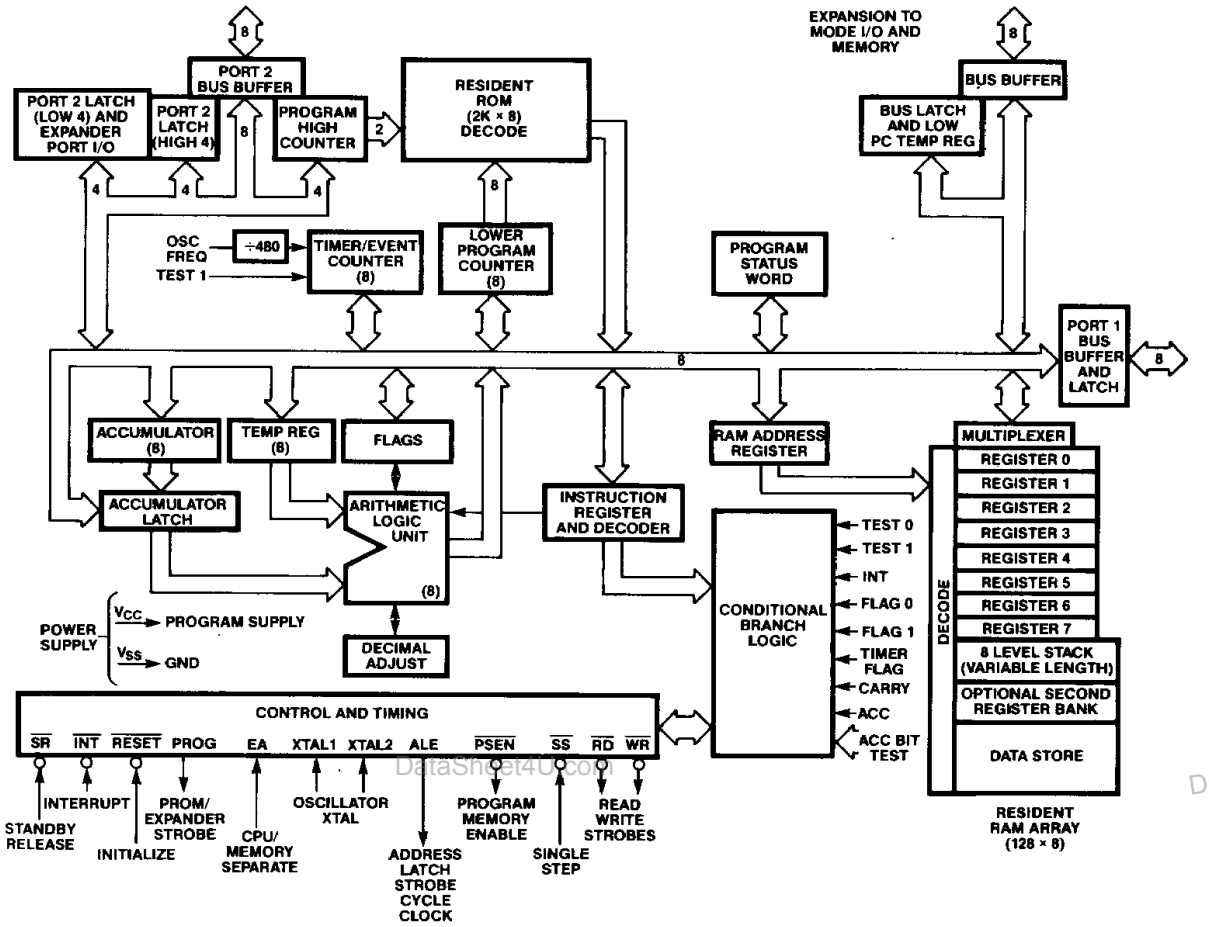




Logic Symbol



Block Diagram



Pin Description

Symbol	Name	Pin No.*	Type	Function
V_{CC}	Power Supply	40 (18)	—	Main power supply; +5V during operation.
V_{SS}	Ground	20 (42)	—	Circuit GND potential.
XTAL1	Crystal 1	2 (21)	I	One side of crystal input for internal oscillator. Also input for external source. (Non-TTL level input)
XTAL2	Crystal 2	3 (22)	I	Other side of crystal input.
PROG	Program	25 (48)	O	Output strobe for MBL82C43 I/O expander.
P10-P17	Port 1**	27-34 (3-6, 8-11)	I/O	8-bit quasi-bidirectional port.
P20-P23	Port 2**	21-24 (44-47)	I/O	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for MBL82C43.
$\overline{P24}$-$\overline{P27}$		35-38 (13-16)		

Pin Description
(Continued)

Symbol	Name	Pin No.*	Type	Function
DB0–DB7	Data Bus	12–19 (33–37, 39–41)	I/O	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	Test 0	1 (20)	I/O	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.
T1	Test 1	39 (17)	I	Input pin testable uising the JT1, and JNT1 instructions. Can be designated as the event counter input using the STRT CNT instruction.
$\overline{INT}$	Interrupt Request	6 (25)	I	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. (Active low). Interrupt must remain low for at least 3 machine cycles to ensure proper operation.
$\overline{RESET}$	Reset**	4 (23)	I	Input used to initialize the processor. (Active low) (Non-TTL level input)
$\overline{RD}$	Read	8 (27)	O	Output strobe activated during a BUS read. Can be used to enable data onto the BUS from an external device. (Active low) Used as a read strobe to external data memory.
$\overline{WR}$	Write	10 (30)	O	Output strobe during a BUS write. (Active low) Used as write strobe to external data memory.
ALE	Address Latch Enable	11 (31)	O	This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	Program Store Enable	9 (29)	O	This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	Single Step**	5 (24)	I	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	External Access	7 (26)	O	External Access forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
$\overline{SR}$	Standby Release**	26 (2)	I	This is the control input for standby operation. A low level on this input releases the MPU from the standby mode.

Note: *Bracketed value is applied to Flat Package.
**These pins are internally pulled up.

2

Functional Description

Architecture

The following sections break the MBL80C49 into function blocks and describe each in detail. See Block Diagram.

Arithmetic Section

The arithmetic section of the processor contains the basic data manipulation functions of the MBL80C49 and can be divided into the following blocks:

- Arithmetic Logic Unit (ALU)
- Accumulator
- Carry Flag
- Instruction Decoder

In a typical operation, data stored in the accumulator is combined in the ALU with data from another source on the internal bus (such as a register or I/O port). The result is stored in the accumulator or another register.

The following is a detailed description of the function of each block.

Instruction Decoder

The operation code (op code) portion of each program instruction is stored in the Instruction Decoder and converted to outputs which control the function of each of the blocks of the Arithmetic Section. These lines control the source of data and the destination register as well as the function performed in the ALU.

Arithmetic Logic Unit

The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under control of the Instruction Decoder. The ALU can perform the following functions:

- Add With or Without Carry
- AND, OR, Exclusive OR
- Increment/Decrement
- Bit Complement
- Rotate Left, Right
- Swap Nibbles
- BCD Decimal Adjust

If the operation performed by the ALU results in a value represented by more than 8 bits (overflow of most significant bit), a Carry Flag is set in the Program Status Word.

Accumulator

The accumulator is the single most important data register in the processor, being one of the sources of input to the ALU and often the destination of the result of operations performed in the ALU. Data to and from I/O ports and memory also normally passes through the accumulator.

Program Memory

Resident program memory consists of 2048 words eight bits wide which are addressed by the program counter. In the MBL80C49 the memory is ROM which is mask programmable at the factory. The MBL80C39 has no internal program memory and is used with external memory devices. Program code is completely interchangeable among the various versions. To access the upper 2K of program memory in the MBL80C49, a select memory bank and a JUMP or CALL instruction must be executed to cross the 2K boundary.

There are three Program Memory locations of special importance as shown in Fig. 1.

Location 0

Activating the Reset line of the processor causes the first instruction to be fetched from location 0.

Location 3

Activating the Interrupt input line of the processor (if interrupt is enabled) causes a jump to subroutine at location 3.

Location 7

A timer/counter interrupt resulting from timer/counter overflow (if enabled) causes a jump to subroutine at location 7.

Therefore, the first instruction to be executed after initialization is stored in location 0, the first word of an external interrupt service subroutine is stored in location 2, and the first word of a timer/counter service routine is stored in location 7. Program memory can be used to store constants as well as program instructions. Instructions such as MOVP and MOVP3 allow easy access to data "look-up" tables.

Data Memory

Resident data memory is organized as 128 words 8-bits wide in the MBL80C49/MBL80C39. All locations are indirectly addressable through either of two RAM Pointer Registers which reside at address 0 and 1 of the register array. In addition, as shown in Fig. 2, the first 8 locations (0–7) of the array are designated as working registers and are directly addressable by several instructions. Since these registers are more easily addressed, they are usually used to store frequently accessed intermediate results. The DJNZ instruction makes very efficient use of the working registers as program loop counters by allowing the programmer to decrement and test the register in a single instruction.

By executing a Register Bank Switch instruction (SEL RB) RAM location 24–31 are designated as the working registers in place of locations 0–7 and are then directly addressable. This second bank of working registers may be used as an extension of the first bank or reserved for use during interrupt service subroutines allowing the registers of Bank 0 used in the main program to be instantly "saved" by a Bank Switch. Note that if this second bank is not used, locations 24–31 are still addressable as general purpose RAM. Since the two RAM pointer Registers R0 and R1 are a part of the working register array, bank switching

Functional Description
(Continued)

effectively creates two more pointer registers (R0/ and R1/) which can be used with R0 and R1 to easily access up to four separate working areas in RAM at one time. RAM locations (8-23) also serve a dual role in that they contain the program counter stack. These locations are addressed by the Stack Pointer during subroutine calls as well as by RAM Pointer Registers R0 and R1. If the level of subroutine nesting is less than 8, all stack registers are not required and can be used as general purpose RAM locations. Each level of subroutine nesting not used provides the user with two additional RAM locations.

Input/Output

The MBL80C49 has 27 lines which can be used for input or output functions. These lines are grouped as 3 ports of 8 lines each which serve as either inputs, outputs of bidirectional ports and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8 bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of special output circuit structure which allows each line to serve as an input, and output, or both even though outputs are statically latched. Each line is continuously pulled up to V_{CC} through a resistive device of relatively high impedance.

This pullup is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. To provide fast switching times in a "0" to "1" transition to relatively low impedance device is switched in momentarily (≈1/5 of a machine cycle) whenever a "1" is written to the line. When a "0" is written to the line a low impedance device overcomes the light pullup and provides TTL current sinking capability. Since the pulldown transistor is a low impedance device a "1" must first be written to any line which is to be used as an input. Reset initializes all lines to the high impedance "1" state.

It is important to note that the ORL and ANL are read/write operations. When executed, the μC "reads" the port, modifies the data according to the instruction, then "writes" the data back to the port. The "writing" (essentially an OUTL instruction) enables the low impedance pullup momentarily again even if the data was unchanged from a "1". This specifially applies to the configurations that have inputs and outputs mixed together on the same port.

Bus

Bus is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, Bus can serve as either a statically latched output port or non-latched input port. Input and output lines on this port cannot be mixed however.

As a static port, data is written and latched using the OUTL instruction and inputted using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$

Figure 1. Program Memory Map

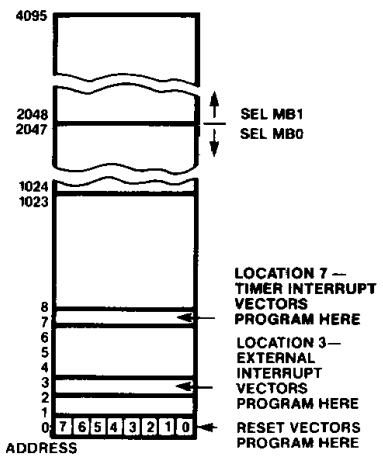
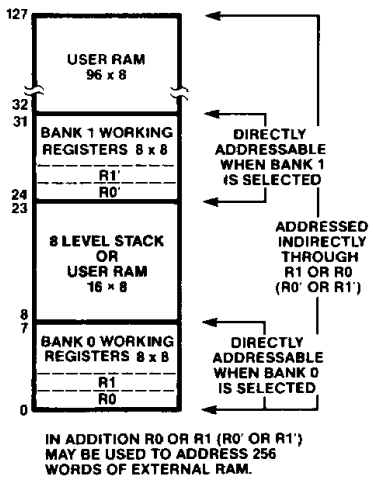


Figure 2. Data Memory Map



Functional Description
(Continued)

output line and input data must be valid at the trailing edge of RD. When not being written or read, the BUS lines are in a high impedance state.

Test and INT Inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These are T0, T1, and INT. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1, and INT pins have other possible functions as well. See the pin description in page 3.

Program Counter and Stack

The Program Counter is an independent counter while the Program Counter Stack is implemented using pairs of registers in the Data Memory Array. Only 11 bits of the Program Counter are used to address the 2048 words of on-board program memory of the MBL80C49, while the most significant bits can be used for external Program Memory fetches. See Fig. 3. The Program Counter is initialized to zero by activating the Reset line.

An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack as shown in Fig. 7. The pair to be used is determined by a 3-bit Stack Pointer which is part of the Program Status Word (PSW).

Data RAM locations 8-23 are available as stack registers and are used to store the Program Counter and 4 bits of PSW as shown in Fig. 7. The Stack Pointer when initialized to 000 points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transfered to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without oveflowing the stack.

If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the Stack Pointer to be decremented and the contents of the resulting register pair to be transferred to the Program Counter.

Program Status Word

An 8-bit status word which can be loaded to and from the accumulator is called the Program Status Word (PSW). Fig. 5 shows the information available in the word. The Program Status Word is actually a collection of flip-flops throughout the machine which can be read or written as a whole. The ability to write to PSW allows for easy restoration of machine status after a power down sequence.

The upper four bits of PSW are stored in the Program Counter Stack with every call to subroutine or interrupt vector and are optionally restored upon return with the RETR instruction. The RET return instruction does not update PSW.

Figure 3. Program Counter

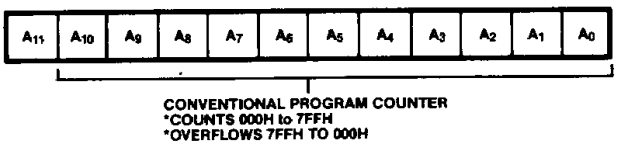
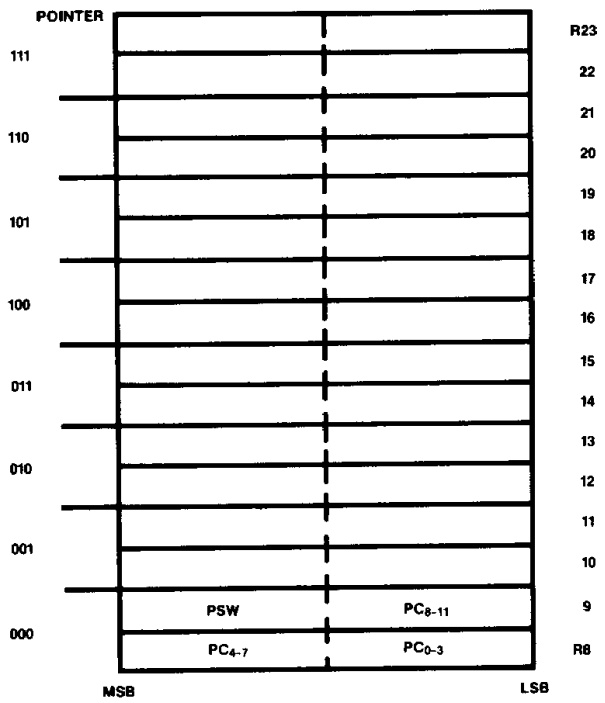


Figure 4. Program Counter Stack



Functional Description
(Continued)

The PSW bit definitions are as follows:

Bits 0-2: Stack Pointer bits (S_0, S_1, S_2)

Bit 3: Not used ("1" level when read)

Bit 4: Working Register Bank Switch Bit (BS)
0 = Bank 0
1 = Bank 1

Bit 5: Flag 0 bit (F0) user controlled flag which can be complemented or cleared, and tested with the conditional jump instruction JF0.

Bit 6: Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DAA.

Bit 7: Carry (CY) carry flag which indicates that the previous operation has resulted in overflow of the accumulator.

Conditional Branch Logic

The conditional branch logic within the processor enables several conditions internal and external to the processor to be tested by the users program. By using the conditional jump instruction the conditions that are listed in Table 2 can effect a change in the sequence of the program execution.

Timer/Counter

The MBL80C49 contains a counter to aid the user in counting external events and generating accurate time delays without placing a burden on the processor for these functions. In both modes the counter operation is the same, the only difference being the source of the input to the counter. The timer/event counter is shown in Fig. 6.

Counter

The 8-bit binary counter is presettable and readable with two MOV instructions which transfer the contents of the accumulator to the counter and vice versa. The counter content may be affected by Reset and should be initialized by software. The counter is stopped by a Reset or STOP TCNT instruction and remains stopped until started as a timer by a START T instruction or as an event counter by a START CNT instruction. Once started the counter will increment to this maximum count (FF) and overflow to zero continuing its count until stopped by a STOP TCNT instruction or Reset.

The increment from maximum count to zero (overflow) results in the setting of an overflow flag flip-flop and in the generation of an interrupt request. The state of the overflow flag is testable with the conditional jump instruction JTF. The flag is reset by executing a JTF or by Reset. The interrupt request is stored in a latch and then ORed with the external interrupt input INT. The timer interrupt may be enabled or disabled independently of external interrupt by the EN TCNTI and DISTCNTI instructions. If enabled, the counter overflow will cause a subroutine call to location 7 where the timer or counter service routine may be stored.

If timer and external interrupts occur simultaneously, the external source will be recognized and the Call will be to location 3. Since the timer interrupt is latched it will remain pending until the external device is serviced and immediately be recognized upon return from the service routine. The pending timer interrupt is reset by the Call to location 7 or may be removed by executing a DIS TCNTI instruction.

As An Event Counter

Execution of a START CNT instruction connects the T1 input pin to the counter input and enables the counter. The T1 input is sampled at the beginning of state 3 or in state time 4. Subsequent high to low transistions on T1 will cause the counter to increment. T1 must be held low for at least 1 machine cycle to insure it won't be missed. The maximum

Figure 5. Program Status Word (PSW)

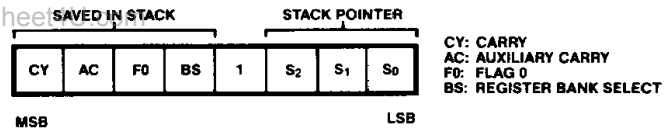


Figure 6. Timer/Event Counter

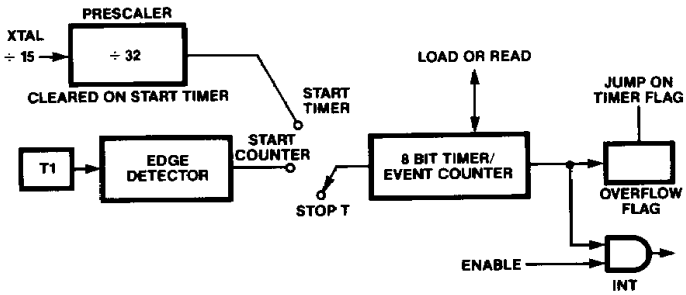


Table 2

Device Testable	Jump Conditions (Jump On)	
Accumulator	All zeros	Not all zeros
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	—

Functional Description
(Continued)

rate at which the counter may be incremented is once per three instruction cycles (every 5.7 μsec when using an 8 MHz crystal)—there is no minimum frequency. T1 input must remain stable for at least 1/5 machine cycle after each transition.

As A Timer

Execution of a START T instruction connects an internal clock to the counter input and enables the counter. The internal clock is derived by passing the basic machine cycle clock through a ÷ 32 prescaler. The prescaler is reset during the START T instruction. The resulting clock increments the counter every 32 machine cycle. Various delays from 1 to 256 counts can be obtained by presetting the counter and detecting overflow. Times longer than 256 counts may be achieved by accumulating multiple overflows in a register under software control. For time resolution less than 1 count, an external clock can be applied to the T1 input and the counter operated in the event counter mode. ALE divided by 3 or more can serve as this external clock. Very small delays or "fine tuning" of larger delays can be easily accomplished by software delay loops.

Clock and Timing Circuits

Timing generation for the MBL80C49 is completely self-contained with the exception of a frequency reference which can be XTAL, ceramic resonator, or external clock source. The clock and Timing circuitry can be divided into the following functional blocks.

Oscillator

The on-board oscillator is a high gain parallel resonant circuit with a frequency range of 1 to 11 MHz. The X1 external pin is the input to the amplifier stage while X2 is the output. A crystal or ceramic resonator connected between X1 and X2 provides the feedback and phase shift required for oscillatin. If an accurate frequency reference is not required, ceramic resonator may be used in place of the crystal.

For accurate clocking a crystal should be used. An externally generated clock may also be applied to X1-X2 as the frequency source.

State Counter

The output of the oscillator is divided by 3 in the State counter to create a clock which defines the state times of the machine (CLK). CLK can be made available on the external pin T0 by executing an ENT0 CLK instruction. The output of CLK on T0 is disabled by Reset of the processor.

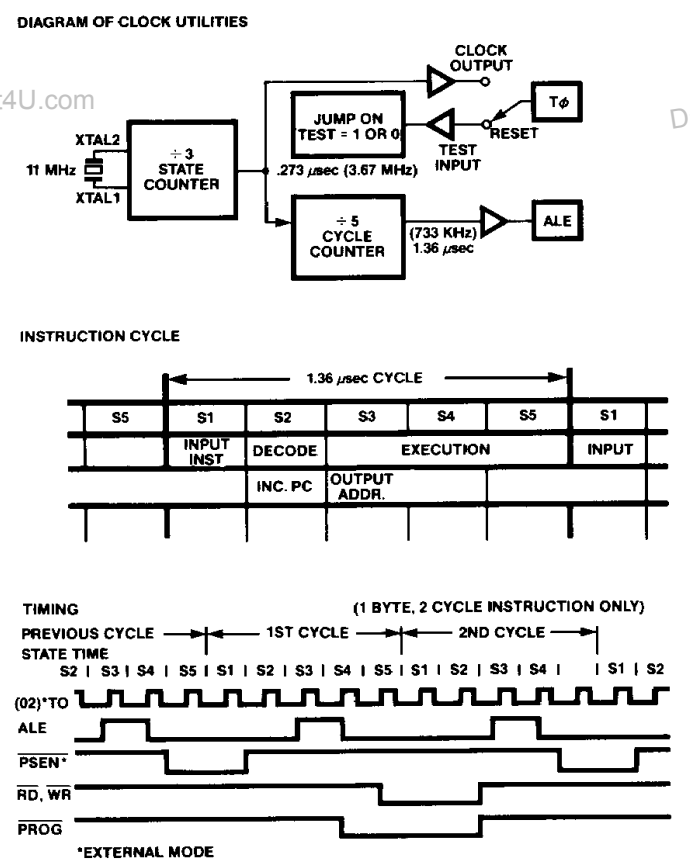
Cycle Counter

CLK is then divided by 5 in the Cycle Counter to provide a clock which defines a machine cycle consisting of 5 machine states as shown in Fig. 7. This clock is called Address Latch Enable (ALE) because of its function in MBL80C49 with external memory. It is provided continously on the ALE output pin.

Reset

The reset input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pull-up device, which in combination with an external 1μfd capacitor, provides an internal reset pulse of sufficient length to guarantee all circuitry is reset, as shown in Fig. 8. If the reset pulse is generated externally the RESET pin must be held low for at least 10 milliseconds after the power supply is within tolerance. Only 5 machine cycles (6.8 μs @ 11 MHz) are required if power is already on and the oscillator has stablized. ALE and PSEN (if EA = 1) are active while in Reset.

Figure 7. Timing Generation and Cycle Timing



Functional Description
(Continued)

Reset performs the following functions:

1) Sets program counter to zero.
2) Sets stack pointer to zero.
3) Selects register bank 0.
4) Selects memory bank 0.
5) Sets BUS to high impedance state (except when EA = 5V).
6) Sets Ports 1 and 2 to input mode.
7) Disables interrupts (timer and external).
8) Stops timer.
9) Clears timer flag.
10) Clears F0 and F1.
11) Disables clock output from T0.

Interrupt

An interrupt sequence is initiated by applying a low ("0") level input to the $\overline{INT}$ pin. Interrupt is level triggered and active low to allow "WIRE ORing" of several interrupt sources at the input pin. Fig. 9 shows the interrupt logic of the MBL80C49. The interrupt line is sampled every instruction cycle and when detected causes a "call to subroutine" at location 3 in program memory as soon as all cycles of the current instruction are complete. On 2-cycle instructions, the interrupt line is sampled on the 2nd cycle only. $\overline{INT}$ must be held low for at least 3 machine cycles to ensure proper interrupt operations. As in any CALL to subroutine, the Program Counter and Program Status word are saved in the stack. For a description of this operation see the previous section, Program Counter and Stack. Program Memory location 3 usually contains an unconditional jump to an interrupt service subroutine elsewhere in program memory. The end of an interrupt service subroutine is signalled by the execution of a Return and Restore Status instruction RETR. The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR reenables the interrupt input logic. This occurs at the beginning of the second cycle of the RETR instruction. This sequence holds true also for an internal interrupt generated by timer overflow. If an internal timer/counter generated interrupt and an external interrupt are detected at the same time, the external source will be recognized. See the following Timer/Counter section for a description of timer interrupt. If needed, a second external interrupt can be created by enabling the timer/counter interrupt, loading FFH in the Counter (one less than terminal count), and enabling the event counter mode. A "1" to "0" transition on the T1 input will then cause an interrupt vector to location 7.

Interrupt Timing

The interrupt input may be enabled or disabled under Program Control using the EN I and DIS I instructions. Interrupts are disabled by Reset and remain so until enabled by the users program. An interrupt request must be removed before the RETR instruction is executed upon return from the service routine

Figure 8. Reset Circuit

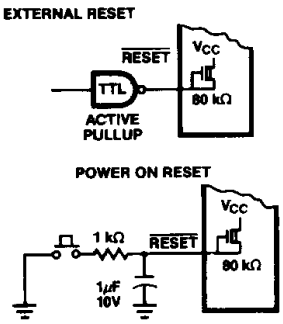
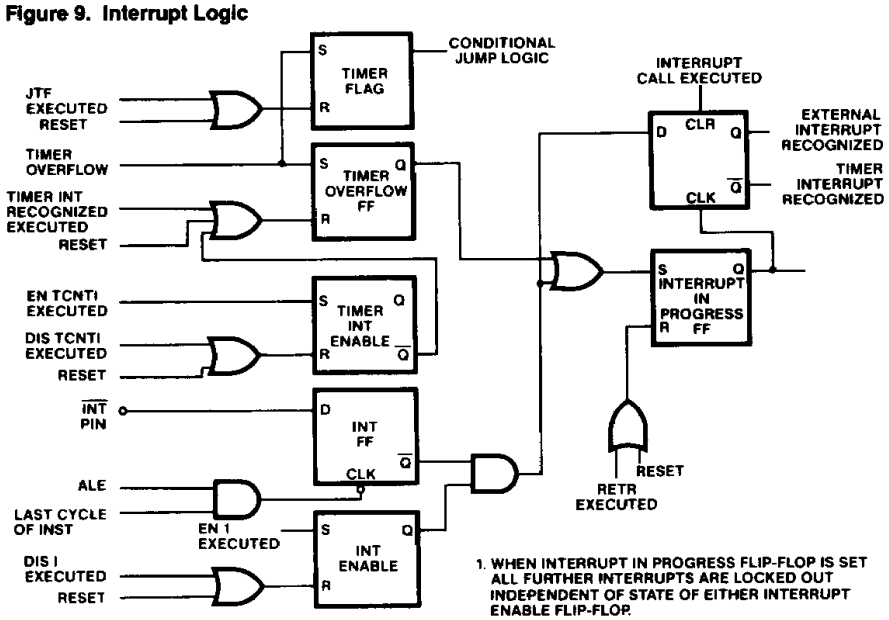


Figure 9. Interrupt Logic



1. WHEN INTERRUPT IN PROGRESS FLIP-FLOP IS SET ALL FURTHER INTERRUPTS ARE LOCKED OUT INDEPENDENT OF STATE OF EITHER INTERRUPT ENABLE FLIP-FLOP.
2. WHILE TIMER INTERRUPTS ARE DISABLED TIMER OVERFLOW I/I WILL NOT STORE ANY OVERFLOW THAT OCCURS. TIMER FLAG WILL BE SET, HOWEVER.

Functional Description
(Continued)

otherwise the processor will re-enter the service routine immediately. Many peripheral devices prevent this situation by resetting their interrupt request line whenever the processor accesses (Reads or Writes) the peripherals data buffer register. If the interrupting device does not require access by the processor, one output line of the MBL80C49 may be designated as an "interrupt acknowledge" which is activated by the service subroutine to reset the interrupt request. The $\overline{INT}$ pin may also be tested using the conditional jump instruction JNI. This instruction may be used to detect the presence of a pending interrupt before interrupts are enabled. If interrupt is left disabled $\overline{INT}$ may be used as another test input like T0 and T1.

Single-Step

This feature, as pictured in Fig. 10, provides the user with a debug capability in that the processor can be stepped through the program one instruction at a time. When stopped, the address of the next instruction to be fetched is available concurrently on BUS and the lower half of Port 2. The user can therefore follow the program through each of the instruction steps. A timing diagram, showing the interaction between output ALE and input $\overline{SS}$, is shown. The BUS buffer contents are lost during single step; however, a latch may be added to reestablish the lost I/O capability if needed. Data is valid at the leading edge of ALE.

Timing

The MBL80C49 operates in a single-step mode as follows:
1) The processor is requested to stop by applying a low level on $\overline{SS}$.
2) The processor responds by stopping during the address fetch portion of the next instruction. If a double cycle instruction is in progress when the single step command is received, both cycles will be completed before stopping.
3) The processor acknowledges it has entered the stopped state by raising ALE high. In this state (which can be maintained indefinitely) the address of the next instruction to be fetched is present on BUS and the lower half of port 2.
4) $\overline{SS}$ is then raised high to bring the processor out of the stopped mode allowing it to fetch the next instruction. The exit from stop is indicated by the processor bringing ALE low.
5) To stop the processor at the next instruction $\overline{SS}$ must be brought low again soon after ALE goes low. If $\overline{SS}$ is left high the processor remains in a "Run" mode.

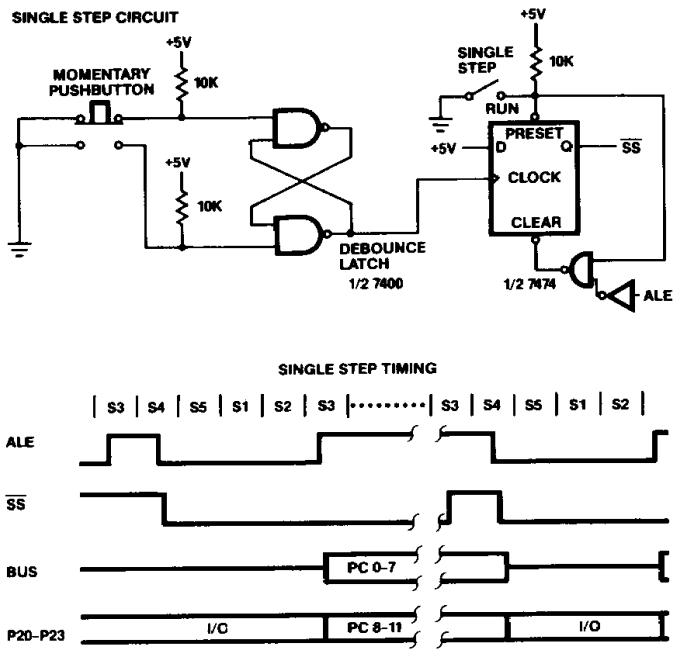
A diagram for implementing the single-step function of the MBL80C49 is shown in Fig. 10. A D-type flip-flop with preset and clear is used to generate $\overline{SS}$. In the run mode, $\overline{SS}$ is held high by keeping the flip-flop preset (preset has precedence over the clear input). To enter single step, preset is removed allowing ALE to bring $\overline{SS}$ low via the clear input. ALE should be buffered since the clear input of an SN7474 is the equivalent of 3 TTL loads. The processor is now in the stopped state. The next instruction is initiated by clocking a "1" into the flip-flop. This "1" will not appear on $\overline{SS}$ unless ALE is high, removing clear from the flip-flop. In response to $\overline{SS}$ going high the processor begins an instruction fetch which brings ALE low, resetting $\overline{SS}$ through the clear input and causing the processor to again enter the stopped state.

External Access Mode

Normally the first 2K words of program memory are automatically fetched from internal ROM. The EA input pin however allows the user to effectively disable internal program memory by forcing all program memory fetches to reference external memory. The following chapter explains how access to external program memory is accomplished.

The External Access mode is very useful in system test and debug because it allows the user to disable his internal applications program and substitute an external program of his choice—a diagnostic routine for instance. In addition, the section on Test

Figure 10. Single Step Operation



Functional Description
(Continued)

and Debug explains how internal program memory can be read externally, independent of the processor. A "1" level on EA initiates the external access mode. For proper operation, Reset should be applied while the EA input is changed.

Low Power Standby Operation

MBL80C49 has two low-power standby modes, HALT and STOP modes. Both are initiated by software and are released by hardware.

Halt Mode

This mode is initiated by the HALT instruction (OP code: 01H). When the HALT instruction is executed, the device enters the HALT mode and stops executing instructions after that.

During the HALT mode all the other internal circuits stop, except the on-chip oscillator. The oscillator idles and the clock is inhibited to the internal circuits. This time the program counter holds the address next to the HALT instruction, and the internal RAM, internal registers, and flags keep the states executing the HALT instruction. The input/output ports hold the states shown below:

Type	Pin	State
Input	XTAL1, XTAL2, $\overline{EA}$, $\overline{RESET}$, $\overline{INT}$, $\overline{SR}$	Active
	T1, $\overline{SS}$	Inactive
Output	$\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	High Level
	P10–17, P20–27, DB0–DB7, T0, $\overline{PROG}$	High Impedance

In this mode, the supply voltage, V_{CC} can be lowered to 3.5V, and the supply current reduced to 2mA for N version and 4mA for H version.

A negative pulse at one of $\overline{RESET}$, $\overline{INT}$, and $\overline{SR}$ inputs releases the device from the HALT mode. When a pulse is applied, the device restarts executing instructions from an address shown below, depending on the applied pin. Before applying the negative pulse, the supply voltage must be returned to the value (+5V ±10%) to guarantee normal operation.

$\overline{RESET}$ (Low level sense):

A negative pulse with minimum 12 instruction cycle width is needed for the standby release. The low level releases the device from the standby state, and initializes the device to the reset state. Approximately 8200 instruction cycles later $\overline{RESET}$ returns to high and the device restarts the execution from address #0.

$\overline{INT}$ (Low level sense):

The low level releases the device from the standby state, and after a dummy cycle of approximately 5 instruction cycles the device restarts the execution from an address next to the HALT instruction executed. If the external interrupt is enabled, the control jumps at address #3 after executing the instruction next to the HALT instruction executed. The $\overline{INT}$ pulse must be kept low until the interrupt is accepted.

$\overline{SR}$ (Low level sense):

A negative pulse with minimum 2 instruction cycle is required for the standby release. The low level releases the device from the standby state, and after a dummy cycle of approximately 5 instruction cycles the device restarts the execution from an address next to the HALT instruction executed.

Stop Mode

This mode is initiated by the STOP instruction (OP code: C1H). When the STOP instruction is executed, the device enters the STOP mode and stops executing instructions after that.

During the STOP mode all the internal circuits stop. Only the internal RAM is retained. The internal registers and flags are not kept. The input/output ports hold the states shown below:

Type	Pin	State
Input	$\overline{EA}$, $\overline{RESET}$, $\overline{INT}$, $\overline{SR}$	Active
	XTAL1, XTAL2, T1, $\overline{SS}$	Inactive
Output	$\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	High Level
	P10–17, P20–27, DB0–DB7, T0, $\overline{PROG}$	High Impedance

In this mode, the supply voltage, V_{CC} can be lowered to 2.0V, and the supply current reduced to 50μA.

A negative pulse at $\overline{RESET}$ input releases the device from the STOP mode. When a $\overline{RESET}$ pulse is applied, the device restarts executing instructions from an address #0. Before applying the negative pulse, the supply voltage must be returned to the value (+5V ±10%) to guarantee normal operation.

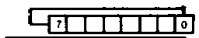
$\overline{RESET}$ (Low level sense):

A negative pulse with minimum 12 instruction cycle width is needed for the standby release. The low level releases the device from the standby state, and initializes the device to the reset state. Approximately 8200 instruction cycles after $\overline{RESET}$ returns to high, the device restarts the execution from address #0.

2

Instruction Set Summary†

Accumulator and Memory Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Add Register to A	ADD A, Rr	6X	1	1	*	*	—	—	(A) ← (A) + (Rr)
Add data memory to A	ADD A, @R0	60	1	1	*	*	—	—	(A) ← (A) + ((R0))
	ADD A, @R1	61	1	1	*	*	—	—	(A) ← (A) + ((R1))
Add immediate to A	ADD A, #data	03	2	2	*	*	—	—	(A) ← (A) + data
Add register to A with carry	ADDC A, Rr	7X	1	1	*	*	—	—	(A) ← (A) + (Rr) + (C)
Add data memory to A with carry	ADDC A, @R0	70	1	1	*	*	—	—	(A) ← (A) + ((R0)) + (C)
	ADDC A, @R1	71	1	1	*	*	—	—	(A) ← (A) + ((R1)) + (C)
Add immediate to A with carry	ADDC A, #data	13	2	2	*	*	—	—	(A) ← (A) + data + (C)
And register to A	ANL A, Rr	5X	1	1	—	—	—	—	(A) ← (A) ∩ (Rr)
And data memory to A	ANL A, @R0	50	1	1	—	—	—	—	(A) ← (A) ∩ ((R0))
	ANL A, @R1	51	1	1	—	—	—	—	(A) ← (A) ∩ ((R1))
And immediate to A	ANL A, #data	53	2	2	—	—	—	—	(A) ← (A) ∩ data
Clear A	CLR A	27	1	1	—	—	—	—	(A) ← 0
Complement A	CPL A	37	1	1	—	—	—	—	(A) ← (A)
Decimal Adjust A	DA A	57	1	1	*	—	—	—	(Note 1)
Decrement A	DEC A	07	1	1	—	—	—	—	(A) ← (A) - 1
Increment A	INC A	17	1	1	—	—	—	—	(A) ← (A) + 1
Or register to A	ORL A, Rr	4X	1	1	—	—	—	—	(A) ← (A) ∪ (Rr)
Or data memory to A	ORL A, @R0	40	1	1	—	—	—	—	(A) ← (A) ∪ ((R0))
	ORL A, @R1	41	1	1	—	—	—	—	(A) ← (A) ∪ ((R1))
Or immediate to A	ORL A, #data	43	2	2	—	—	—	—	(A) ← (A) ∪ data
Rotate A left	RL A	E7	1	1	—	—	—	—	
Rotate A left through carry	RLC A	F7	1	1	*	—	—	—	
Rotate A right	RR A	77	1	1	—	—	—	—	
Rotate A right through carry	RRC A	67	1	1	*	—	—	—	
SWAP nibbles of A	SWAP A	47	1	1	—	—	—	—	$(A_{4-7}) = (A_{0-3})$
Exclusive Or register to A	XRLA, Rr	DX	1	1	—	—	—	—	(A) ← (A) + (Rr)
Exclusive Or data memory to A	XRL A, @R0	D0	1	1	—	—	—	—	(A) ← (A) + ((R0))
	XRL A, @R1	D1	1	1	—	—	—	—	(A) ← (A) + ((R1))
Exclusive Or immediate to A	XRL A, #data	D3	2	2	—	—	—	—	(A) ← (A) + data

†Mnemonic copyright Intel Corporation 1983. (except HALT and STOP)
Operation Code X: See Tables 1 and 2.
%: See Table 3.
Flags*: This flag is set or reset in the state after executed instruction.
Z: This flag is reset.
CP: This flag is complemented.
Note: (1) The accumulator value is adjusted to form BCD digits following the binary addition of BCD number.

Instruction Set Summary
(Continued)

Input/Output Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
And immediate to BUS	ANL BUS, #data	98	2	2	—	—	—	—	(BUB) ← (BUS) ∩ data
And immediate to P1	ANL P1, #data	99	2	2	—	—	—	—	(P1) ← (P1) ∩ data
And immediate to P2	ANL P2, #data	9A	2	2	—	—	—	—	(P2) ← (P2) ∩ data
And A to Expander Port	ANLD P_P, A	9X	1	2	—	—	—	—	(P_P) ← (P_P) ∩ (A3~0)
Input BUS to A	INS A, BUS	08	1	2	—	—	—	—	(A) ← (BUS)
Input P1 to A	IN A, P1	09	1	2	—	—	—	—	(A) ← (P1)
Input P2 to A	IN A, P2	0A	1	2	—	—	—	—	(A) ← (P2)
Input Expander Port to A	MOVD A, P_P	0X	1	2	—	—	—	—	(A3~0) ← (P_P), (A7~4) ← 0
Or immediate to BUS	ORL BUS, #data	88	2	2	—	—	—	—	(BUS) ← (BUS) ∪ data
Or immediate to P1	ORL P1, #data	89	2	2	—	—	—	—	(P1) ← (P1) ∪ data
Or immediate to P2	ORL P2, #data	8A	2	2	—	—	—	—	(P2) ← (P2) ∪ data
Or A to Expander Port	ORLD P_P, A	8X	1	2	—	—	—	—	(P_P) ← (P_P) ∪ (A3~0)
Output A to BUS	OUTL BUS, A	02	1	2	—	—	—	—	(BUS) ← (A)
Output A to P1	OUTL P1, A	39	1	2	—	—	—	—	(P1) ← (A)
Output A to P2	OUTL P2, A	3A	1	2	—	—	—	—	(P2) ← (A)
Output A to Expander Port	MOVD P_P, A	3X	1	2	—	—	—	—	(P_P) ← (A3~0)

Data Move Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Move register to A	MOV A, Rr	FX	1	1	—	—	—	—	(A) ← (Rr)
Move data memory to A	MOV A, @R0	F0	1	1	—	—	—	—	(A) ← ((R0))
	MOV A, @R1	F1	1	1	—	—	—	—	(A) ← ((R1))
Move immediate to A	MOV A, #data	23	2	2	—	—	—	—	(A) ← data
Move A to register	MOV Rr, A	AX	1	1	—	—	—	—	(Rr) ← (A)
Move A to data memory	MOV @R0, A	A0	1	1	—	—	—	—	((R0)) ← (A)
	MOV @R1, A	A1	1	1	—	—	—	—	((R1)) ← (A)
Move immediate to register	MOV Rr, #data	BX	2	2	—	—	—	—	(Rr) ← data
Move immediate to data memory	MOV @R0, #data	B0	2	2	—	—	—	—	((R0)) ← data
	MOV @R1, #data	B1	2	2	—	—	—	—	((R1)) ← data
Move PSW to A	MOV A, PSW	C7	1	1	—	—	—	—	(A) ← (PSW)
Move A to PSW	MOV PSW, A	D7	1	1	•	•	•	—	(PSW) ← (A)
Move external data memory to A	MOVX A, @R0	80	1	2	—	—	—	—	(A) ← ((R0))
	MOVX A, @R1	81	1	2	—	—	—	—	(A) ← ((R1))
Move A to external data memory	MOVX @R0, A	90	1	2	—	—	—	—	((R0)) ← (A)
	MOVX @R1, A	91	1	2	—	—	—	—	((R1)) ← (A)
Move to A from current page	MOVP A, @A	A3	1	2	—	—	—	—	(PC7~0) ← (A), (A) ← ((PC))
Move to A from page 3	MOVP3 A, @A	E3	1	2	—	—	—	—	(PC7~0) ← (A), (PC11~8) ← 3, (A) ← ((PC))
Exchange A and register	XCH A, Rr	2X	1	1	—	—	—	—	(A) ⇄ (Rr)
Exchange A and data memory	XCH A, @R0	20	1	1	—	—	—	—	(A) ⇄ ((R0))
	XCH A, @R1	21	1	1	—	—	—	—	(A) ⇄ ((R1))
Exchange nibble of A and data memory	XCHD A, @R0	30	1	1	—	—	—	—	(A3~0) ⇄ ((R0)3~0)
	XCHD A, @R1	31	1	1	—	—	—	—	(A3~0) ⇄ ((R1)3~0)

Instruction Set Summary
(Continued)

Branch and Jump Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Decrement register and test	DJNZ Rr, addr	EX	2	2	—	—	—	—	(Rr) ≠ 0 Note (1)
Jump unconditional	JMP addr	%4	2	2	—	—	—	—	Unconditional Branch
Jump indirect	JMPP @A	B3	1	2	—	—	—	—	Unconditional Branch Note (2)
Jump on carry = 1	JC addr	F6	2	2	—	—	—	—	(C) = 1
Jump on carry = 0	JNC addr	E6	2	2	—	—	—	—	(C) = 0
Jump on A zero	JZ addr	C6	2	2	—	—	—	—	(A) = 0
Jump on A no zero	JNZ addr	96	2	2	—	—	—	—	(A) ≠ 0
Jump on T0 = 1	JT0 addr	36	2	2	—	—	—	—	(T0) = 1
Jump on T0 = 0	JNT0 addr	26	2	2	—	—	—	—	(T0) = 0
Jump on T1 = 1	JT1 addr	56	2	2	—	—	—	—	(T1) = 1
Jump on T1 = 0	JNT1 addr	46	2	2	—	—	—	—	(T1) = 0
Jump on F0 = 1	JF0 addr	B6	2	2	—	—	—	—	(F0) = 1
Jump on F1 = 1	JF1 addr	76	2	2	—	—	—	—	(F1) = 1
Jump on timer flag, Clear flag	JTF addr	16	2	2	—	—	—	—	(TF) = 1
Jump on $\overline{INT}$ = 0	JNI addr	86	2	2	—	—	—	—	($\overline{INT}$) = 0
Jump on accumulator bit	JBb addr	%2	2	2	—	—	—	—	(Ab) = 1

Subroutine Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Jump to subroutine	CALL addr	%4	2	2	—	—	—	—	Note (3)
Return	RET	83	1	2	—	—	—	—	Note (4)
Return and restore status	RETR	93	1	2	*	*	*	—	Note (5)

Flags Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Clear carry	CLR C	97	1	1	Z	—	—	—	(C) ← 0
Complement carry	CPL C	A7	1	1	CP	—	—	—	(C) ← ($\overline{C}$)
Clear flag 0	CLR F0	85	1	1	—	—	Z	—	(F0) ← 0
Complement flag 0	CPL F0	95	1	1	—	—	CP	—	(F0) ← ($\overline{F0}$)
Clear flag 1	CLR F1	A5	1	1	—	—	—	Z	(F1) ← 0
Complement flag 1	CPL F1	B5	1	1	—	—	—	CP	(F1) ← ($\overline{F1}$)

(1) DJNZ Rr, Addr: (Rr) - 1 → (Rr)
if (Rr) ≠ 0, addr → (PC_0 to PC_7).
if (R_r) = 0, execute next instruction.
(2) JMPP @ A: ((A)) → (PC_0 to PC_7)

Register Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Decrement register	DEC Rr	CX	1	1	—	—	—	—	(Rr) ← (Rr) - 1
Increment register	INC Rr	1X	1	1	—	—	—	—	(Rr) ← (Rr) + 1
Increment data memory	INC @R0	10	1	1	—	—	—	—	((R0)) ← ((R0)) + 1
	INC @R1	11	1	1	—	—	—	—	((R1)) ← ((R1)) + 1

Instruction Set Summary
(Continued)

Timer/Counter Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Disable Timer/ Counter Interrupt	DIS TCNTI	35	1	1	—	—	—	—	
Enable Timer/ Counter Interrupt	EN TCNTI	25	1	1	—	—	—	—	
Read Timer/Counter	MOV A, T	42	1	1	—	—	—	—	(A) ← (T)
Load Timer/Counter	MOV T, A	62	1	1	—	—	—	—	(T) ← (A)
Start Timer	STRT T	55	1	1	—	—	—	—	
Start Counter	STRT CNT	45	1	1	—	—	—	—	
Stop Timer/Counter	STOP TCNT	65	1	1	—	—	—	—	

Control Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Disable external Interrupt	DIS I	15	1	1	—	—	—	—	
Enable external Interrupt	EN I	05	1	1	—	—	—	—	
Enable Clock output on T0	ENT0 CLK	75	1	1	—	—	—	—	
No Operation	NOP	00	1	1	—	—	—	—	
Select register bank 0	SEL RB0	C5	1	1	—	—	—	—	(BS) ← 0
Select register bank 1	SEL RB1	D5	1	1	—	—	—	—	(BS) ← 1
Select memory bank 0	SEL MB0	E5	1	1	—	—	—	—	(MBF) ← 0
Select memory bank 1	SEL MB1	F5	1	1	—	—	—	—	(MBF) ← 1

Standby Instructions

Operation	Mnemonic	OP Code	Byte	Cycle	Flag CY	HC	F_0	F_1	Note
Halt	HALT	01	1	1	—	—	—	—	
Stop	STOP	C1	1	1	—	—	—	—	

Instruction Set Summary
(Continued)

Table 1. O.P. Code of Register Access Instruction

Mnemonic	Rr	R0	R1	R2	R3	R4	R5	R6	R7
INC Rr		18	19	1A	1B	1C	1D	1E	1F
XCH A, Rr		28	29	2A	2B	2C	2D	2E	2F
ORL A, Rr		48	49	4A	4B	4C	4D	4E	4F
ANL A, Rr		58	59	5A	5B	5C	5D	5E	5F
ADD A, Rr		68	69	6A	6B	6C	6D	6E	6F
ADDC A, Rr		78	79	7A	7B	7C	7D	7E	7F
MOV Rr, A		A8	A9	AA	AB	AC	AD	AE	AF
MOV Rr, #data		B8	B9	BA	BB	BC	BD	BE	BF
DEC Rr		C8	C9	CA	CB	CC	CD	CE	CF
XRL A, Rr		D8	D9	DA	DB	DC	DD	DE	DF
DJNZ Rr, M		E8	E9	EA	EB	EC	ED	EE	EF
MOV A, Rr		F8	F9	FA	FB	FC	FD	FE	FF

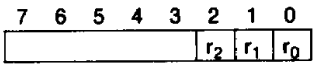
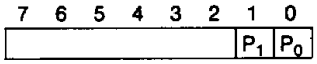
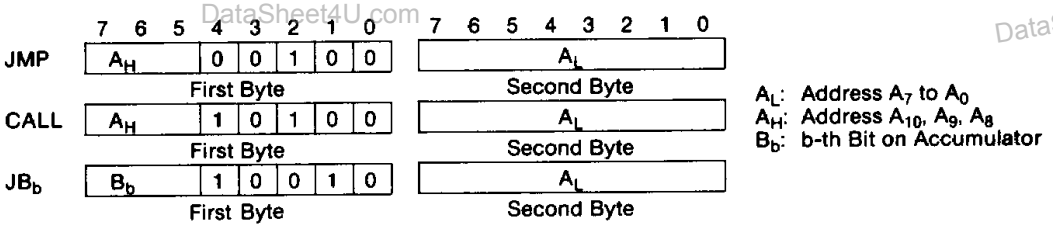


Table 2. OP Code Of Expander Port Instruction

Mnemonic	Pp	P4	P5	P6	P7
MOVD A.P_P		0C	0D	0E	0F
MOVD P_P, A		3C	3D	3E	3F
ORLD P_P, A		8C	8D	8E	8F
ANLD P_P, A		9C	9D	9E	9F



OP Code of JMP/CALL/JB_b



A_L: Address A_7 to A_0
A_H: Address A_{10}, A_9, A_8
B_b: b-th Bit on Accumulator

Notes:

(3) CALL addr:
(PC_0 to PC_7) → ((SP))
(PC_8 to PC_{11}), (MBF), (PSW_5 to PSW_7) → ((SP))
(SP) + 1 → (SP)
A_L → (PC_0 to PC_7)
A_H → (PC_8 to PC_{10})
MBF → (PC_{11})

(4) RET:
(SP) - 1 → (SP)
((SP)) → (PC_0 to PC_7), (PC_8 to PC_{11})

(5) RETR:
(SP) -1 → (SP)
$((SP))_0$ to $(SP)_3$ → (PC_8 to PC_{11})
$((SP))_4$ to $(SP)_7$ → (PSW_4 to PSW_7)
((SP)) → (PC_0 to PC_7), (PC_8 to PC_{11})
((SP)) → (PSW_5 to PSW_7), (MBF)

Instruction Codes

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	HALT	OUT (BSU, A)	ADD (A,#)	JMP 0	EN I		DEC (A)	INS (A,BUS)	IN (A,P1)	IN (A,P2)		MOVD (A,P4)	MOVD (A,P5)	MOVD (A,P6)	MOVD (A,P7)
1	INC @R0	INC @R1	JB 0	ADDC (A,#)	CALL 0	DIS I	JTF	INC (A)	INC (R0)	INC (R1)	INC (R2)	INC (R3)	INC (R4)	INC (R5)	INC (R6)	INC (R7)
2	XCH (A,@R0)	XCH (A,@R1)		MOV (A,#)	JMP 1	EN TCNTI	JNT0	CLR (A)	XCH (A,R0)	XCH (A,R1)	XCH (A,R2)	XCH (A,R3)	XCH (A,R4)	XCH (A,R5)	XCH (A,R6)	XCH (A,R7)
3	XCHD (A, @R0)	XCHD (A, @R1)	JB 1		CALL 1	DIS TCNTI	JT0	CPL (A)		OUTL (P1,A)	OUTL (P2,A)		MOVD (P4,A)	MOVD (P5,A)	MOVD (P6,A)	MOVD (P7,A)
4	ORL (Ac@R0)	ORL (A, @R1)	MOV (A,T)	ORL (A,#)	JMP 2	STRT CNT	JNT1	SWAP (A)	ORL (A,R0)	ORL (A,R1)	ORL (A,R2)	ORL (A,R3)	ORL (A,R4)	ORL (A,R5)	ORL (A,R6)	ORL (A,R7)
5	ANL (A,@R0)	ANL (A,@R1)	JB 2	ANL (A,#)	CALL 2	STRT T	JT1	DA (A)	ANL (A,R0)	ANL (A,R1)	ANL (A,R2)	ANL (A,R3)	ANL (A,R4)	ANL (A,R5)	ANL (A,R6)	ANL (A,R7)
6	ADD (A,@R0)	ADD (A,@R1)	MOV (T,A)		JMP 3	STOP TCNT		RRC (A)	ADD (A,R0)	ADD (A,R1)	ADD (A,R2)	ADD (A,R3)	ADD (A,R4)	ADD (A,R5)	ADD (A,R6)	ADD (A,R7)
7	ADDC (A,@R0)	ADDC (A,@R1)	JB 3		CALL 3	ENT0 CLK	JF1	RR (A)	ADDC (A,R0)	ADDC (A,R1)	ADDC (A,R2)	ADDC (A,R3)	ADDC (A,R4)	ADDC (A,R5)	ADDC (A,R6)	ADDC (A,R7)
8	MOVX (A,@R0)	MOVX (A,@R1)		RET	JMP 4	CLR F0	JNI		ORL (BUS,#)	ORL (P1,#)	ORL (P2,#)		ORLD (P4,A)	ORLD (P5,A)	ORLD (P6,A)	ORLD (P7,A)
9	MOVX (@R0,A)	MOVX (@R1,A)	JB 4	RETR	CALL 4	CPL F0	JNZ	CLR C	ANL (BUS,#)	ANL (P1,#)	ANL (P2,#)		ANLD (P4,A)	ANLD (P5,A)	ANLD (P6,A)	ANLD (P7,A)
A	MOV (@R0,A)	MOV (@R1,A)		MOVP (A,@A)	JMP 5	CLR F1		CPL C	MOV (R0,A)	MOV (R1,A)	MOV (R2,A)	MOV (R3,A)	MOV (R4,A)	MOV (R5,A)	MOV (R6,A)	MOV (R7,A)
B	MOV (@R0,#)	MOV (@R1,#)	JB 5	JMPP (@A)	CALL 5	CPL F1	JF0		MOV (R0,#)	MOV (R1,#)	MOV (R2,#)	MOV (R3,#)	MOV (R4,#)	MOV (R5,#)	MOV (R6,#)	MOV (R7,#)
C		STOP			JMP 6	SEL RB0	JZ	MOV (A,PSW)	DEC (R0)	DEC (R1)	DEC (R2)	DEC (R3)	DEC (R4)	DEC (R5)	DEC (R6)	DEC (R7)
D	XRL (A,@R0)	XRL (A,@R1)	JB 6	XRL (A,#)	CALL 6	SEL RB1		MOV (PSW,A)	XRL (A,R0)	XRL (A,R1)	XRL (A,R2)	XRL (A,R3)	XRL (A,R4)	XRL (A,R5)	XRL (A,R6)	XRL (A,R7)
E				MOVP3 (A,@A)	JMP 7	SEL MB0	JNC	RL (A)	DJNZ (R0,M)	DJNZ (R1,M)	DJNZ (R2,M)	DJNZ (R3,M)	DJNZ (R4,M)	DJNZ (R5,M)	DJNZ (R6,M)	DJNZ (R7,M)
F	MOV (A,@R0)	MOV (A,@R1)	JB 7		CALL 7	SEL MB1	JC	RLC (A)	MOV (A,R0)	MOV (A,R1)	MOV (A,R2)	MOV (A,R3)	MOV (A,R4)	MOV (A,R5)	MOV (A,R6)	MOV (A,R7)

#: Immediate data
H; Higher 4 Bits
L; Lower Bits

1 Byte, 1 Cycle Instruction

1 Byte, 2 Cycles Instruction

2 Byte, 2 Cycles Instruction

Typical Application

Figure 15. Stand Alone System

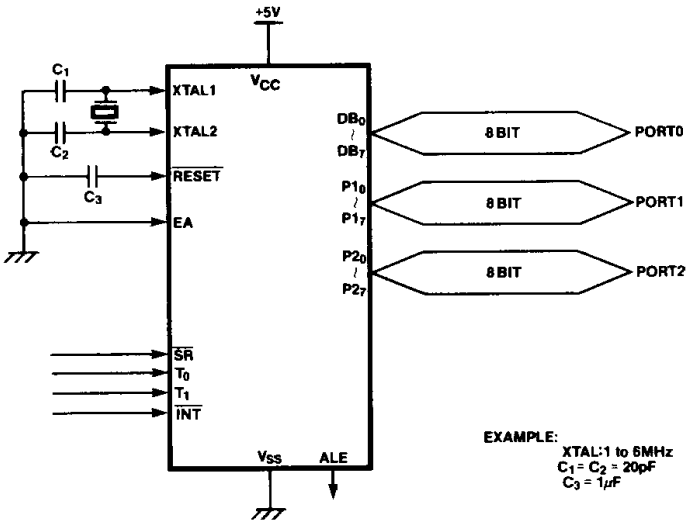
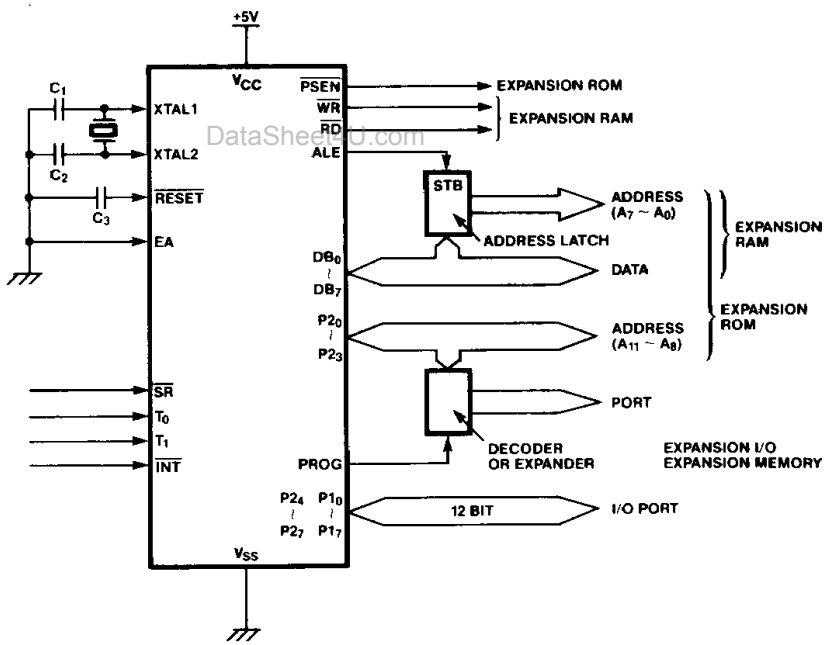


Figure 16. Expanded System



Absolute Maximum Ratings

Parameter	Symbol	Value Min.	Value Max.	Unit
Supply Voltage	V_{CC}	V_{SS} - 0.3	V_{SS} + 7.0	V
Input Voltage	V_{IN}	V_{SS} - 0.3	V_{CC} + 0.3	V
Output Voltage	V_{OUT}	V_{SS} - 0.3	V_{CC} + 0.3	V
Power Dissipation	P_D	—	600	mW
Operating Temperature	T_A*	-40	+85	°C
Storage Temperature	T_{STG}	-55	+150	°C

Note:
*T_A = 0°C to 70°C for H version.
**Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Exposure to ABSOLUTE MAXIMUM RATINGS for extended period may affect device reliability.

Recommended Operating Conditions

Parameter	Symbol	Value Min.	Value Typ.	Value Max.	Unit	Note	
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	Active Mode (Normal Operation)	
		3.5		6.0		Standby Mode	All internal states are retained
		2.0		6.0			Only internal RAM data are retained
	V_{SS}		0				
Operating Temperature	T_A	-40		+85	°C	MBL80C49N/C39N	
		0		+70		MBL80C49H/C39H	

2

DC Characteristics

Recommended operating conditions, unless otherwise noted.

(T_A = -40°C to +85°C, V_{CC} = 5V ±10%, V_{SS} = 0V for MBL80C49N/MBL80C39N)
(T_A = 0°C to +70°C, V_{CC} = 5V ±10%, V_{SS} = 0V for MBL80C49H/MBL80C39H)

Parameter	Applicable Pin/Device	Symbol	Value Min.	Value Max.	Unit	Test Conditions
Input Low Voltage	All except XTAL1, XTAL2, $\overline{RESET}$	V_{IL}	-0.3	0.8	V	
	XTAL1, XTAL2, $\overline{RESET}$	V_{IL1}	-0.3	0.6	V	
Input High Voltage	All except XTAL1, XTAL2, $\overline{RESET}$	V_{IH}	2.2	V_{CC}	V	
	XTAL1, XTAL2, $\overline{RESET}$	V_{IH1}	3.8	V_{CC}	V	
Output Low Voltage	BUS	V_{OL}		0.45	V	I_{OL} = 2.0mA
	$\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	V_{OL1}		0.45	V	I_{OL} = 2.0mA
	PROG	V_{OL2}		0.45	V	I_{OL} = 1.0mA
	All Other Outputs	V_{OL3}		0.45	V	I_{OL} = 1.6mA
Output High Voltage	BUS	V_{OH}	2.4		V	I_{OH} = -400μA
	$\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE, PROG, T0	V_{OH1}	2.4		V	I_{OH} = -100μA
	P10–P17, P20–P27	V_{OH2}	2.4		V	I_{OH} = -40μA
Input Leakage Current	$\overline{RESET}$, $\overline{SS}$, $\overline{SR}$	I_{LI}		-100	μA	V_{SS} + 0.45≦V_{IN}≦V_{CC}
	T1, $\overline{INT}$, EA	I_{LI1}		±10	μA	V_{SS}≦V_{IN}≦V_{CC}
	P10–P17, P20–P27	I_{LI2}		-200	μA	V_{SS} + 0.45≦V_{IN}≦V_{CC}
Output Leakage Current	BUS, T0	I_{LO}		±10	μA	V_{SS} + 0.45≦V_{IN}≦V_{CC} High-Impedance Mode
Standby Supply Current	MBL80C49N/C39N	I_{CCH}		2	mA	V_{CC} = 3.5V, 6MHz, HALT Mode
	MBL80C49H/C49H	I_{CCH}		4	mA	V_{CC} = 3.5V, 11MHz, HALT Mode
	MBL80C49N/H, MBL80C39N/H	I_{CCS}		50	μA	V_{CC} = 2.0V, STOP Mode
Active Supply Current	MBL80C49N/C39N	I_{CC}		10	mA	All Outputs Open
	MBL80C49H/C39H	I_{CC}		18	mA	All Outputs Open

AC Characteristics

Recommended operating conditions, unless otherwise noted.

(T_A = -40°C to +85°C, V_{CC} = +5V ±10%, V_{SS} = 0V for MBL80C49N/MBL80C39N)
(T_A = 0°C to +70°C, V_{CC} = +5V ±10%, V_{SS} = 0V for MBL80C49H/MBL80C39H)

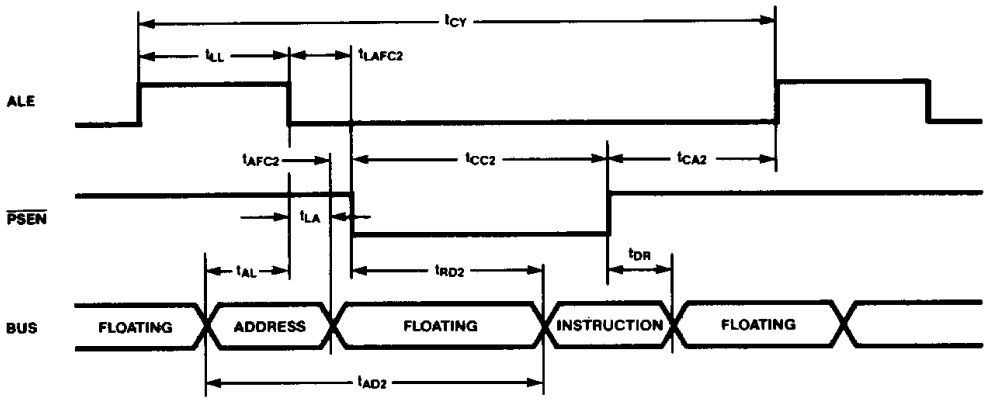
Parameter	Symbol	MBL80C49H/C39H		MBL80C49N/C39N		Unit	Test Conditions
		Min.	Max.	Min.	Max.		
ALE Pulse Width	t_{LL}	150		410		ns	
Address Setup Time (to ALE↓)	t_{AL}	70		230		ns	
Address Hold Time (from ALE↓)	t_{LA}	50		120		ns	Note 1
$\overline{RD}$ & $\overline{WR}$ Pulse Width	t_{CC1}	480		1050		ns	
$\overline{PSEN}$ Pulse Width	t_{CC2}	350		800		ns	
Data Setup Time (to $\overline{WR}$↑)	t_{DW}	390		880		ns	
Data Hold Time (from $\overline{WR}$↑)	t_{WD}	40		120		ns	Note 2
Data Hold Time (from $\overline{RD}$↑, $\overline{PSEN}$↑)	t_{DR}	0	110	0	220	ns	
Data Delay Time (from $\overline{RD}$↓)	t_{RD1}		350		800	ns	
Data Delay Time (from $\overline{PSEN}$↓)	t_{RD2}		210		550	ns	
Address Setup Time (to $\overline{WR}$↓)	t_{AW}	310		680		ns	
Data Delay Time ($\overline{RD}$)	t_{AD1}		760		1590	ns	
Data Delay Time ($\overline{PSEN}$)	t_{AD2}		480		1090	ns	
Address Floating Time (to $\overline{RD}$↑, $\overline{WR}$↓)	t_{AFC1}	140		290		ns	
Address Floating Time (to $\overline{PSEN}$↓)	t_{AFC2}	10		40		ns	
$\overline{RD}$, $\overline{WR}$ Output Delay Time (from ALE↓)	t_{LAFC1}	200		420		ns	
$\overline{PSEN}$ Output Delay Time (from ALE↓)	t_{LAFC2}	60		170		ns	
ALE Delay Time (from $\overline{RD}$↑, $\overline{WR}$↑, PROG↑)	t_{CA1}	50		120		ns	
ALE Delay Time (from $\overline{PSEN}$↑)	t_{CA2}	320		620		ns	Note 1
Port Control Setup Time (to PROG↓)	t_{CP}	100		250		ns	
Port Control Hold Time (from PROG↓)	t_{PC}	160		460		ns	
Port 2 Input Data Delay Time (from PROG↓)	t_{PR}		700		1380	ns	
Port 2 Input Data Hold Time (from PROG↑)	t_{PF}	0	140	0	250	ns	
Output Data Setup Time (to PROG↑)	t_{DP}	400		850		ns	
Output Data Hold Time (from PROG↑)	t_{PD}	90		200		ns	
PROG Pulse Width	t_{PP}	700		1500		ns	
Port 2 I/O Data Setup Time (to ALE↑)	t_{PL}	160		460		ns	
Port 2 I/O Data Hold Time (from ALE↑)	t_{LP}	40		80		ns	
Port Data Output Time (from ALE↓)	t_{PV}		510		850	ns	
Cycle Time	t_{CY}	1.36		2.5		μs	
T0 Output Frequency	t_{OPRR}	270		500		ns	

Notes:
1. Load Conditions: BUS: C_L = 150pF, Other Outputs: C_L = 80pF, 1TTL
2. Load Conditions: BUS: C_L = 20pF, High impedance
↓: Falling edge
↑: Rising edge

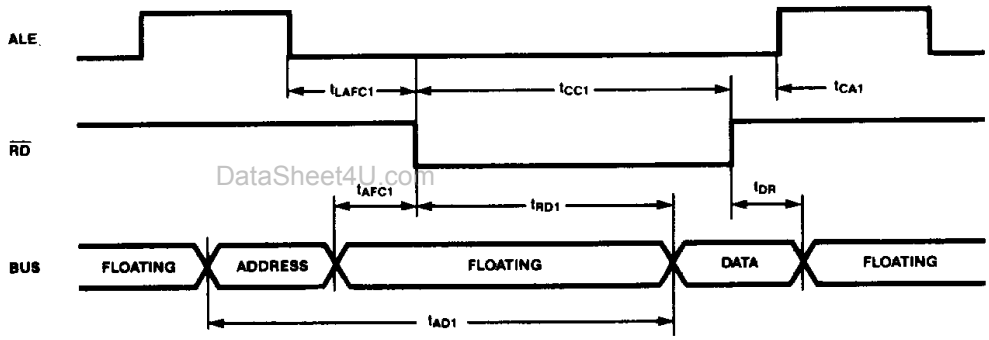
Timing Diagram

Figure 11. Timing Diagram (1)

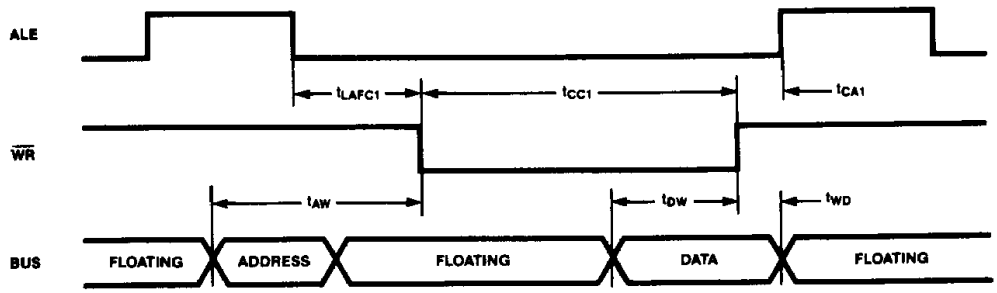
Instruction Fetch From External Program Memory



Read From External Data Memory



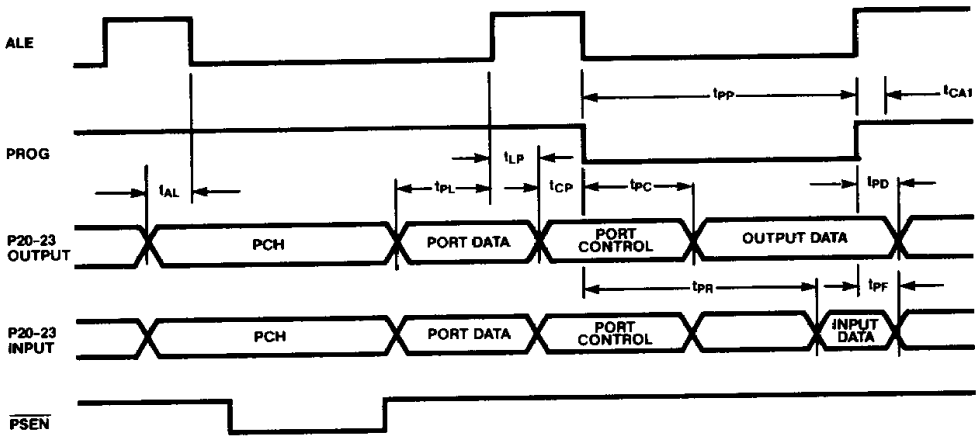
Write To External Data Memory



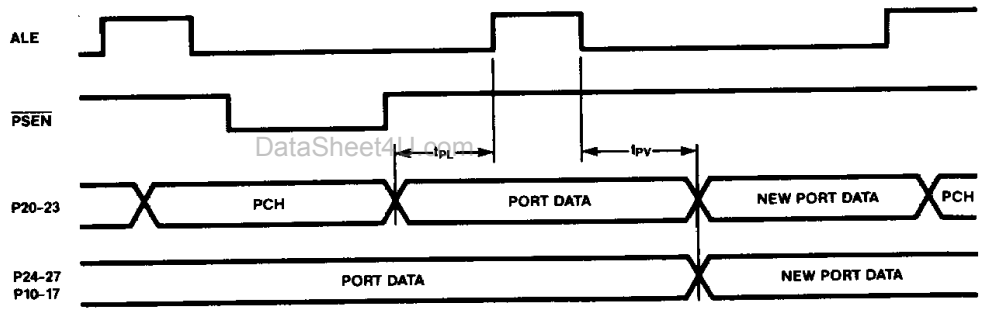
Timing Diagram
(Continued)

Figure 12. Timing Diagram (2)

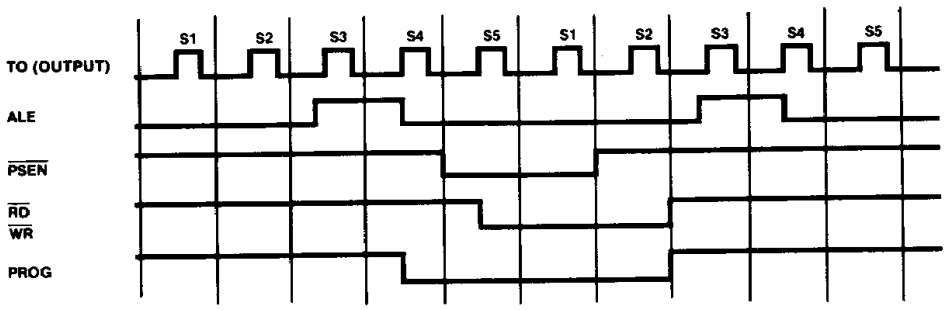
P20-P23 Input/Output for Use of External Program Memory and Expander I/O Port



Port 1/Port 2 Outputs



Clock Outputs

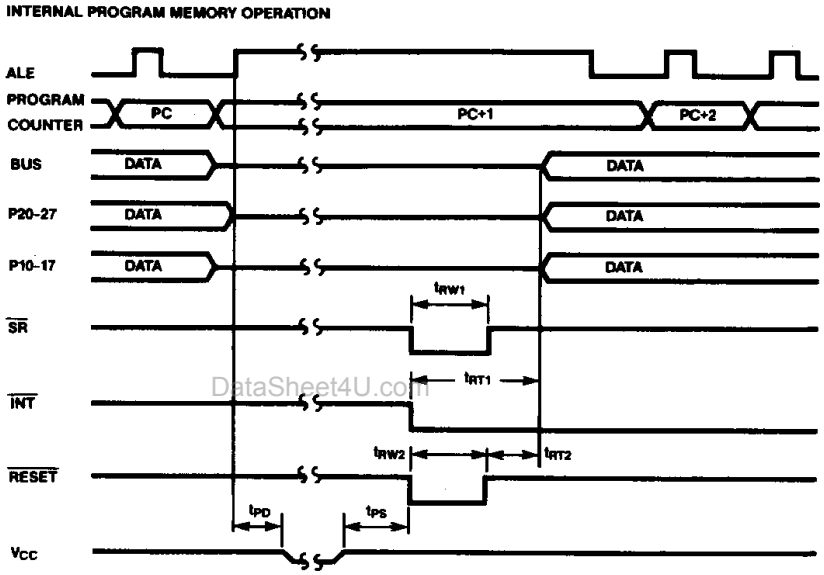


AC Characteristics for Standby Operation

Parameter		Symbol	Value Min.	Typ.	Max.
Standby Release Pulse Width	$\overline{SR}$	t_{RW1}	$2 \cdot t_{CY}$	—	—
	$\overline{RESET}$	t_{RW2}	$12 \cdot t_{CY}$	—	—
Standby Release Time	HALT Mode	t_{RT1}	$4 \cdot t_{CY}$	$5 \cdot t_{CY}$	$6 \cdot t_{CY}$
		t_{RT2}	$4 \cdot t_{CY}$	$5 \cdot t_{CY}$	$6 \cdot t_{CY}$
	STOP Mode	t_{RT2}	—	$8197 \cdot t_{CY}$	$8200 \cdot t_{CY}$
V_{CC} Hold Time		t_{PD}	$5 \cdot t_{CY}$	—	—
V_{CC} Setup Time		t_{PS}	$5 \cdot t_{CY}$	—	—

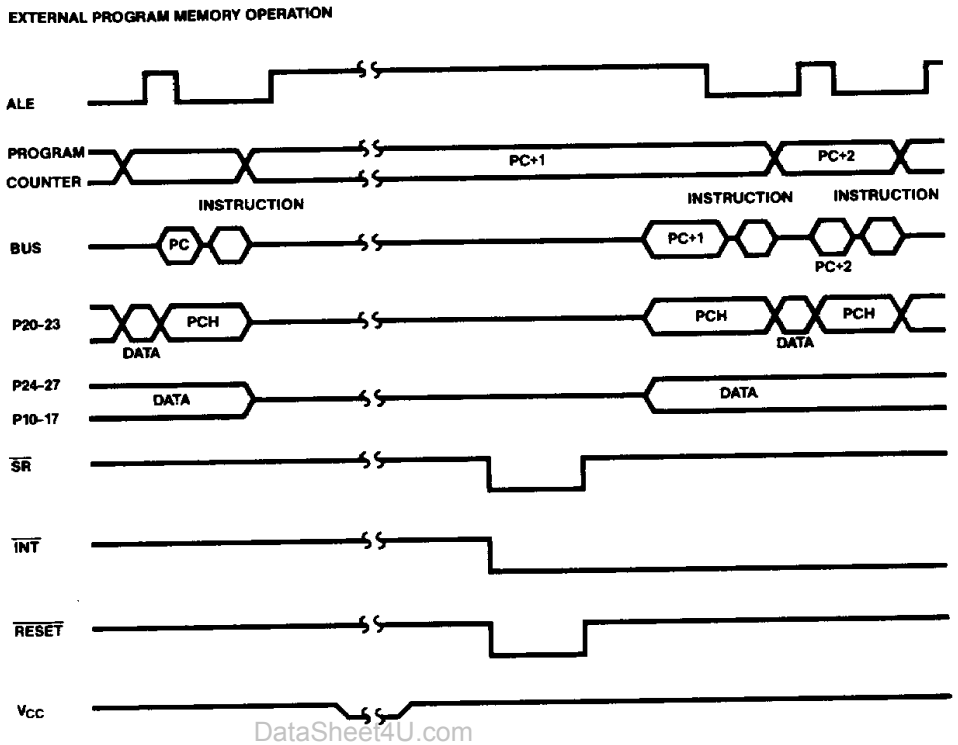
Timing Diagram

Figure 13. Standby Operation Timing (1)

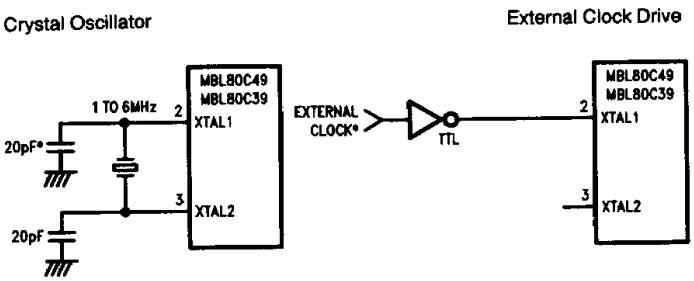


Timing Diagram
(Continued)

Figure 14. Standby Operation Timing (2)



Oscillation Circuits



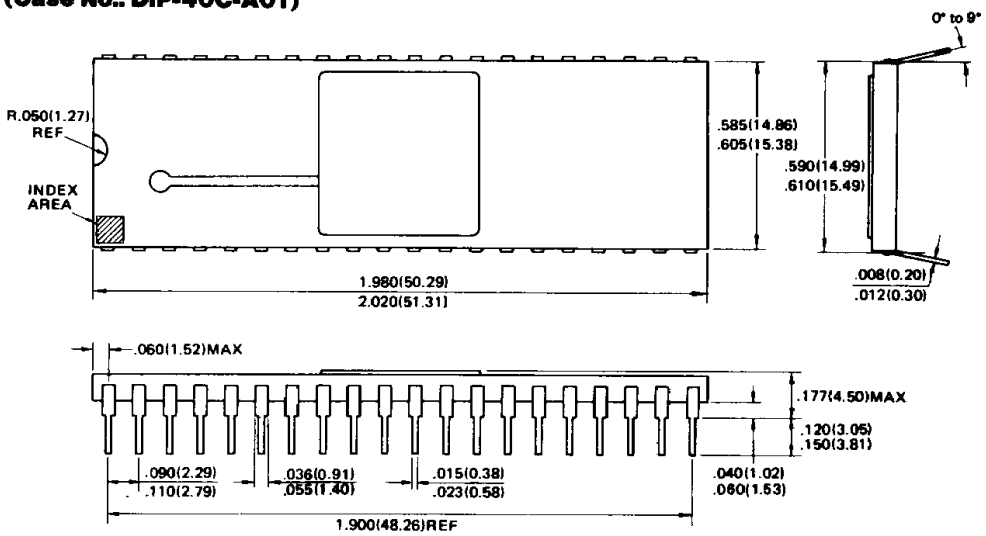
*Including stray capacitances

*Both high and low times should be more than 35% of the cycle time, and the rise and fall times should be less than 20ns.

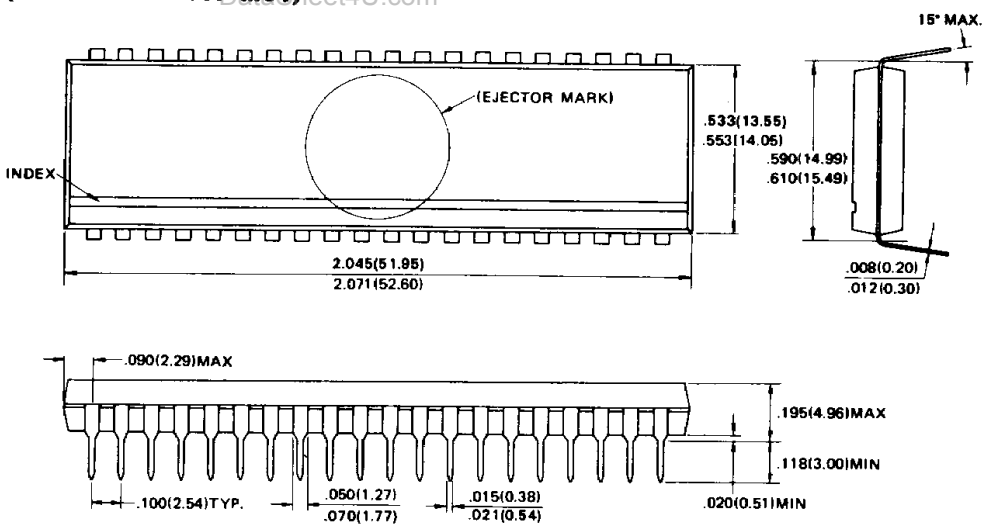
Package Dimensions

Dimensions in inches
(millimeters)

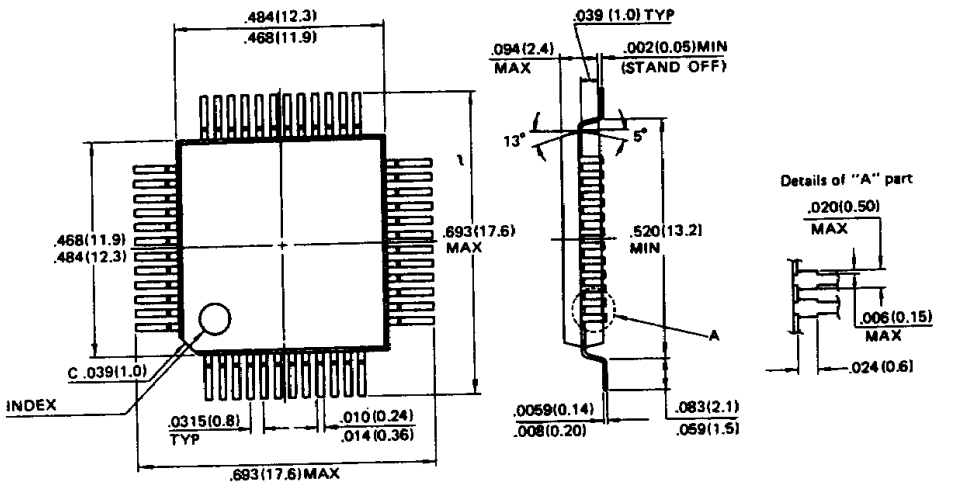
40-Lead Ceramic
(Metal Seal)
Dual In-Line Package
(Case No.: DIP-40C-A01)



40-Lead Plastic
Dual In-Line Package
(Case No.: DIP-40P-M01)



48-Lead Plastic
Flat Package
(CASE NO.:FPT-48P-M02)



Dimensions in inches (millimeters)

2