FUJITSU

NMOS 8-BIT MICROPROCESSOR

MBL 8088
MBL 8088-2
MBL 8088-1

February 1985
Edition 4.0

NMOS 8-BIT MICROPROCESSOR

The Fujitsu MBL 8088 is a new generation, high performance microprocessor implemented in N-channel, depletion load, silicon gate technology (NMOS), and packaged in a 40-pin ceramic or plastic DIP. The processor has attributes of both 8- and 16-bit microprocessors. It is directly compatible with MBL 8086 software and Intel 8080/8085 hardware and peripherals.

- **8-Bit Data Bus Interface**
- **16-Bit Internal Architecture**
- **Direct Addressing Capability to 1 Mbyte of Memory**
  Direct Software Compatibility with MBL 8086 CPU
  14-Word by 16-Bit Register Set with Symmetrical Operations
- **24 Operand Addressing Modes**
- **Byte, Word, and Block Operations**
- **8-Bit and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal, Including Multiply and Divide**
- **Compatible with 8155-2, 8755A-2 and 8185-2 Multiplexed Peripherals**
- **Two Clock Rates:**
  5MHz for MBL 8088,
  8MHz for MBL 8088-2,
  10MHz for MBL 8088-1
- **40-Pin DIP:**
  Ceramic DIP (Suffix: -C)
  Plastic DIP (Suffix: -P)

Fig. 1 – BLOCK DIAGRAM

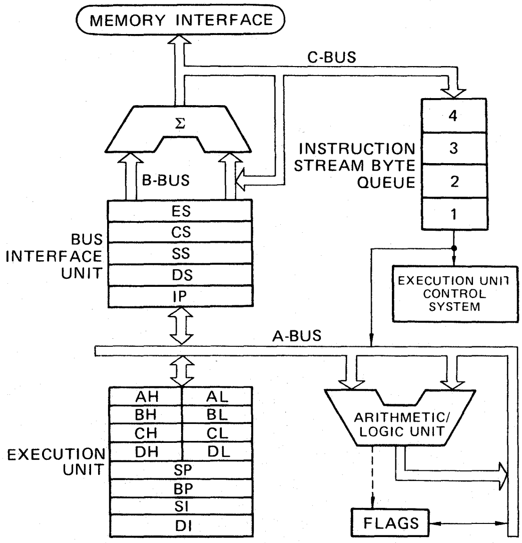


Fig. 2 – PIN CONFIGURATION

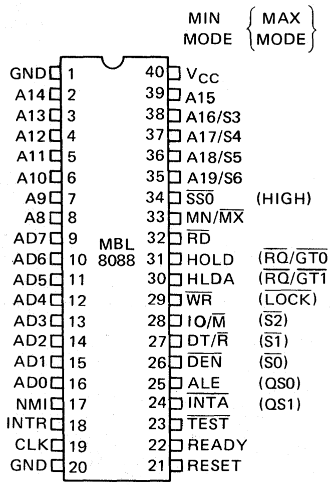


TABLE 1 — PIN DESCRIPTION

The following pin function descriptions are for MBL 8088 systems in either minimum or maximum mode. The "local bus" in these descriptions is the direct multiplexed bus interface connection to the MBL 8088 (without regard to additional bus buffers).

<table>
<tr><th>Symbol</th><th>Pin No.</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>AD_7-AD_0</td><td>9-16</td><td>I/O</td><td><b>Address Data Bus:</b> These lines constitute the time multiplexed memory/IO address (T_1) and data (T_2, T_3, Tw, and T_4) bus. These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".</td></tr>
<tr><td>$A_{15}-A_8$</td><td>2-8, 39</td><td>O</td><td><b>Address Bus:</b> These lines provide address bits 8 through 15 for the entire bus cycle (T_1-T_4). These lines do not have to be latched by ALE to remain valid. $A_{15}-A_8$ are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".</td></tr>
<tr><td>A_{19}/S_6,
A_{18}/S_5,
A_{17}/S_4,
A_{16}/S_3</td><td>35-38</td><td>O</td><td><b>Address/Status:</b> During T_1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is avalable on these lines during T_2, T_3, Tw, and T_4. S_6 is always low. The status of the interrupt enable flag bit (S_5) is updated at the beginning of each clock cycle. S_4 and S_3 are encoded as shown.
This information indicates which segment register is presently being used for data accessing.
These lines float to 3-state OFF during local bus "hold acknowledge".
<table>
<tr><th>S_4</th><th>S_3</th><th>Characteristics</th></tr>
<tr><td>0 (LOW)</td><td>0</td><td>Alternate Data</td></tr>
<tr><td>0</td><td>1</td><td>Stack</td></tr>
<tr><td>1 (HIGH)</td><td>0</td><td>Code or None</td></tr>
<tr><td>1</td><td>1</td><td>Data</td></tr>
<tr><td>S_6 is 0 (LOW)</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>$\overline{RD}$</td><td>32</td><td>O</td><td><b>Read:</b> Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the $IO/\overline{M}$ pin or S_2. This signal is used to read devices which reside on the MBL 8088 local bus. $\overline{RD}$ is active LOW during T_2, T_3 and Tw of any read cycle, and is guaranteed to remain HIGH in T_2 until the MBL 8088 local bus has floated.
This signal floats to 3-state OFF in "hold acknowledge".</td></tr>
<tr><td>READY</td><td>22</td><td>I</td><td><b>READY:</b> is the acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the MBL 8284A clock generator to form READY. This signal is active HIGH. The MBL 8088 READY input is not synchronized. Correct operation is not guaranteed if the set up and hold times are not met.</td></tr>
<tr><td>INTR</td><td>18</td><td>I</td><td><b>Interrupt Request:</b> is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.</td></tr>
<tr><td>$\overline{TEST}$</td><td>23</td><td>I</td><td><b>$\overline{TEST}$:</b> input is examined by the "wait for test" instruction. If the $\overline{TEST}$ input is LOW, execution continues, otherwise the processor waits in an "idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.</td></tr>
<tr><td>NMI</td><td>17</td><td>I</td><td><b>Non-Maskable Interrupt:</b> is an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.</td></tr>
</table>

TABLE 1 — PIN DESCRIPTION (Continued)

Symbol	Pin No.	Type	Name and Function
RESET	21	I	**RESET:** causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.
CLK	19	I	**Clock:** provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
V_{CC}	40		**V_{CC} :** is the +5V ±10% power supply pin.
GND	1, 20		**GND:** are the ground pins.
MN/$\overline{MX}$	33	I	**Minimum/Maximum:** indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

The following pin function descriptions are for the MBL 8088 minimum mode (i.e., MN/$\overline{MX}$ = V_{CC}). Only the pin functions which are unique to minimum mode are described; all other pin functions are as described above.

IO/$\overline{M}$	28	O	**Status Line:** is an inverted maximum mode $\overline{S_2}$. It is used to distinguish a memory access from an I/O access. IO/$\overline{M}$ becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (I/O=HIGH, $\overline{M}$=LOW). IO/$\overline{M}$ floats to 3-state OFF in local bus "hold acknowledge".
$\overline{WR}$	29	O	**Write:** strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/$\overline{M}$ signal. $\overline{WR}$ is active for T_2, T_3, and Tw of any write cycle. It is active LOW, and floats to 3-state OFF in local bus "hold acknowledge".
$\overline{INTA}$	24	O	**INTA:** is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3, and Tw of each interrupt acknowledge cycle.
ALE	25	O	**Address Latch Enable:** is provided by the processor to latch the address into the MBL 8282/8283 address latch. It is a HIGH pulse active during clock low of T_1 of any bus cycle. Note that ALE is never floated.
DT/$\overline{R}$	27	O	**Data Transmit/Receive:** is needed in a minimum system that desires to use an MBL 8286/8287 data bus transceiver. It is used to control to direction of data flow through the transceiver. Logically, DT/$\overline{R}$ is equivalent to S_1 in the maximum mode, and its timing is the same as for IO/$\overline{M}$ (T=HIGH, $\overline{R}$=LOW). This signal floats to 3-state OFF in local "hold acknowledge".
$\overline{DEN}$	26	O	**Data Enable:** is provided as an output enable for the MBL 8286/8287 in a minimum system which uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access, and for $\overline{INTA}$ cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T_2 until the middle of T_4, while for a write cycle, it is active from the beginning of T_2 until the middle of T_4. $\overline{DEN}$ floats to 3-state OFF during local bus "hold acknowledge".
HOLD, HLDA	30, 31	I, O	**HOLD:** indicates that another master is requesting a local bus "hold". To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement, in the middle of a T_4 or T_1 clock cycle. Simultaneous with the issuance of HLDA the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. Hold is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set up time.

TABLE 1 — PIN DESCRIPTION (Continued)

<table>
<tr><th>Symbol</th><th>Pin No.</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>$\overline{\text{SSO}}$</td><td>34</td><td>O</td><td><b>Status line:</b> is logically equivalent to $\overline{\text{SO}}$ in the maximum mode. The combination of $\overline{\text{SSO}}$, IO/$\overline{\text{M}}$ and DT/$\overline{\text{R}}$ allows the system to completely decode the current bus cycle status.
<table>
<tr><th>IO/$\overline{\text{M}}$</th><th>DT/$\overline{\text{R}}$</th><th>$\overline{\text{SS0}}$</th><th>Characteristics</th></tr>
<tr><td>1 (HIGH)</td><td>0</td><td>0</td><td>Interrupt acknowledge</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Read I/O Port</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Write I/O Port</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Halt</td></tr>
<tr><td>0 (LOW)</td><td>0</td><td>0</td><td>Code Access</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Read Memory</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Write Memory</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Passive</td></tr>
</table>
</td></tr>
</table>

The following pin function descriptions are for the MBL 8088, 8228 system in maximum mode (i.e., MN/$\overline{\text{MX}}$=GND.) Only the pin functions which are unique to maximum mode are described; all other pin functions are as described above.

<table>
<tr><th></th><th></th><th></th><th></th></tr>
<tr><td>$\overline{\text{S2}}$, $\overline{\text{S1}}$, $\overline{\text{S0}}$</td><td>26-28</td><td>O</td><td><b>Status:</b> is active during clock high of T_4, T_1, and T_2, and is returned to the passive state (1,1,1) during T_3 or during Tw when READY is HIGH. This status is used by the MBL 8288 bus controller to generate all memory and I/O access control signals. Any change by $\overline{\text{S2}}$, $\overline{\text{S1}}$, or $\overline{\text{S0}}$ during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or Tw is used to indicate the end of a bus cycle.
These signals float to 3-state OFF during "hold acknowledge". During the first clock cycle after RESET becomes active, these signals are active HIGH. After this first clock they float to 3-state OFF.
<table>
<tr><th>$\overline{\text{S2}}$</th><th>$\overline{\text{S1}}$</th><th>$\overline{\text{S0}}$</th><th>Characteristics</th></tr>
<tr><td>0 (LOW)</td><td>0</td><td>0</td><td>Interrupt acknowledge</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Read I/O Port</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Write I/O Port</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Halt</td></tr>
<tr><td>1 (HIGH)</td><td>0</td><td>0</td><td>Code Access</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Read Memory</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Write Memory</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Passive</td></tr>
</table>
</td></tr>
<tr><td>$\overline{\text{RQ}}/\overline{\text{GT0}}$,
$\overline{\text{RQ}}/\overline{\text{GT1}}$</td><td>30, 31</td><td>I/O</td><td><b>Request/Grant:</b> pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{\text{RQ}}/\overline{\text{GT}}_0$ having higher priority than $\overline{\text{RQ}}/\overline{\text{GT}}_1$. $\overline{\text{RQ}}/\overline{\text{GT}}$ has an internal pull-up resistor, so may be left unconnected. The request/grant sequence is as follows (See Fig. 8):
1. A pulse of one CLK wide from another local bus master indicates a local bus request ("hold") to the MBL 8088 (pulse 1).
2. During a T_4 or T_1 clock cycle, a pulse one clock wide from the MBL 8088 to the requesting master (pulse 2), indicates that the MBL 8088 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge". The same rules as for HOLD/HOLDA apply as for when the bus is released.
3. A pulse one CLK wide from the requesting master indicates to the MBL 8088 (pulse 3) that the "hold" request is about to end and that the MBL 8088 can reclaim the local bus at the next CLK. The CPU then enters T_4.</td></tr>
</table>

TABLE 1 – PIN DESCRIPTION (Continued)

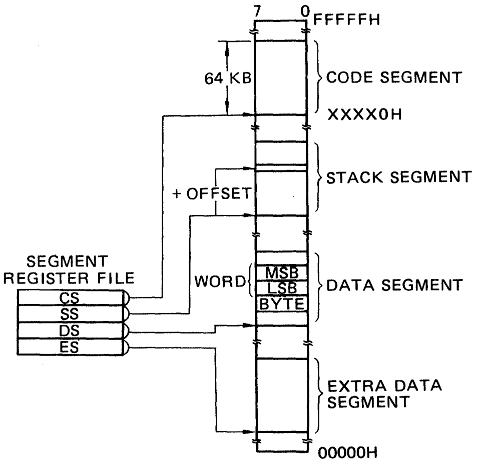
<table>
<tr><th>Symbol</th><th>Pin No.</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>$\overline{\text{RQ}/\text{GT0}}$,
$\overline{\text{RQ}/\text{GT1}}$</td><td>30, 31</td><td>I/O</td><td>Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active LOW.
If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met:
1. Request occurs on or before T_2.
2. Current cycle is not the low byte of a word.
3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence.
4. A locked instruction is not currently executing.
If the local bus is idle when the request is made the two possible events will follow:
1. Local bus will be released during the next clock.
2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.</td></tr>
<tr><td>$\overline{\text{LOCK}}$</td><td>29</td><td>O</td><td><b>$\overline{\text{LOCK}}$:</b> indicates that other system bus masters are not to gain control of the system bus while $\overline{\text{LOCK}}$ is active (LOW). The $\overline{\text{LOCK}}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to 3-state off in "hold acknowledge".</td></tr>
<tr><td>QS1, QS0</td><td>24, 25</td><td>O</td><td><b>Queue Status:</b> provide status to allow external tracking of the internal MBL 8088 instruction queue.
The queue status is valid during the CLK cycle after which the queue operation is performed.
<table>
<tr><th>QS1</th><th>QS0</th><th>Characteristics</th></tr>
<tr><td>0 (LOW)</td><td>0</td><td>No operation</td></tr>
<tr><td>0</td><td>1</td><td>First byte of opcode from queue</td></tr>
<tr><td>1 (HIGH)</td><td>0</td><td>Empty the queue</td></tr>
<tr><td>1</td><td>1</td><td>Subsequent byte from queue</td></tr>
</table></td></tr>
<tr><td>–</td><td>34</td><td>O</td><td>Pin 34 is always high in the maximum mode.</td></tr>
</table>

FUNCTIONAL DESCRIPTION

MEMORY ORGANIZATION

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra data, and stack segments of up to 64K bytes each, with each segment falling on 16-byte boundaries. (See Fig. 3.)

Fig. 3 – MEMORY ORGANIZATION

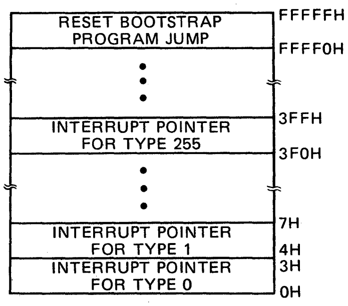


All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the rules of the following table. All information in one segment type share the same logical attributes (e.g. code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster, and more structured.

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location. The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Certain locations in memory are reserved for specific CPU operations. (See Fig. 4.) Locations from addresses FFFF0H through FFFFFH are reserved for operations including a jump to the initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations 00000H through 003FFH are reserved for interrupt operations. Four-byte pointers consisting of a 16-bit segment address and a 16-bit offset address direct program flow to one of the 256 possible interrupt service routines. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

Fig. 4 – RESERVED MEMORY LOCATIONS



MINIMUM AND MAXIMUM MODES

The requirements for supporting minimum and maximum MBL 8088 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the MBL 8088 is equipped with a strap pin (MN/$\overline{\text{MX}}$) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/$\overline{\text{MX}}$ pin is strapped to GND, the MBL 8088 defines pins 24 through 31 and 34 in maximum mode. When the MN/$\overline{\text{MX}}$ pin is strapped to V_{CC}, the MBL 8088 generates bus control signals itself on pins 24 through 31 and 34.

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when: relative to stack, destination of string operation, or explicity overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicity selected using a segment override.

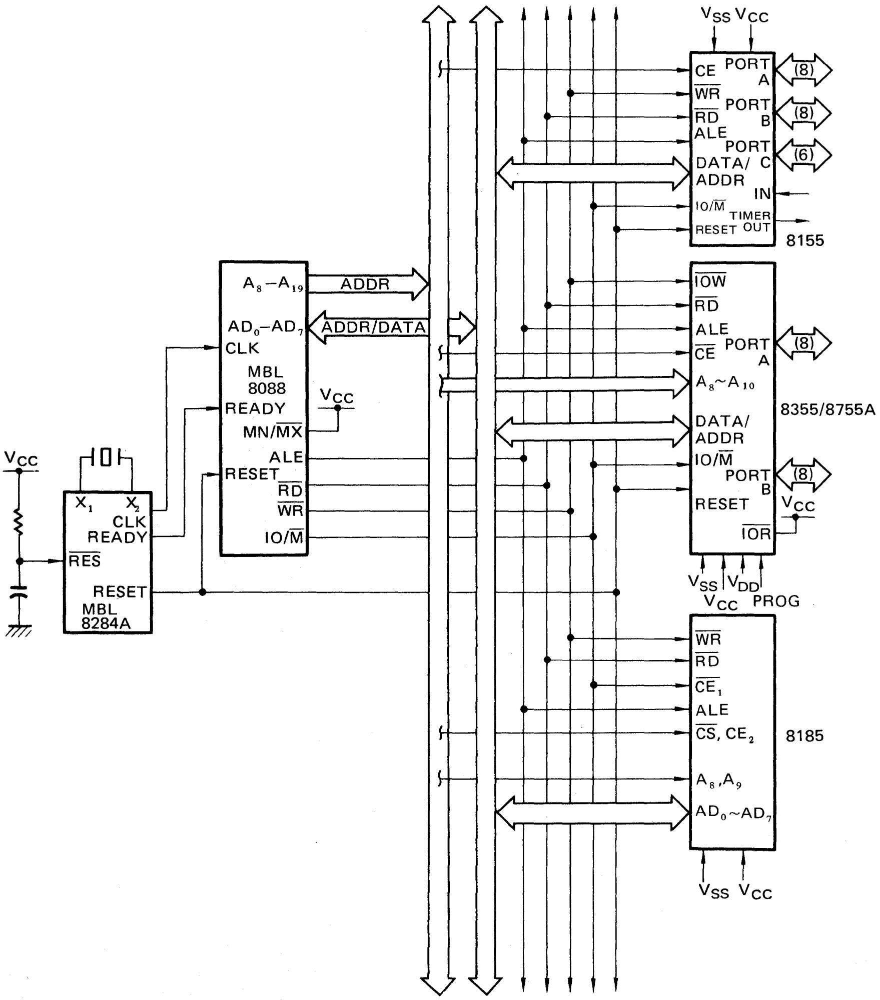
The minimum mode MBL 8088 can be used with either a multiplexed or demultiplexed bus. The multiplexed bus configuration is compatible with the MCS-85* multiplexed bus peripherals (Intel 8155, 8156, 8355, 8755A, and 8185). This configuration (See Fig. 5.) provides the user with a minimum chip count system. This architecture provides the MBL 8088 processing power in a highly integrated form.

The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). A third latch can be used for buffering if the address bus loading requires it. An MBL 8286 or MBL 8287 transceiver can also be used if data bus buffering is required. (See Fig. 6.) The MBL 8088 provides $\overline{\text{DEN}}$ and DT/$\overline{\text{R}}$ to control the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the MBL 8288 bus controller. (See Fig. 7.) The MBL 8288 decodes status lines $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$, and provides the system with all bus control signals. Moving the bus control to the MBL 8288 provides better source and sink current capability to the control lines, and frees the MBL 8088 pins for extended large system features. Hardware lock, queue status, and two request/grant interfaces are provided by the MBL 8088 in maximum mode. These features allow coprocessors in local bus and remote bus configurations.

*Trade Mark of Intel Corporation, USA

Fig. 5 – MULTIPLEXED BUS CONFIGURATION



1

Fig. 6 — DEMULTIPLEXED BUS CONFIGRATION EXAMPLES (MINIMUM MODE)

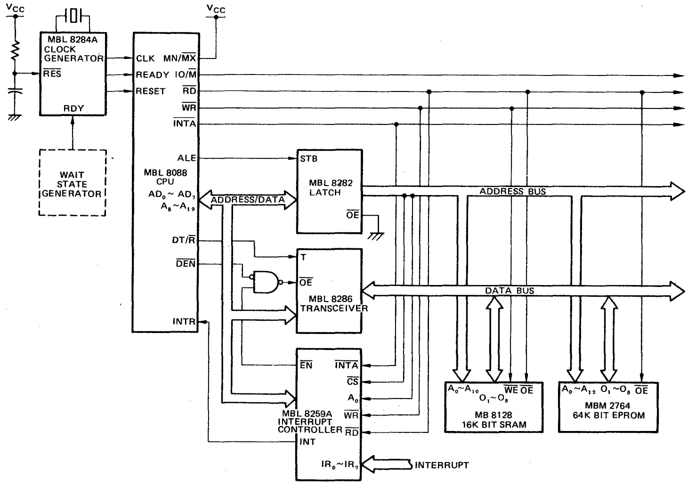
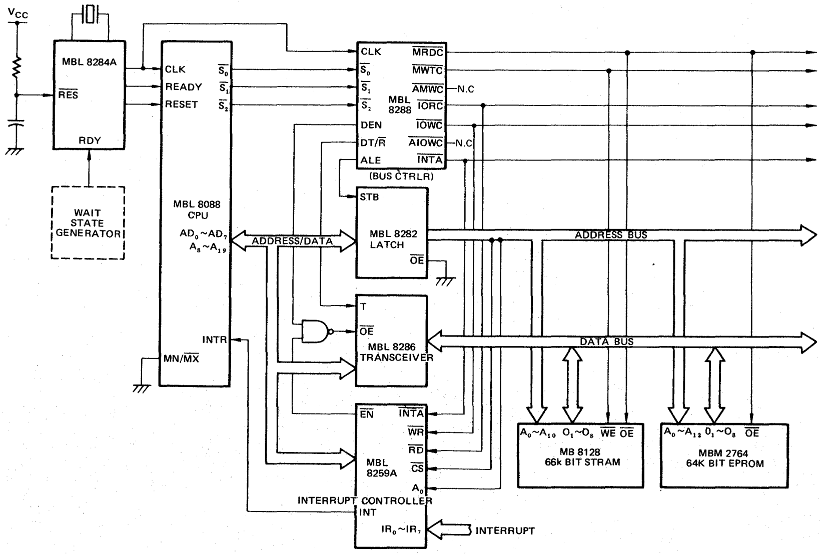


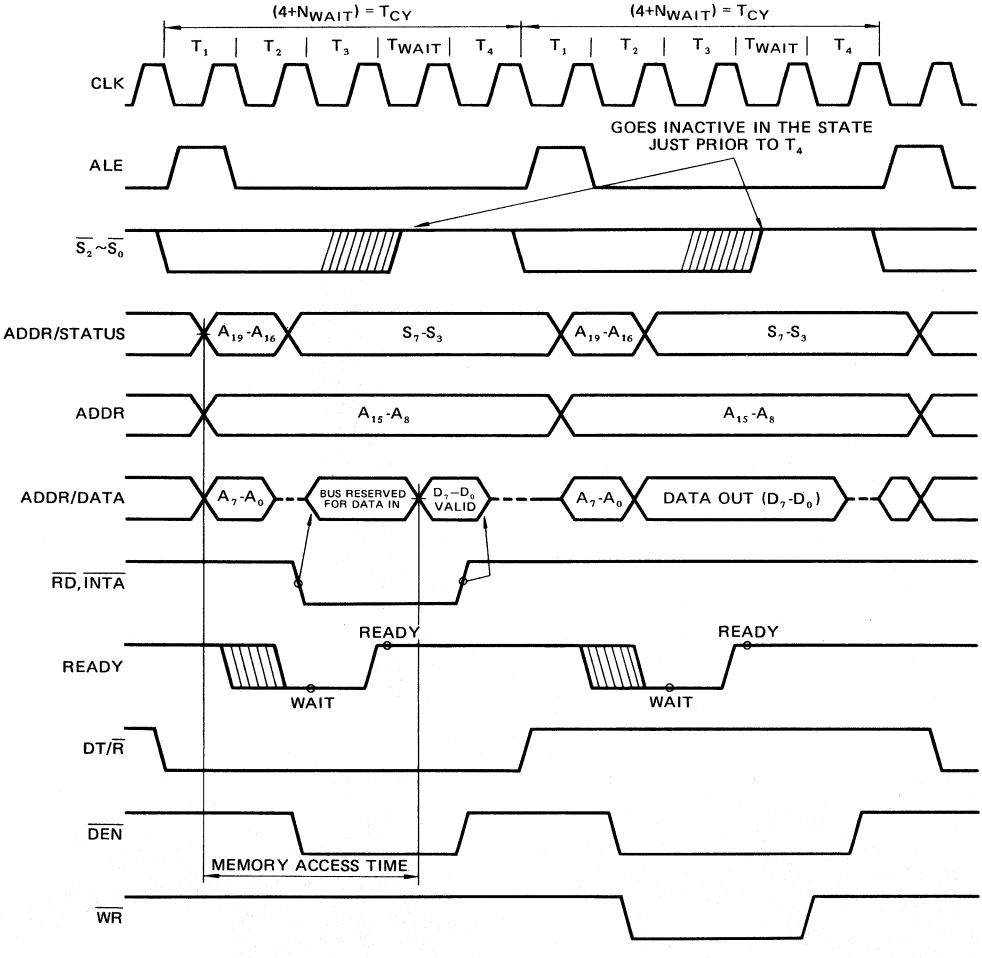
Fig. 7 — FULLY BUFFERED SYSTEM EXAMPLE (MAXIMUM MODE)



BUS OPERATION

The MBL 8088 address/data bus is broken into three parts – the lower eight address/data bits (AD_0–AD_7), the middle eight address bits (A_8 – A_{15}), and the upper four address bits (A_{16} –A_{19}). The address/data bits and the highest four address bits are time multiplexed. This technique provides the most efficient use of pins on the processor, permitting the use of a standard 40 lead package. The middle eight address bits are not multiplexed, i.e. they remain valid throughout each bus cycle. In addition, the bus can be demultiplexed at the processor with a single address latch if a standard, non-multiplexed bus is desired for the system.

Fig. 8 – BASIC SYSTEM TIMING



Each processor bus cycle consists of at least four CLK cycles. These are referred to as T_1, T_2, T_3, and T_4. (See Fig. 8). The address is emitted from the processor during T_1 and data transfer occurs on the bus during T_3 and T_4. T_2 is used primarily for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "wait" states (Tw) are inserted between T_3 and T_4. Each inserted "wait" state is of the same duration as a CLK cycle. Periods can occur between MBL 8088 driven bus cycles. These are referred to as "idle" states (Ti), or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

During T_1 of any bus cycle, the ALE (address latch enable) signal is emitted (by either the processor or the MBL 8288 bus controller, depending on the MN/$\overline{\text{MX}}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$ are used by the bus controller, in maximum mode, to identify the type of bus transaction according to the following table:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	CHARACTERISTICS
0 (Low)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (High)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S_3 through S_6 are multiplexed with high order address bits and are therefore valid during T_2 through T_4, S_3 and S_4 indicate which segment register was used for this bus cycle in forming the address according to the following table:

S_4	S_3	CHARACTERISTICS
0 (Low)	0	Alternate Data (Extra Segment)
0	1	Stack
1 (High)	0	Code or None
1	1	Data

S_5 is a reflection of the PSW interrupt enable bit. S_6 is always equal to 0.

I/O ADDRESSING

In the MBL 8088, I/O operations can address up to a maximum of 64K I/O registers. The I/O address appears in the same format as the memory address on bus lines $A_{15} - A_0$. The address lines $A_{19} - A_{16}$ are zero in I/O operations. The variable I/O instructions, which use register DX as a pointer, have full address capability, while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

Designers familiar with the Intel 8085 or upgrading an 8085 design should note that the 8085 addresses I/O with an 8-bit address on both halves of the 16-bit address bus. The MBL 8088 uses a full 16-bit address on its lower 16 address lines.

EXTERNAL INTERFACF

PROCESSOR RESET AND INITIALIZATION

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The MBL 8088 RESET is required to be HIGH for greater than four clock cycles. The MBL 8088 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 clock cycles. After this interval the MBL 8088 operates normally, beginning with the instruction in absolute location FFFFOH. (See Fig. 4.) The RESET input is internally synchronized to the processor clock. At initialization, the HIGH to LOW transition of RESET must occur no sooner than 50 μs after power up, to allow complete initialization of the MBL 8088.

If INTR is asserted sooner than nine clock cycles after the end of RESET, the processor may execute one instruction before responding to the interrupt.

All 3-state outputs float to 3-state OFF during RESET. Status is active in the idle state for the first clock after RESET becomes active and then floats to 3-state OFF.

INTERRUPT OPERATIONS

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the instruction set description in the MBL 8086 Family

User's Manual. Hardware interrupts can be classified as non-maskable or maskable.

Interrupts result in a transfer of control to a new program location. A 256 element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Fig. 4), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type". An interrupting device supplies an 8-bit type number, during the interrupt acknowledge sequence, which is used to vector through the appropriate element to the new interrupt service program location.

NON-MASKABLE INTERRUPT (NMI)

The processor provides a single non-maskable interrupt (NMI) pin which has higher priority than the maskable interrupt request (INTR) pin. A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW to HIGH transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than two clock cycles, but is not required to be synchronized to the clock. Any higher going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves (2 bytes in the case of word moves) of a block type instruction. Worst case response to NMI would be for multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

MASKABLE INTERRUPT (INTR)

The MBL 8088 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable (IF) flag bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block type instruction. During interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single step), although the FLAGS register which is automatically pushed onto the stack reflects the state of the processor prior to the interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (See Fig. 9), the processor executes two successive (back to back) interrupt acknowledge cycles. The MBL 8088 emits the LOCK signal (maximum mode only) from T_2 of the first bus cycle until T_2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interrupt system (e.g., MBL 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The interrupt return instruction includes a flags pop which returns the status of the original interrupt enable bit when it restores the flags.

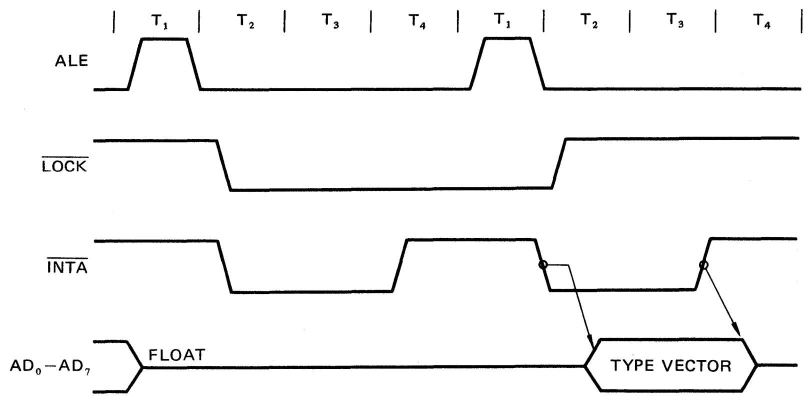
HALT

When a software HALT instruction is executed, the processor indicates that it is entering the HALT state in one of two ways, depending upon which mode is strapped. In minimum mode, the processor issues ALE, delayed by one clock cycle, to allow the system to latch the halt status. Halt status is available on IO/$\overline{\text{M}}$, DT/$\overline{\text{R}}$, and $\overline{\text{SSO}}$. In maximum mode, the processor issues appropriate HALT status on $\overline{S}_2$, $\overline{S}_1$, and $\overline{S}_0$, and the MBL 8288 bus controller issues one ALE. The MBL 8088 will not leave the HALT state when a local bus hold is entered while in HALT. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An interrupt request or RESET will force the MBL 8088 out of the HALT state.

READ/MODIFY/WRITE (SEMAPHORE) OPERATIONS VIA LOCK

The $\overline{\text{LOCK}}$ status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This allows the processor to perform read/modify/write operations on memory (via the "exchange register with memory" instruction), without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{\text{LOCK}}$ signal is activated (LOW) in the clock cycle following decoding of the LOCK prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the LOCK prefix. While $\overline{\text{LOCK}}$ is active, a request on a $\overline{\text{RQ}}/\overline{\text{GT}}$ pin will be recorded, and then honored at the end of the $\overline{\text{LOCK}}$.

Fig. 9 – INTERRUPT ACKNOWLEDGE SEQUENCE



EXTERNAL SYNCHRONIZATION VIA TEST

As an alternative to interrupts, the MBL 8088 provides a single software-testable input pin (TEST). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the TEST input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the MBL 8088 3-states all output drivers. If interrupts are enabled, the MBL 8088 will recognize interrupts and process them. The WAIT instruction is then refetched, and reexecuted.

BASIC SYSTEM TIMING

In minimum mode, the NM/MX pin is strapped to V_{CC} and the processor emits bus control signals compatible with the 8085 bus structure. In maximum mode, the MN/MX pin is strapped to GND and the processor emits coded status information which the MBL 8288 bus controller uses to generate MULTIBUS* compatible bus control signals.

SYSTEM TIMING – MINIMUM SYSTEM

(See Fig. 8)

The read cycle begins in T_1 with the assertion of the address latch enable (ALE) signal. The trailing (low going) edge of this signal is used to latch the address information, which is valid on the address/data bus (AD_0–AD_7) at this time, into the MBL 8282/8283 latch. Address lines A_8 through A_{15} do not need to be latched because they remain valid throughout the bus cycle. From T_1 to T_4 the IO/M signal indicates a memory or I/O operation. At T_2 the address is removed from the address/data bus and the bus goes to a high impedance state. The read control signal is also asserted at T_2. The read (RD) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver (MBL 8286/8287) is required to buffer the MBL 8088 local bus, signals DT/R and DEN are provided by the MBL 8088.

A write cycle also begins with the assertion of ALE and the emission of the address. The IO/M signal is again asserted to indicate a memory or I/O write operation. In T_2, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T_4. During T_2, T_3, and T_W, the processor asserts the write control signal. The write (WR) signal becomes active at the beginning of T_2, as opposed to the read, which is delayed somewhat into T_2 to provide time for the bus to float.

*Trade Mark of Intel Corporation, USA

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge ($\overline{\text{INTA}}$) signal is asserted in place of the read ($\overline{\text{RD}}$) signal and the address bus is floated. (See Fig. 9.) In the second of two successive $\overline{\text{INTA}}$ cycles, a byte of information is read from the data bus, as supplied by the interrupt system logic (i.e. MBL 8259A priority interrupt controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into the interrupt vector lookup table, as described earlier.

BUS TIMING – MEDIUM COMPLEXITY SYSTEMS

(See Fig. 10)

For medium complexity systems, the MN/$\overline{\text{MX}}$ pin is connected to GND and the MBL 8288 bus controller is added to the system, as well as an MBL 8282/8283 latch for latching the system address, and an MBL 8286/8287 transceiver to allow for bus loading greater than the MBL 8088 is capable of handling. Signals ALE, $\overline{\text{DEN}}$, and DT/$\overline{\text{R}}$ are generated by the MBL 8288 instead of the processor in this configuration, although their timing remains relatively the same. The MBL 8088 status outputs ($\overline{S_2}$, $\overline{S_1}$, and $\overline{S_0}$) provide type of cycle information and become MBL 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The MBL 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The MBL 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The MBL 8286/8287 transceiver receives the usual T and $\overline{\text{OE}}$ inputs from the MBL 8288's DT/$\overline{\text{R}}$ and $\overline{\text{DEN}}$ outputs.

The pointer into the interrupt vector table, which is passed during the second $\overline{\text{INTA}}$ cycle, can derive from an MBL 8259A located on either the local bus or the system bus. If the master MBL 8259A priority interrupt controller is positioned on the local bus, a TTL gate is required to disable the MBL 8286/8287 transceiver when reading from the master MBL 8259A during the interrupt acknowledge sequence and knowledge sequence and software "poll".

THE MBL 8088 COMPARED TO THE MBL 8086

The MBL 8088 CPU is an 8-bit processor designed around the MBL 8086 internal structure. Most internal functions of the MBL 8088 are identical to the equivalent MBL 8086 functions. The MBL 8088 handles the external bus the same way the MBL 8086 does with the distinction of handling only 8 bits at a time. Sixteen-bit operands are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time. The internal register structure is identical and all instructions have the same end result. The differences between the MBL 8088 and MBL 8086 are outlined below. The engineer who is unfamiliar with the MBL 8086 is referred to the MBL 8086 Family User's Manual, Chapters 2 and 4, for function description and instruction set information.

Internally, there are three differences between the MBL 8088 and the MBL 8086. All changes are related to the 8-bit bus interface.

- The queue length is 4 bytes in the MBL 8088, whereas MBL 8086 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The MBL 8088 BIU will fetch a new instruction to load into the queue each time there is a 1 byte hole (space available) in the queue. The MBL 8086 waits until a 2-byte space is available.
- The internal execution time of the instruction set is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU is also limited by the speed of instruction fetches. This latter problem only occurs when a series of simple operations occur. When the more sophisticated instructions of the MBL 8088 are being used, the queue has time to fill and the execution proceeds as fast as the execution unit will allow.

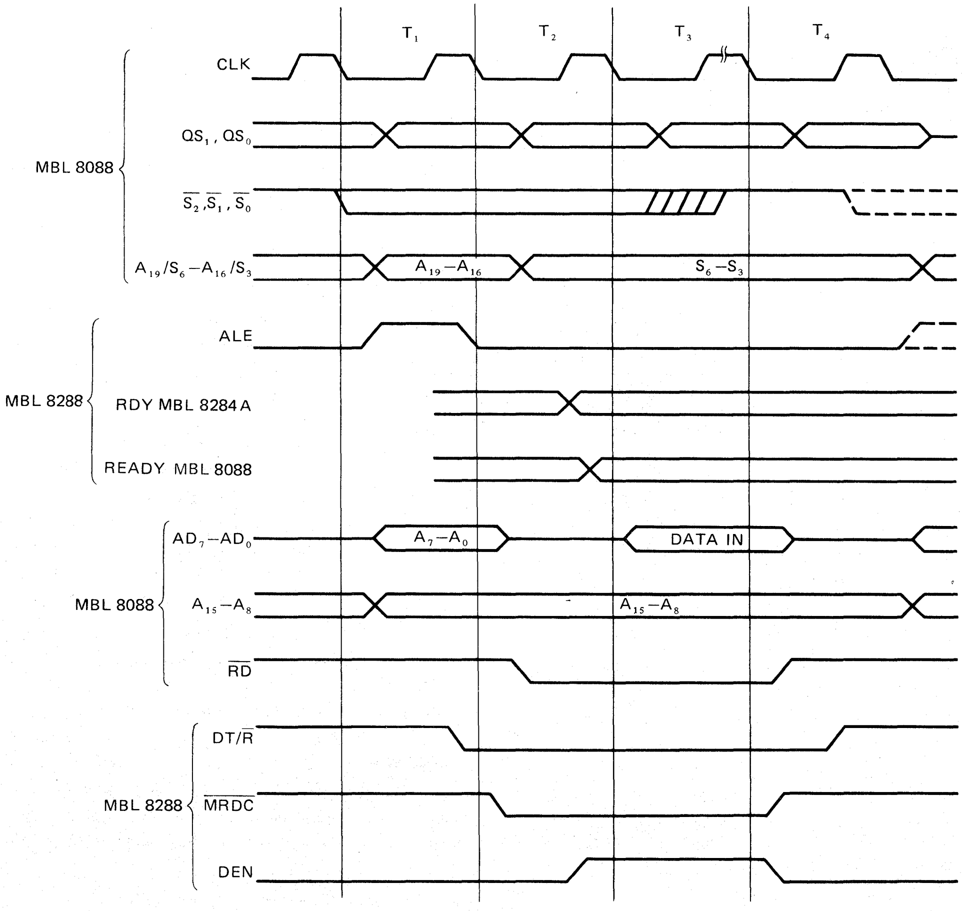
The MBL 8088 and MBL 8086 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally as well on an MBL 8088 or an MBL 8086.

The hardware interface of the MBL 8088 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes:

- A_8–A_{15} – These pins are only address outputs on the MBL 8088. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- $\overline{\text{BHE}}$ has no meaning on the MBL 8088 and has been eliminated.

- $\overline{SSO}$ provides the $\overline{SO}$ status information in the minimum mode. This output occurs on pin 34 in minimum mode only. DT/$\overline{R}$, IO/$\overline{M}$, and $\overline{SSO}$ provide the complete bus status in minimum mode.
- IO/$\overline{M}$ has been inverted to be compatible with the MCS-85* bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with ALE.

Fig. 10 – MEDIUM COMPLEXITY SYSTEM TIMING



*Trade Mark of Intel Corporation, USA

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .... 0°C to 70°C
Storage Temperature .... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground .... −0.5 to +7V
Power Dissipation .... 2.5 Watt

*NOTE: 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 rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS (MBL 8088: V_{CC} = 5 V ± 10%, T_A = 0°C to 70°C)
(MBL 8088-2: V_{CC} = 5 V ± 5%, T_A = 0°C to 70°C)
(MBL 8088-1: V_{CC} = 5 V ± 5%, T_A = 0°C to 70°C)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400 μA
I_{CC}	Power Supply Current: MBL 8088 MBL 8088-2 MBL 8088-1		340 350 360	mA	T_A = 25°C
I_{LI}	Input Leakage Current		±10	μA	$0V \leqslant V_{IN} \leqslant V_{CC}$
I_{LO}	Output Leakage Current		±10	μA	$0.45V \leqslant V_{OUT} \leqslant V_{CC}$
V_{CL}	Clock Input Low Voltage	-0.5	+0.6	V	
V_{CH}	Clock Input High Voltage	3.9	V_{CC} + 1.0	V	
C_{IN}	Capacitance of Input Buffer (All input except AD_0–AD_7 $\overline{RQ}/\overline{GT}$)		15	pF	fc = 1 MHz
C_{IO}	Capacitance of I/O Buffer (AD_0–AD_7 $\overline{RQ}/\overline{GT}$)		15	pF	fc = 1 MHz

A.C. CHARACTERISTICS (MBL 8088: V_{CC} = 5 V ± 10%, T_A = 0°C to 70°C)

(MBL 8088-2: V_{CC} = 5 V ± 5%, T_A = 0°C to 70°C)
(MBL 8088-1: V_{CC} = 5 V ± 5% T_A = 0°C to 70°C)

MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

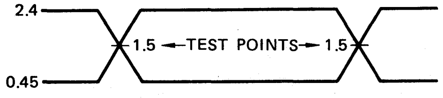
Symbol	Parameter	MBL 8088		MBL 8088-2		MBL 8088-1 (Preliminary)		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
TCLCL	CLK Cycle Period	200	500	125	500	100	500	ns	
TCLCH	CLK Low Time	118		68		53		ns	
TCHCL	CLK High Time	69		44		39		ns	
TCH1CH2	CLK Rise Time		10		10		10	ns	From 1.0 V to 3.5 V
TCL2CL2	CLK Fall Time		10		10		10	ns	From 3.5 V to 1.0 V
TDVCL	Data in Setup Time	30		20		5		ns	
TCLDX	Data in Hold Time	10		10		10		ns	
TR1VCL	RDY Setup Time into MBL 8284A (See Notes 1, 2)	35		35		35		ns	
TCLR1X	RDY Hold Time into MBL 8284A (See Notes 1, 2)	0		0		0		ns	
TRYHCH	READY Setup Time into MBL 8088	118		68		53		ns	
TCHRYX	READY Hold Time into MBL 8088	30		20		20		ns	
TRYLCL	READY Inactive to CLK (See Note 3)	-8		-8		-10		ns	
THVCH	HOLD Setup Time	35		20		20		ns	
TINVCH	INTR, NMI, $\overline{\text{TEST}}$ Setup Time (See Note 2)	30		15		15		ns	
TILIH	Input Rise Time (Except CLK)		20		20		20	ns	From 0.8 V to 2.0 V
TIHIL	Input Fall Time (Except CLK)		12		12		12	ns	From 2.0 V to 0.8 V

A.C. CHARACTERISTICS (Continued)

TIMING RESPONSES

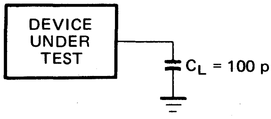
Symbol	Parameter	MBL 8088		MBL 8088-2		MBL 8088-1 (Preliminary)		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
TCLAV	Address Valid Delay	10	110	10	60	10	50	ns	C_L=20–100pF for all MBL 8088 Outputs in addition to interna internal loads
TCLAX	Address Hold Time	10		10		10		ns	
TCLAZ	Address Float Delay	TCLAX	80	TCLAX	50	10		ns	
TLHLL	ALE Width	TCLCH−20		TCLCH−10		TCLCH−10		ns	
TCLLH	ALE Active Delay		80		50		40	ns	
TCHLL	ALE Inactive Delay		85		55		45	ns	
TLLAX	Address Hold Time to ALE Inactive	TCHCL−10		TCHCL−10		TCHCL−10		ns	
TCLDV	Data Valid Delay	10	110	10	60	10	50	ns	
TCHDX	Data Hold Time	10		10		10		ns	
TWHDX	Data Hold Time After WR	TCLCH−30		TCLCH−30		TCLCH−25		ns	
TCVCTV	Control Active Delay 1	10	110	10	70	10	50	ns	
TCHCTV	Control Active Delay 2	10	110	10	60	10	45	ns	
TCVCTX	Control Inactive Delay	10	110	10	70	10	50	ns	
TAZRL	Address Float to READ Active	0		0		0		ns	
TCLRL	RD Active Delay	10	165	10	100	10	70	ns	
TCLRH	RD Inactive Delay	10	150	10	80	10	60	ns	
TRHAV	RD Inactive to Next Address Active	TCLCL−45		TCLCL−40		TCLCL−35		ns	
TCLHAV	HLDA Valid Delay	10	160	10	100	10	60	ns	
TRLRH	RD Width	2TCLCL−75		2TCLCL−50		2TCLCL−40		ns	
TWLWH	WR Width	2TCLCL−60		2TCLCL−40		2TCLCH−35		ns	
TAVAL	Address Valid to ALE Low	TCLCH−60		TCLCH−40		TCLCH−35		ns	
TOLOH	Output Rise Time		20		20		20	ns	From 0.8 V to 2.0 V
TOHOL	Output Fall Time		12		12		12	ns	From 2.0 V to 0.8 V

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". THE CLOCK IS DRIVEN AT 4.3V AND 0.25V TIMING MEASUREMENTS ARE MADE AT 1.5V FOR BOTH A LOGIC "1" AND "0".

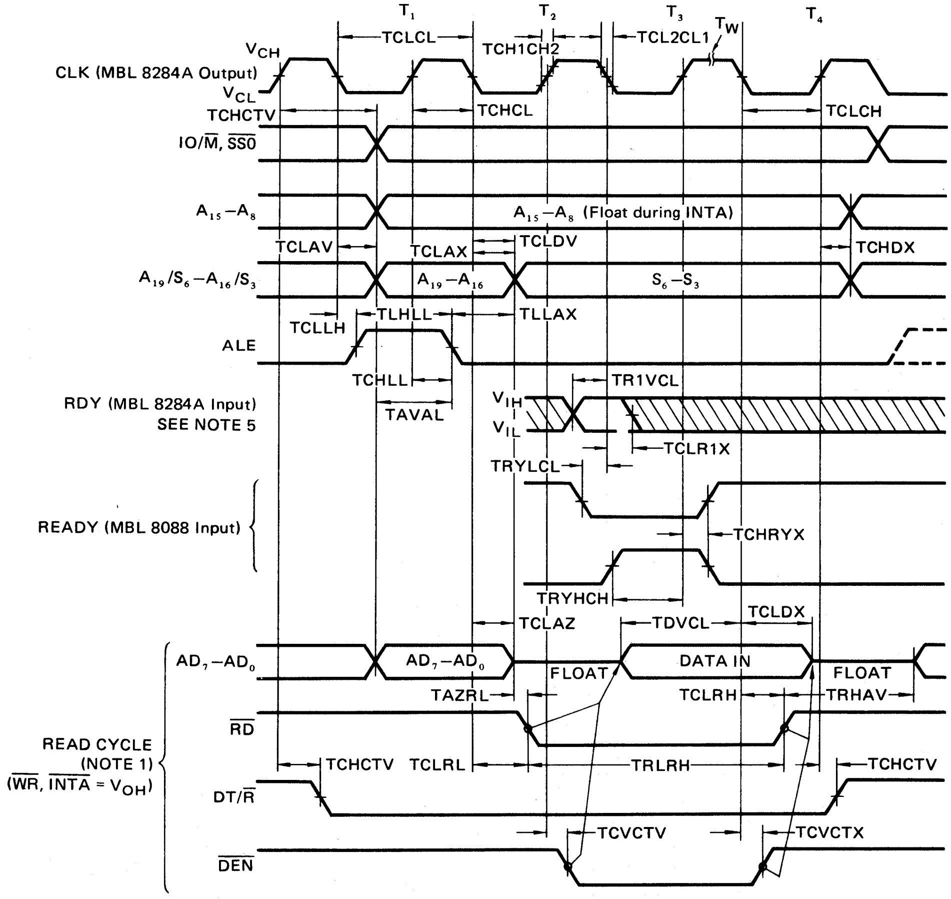
A.C. TESTING LOAD CIRCUIT



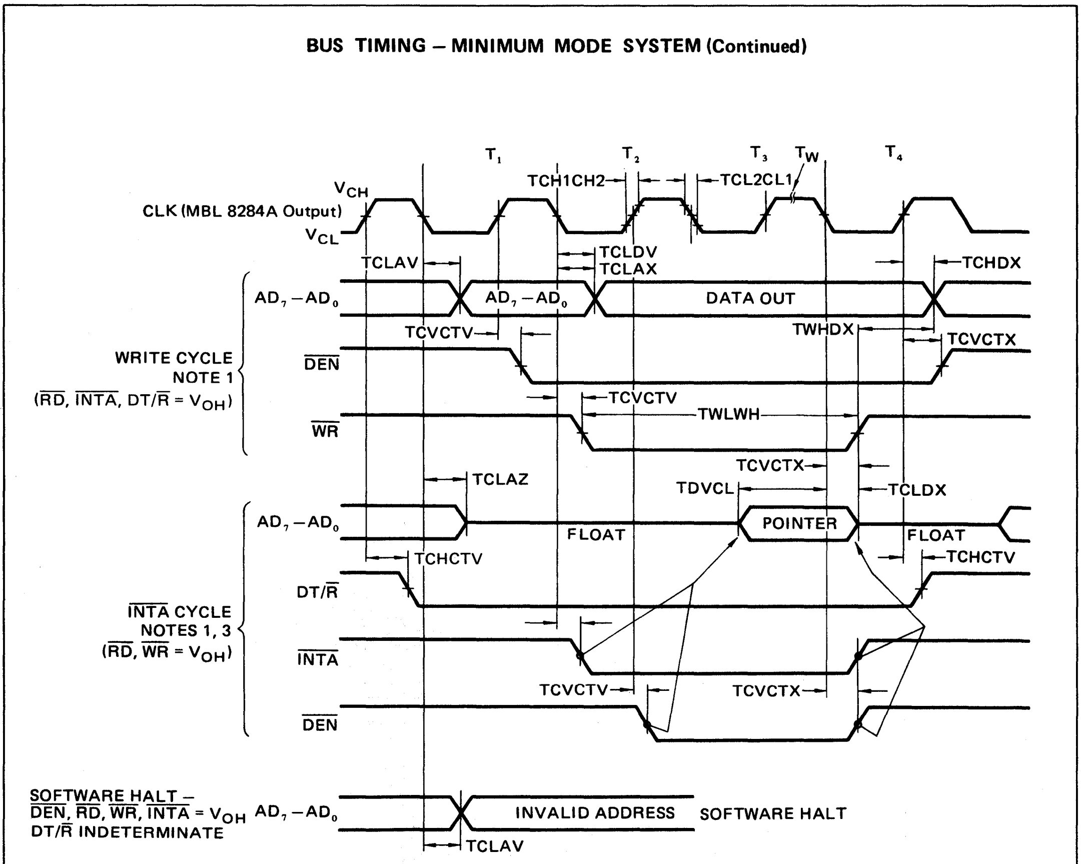
C_L: INCLUDES JIG CAPACITANCE

WAVEFORMS

BUS TIMING – MINIMUM MODE SYSTEM



WAVEFORMS (Continued)



NOTES:
1. ALL SIGNALS SWITCH BETWEEN V_{OH} AND V_{OL} UNLESS OTHERWISE SPECIFIED.
2. RDY IS SAMPLED NEAR THE END OF T_2, T_3, T_W TO DETERMINE IF T_W MACHINES STATES ARE TO BE INSERTED.
3. TWO INTA CYCLES RUN BACK-TO-BACK. THE MBL 8088 LOCAL ADDR/DATA BUS IS FLOATING DURING BOTH INTA CYCLES. CONTROL SIGNALS ARE SHOWN FOR THE SECOND INTA CYCLE.
4. SIGNALS AT MBL 8284A ARE SHOWN FOR REFERENCE ONLY.
5. ALL TIMING MEASUREMENTS ARE MADE AT 1.5V UNLESS OTHERWISE NOTED.

A.C. CHARACTERISTICS (Continued)

MAX MODE SYSTEM (USING MBL 8288 BUS CONTROLLER) TIMING REQUIREMENTS

Symbol	Parameter	MBL 8088		MBL 8088-2		MBL 8088-1 (Preliminary)		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
TCLCL	CLK Cycle Period	200	500	125	500	100	500	ns	
TCLCH	CLK Low Time	118		68		53		ns	
TCHCL	CLK High Time	69		44		39		ns	
TCH1CH2	CLK Rise Time		10		10		10	ns	From 1.0 V to 3.5 V
TCL2CL1	CLK Fall Time		10		10		10	ns	From 3.5 V to 1.0 V
TDVCL	Data in Setup Time	30		20		5		ns	
TCLDX	Data in Hold Time	10		10		10		ns	
TR1VCL	RDY Setup Time into MBL 8284A (See Notes 1, 2)	35		35		35		ns	
TCLR1X	RDY Hold Time into MBL 8284A (See Notes 1, 2)	0		0		0		ns	
TRYHCH	READY Setup Time into MBL 8088	118		68		53		ns	
TCHRYX	READY Hold Time into MBL 8088	30		20		20		ns	
TRYLCL	READY Inactive to CLK (See Note 2)	-8		-8		-10		ns	
TINVCH	Setup Time for Recognition (INTR, NMI, TEST) (See Note 2)	30		15		15		ns	
TGVCH	RQ/GT Setup Time	30		15		12		ns	
TCHGX	RQ Hold Time into MBL 8086	40		30		20		ns	
TILIH	Input Rise Time (Except CLK)		20		20		20	ns	From 0.8 V to 2.0 V
TIHIL	Input Fall Time (Except CLK)		12		12		12	ns	From 2.0 V to 0.8 V

A.C. CHARACTERISTICS (Continued)

TIMING RESPONSES

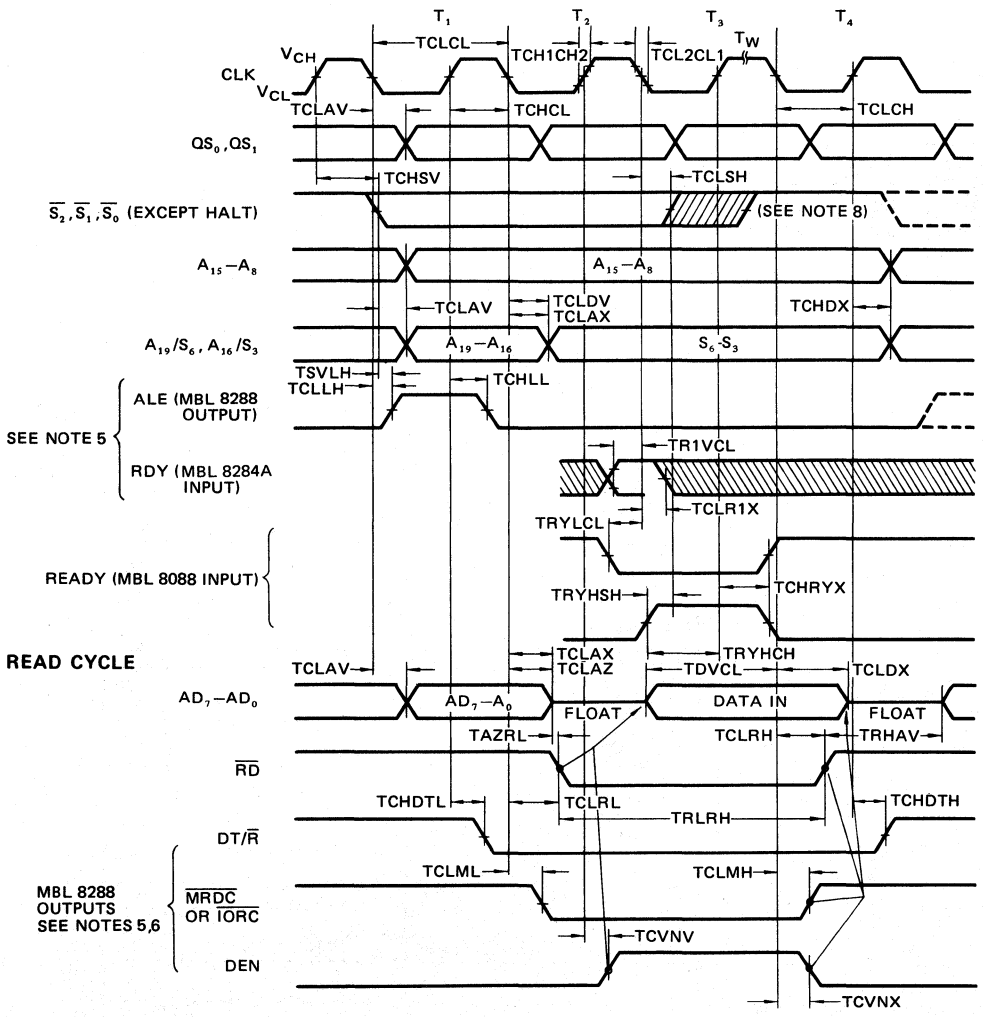
Symbol	Parameter	MBL 8088		MBL 8088-2		MBL 8088-1 (Preliminary)		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
TCLML	Command Active Delay (See Note 1)	10	35	10	35	10	35	ns	
TCLMH	Command Inactive Delay (See Note 1)	10	35	10	35	10	35	ns	C_L=20–100pF for all MBL 8088 Outputs in addition to internal loads
TRYHSH	READY Active to Status Passive (See Note 3)		110		65		45	ns	
TCHSV	Status Active Delay	10	110	10	60	10	45	ns	
TCLSH	Status Inactive Delay	10	130	10	70	10	55	ns	
TCLAV	Address Valid Delay	10	110	10	60	10	50	ns	
TCLAX	Address Hold Time	10		10		10		ns	
TCLAZ	Address Float Delay	TCLAX	80	TCLAX	50	10	40	ns	
TSVLH	Status Valid to ALE High (See Note 1)		15		15		15	ns	
TSVMCH	Status Valid to MCE High (See Note 1)		15		15		15	ns	
TCLLH	CLK Low to ALE Valid (See Note 1)		15		15		15	ns	
TCLMCH	CLK Low to MCE High (See Note 1)		15		15		15	ns	
TCHLL	ALE Inactive Delay (See Note 1)		15		15		15	ns	
TCLMCL	MCE Inactive Delay (See Note 1)		15		15		15	ns	
TCLDV	Data Valid Delay	10	110	10	60	10	50	ns	
TCHDX	Data Hold Time	10		10		10		ns	
TCVNV	Control Active Delay (See Note 1)	5	45	5	45	5	45	ns	
TCVNX	Control Inactive Delay (See Note 1)	10	45	10	45	10	45	ns	
TAZRL	Address Float to Read Active	0		0		0		ns	
TCLRL	RD Active Delay	10	165	10	100	10	70	ns	
TCLRH	RD Inactive Delay	10	150	10	80	10	60	ns	
TRHAV	RD Inactive to Next Address Active	TCLCL–45		TCLCL–40		TCLCL–35		ns	
TCHDTL	Direction Control Active Delay (See Note 1)		50		50		50	ns	
TCHDTH	Direction Control Inactive Delay (See Note 1)		30		30		30	ns	
TCLGL	GT Active Delay		85		50	0	45	ns	
TCLGH	GT Inactive Delay		85		50	0	45	ns	
TRLRH	RD Width	2TCLCL–75		2TCLCL–50		2TCLCL–40		ns	
TOLOH	Output Rise Time		20		20		20	ns	From 0.8 V to 2.0 V
TOHOL	Output Fall Time		12		12		12	ns	From 2.0 V to 0.8 V

NOTES:

1. Signal at MBL 8284A or MBL 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T_2 state (8 ns into T_3 state).

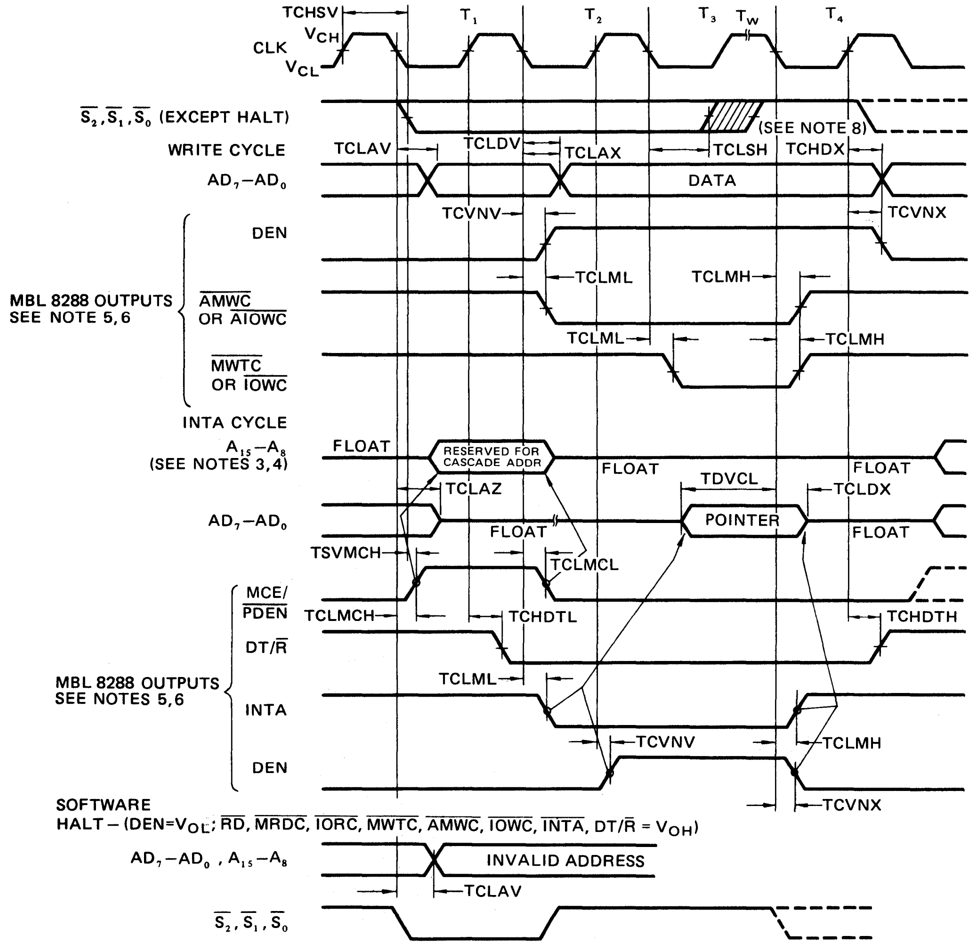
WAVEFORMS (Continued)

BUS TIMING—MAXIMUM MODE SYSTEM (USING MBL 8288)



WAVEFORMS (Continued)

BUS TIMING—MAXIMUM MODE SYSTEM (USING MBL 8288)

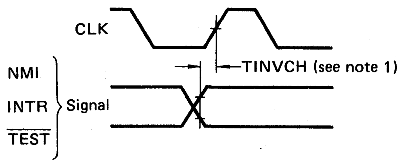


NOTES:

1. ALL SIGNALS SWITCH BETWEEN V_{OH} AND V_{OL} UNLESS OTHERWISE SPECIFIED.
2. RDY IS SAMPLED NEAR THE END OF T_2, T_3, T_W TO DETERMINE IF T_W MACHINES STATES ARE TO BE INSERTED.
3. CASCADE ADDRESS IS VALID BETWEEN FIRST AND SECOND $\overline{\text{INTA}}$ CYCLES.
4. TWO $\overline{\text{INTA}}$ CYCLES RUN BACK-TO-BACK. THE MBL 8088 LOCAL ADDR/DATA BUS IS FLOATING DURING BOTH $\overline{\text{INTA}}$ CYCLES. CONTROL FOR POINTER ADDRESS IS SHOWN FOR SECOND $\overline{\text{INTA}}$ CYCLE.
5. SIGNALS AT MBL 8284A OR MBL 8288 ARE SHOWN FOR REFERENCE ONLY.
6. THE ISSUANCE OF THE MBL 8288 COMMAND AND CONTROL SIGNALS ($\overline{\text{MRDC}}$, $\overline{\text{MWTC}}$, $\overline{\text{AMWC}}$, $\overline{\text{IORC}}$, $\overline{\text{IOWC}}$, $\overline{\text{AIOWC}}$, $\overline{\text{INTA}}$ AND DEN) LAGS THE ACTIVE HIGH MBL 8288 CEN.
7. ALL TIMING MEASUREMENTS ARE MADE AT 1.5V UNLESS OTHERWISE NOTED.
8. STATUS INACTIVE IN STATE JUST PRIOR TO T_4.

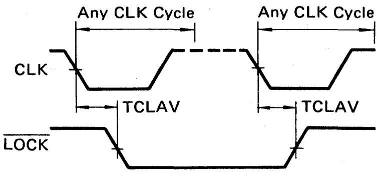
WAVEFORMS (Continued)

ASYNCHRONOUS SIGNAL RECOGNITION

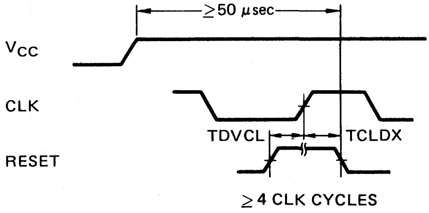


NOTE: 1. SETUP REQUIREMENTS FOR ASYNCHRONOUS SIGNALS ONLY TO GUARANTEE RECOGNITION AT NEXT CLK

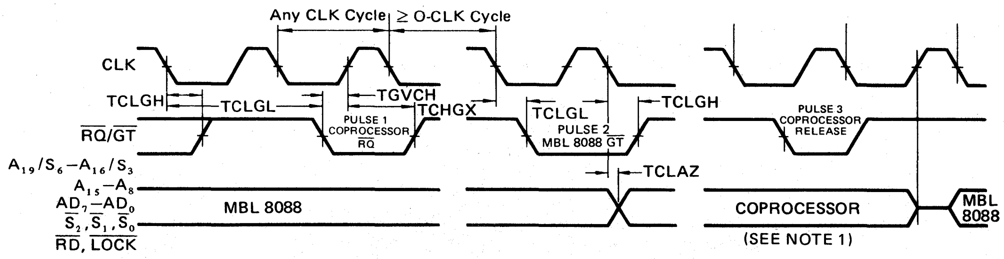
BUS LOCK SIGNAL TIMING (MAXIMUM MODE ONLY)



RESET TIMING



REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)



NOTE: 1. THE COPROCESSOR MAY NOT DRIVE THE BUSSES OUTSIDE THE REGION SHOWN WITHOUT RISKING CONTENTION.

HOLD/HOLD ACKNOWLEDGE TIMING (MINIMUM MODE ONLY)

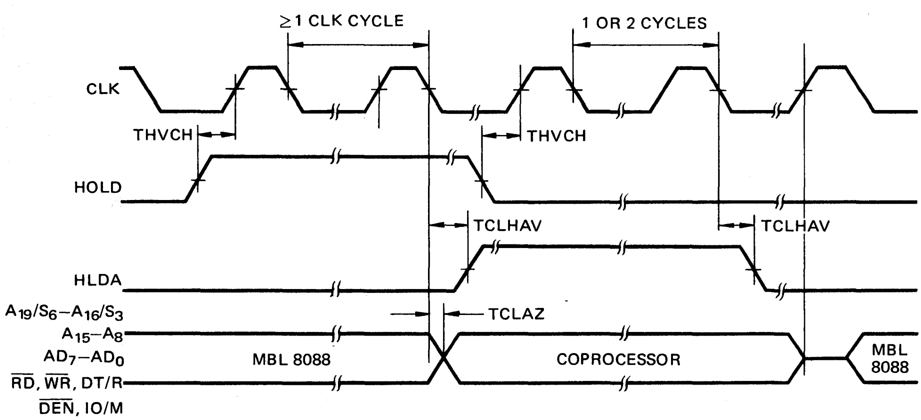


TABLE 2 – INSTRUCTION SET SUMMARY*

DATA TRANSFER				
MOV = Move:	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory to/from register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w=1
Immediate to register	1 0 1 1 w reg	data	data if w=1	
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment register	0 0 0 reg 1 1 1			
XCHG = Exchange:				
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with accumulator	1 0 0 1 0 reg			
IN = Input from:				
Fixed port	1 1 1 0 0 1 0 w	port		
Variable port	1 1 1 0 1 1 0 w			
OUT = Output to:				
Fixed port	1 1 1 0 0 1 1 w	port		
Variable port	1 1 1 0 1 1 1 w			
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1			
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m		
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m		
LAHF = Load AH with flags	1 0 0 1 1 1 1 1			
SAHF = Store AH into flags	1 0 0 1 1 1 1 0			
PUSHF = Push flags	1 0 0 1 1 1 0 0			
POPF = Pop flags	1 0 0 1 1 1 0 1			
ARITHMETIC				
ADD = Add:				
Reg./memory with register to either	0 0 0 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w=01
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w=1	
ADC = Add with carry:				
Reg./memory with register to either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w=01
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w=1	
INC = Increment:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1			
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1			
SUB = Subtract:				
Reg./memory and register to either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w=01
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w=1	
SBB = Subtract with borrow				
Reg./memory and register to either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w=01
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w=1	

DEC = Decrement:	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
NEG=Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		
CMP = Compare:				
Register/memory and register	0 0 1 1 1 0 d w	mod reg r/m		
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w=01
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w=1	
AAS =ASCII adjust for subtract	0 0 1 1 1 1 1 1			
DAS =Decimal adjust for subtract	0 0 1 0 1 1 1 1			
MUL =Multiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
IMUL=Integer multiply (signed)	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
AAM=ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
DIV = Divide (unsigned)	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
IDIV =Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
AAD =ASCII adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
CBW = Convert byte to word	1 0 0 1 1 0 0 0			
CWD =Convert word to double word	1 0 0 1 1 0 0 1			
LOGIC				
NOT =Invert	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
SHL/SAL = Shift logical/arithmetic left	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
SHR = Shift logical right	1 1 0 1 0 0 v w	mod 1 0 1 r/m		
SAR = Shift arithmetic right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
ROL = Rotate left	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
ROR =Rotate right	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
RCL =Rotate through carry flag left	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
RCR = Rotate through carry flag right	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
AND = And:				
Reg./memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w=1
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w=1	
TEST = And function to flags. no result:				
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w=1
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w=1	
OR = Or:				
Reg./memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w=1
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w=1	
XOR = Exclusive or:				
Reg./memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w=1
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w=1	
STRING MANIPULATION				
REP = Repeat	1 1 1 1 0 0 1 z			
MOVS = Move byte/word	1 0 1 0 0 1 0 w			
CMPS = Compare byte/word	1 0 1 0 0 1 1 w			
SCAS = Scan byte/word	1 0 1 0 1 1 1 w			
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w			
STOS = Store byte/wd from AL/A	1 0 1 0 1 0 1 w			

*Mnemonics © Intel Corporation, 1978

TABLE 2 – INSTRUCTION SET SUMMARY (Continued)*

CONTROL TRANSFER

CALL = Call:	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high
Indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m	
Direct intersegment	1 0 0 1 1 0 1 0	offset-low	offset-high
		seg-low	seg-high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m	
JMP = Unconditional Jump:			
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high
Direct within segment-short	1 1 1 0 1 0 1 1	disp	
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m	
Direct intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high
		seg-low	seg-high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m	
RET = Return from CALL:			
Within segment	1 1 0 0 0 0 1 1		
Within seg, adding immed, to SP	1 1 0 0 0 0 1 0	data-low	data-high
Intersegment	1 1 0 0 1 0 1 1		
Intersegment, adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high
JE/JZ = Jump on equal/zero	0 1 1 1 0 1 0 0	disp	
JL/JNGE = Jump on less/not greater or equal	0 1 1 1 1 1 0 0	disp	
JLE/JNG = Jump on less or equal/not greater	0 1 1 1 1 1 1 0	disp	
JB/JMAE = Jump on below/not above or equal	0 1 1 1 0 0 1 0	disp	
JBE/JNA = Jump on below or equal/not above	0 1 1 1 0 1 1 0	disp	
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0	disp	
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp	
JS = Jump on sign	0 1 1 1 1 0 0 0	disp	
JNE/JNZ = Jump on not equal/not zero	0 1 1 1 0 1 0 1	disp	
JNL/JGE = Jump on not less/greater or equal	0 1 1 1 1 1 0 1	disp	
JNLE/JG = Jump on not less or equal/greater	0 1 1 1 1 1 1 1	disp	

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
JNB/JAE = Jump on not below/above or equal	0 1 1 1 0 0 1 1	disp
JNBE/JA = Jump on not below or equal/above	0 1 1 1 0 1 1 1	disp
JNP/JPO = Jump on not par/par odd	0 1 1 1 1 0 1 1	disp
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp
LOOPZ/LOOPE = Loop while zero/equal	1 1 1 0 0 0 0 1	disp
LOOPNZ/LOOPNE = Loop while not zero/equal	1 1 1 0 0 0 0 0	disp
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp
INT = Interrupt		
Type specified	1 1 0 0 1 1 0 1	type
Type 3	1 1 0 0 1 1 0 0	
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0	
IRET = Interrupt return	1 1 0 0 1 1 1 1	

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0	
CMC = Complement carry	1 1 1 1 0 1 0 1	
STC = Set carry	1 1 1 1 1 0 0 1	
CLD = Clear direction	1 1 1 1 1 1 0 0	
STD = Set direction	1 1 1 1 1 1 0 1	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	
STI = Set interrupt	1 1 1 1 1 0 1 1	
HLT = Halt	1 1 1 1 0 1 0 0	
WAIT = Wait	1 0 0 1 1 0 1 1	
ESC = Escape (to external device)	1 1 0 1 1 x x x	mod x x x r/m
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	
NOP = No operation	1 0 0 1 0 0 0 0	

Footnotes:

AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS = Data segment
ES = Extra segment
Above/below refers to unsigned value
Greater = more positive
Less = less positive (more negative) signed values
if d = 1 then "to" reg; if d = 0 then "from" reg
if w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low
if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP
DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low

if s:w = 01 then 16 bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to
form the 16-bit operand.
if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
x = don't care
z is used for string primitives for comparison with ZF FLAG
SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

REG is assigned according to the following table:

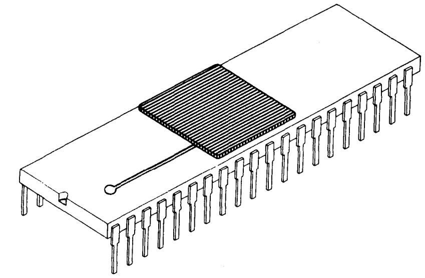
16-Bit [w = 1]	8-Bit [w = 0]	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register file as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = X:X:X:X:(OF):(DF):(IF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

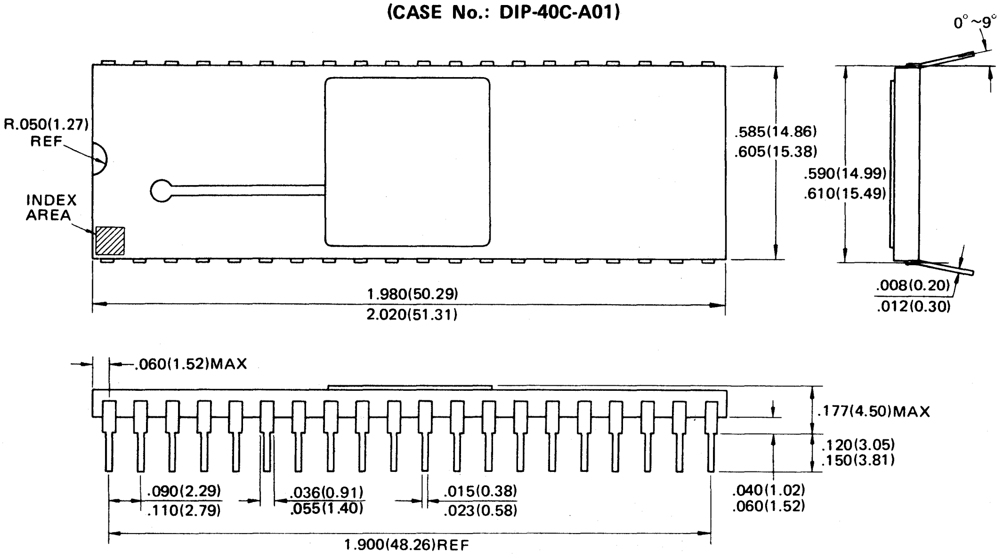
PACKAGE ILLUSTRATION

CERAMIC PACKAGE
(METAL SEAL)



PACKAGE DIMENSIONS (Suffix: -C)

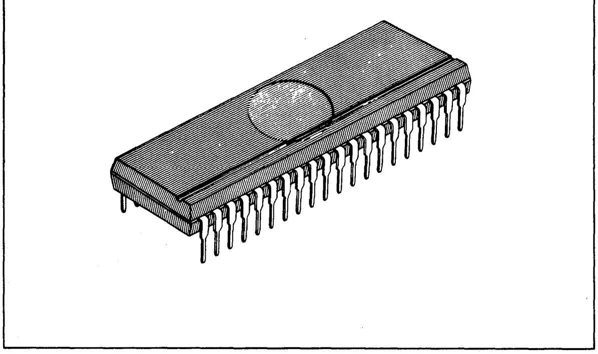
40-LEAD CERAMIC (METAL SEAL) DUAL IN-LINE PACKAGE
(CASE No.: DIP-40C-A01)



Dimensions in
inches (millimeters)

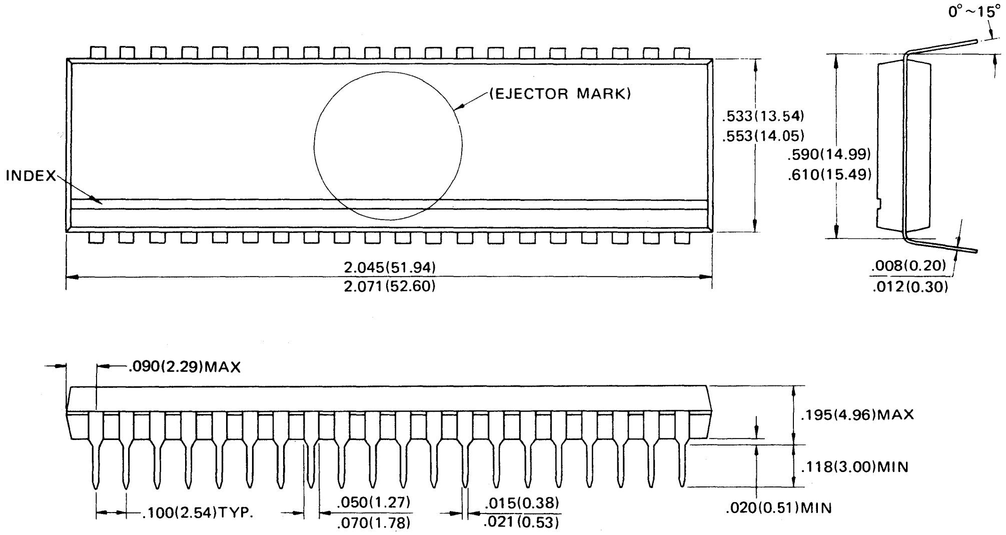
PACKAGE ILLUSTRATION

PLASTIC PACKAGE



PACKAGE DIMENSIONS (Suffix: -P)

40-LEAD PLASTIC DUAL IN-LINE PACKAGE
(CASE No.: DIP-40P-M01)



Dimensions in inches (millimeters)