

# Zilog Product Specification

T-52-33-25

## Z8010 Z8000<sup>®</sup> MMU Memory Management Unit

October 1988

**Features**

- Dynamic segment relocation makes software addresses independent of physical memory addresses.
- Sophisticated memory-management features include access validation that protects memory areas from unauthorized or unintentional access, and a write-warning indicator that predicts stack overflow.
- For use with both Z8001 and Z8003 CPU.
- 64 variable-sized segments from 256 to 65,536 bytes can be mapped into a total physical address space of 16M bytes; all 64 segments are randomly accessible.
- Multiple MMUs can support several translation tables for each Z8001 address space.
- MMU architecture supports multi-programming systems and virtual memory implementations.

**General Description**

The Z8010 Memory Management Unit (MMU) manages the large 8M byte addressing spaces of the Z8001 CPU. The MMU provides dynamic segment relocation as well as numerous memory protection features.

Dynamic segment relocation makes user software addresses independent of the physical memory addresses, thereby freeing the user from specifying where information is actually

located in the physical memory. It also provides a flexible, efficient method for supporting multi-programming systems. The MMU uses a translation table to transform the 23-bit logical address output from the Z8001 CPU into a 24-bit address for the physical memory. (Only logical memory addresses go to an MMU for translation; I/O addresses and data, in general, must by pass this component.)

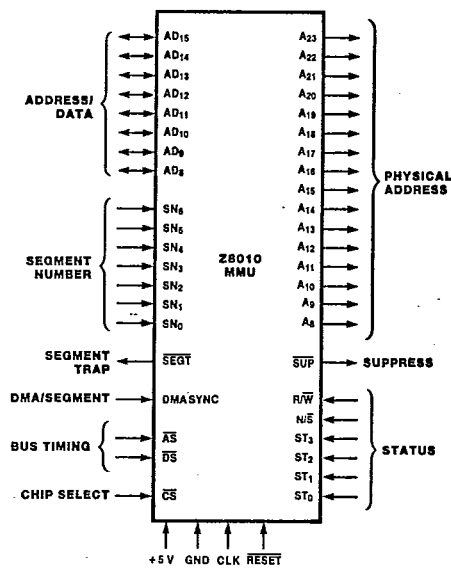


Figure 1. Pin Functions

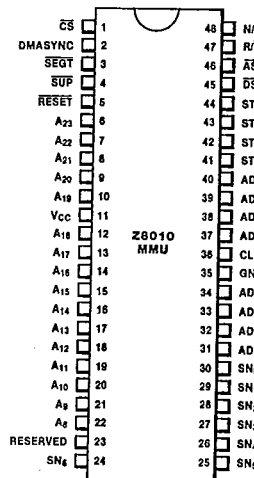


Figure 2. 48-pin Dual-In-Line Package (DIP), Pin Assignments

**General  
Description  
(Continued)**

Memory segments are variable in size from 256 bytes to 64K bytes, in increments of 256 bytes. Pairs of MMUs support the 128 segment numbers available for the various Z8001 CPU address spaces. Within an address space, any number of MMUs can be used to accommodate multiple translation tables for System and Normal operating modes, or to support more sophisticated memory-management systems.

MMU memory-protection features safeguard memory areas from unauthorized or unintended access by associating special access restrictions with each segment. A segment is assigned a number of attributes when its descriptor is entered into the MMU. When a memory reference is made, these attributes are checked against the status information supplied by the Z8001 CPU. If a mismatch oc-

curs, a trap is generated and the CPU is interrupted. The CPU can then check the status registers of the MMU to determine the cause.

Segments are protected by modes of permitted use, such as read only, system only, execute only and CPU-access only. Other segment management features include a write-warning zone useful for stack operations and status flags that record read or write accesses to each segment.

The MMU is controlled via 22 Special I/O instructions from the Z8000 CPU in System mode. With these instructions, system software can assign program segments to arbitrary memory locations, restrict the use of segments and monitor whether segments have been read or written.

General Description  
(Continued)

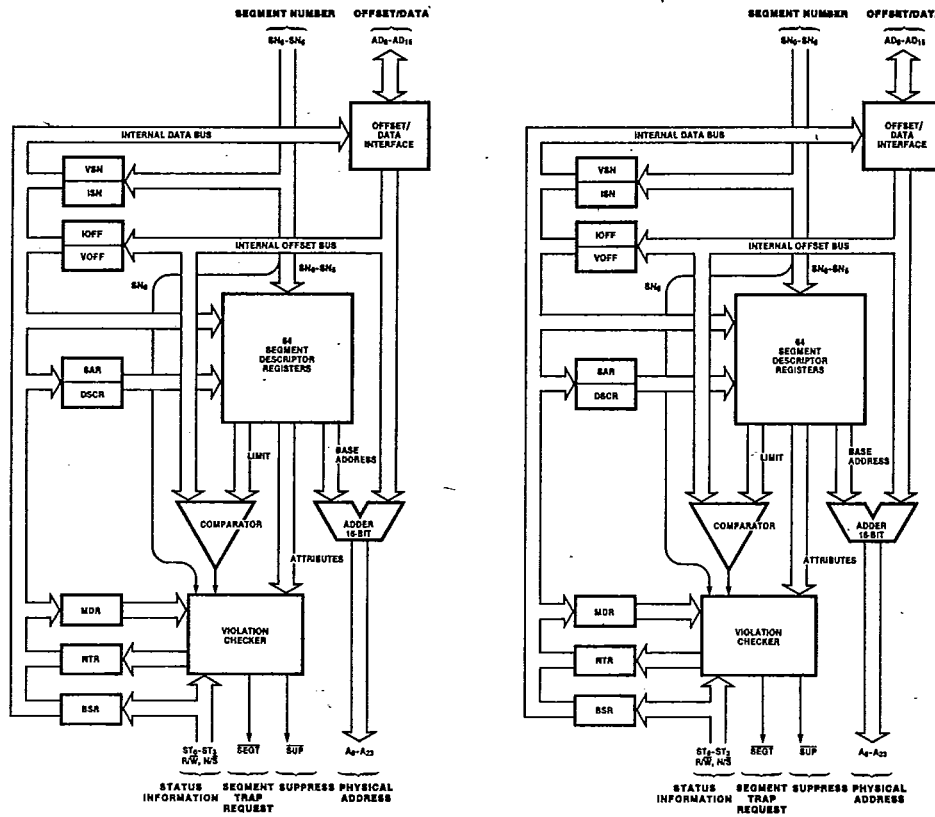


Figure 3. The shaded areas in these block diagrams illustrate the resources used in the two modes of MMU operation. In the Address Translation Mode shown on the left, addresses are translated automatically. In the Command Mode shown on the right, specific registers are accessed using Special I/O commands.

**Segmented Addressing**

A segmented addressing space—compared with linear addressing—is closer to the way a programmer uses memory because each procedure and data set can reside in its own segment.

The 8M byte Z8001 addressing spaces are divided into 128 relocatable segments of up to 64K bytes each. A 23-bit segmented address uses a 7-bit segment address to point to the segment, and a 16-bit offset to address any byte relative to the beginning of the segment. The two parts of the segmented address may be manipulated separately.

The MMU divides the physical memory into 256-byte blocks. Segments consist of physically contiguous blocks. Certain segments may be designated so that writes into the last block generate a warning trap. If such a segment is used as a stack, this warning can be used to increase the segment size and prevent a stack overflow error.

The addresses manipulated by the programmer, used by instructions and output by the Z8001 are called *logical addresses*. The MMU takes the logical addresses and transforms them into the *physical addresses* required for accessing the memory (Figure 4). This address transformation process is called *relocation*.

The relocation process is transparent to user software. A translation table in the MMU associates the 7-bit segment number with the base address of the physical memory segment. The 16-bit logical address offset is added to the physical base address to obtain the actual physical memory location. Because a base address always has a low byte equal to zero,

only the high-order 16 bits are stored in the MMU and used in the addition. Thus the low-order byte of the physical memory location is the same as the low-order byte of the logical address offset. This low-order byte therefore bypasses the MMU, thus reducing the number of pins required.

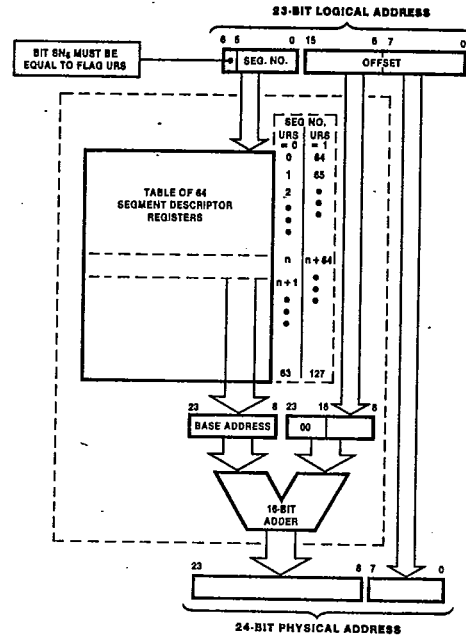


Figure 4. Logical-to-Physical Address Translation

**Memory Protection**

Each memory segment is assigned several attributes that are used to provide memory access protection. A memory request from the Z8001 CPU is accompanied by status information that indicates the attributes of the memory request. The MMU compares the memory request attributes with the segment attributes and generates a Trap Request whenever it detects an attribute violation. Trap Request informs the Z8001 CPU and the system control program of the violation so that appropriate action can be taken to recover. The MMU also generates the Suppress signal SUP in the event of an access violation. Suppress can be used by a memory system to inhibit stores into the memory and thus protect the contents of the memory from erroneous changes.

Five attributes can be associated with each segment. When an attempted access violates any one of the attributes associated with a segment, a Trap Request and a Suppress signal are generated by the MMU. These attributes are read only, execute only, system access only, inhibit CPU accesses and inhibit DMA accesses.

Segments are specified by a base address and a range of legal offsets to this base address. On each access to a segment, the offset is checked against this range to insure that the access falls within the allowed range. If an access that lies outside the segment is attempted, Trap Request and Suppress are generated.

Normally the legal range of offsets within a segment is from 0 to  $256N + 255$  bytes, where  $0 \leq N \leq 255$ . However, a segment may be specified so that legal offsets range from  $256N$  to  $65,535$  bytes, where  $0 \leq N \leq 255$ . The later type of segment is useful for stacks since the Z8000 stack manipulation instructions cause stacks to grow toward lower memory locations. Thus when a stack grows to the limit of its allocated segment, additional memory can be allocated on the correct end of the segment. As an aid in maintaining stacks, the MMU detects when a write is performed to the lowest allocated 256 bytes of these segments and generates a Trap Request. No Suppress signal is generated so the write is allowed to proceed. This write warning can then be used to indicate that more memory should be allocated to the segment.

**MMU Register Organization**

The MMU contains three types of registers: Segment Descriptor, Control and Status. A set of 64 Segment Descriptor Registers supplies the information needed to map logical memory addresses to physical memory locations. The segment number of a logical address determines which Segment Descriptor Register is used in address translation. Each Descriptor Register also contains the necessary information for checking that the segment location referenced is within the bounds of the segment and that the type of reference is permitted. It also indicates whether the segment has been read or written.

In addition to the Segment Descriptor Registers, the Z8010 MMU contains three 8-bit control registers for programming the device and six 8-bit status registers that record information in the event of an access violation.

**Segment Descriptor Registers.** Each of the 64 Descriptor Registers contains a 16-bit base address field, an 8-bit limit field and an 8-bit attribute field (Figure 5). The base address field is subdivided into high- and low-order bytes that are loaded one byte at a time when the descriptor is initialized. The limit field contains a value N that indicates N + 1 blocks of 256 bytes have been allocated to the segment.\*

The attribute field contains eight flags (Figure 6). Five are related to protecting the segment against certain types of access, one indicates the special structure of the segment, and two encode the types of accesses that have been made to the segment. A flag is set when its value is 1. The following brief descriptions indicate how these flags are used.

**Read-Only (RD).** When this flag is set, the segment is read only and is protected against any write access.

**System-Only (SYS).** When this flag is set, the segment can be accessed only in System mode, and is protected against any access in Normal mode.

**CPU-Inhibit (CPUI).** When this flag is set, the segment is not accessible to the currently executing process, and is protected against any memory access by the CPU. The segment is, however, accessible under DMA.

**Execute-Only (EXC).** When this flag is set, the segment can be accessed only during an instruction fetch or access by the relative addressing mode cycle, and thus is protected against any access during other cycles.

**DMA-Inhibit (DMAI).** When this flag is set, the segment can be accessed only by the CPU, and thus is protected against any access under DMA.

**Direction and Warning (DIRW).** When this flag is set, the segment memory locations are considered to be organized in descending order and each write to the segment is checked for access to the last 256-byte block. Such an access generates a trap to warn of potential segment overflow, but no Suppress signal is generated.

**Changed (CHG).** When this flag is set, the segment has been changed (written). This bit is set automatically during any write access to this segment if the write access does not cause any violation.

**Referenced (REF).** When this flag is set, the segment has been referenced (either read or written). This bit is set automatically during any access to the segment if the access does not cause a violation.

\*In the stack mode, segment size is 64K-256N.

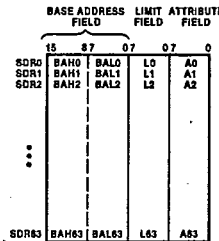


Figure 5. Segment Descriptor Registers

T-52-33-25

Figure 6. Attribute Field in Segment Descriptor Register

**Control Registers.** The three user-accessible 8-bit control registers in the MMU direct the functioning of the MMU (Figure 7). The Mode Register provides a sophisticated method for selectively enabling MMUs in multiple-MMU configurations. The Segment Address Register (SAR) selects a particular Segment Descriptor Register to be accessed during a control operation. The Descriptor Selection Counter Register points to a byte within the Segment Descriptor Register to be accessed during a control operation.

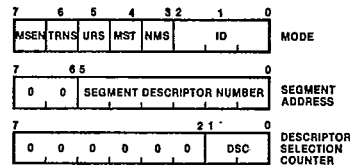


Figure 7. Control Registers

The Mode Register contains a 3-bit identification field (ID) that distinguishes among eight enabled MMUs in a multiple-MMU configuration. This field is used during the segment trap acknowledge sequence (refer to the section on Segment Trap and Acknowledge). In addition, the Mode Register contains five flags.

**Multiple Segment Table (MST).** This flag indicates whether multiple segment tables are present in the hardware configuration. When this flag is set, more than one table is present and the N/S line must be used to determine whether the MMU contains the appropriate table.

**Normal Mode Select (NMS).** This flag indicates whether the MMU is to translate addresses when the N/S line is High or Low. If the MST flag is set, the N/S line must match the NMS flag for the MMU to translate segment addresses, otherwise the MMU Address lines remain 3-stated.

**MMU Register Organization**  
(Continued)

**Upper Range Select (URS).** This flag is used to indicate whether the MMU contains the lower-numbered segment descriptors or the higher-numbered segment descriptors. The most significant bit of the segment number must match the URS flag for the MMU to translate segment addresses, otherwise the MMU Address lines remain 3-stated.

**Translate (TRNS).** This flag indicates whether the MMU is to translate logical program addresses to physical memory locations or is to pass the logical addresses unchanged to the memory and without protection checking. In the non-translation mode, the most significant byte of the output is the 7-bit segment number and the most significant bit is 0. When this flag is set, the MMU performs address translation and attribute checking.

**Master Enable (MSEN).** This flag enables or disables the MMU from performing its address translation and memory protection functions. When this flag is set, the MMU performs these tasks; when the flag is clear the Address lines of the MMU remain 3-stated.

The Segment Address Register (SAR) points to one of the 64 segment descriptors. Control commands to the MMU that access segment descriptors implicitly use this pointer to select one of the descriptors. This register has an auto-incrementing capability so that multiple descriptors can be accessed in a block read/write fashion.

The Descriptor Selection Counter Register holds a 2-bit counter that indicates which byte in the descriptor is being accessed during the reading or writing operation. A value of zero in this counter indicates the high-order byte of the base address field is to be accessed, one indicates the low-order byte of the base address, two indicates the limit field and three indicates the attribute field.

**Status Registers.** Six 8-bit registers contain information useful in recovering from memory access violations (Figure 8). The Violation Type Register describes the conditions that generated the trap. The Violation Segment Number and Violation Offset Registers record the most-significant 15 bits of the logical address that causes a trap. The Instruction Segment Number and Offset Registers record the most-significant 15 bits of the logical address of the last instruction fetched before the first accessing violation. These two registers can be used in conjunction with external circuitry that records the low-order offset byte. At the time of the addressing violation, the Bus Cycle Status Register records the bus cycle status (status code, read/write mode and normal/system mode).

The MMU generates a Trap Request for two general reasons: either it detects an access

violation, such as an attempt to write into a read-only segment, or it detects a warning condition, which is a write into the lowest 256 bytes of a segment with the DIRW flag set. When a violation or warning condition is detected, the MMU generates a Trap Request and automatically sets the appropriate flags. The eight flags in the Violation Type Register describe the cause of a trap.

**Read-Only Violation (RDV).** Set when the CPU attempts to access a read-only segment and the R/W line is Low.

**System Violation (SYSV).** Set when the CPU accesses a system-only segment and the  $\bar{N}/S$  line is High.

**CPU-Inhibit Violation (CPUIV).** Set when the CPU attempts to access a segment with the CPU-inhibit flag set.

**Execute-Only Violation (EXCV).** Set when the CPU attempts to access an execute-only segment in other than an instruction fetch or load relative instructions cycle.

**Segment Length Violation (SLV).** Set when an offset falls outside of the legal range of a segment.

**Primary Write Warning (PWW).** Set when an access is made to the lowest 256 bytes of a segment with the DIRW flag set.

**Secondary Write Warning (SWW).** Set when the CPU pushes data into the last 256 bytes of the system stack and EXCV, CPUIV, SLV, SYSV, RDV or PWW is set. Once this flag is set, subsequent write warnings for accessing the system stack do not generate a Segment Trap request.

**Fatal Condition (FATL).** Set when any other flag in the Violation Type Register is set and either a violation is detected or a write warning condition occurs in Normal mode. This flag is not set during a stack push in System mode that results in a warning condition. This flag indicates a memory access error has occurred in the trap processing routine. Once set, no Trap Request signals are generated on subsequent violations. However, Suppress signals are generated on this and subsequent CPU violations until the FATL flag has been reset.

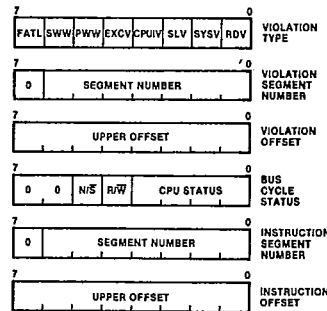


Figure 8. Status Registers

**Segment Trap and Acknowledge**

The Z8010 MMU generates a Segment Trap when it detects an access violation or a write warning condition. In the case of an access violation, the MMU also activates Suppress, which can be used to inhibit memory writes and to flag special data to be returned on a read access. Segment Trap remains Low until a Trap Acknowledge signal is received. If a CPU-generated violation occurs, Suppress is asserted for that cycle and all subsequent CPU instruction execution cycles until the end of the instruction. Intervening DMA cycles are not suppressed, however, unless they generate a violation. Violations detected during DMA cycles cause Suppress to be asserted during that cycle only—no Segment Trap Requests are ever generated during DMA cycles.

Segment traps to the Z8001/3 CPU are handled similarly to other types of interrupts. To service a segment trap, the CPU issues a segment trap acknowledge cycle. The acknowledge cycle is always preceded by an instruction fetch cycle that is ignored (the MMU has been designed so that this dummy cycle is ignored). During the acknowledge cycle all enabled MMUs use the Address/Data lines to indicate their status. An MMU that has generated a Segment Trap Request outputs a 1 on the A/D line associated with the number in its ID field; an MMU that has not generated a segment trap request outputs a 0 on its associated A/D line. A/D lines for which no MMU is associated remain 3-stated. During a

segment trap acknowledge cycle, an MMU uses A/D line  $8+i$  if its ID field is  $i$ .

Following the acknowledge cycle the CPU automatically pushes the Program Status onto the system stack and loads another Program Status from the Program Status Area. The Segment Trap line is reset during the segment trap acknowledge cycle. Suppress is not generated during the stack push. If the store creates a write warning condition, a Segment Trap Request is generated and is serviced at the end of the Program Status swap. The SWW flag is also set. Servicing this second Segment Trap Request also creates a write warning condition, but because the SWW flag is set, no Segment Trap Request is generated. If a violation rather than a write warning occurs during the Program Status swap, the FATL flag is set rather than the SWW flag. Subsequent violations cause Suppress to be asserted but not Segment Trap Request. Without the SWW and FATL flags, trap processing routines that generate memory violations would repeatedly be interrupted and called to process the trap they created.

The CPU routine to process a trap request should first check the FATL flag to determine if a fatal system error has occurred. If not, the SWW flag should be checked to determine if more memory is required for the system stack. Finally, the trap itself should be processed and the Violation Type Register reset.

**Virtual Memory**

Several features of the MMU can be used in conjunction with external circuitry to support virtual memory for the Z8001. Segment Trap Request can be used to signal the CPU in the event that a segment is not in primary memory. The CPU-Inhibit Flag can be used to indicate whether a segment is in the memory or in

secondary storage. The Changed and Altered Flags in the attribute field for each segment can aid in implementing efficient segment management policies. The Status Registers can be used in recovering from virtual memory access faults.

**Multiple MMUs**

MMU architecture directly supports two methods for multiple MMU configurations. The first approach extends single-MMU capability for handling 64 segments to a dual-MMU configuration that manages the 128 different segments the Z8001 can address. This scheme uses the URS flag in the Mode Register in connection with the high-order bit of the segment number (SN<sub>6</sub>).

The second approach uses several MMUs to implement multiple translation tables. Multiple tables can be used to reduce the time required to switch tasks by assigning separate tables to each task. Multiple translation tables for multi-

task environments can use the Master Enable Flag to enable the appropriate MMUs through software. Multiple translation tables may also be used to extend the physical memory size beyond 16 megabytes by separating system from normal memory and/or program from data memory. The MST and NMS flags in the Mode Register can be used in conjunction with the N/S line to select the MMU that contains the appropriate table. Special external circuitry that monitors the CPU Status lines can manipulate the MMU N/S line to perform this selection.

**DMA  
Operation**

Direct memory access operations may occur between Z8001 instruction cycles and can be handled through the MMU. The MMU permits DMA in either the System or Normal mode of operation. For each memory access, the segment attributes are checked and if a violation is detected, Suppress is activated. Unlike a CPU violation that automatically causes Suppress signals to be generated on subsequent memory accesses until the next instruction, DMA violations generate a Suppress only on a per memory access basis.

The DMA device should note the Suppress signal and record sufficient information to enable the system to recover from the access violation. No Segment Trap Request is ever generated during DMA, hence warning conditions are not signaled. Trap Requests are not issued because the CPU cannot acknowledge such a request.

At the start of a DMA cycle, DMASync must go Low for at least two clock cycles, indicating to the MMU the beginning of a DMA cycle. A Low DMASync inhibits the MMU from using an indeterminate segment number on lines SN<sub>0</sub>-SN<sub>6</sub>. When the DMA logical memory address is valid, the DMASync line must be High before a rising edge of Clock and the MMU then performs its address translation and access protection functions. Upon the release of the bus at the termination of the DMA cycle the DMASync line must again be High. After two clock cycles of DMASync High, the MMU assumes that the CPU has control of the bus and that subsequent memory references are CPU accesses. The first instruction fetch occurs at least two cycles after the CPU regains control of the bus. During CPU cycles, DMASync should always be High.

**MMU  
Commands**

The various registers in the MMU can be read and written using Z8001 CPU special I/O commands. These commands have machine cycles that cause the Status lines to indicate an SIO operation is in progress. During these machine cycles the MMU enters command mode. In this mode, the rising edge of the Address Strobe indicates a command is present on the AD<sub>8</sub>-AD<sub>15</sub>. If Chip Select is asserted and if this command indicates that data is to be written into one of the MMU registers, the data is read from AD<sub>8</sub>-AD<sub>15</sub> while Data Strobe is Low. If the command indicates that data is to be read from one of the MMU registers, the data is placed on AD<sub>8</sub>-AD<sub>15</sub> while Data Strobe is Low.

There are ten commands that read or write various fields in the Segment Descriptor Register. The status of the Read/Write line indicates whether the command is a read or a write.

The auto-incrementing feature of the Segment Address Register (SAR) can be used to block load segment descriptors using the repeat forms of the Special I/O instructions. The SAR is autoincremented at the end of the field. In accessing the base field, first the high-order byte is selected and then the low-order byte. The command accessing the entire Descriptor Register references the fields in the order of base address, limit and attribute.

Opcode (Hex)	Instruction
08	Read/Write Base Field
09	Read/Write Limit Field
0A	Read/Write Attribute Field
0B	Read/Write Descriptor (all fields)
0C	Read/Write Base Field; Increment SAR
0D	Read/Write Limit Field; Increment SAR
0E	Read/Write Attribute Field; Increment SAR
0F	Read/Write Descriptor; Increment SAR
15	Set All CPU-Inhibit Attribute Flags
16	Set All DMA-Inhibit Attribute Flags

Three commands are used to read and write the control registers.

Opcode (Hex)	Instruction
00	Read/Write Mode Register
01	Read/Write Segment Address Register
20	Read/Write Descriptor Selector Counter Register

The Status Registers are read-only registers, although the Violation Type Register (VTR) can be reset. Nine instructions access these registers.

Opcode (Hex)	Instruction
02	Read Violation Type Register
03	Read Violation Segment Number Register
04	Read Violation Offset (High-byte) Register
05	Read Bus Status Register
06	Read Instruction Segment Number Register
07	Read Instruction Offset (High-byte) Register
11	Reset Violation Type Register
13	Reset SWW Flag in VTR
14	Reset FATL Flag in VTR

**MMU  
Timing**

The Z8010 translates addresses and checks for access violations by stepping through sequences of basic clock cycles corresponding to the cycle structure of the Z8001 CPU. The following timing diagrams show the relative timing relationships of MMU signals during the basic operations of memory read/write and MMU control commands. For exact timing information, refer to the composite timing diagram.

**Memory Read and Write.** Memory read and instruction fetch cycles are identical, except for the status information on the ST<sub>0</sub>-ST<sub>3</sub> inputs. During a memory read cycle (Figure 9) the 7-bit segment number is input on SN<sub>0</sub>-SN<sub>6</sub> one clock period earlier than the address offset; a High on DMASYNC during T<sub>3</sub> indicates that the segment offset data is valid. The most significant eight bits of the address offset are placed on the AD<sub>0</sub>-AD<sub>15</sub> inputs early in the

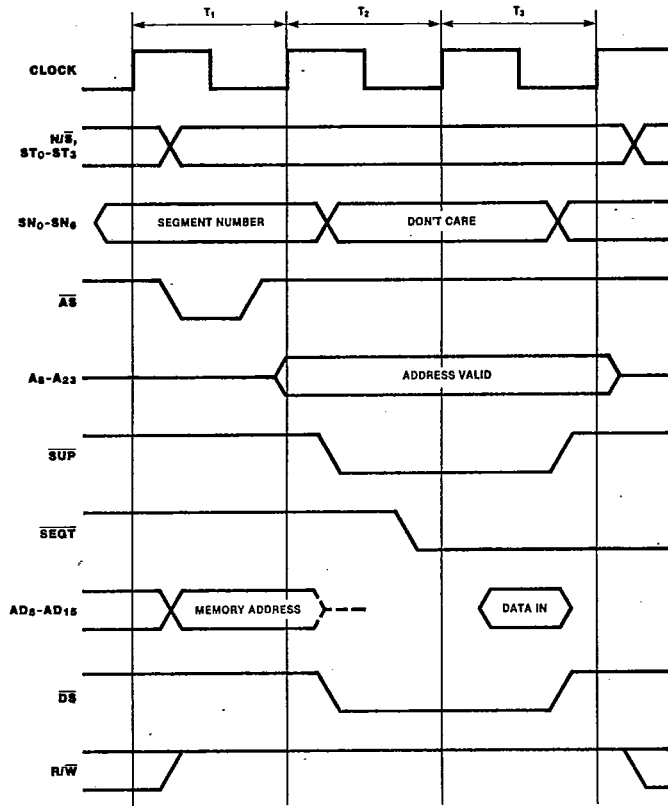


Figure 9. Memory Read Timing

**MMU  
Timing**  
(Continued)

first clock period. Valid address offset data is indicated by the rising edge of Address Strobe. Status and mode information become valid early in the memory access cycle and remain stable throughout. The most significant 16-bits of the address (physical memory location) remain valid until the end of T<sub>3</sub>. Segment Trap Request and Suppress are asserted in T<sub>2</sub>.

Segment Trap Request remains Low until Segment Trap Acknowledge is received. Suppress is asserted during the current machine cycle and terminates during T<sub>3</sub>. Suppress is repeatedly asserted during CPU instruction execution cycles until the current instruction has terminated.

T-52-33-25

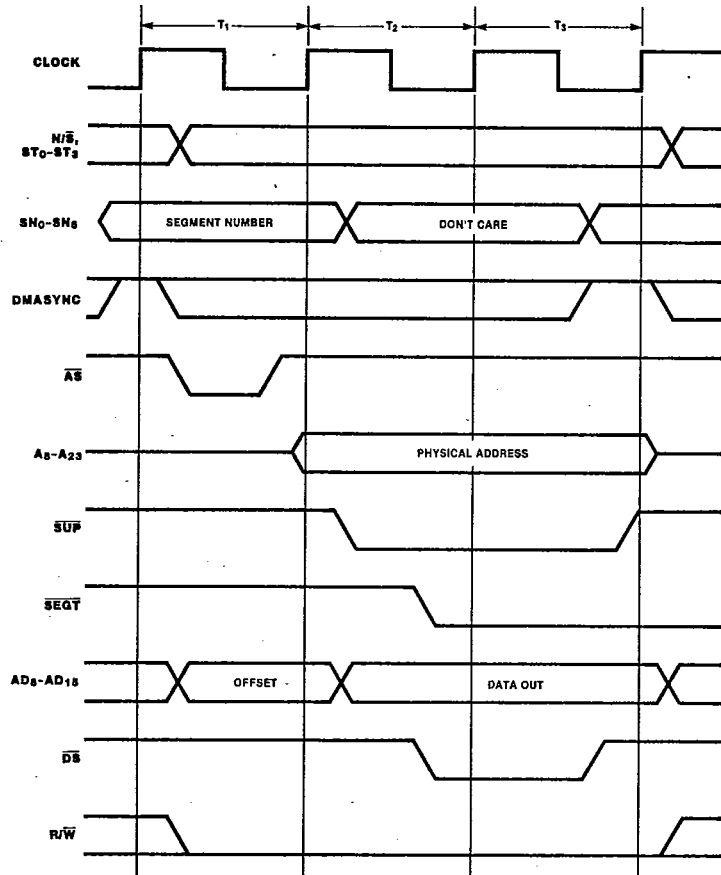


Figure 10. Memory Write Timing

**MMU  
Timing  
(Continued)**

**MMU Command Cycle.** During the command cycle of the MMU (Figure 11), commands are placed on the Address/Data lines during T<sub>1</sub>. The Status lines indicate that a Special I/O instruction is in progress, and the Chip Select line enables the appropriate MMU for that command. Data to be written to a register in the MMU must be valid on the Address/Data lines late in T<sub>2</sub>. Data read from the MMU is

placed on the Address/Data lines late in the T<sub>WA</sub> cycle.

**Input/Output and Refresh.** Input/Output and Refresh operations are indicated by the status lines ST<sub>0</sub>-ST<sub>3</sub>. During these operations, the MMU refrains from any address translation or protection checking. The address lines A<sub>8</sub>-A<sub>23</sub> remain 3-stated.

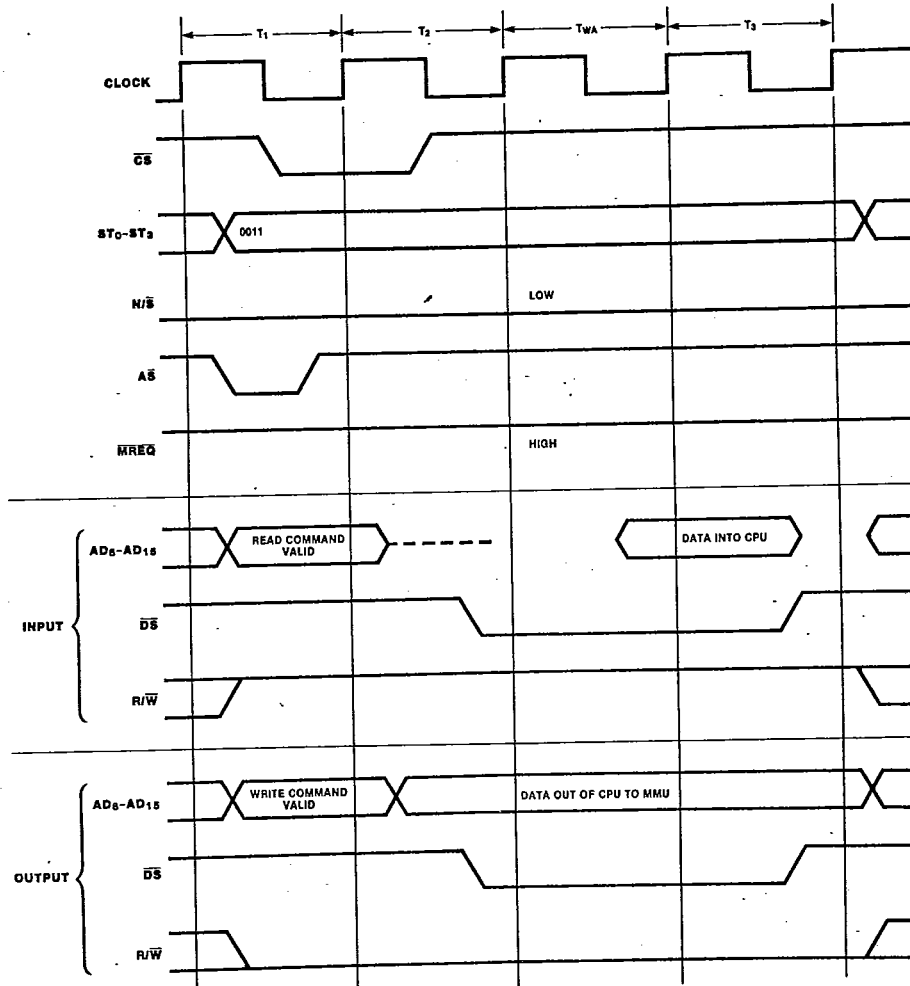


Figure 11. I/O Command Timing

**MMU Timing**  
(Continued)

**Reset.** The MMU can be reset by either hardware or software mechanisms. A hardware reset occurs on the falling edge of the Reset signal; a software reset is performed by a Z8000 Special I/O command. A hardware reset clears the Mode Register, Violation Type Register and Descriptor Selection Counter. If the Chip Select line is Low, the Master Enable Flag in the Mode Register is set to 1. All other registers are undefined. After reset, the AD<sub>8</sub>-AD<sub>15</sub> and A<sub>8</sub>-A<sub>23</sub> lines are 3-stated. The SUP and SEGT open-drain outputs are not driven. If the Master Enable flag is not set during reset, the MMU does not respond to subsequent addresses on its A/D lines. To enable an MMU after a hardware reset, an MMU command must be used in conjunction with the Chip Select line.

A software reset occurs when the Reset Violation Type Register command is issued. This command clears the Violation Type Register and returns the MMU to its initial state (as if no violations or warnings had occurred). Note that the hardware and software resets have different effects.

**Segment Trap and Acknowledge.** The Z8010 MMU generates a segment trap whenever it detects an access violation or a write into the lowest block of a segment with the DIRW flag

set. In the case of an access violation, the MMU also activates Suppress. This Suppress signal can be used to inhibit memory writes. The Segment Trap remains Low until a Trap Acknowledge signal is received. If a violation occurs, Suppress is asserted for that cycle and all subsequent CPU cycles until the end of the instruction; intervening DMA cycles are not suppressed, however, unless they generate a violation. Violations detected during DMA cycles cause Suppress to be asserted during that cycle only, but no Trap Request is generated.

When the MMU issues a Segment Trap Request it awaits a Segment Trap Acknowledge. Subsequent violations occurring before the Trap Acknowledge is received are still detected and handled appropriately. During the Segment Trap Acknowledge cycle, the MMU drives one of its Address/Data lines High; the particular line selected is a function of the identification field of the mode register. After the Segment Trap has been acknowledged by the Z8001 CPU, the Violation Status Register should be read via the Special I/O commands in order to determine the cause of the trap. The Trap Type Register should also be reset so that subsequent traps will be recorded correctly.

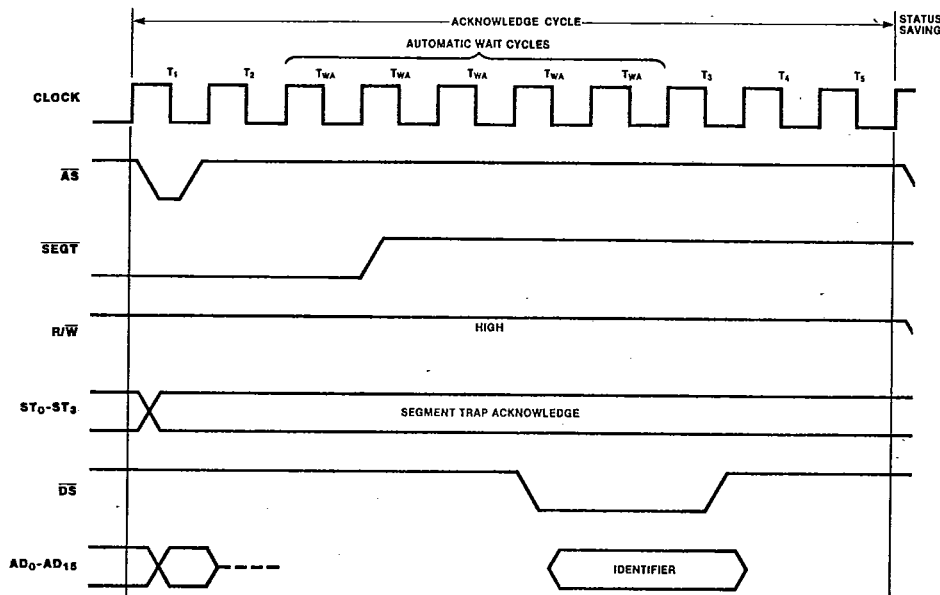


Figure 12. Segment Trap and Acknowledge Timing

**Pin Description**

**A<sub>8</sub>-A<sub>23</sub>.** *Address Bus* (outputs, active High, 3-state). These address lines are the 16 most-significant bits of the physical memory location.

**AD<sub>8</sub>-AD<sub>15</sub>.** *Address/Data Bus* (inputs/outputs, active High, 3-state). These multiplexed address and data lines are used both for commands and for logical addresses intended for translation.

**AS.** *Address Strobe* (input, active Low). The rising edge of AS indicates that AD<sub>0</sub>-AD<sub>15</sub>, ST<sub>0</sub>-ST<sub>3</sub>, R/W and N/S are valid.

**CLK.** *System Clock* (input). CLK is the 5 V single-phase time-base input used for both the CPU and MMU.

**CS.** *Chip Select* (input, active Low). This line selects an MMU for a control command.

**DMASYNC.** *DMA/Segment Number Synchronization Strobe* (input, active High). A Low on this line indicates that the segment number lines are 3-state; a High indicates that the segment number is valid. It must always be High during CPU cycles. If a DMA device does not use the MMU for address translation, the BUSACK signal from the CPU may be used as an input to DMASYNC.

**DS.** *Data Strobe* (input, active Low). This line provides timing for the data transfer between the MMU and the Z8001 CPU.

**N/S.** *Normal/System Mode* (input, Low = System Mode). N/S indicates the Z8001 CPU or Z8016 DMA is in the Normal or System Mode. The signal can also be used to switch between MMUs during different phases of an instruction.

**Reserved.** Do not connect.

**RESET.** *Reset* (input, active Low). A Low on this line resets the MMU.

**R/W.** *Read/Write* (input, Low = write). R/W indicates the Z8001 CPU or Z8016 DTC is reading from or writing to memory or the MMU.

**SEGT.** *Segment Trap Request* (output, active Low, open drain). The MMU interrupts the Z8001 CPU with a Low on this line when the MMU detects an access violation or write warning.

**SN<sub>0</sub>-SN<sub>6</sub>.** *Segment Number* (inputs, active High). The SN<sub>0</sub>-SN<sub>5</sub> lines are used to address one of 64 segments in the MMU; SN<sub>6</sub> is used to selectively enable the MMU.

**ST<sub>0</sub>-ST<sub>3</sub>.** *Status* (inputs, active High). These lines specify the Z8001 CPU status.

ST <sub>3</sub> -ST <sub>0</sub>	Definition
0 0 0 0	Internal operation
0 0 0 1	Memory refresh
0 0 1 0	I/O reference
0 0 1 1	Special I/O reference (e.g., to an MMU)
0 1 0 0	Segment trap acknowledge
0 1 0 1	Nonmaskable interrupt acknowledge
0 1 1 0	Nonvectored interrupt acknowledge
0 1 1 1	Vectored interrupt acknowledge
1 0 0 0	Data memory request
1 0 0 1	Stack memory request
1 0 1 0	Data memory request (EPU)
1 0 1 1	Stack memory request (EPU)
1 1 0 0	Instruction space access
1 1 0 1	Instruction fetch, first word
1 1 1 0	Extension processor transfer

**SUP.** *Suppress* (output, active Low, open drain). This signal is asserted during the current bus cycle when any access violation except write warning occurs.

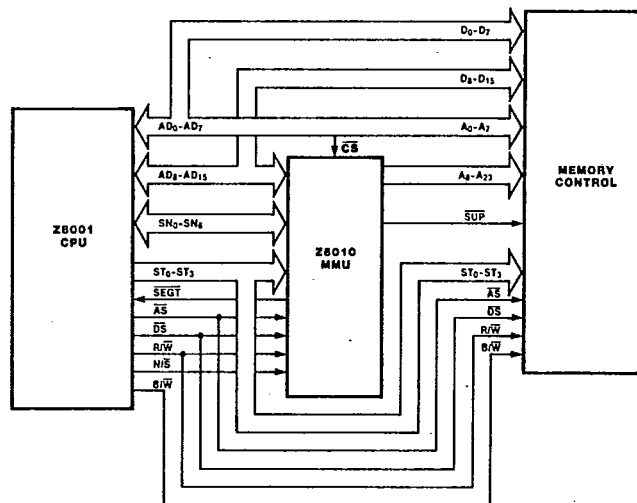
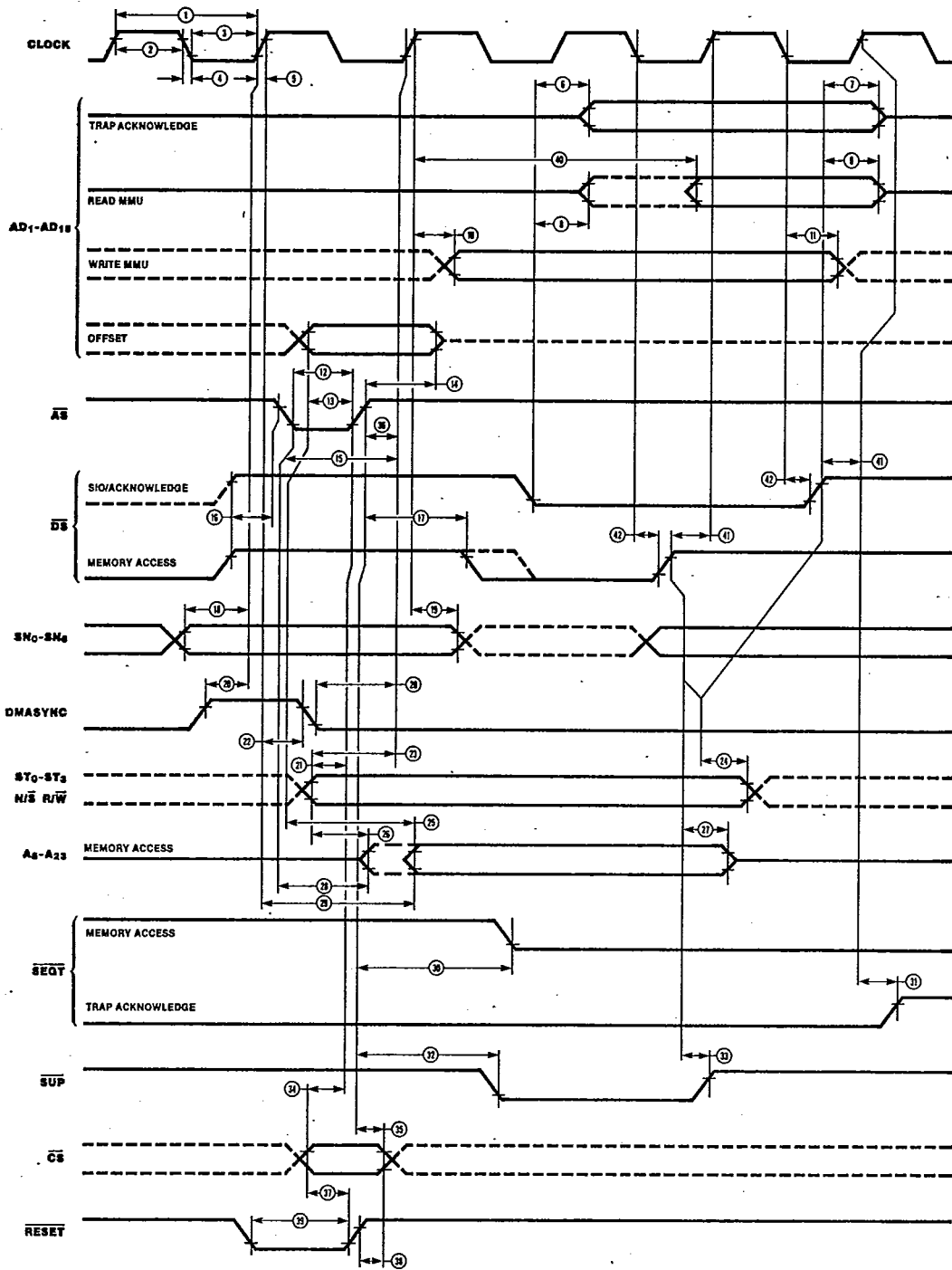


Figure 13. The MMU in a Z8001 System



AC Characteristics

No.	Symbol	Parameter	Z8010 6 MHz		Z8010 10 MHz		Notes*
			Min	Max	Min	Max	
1	TcC	Clock Cycle Time	165		100		
2	TwCh	Clock Width (High)	70		40		
3	TwCl	Clock Width (Low)	70		40		
4	TfC	Clock Fall Time		10		10	
5	TrC	Clock Rise Time		15		10	
6	TdDSA(RDv)	DS ↑ (Acknowledge) to Read Data Valid Delay		80		60	1
7	TdDSA(RDf)	DS ↑ (Acknowledge) to Read Data Float Delay		60		45	1
8	TdDSR(RDv)	DS ↑ (Read) to AD Output Driven Delay		80		60	1
9	TdDSR(RDf)	DS ↑ (Read) to Read Data Float Delay	60	60		45	1
10	TdC(WDv)	CLK ↑ to Write Data Valid Delay		80		50	
11	ThC(WDn)	CLK ↑ to Write Data Not Valid Hold Time	20		10		
12	TwAS	Address Strobe Width	50		30		
13	TsOFF(AS)	Offset Valid to AS ↑ Setup Time	35		20		
14	ThAS(OFFn)	AS ↑ to Offset Not Valid Hold Time	40		20		
15	TdAS(C)	AS ↑ to CLK ↑ Delay	90		50		
16	TdDS(AS)	DS ↑ to AS ↑ Delay	30		15		
17	TdAS(DS)	AS ↑ to DS ↑ Delay	40		30		
18	TsSN(C)	SN Data Valid to CLK ↑ Setup Time	40		30		
19	ThC(SNn)	CLK ↑ to SN Data not Valid Hold Time	0		0		
20	TdDMAS(C)	DMASYNC Valid to CLK ↑ Delay	80		60		
21	TdSTNR(AS)	Status (ST <sub>0</sub> -ST <sub>3</sub> , N/S, R/W) Valid to AS ↑ Delay	30		10		
22	TdC(DMA)	CLK ↑ to DMASYNC ↑ Delay	15		10		
23	TdST(C)	Status (ST <sub>0</sub> -ST <sub>3</sub> ) Valid to CLK ↑ Delay	60		30		
24	TdDS(STn)	DS ↑ to Status Not Valid Delay	0		0		
25	TdOFF(Av)	Offset Valid to Address Output Valid Delay		90		60	1
26	TdST(Ad)	Status Valid to Address Output Driven Delay		75		45	1
27	TdDS(Af)	DS ↑ to Address Output Float Delay		130		100	1
28	TdAS(Ad)	AS ↑ to Address Output Driven Delay		70		40	1
29	TdC(Av)	CLK ↑ to Address Output Valid Delay		155		100	1
30	TdAS(SEGT)	AS ↑ to SEGT ↑ Delay		100		60	1,2
31	TdC(SEGT)	CLK ↑ to SEGT ↑ Delay		200		100	1,2
32	TdAS(SUP)	AS ↑ to SUP ↑ Delay		90		55	1,2
33	TdDS(SUP)	DS ↑ to SUP ↑ Delay		100		60	1,2
34	TsCS(AS)	Chip Select Input Valid to AS ↑ Setup Time	10		10		
35	ThAS(CSn)	AS ↑ to Chip Select Input Not Valid Hold Time	40		20		
36	TdAS(C)	AS ↑ to CLK ↑ Delay	10		10		
37	TsCS(RST)	Chip Select Input Valid to RESET ↑ Setup Time	100		60		
38	ThRST(CSn)	RESET ↑ to Chip Select Input Not Valid Hold Time	0		0		
39	TwRST	RESET Width (Low)	2TcC		2TcC		
40	TdC(RDv)	CLK ↑ to Read Data Valid Delay	460	300		190	
41	TdDS(C)	DS ↑ to CLK ↑ Delay	20		10		
42	TdC(DS)	CLK ↑ to DS ↑ Delay	0		0		

NOTES:

- 1. 50 pf Load.
- 2. 2.2K Pull-up.
- \* Units in nanoseconds (ns).

Timing measurements are made at the following voltages:

	High	Low
Clock	4.0 V	0.8 V
Output	2.0 V	0.8 V
Input	2.0 V	0.8 V
Float	ΔV	±0.5 V

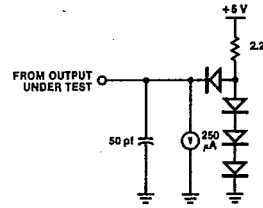
**Absolute Maximum Ratings**  
 Voltages on all pins with respect to GND.....-0.3V to +7.0V  
 Operating Ambient Temperature.....See Ordering Information  
 Storage Temperature.....-65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

T-52-33-25

**Standard Test Conditions**  
 The DC characteristics and capacitance section below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin.  
 Standard conditions are as follows:

- $+4.75\text{ V} \leq V_{CC} \leq +5.25\text{ V}$
- $\text{GND} = 0\text{ V}$
- $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$

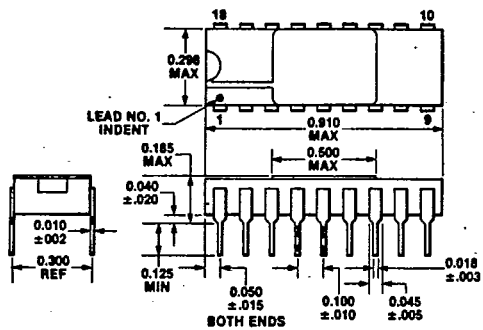


DC Characteristics	Symbol	Parameter	Min	Max	Unit	Condition
	V <sub>CH</sub>	Clock Input High Voltage	V <sub>CC</sub> -0.4	V <sub>CC</sub> +0.3	V	Driven by External Clock Generator
	V <sub>CL</sub>	Clock Input Low Voltage	-0.3	0.45	V	Driven by External Clock Generator
	V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub> +0.3	V	
	V <sub>IL</sub>	Input Low Voltage	-0.3	0.8	V	
	V <sub>OH</sub>	Output High Voltage	2.4		V	I <sub>OH</sub> = -250 μA
	V <sub>OL</sub>	Output Low Voltage		0.4	V	I <sub>OL</sub> = +2.0 mA
	I <sub>IL</sub>	Input Leakage		±10	μA	0.4 ≤ V <sub>IN</sub> ≤ +2.4 V
	I <sub>OL</sub>	Output Leakage		±10	μA	0.4 ≤ V <sub>IN</sub> ≤ +2.4 V
	I <sub>CC</sub>	V <sub>CC</sub> Supply Current		300	mA	

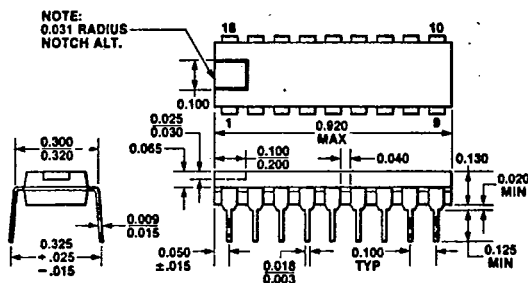
NOTE: The on-chip back-bias voltage generator takes approximately 20 ms to pump the back-bias voltage to -2.5 V after the power has been turned on. The performance of the Z8010 Z-MMU is not guaranteed during this period.

PACKAGE INFORMATION

T-90-20

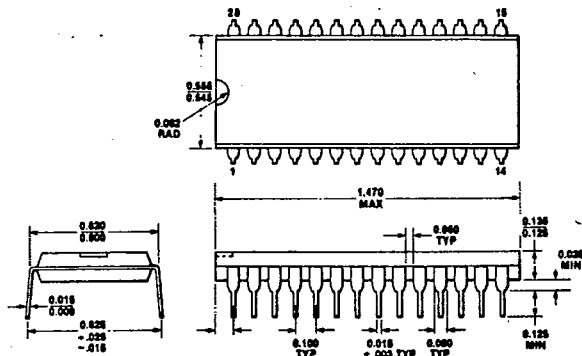


18-Pin Ceramic Package



18-Pin Plastic Package

NOTE: Package dimensions are given in inches. To convert to millimeters, multiply by 25.4

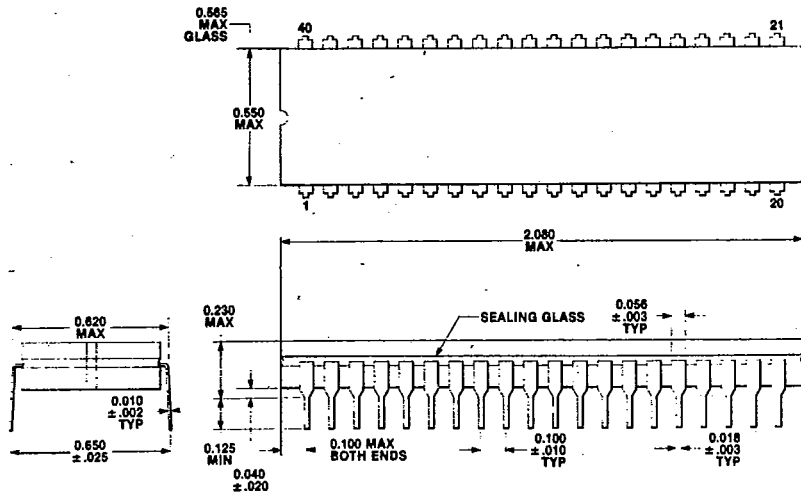


24-Pin Plastic Package

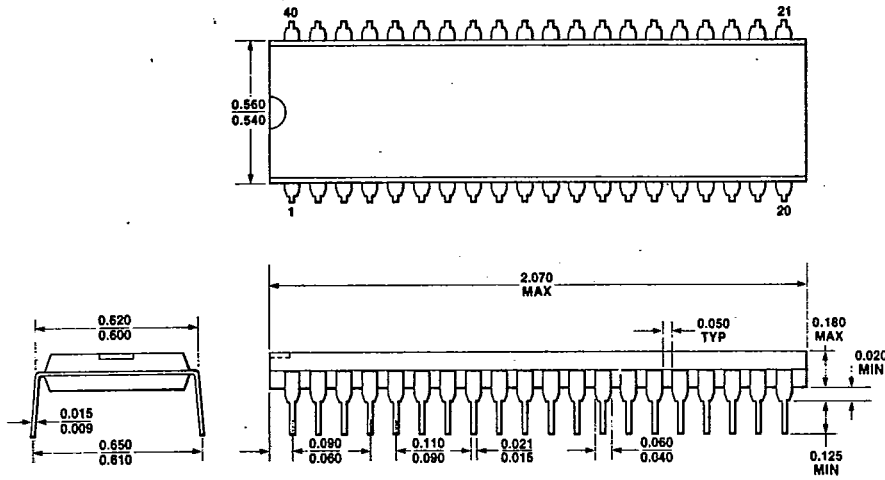
NOTE: Package dimensions are given in inches. To convert to millimeters, multiply by 25.4.

T-90-20

PACKAGE INFORMATION (Continued)



40-Pin Dual-in-Line Package (DIP),  
Cerdip

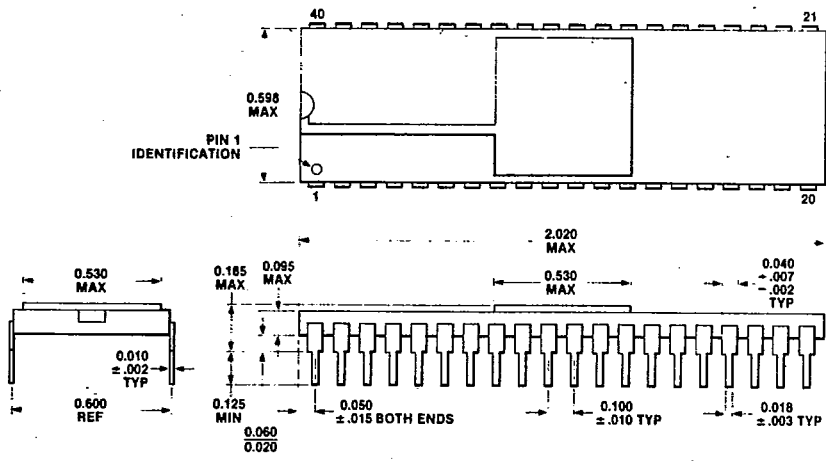


40-Pin Dual-in-Line Package (DIP),  
Plastic

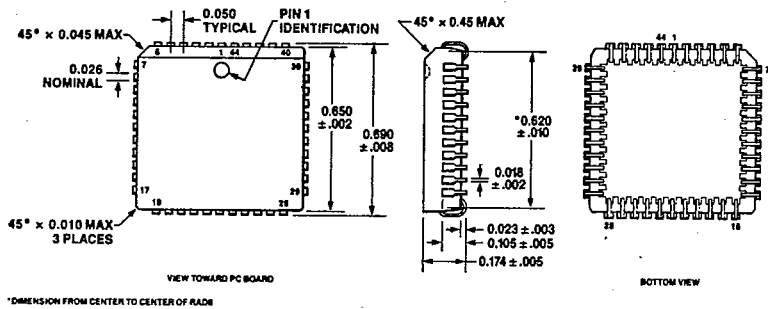
NOTE: Package dimensions are given in inches. To convert to millimeters, multiply by 25.4.

T-90-20

PACKAGE INFORMATION (Continued)



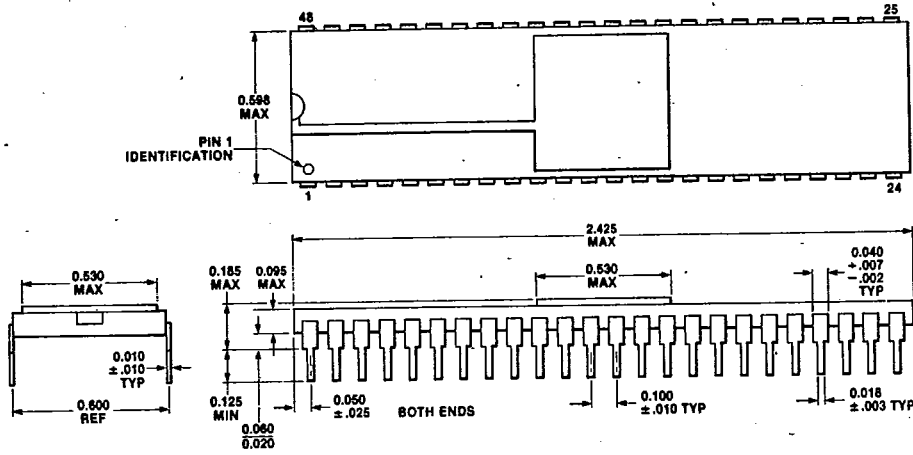
40-Pin Dual-In-Line Package (DIP),  
Ceramic



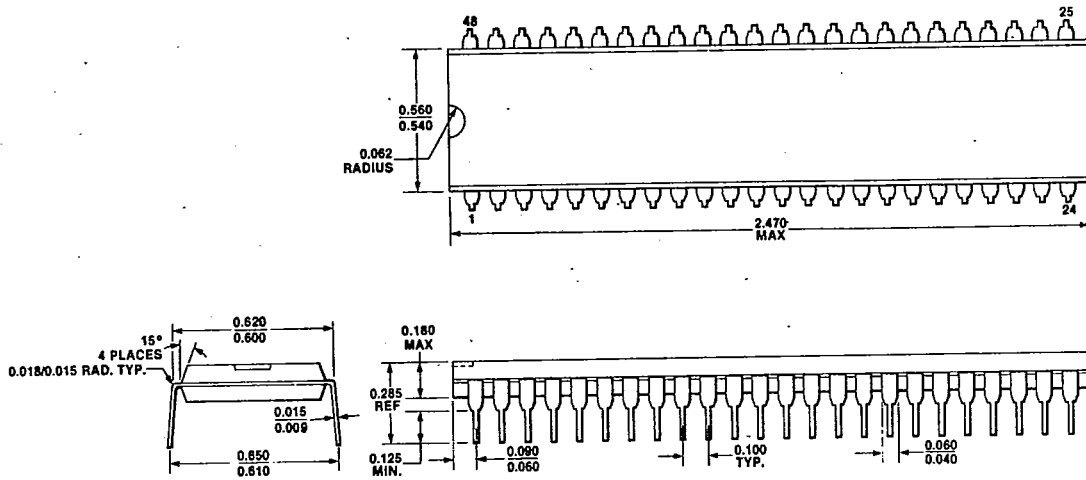
44-Pin Plastic Chip Carrier (PCC)

T-90-20

PACKAGE INFORMATION (Continued)



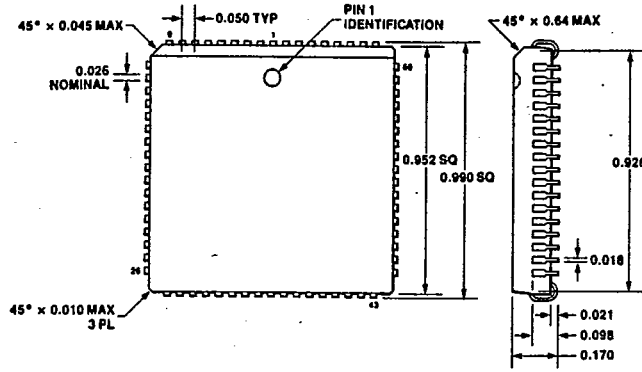
48-Pin Dual-in-Line Package (DIP),  
Ceramic



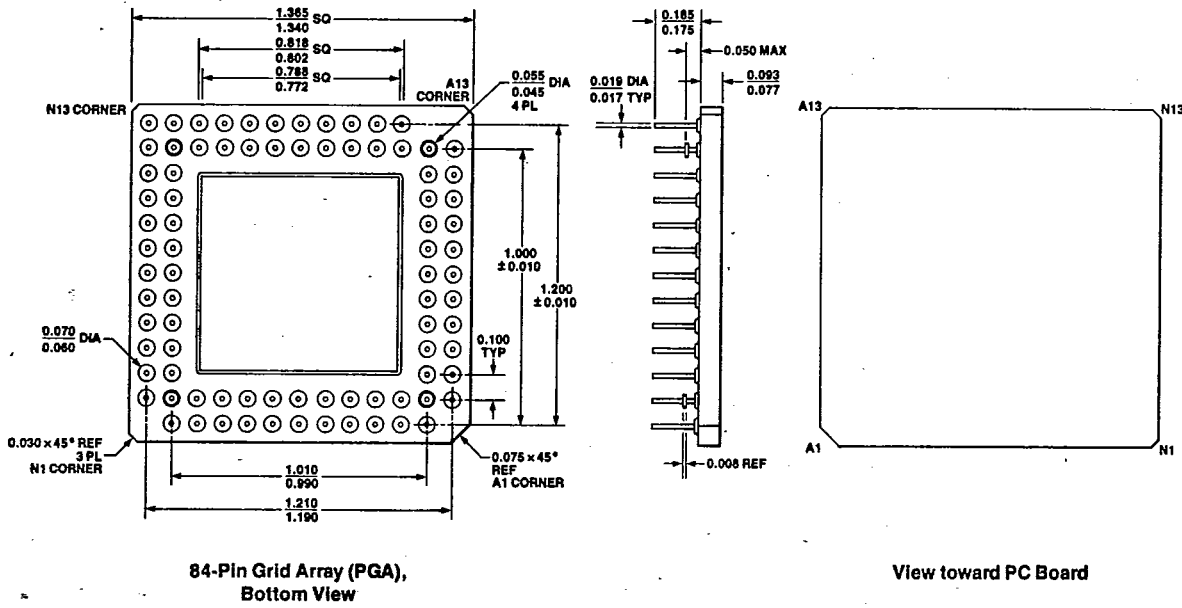
48-Pin Dual-in-Line Package (DIP),  
Plastic

PACKAGE INFORMATION (Continued)

T-90-20



68-Pin Plastic Chip Carrier (PCC)



84-Pin Grid Array (PGA),  
Bottom View

View toward PC Board

NOTE: Package dimensions are given in inches. To convert to millimeters, multiply by 25.4.