AmZ8010
Memory Management Unit

ADVANCED MICRO DEVICES

Product Specification
October 1980

DISTINCTIVE CHARACTERISTICS

- **Complete Memory Management Function**
  Single LSI chip provides full memory management for AmZ8001 CPU.
- **Translates Logical Addresses to Physical Addresses**
  Segments from AmZ8001 CPU can be placed anywhere in memory. Dynamic relocation is possible. Address translation occurs in real-time with no wait state required.
- **Defines and Protects Segment Sizes**
  Each segment can be assigned from 256 to 64K bytes of memory. References outside the limit are trapped for CPU error processing.
- **Six Software Security Checks**
  Each memory access is checked against security flags. Illegal accesses are inhibited and trap to the CPU for error processing.
- **Usage History Monitor Built In**
  MMU automatically records segment usage to aid in swapping segments between memory and disk.

GENERAL DESCRIPTION

The AmZ8010 Memory Management Unit (MMU) is a high-performance LSI product that adds sophisticated address translation and memory protection capabilities to AmZ8001 CPU systems. Addresses output by the CPU consist of a 7-bit segment number and a 16-bit offset. The MMU uses the segment number to index an address translation table to obtain a segment base address. The offset is added to the segment base to form the physical address. A separate table allows the user to individually program each segment size from 256 to 64K bytes.

The MMU also contains a table of access attributes that are individually programmable for each segment. Attributes provided are read-only, system-mode-only, DMA-only, execute-only, and CPU-only (exclude DMA). A trap is issued to the CPU and writes to memory are suppressed if an access is attempted that is prohibited by the attributes or which falls outside of the programmed segment size.

MMU BLOCK INTERFACE

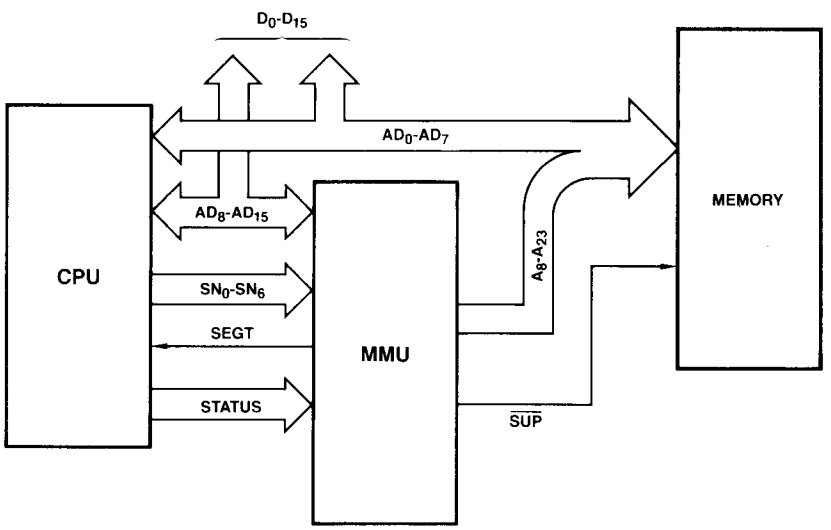


Figure 1.

OTHER LITERATURE:

For more information on this product, refer to:

- *AmZ8010 Memory Management Unit Product Description* – AMPUB-121
- *AmZ8010 MMU Technical Manual*

TABLE OF CONTENTS

GENERAL BLOCK DIAGRAM

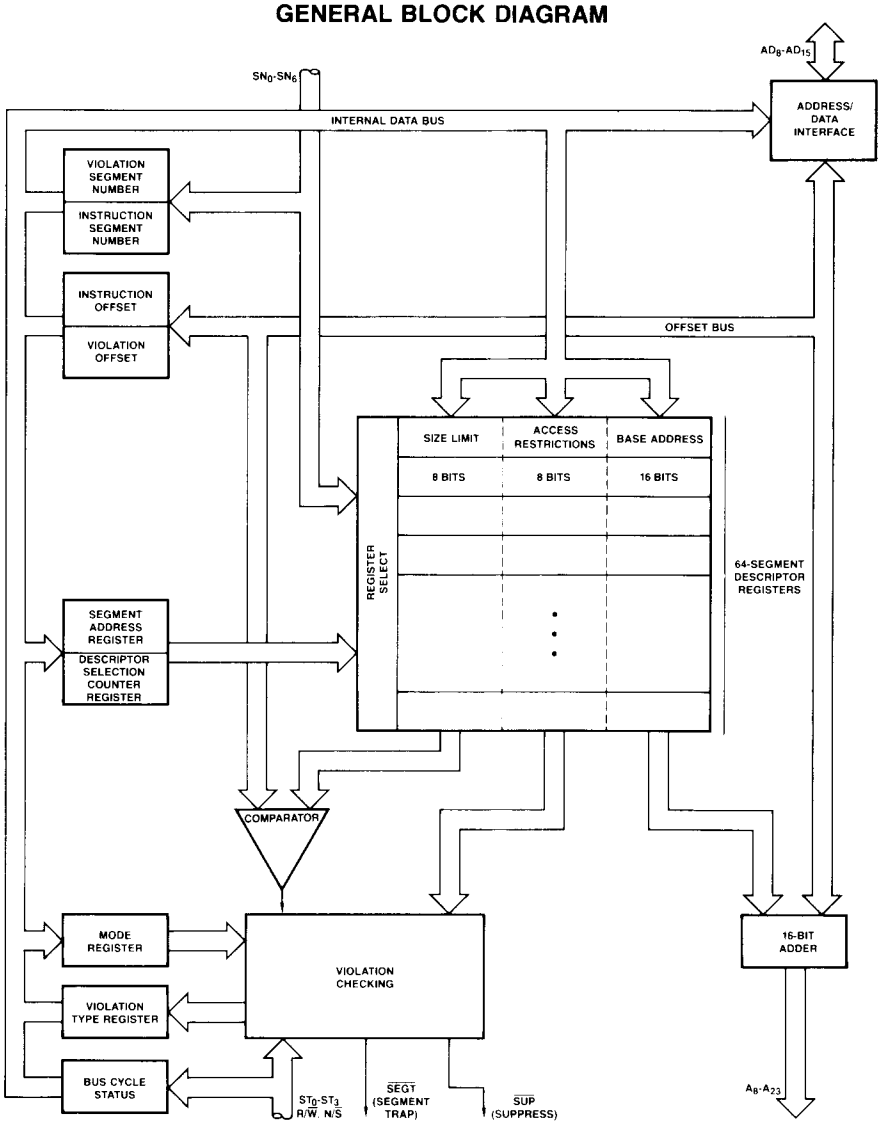


Figure 2.

MMU BLOCK DIAGRAM

The AmZ8010 block diagram (Figure 2) indicates the interface signals and the basic flow of information in the MMU. All control and status registers are connected to a common internal 8-bit bus. Information transfer to the internal data bus is accomplished across the multiplexed Address/Data inputs, AD_8-AD_{15}. The Address/Data lines also provide the upper byte of the address offset for logical-to-physical address mapping and violation checking. The Segment Number input is interfaced directly to the 64 Segment Descriptor Registers for physical base address look-up and attribute assignment.

The segment number addresses the Segment Descriptor Register, which holds the base address for logical-to-physical address translation. The base address is added to the 8-bit offset with a 16-bit adder. Output of the adder is then interfaced to the outputs, A_8-A_{23}, which carry the most significant 16 bits of the 24-bit physical address. If the requested offset is not within the segment boundary defined in the Segment Descriptor Register, an indication is made available to the CPU through the Segment Trap signal and a memory access is inhibited through the Suppress signal.

Each Segment Descriptor Register also includes an attribute field. The attribute field contains flags for specifying segment protection against certain types of access, warning when the end of a segment is approached, and for recording the types of accesses made to the segment. The attribute field is interfaced to the Violation Checking control block. Memory requests by the CPU are accompanied by status information, also interfaced to the Violation Checking control block. The control block compares the memory request attributes with the addressed segment's attributes and generates a Segment Trap signal to the CPU whenever is detects an attribute violation. A Memory Suppress signal may also be generated on access violations.

Attribute violation checking is enabled when the current requested memory location lies within the defined segment boundary. A comparator is used to compare the requested memory address offset and the offset limit for the addressed segment. The comparator generates the violation checking enable.

Among the registers that are accessible through the internal data bus and the Address/Data lines are six status registers and three control registers. The status registers, useful for recovery from a Segment Trap, are the Violation Segment Number register, Instruction Segment Number register, Instruction Offset register, Violation Offset register, Violation Type register and Bus Status register. The control registers are used for directing the functioning and internal addressing of the MMU. They are the Mode register, the Segment Descriptor Address register, and the Descriptor Selection Counter register.

FUNCTIONAL DESCRIPTION

The AmZ8010 Memory Management Unit (MMU) manages the large, 8M byte addressing spaces of the AmZ8001 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 multiprogramming systems. The MMU uses a translation table to transform the 23-bit logical address output from the AmZ8001 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 bypass this component.)

Memory segments can vary 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 AmZ8001 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 initially entered into the MMU. When a memory reference is made, these attributes are checked against the status information supplied by the AmZ8001 CPU. If a mismatch occurs, 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 AmZ8001 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.

SEGMENTED ADDRESSING

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

The 8M byte AmZ8001 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.

Addresses manipulated by the programmer, used by instructions, and output by the AmZ8001 are called *logical addresses*. The MMU takes the logical addresses and transforms them into the *physical addresses* required for accessing the memory (Figure 3). This address transformation process is called *relocation*.

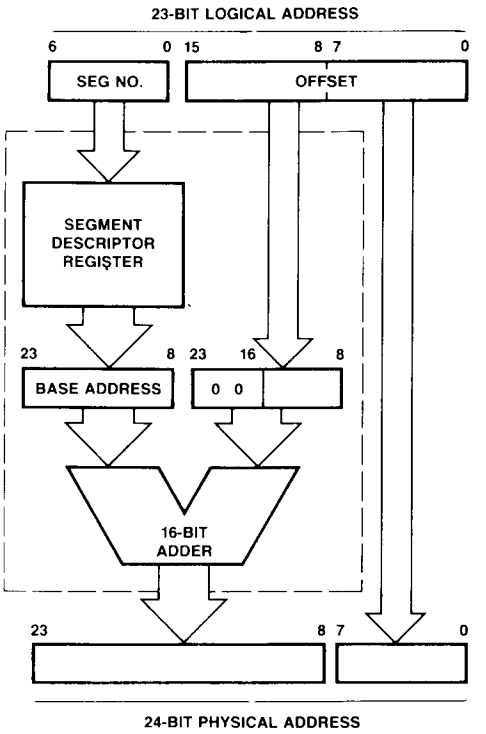


Figure 3. Logical-to-Physical Address Translation

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 address is the same as the low-order byte of the logical address offset. This low-order byte bypasses the MMU, reducing the number of pins required.

MEMORY PROTECTION

Each memory segment is assigned several attributes that are used to provide memory access protection. A memory request from the AmZ8001 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 AmZ8001 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 $\overline{SUP}$ in the event of an access violation. Suppress can be used by a memory system to inhibit stores and reads 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 an offset to this base address. On each access to a segment, the offset is checked against this range to ensure 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 $0 \leq N \leq 255$. This latter type of segment is useful for stacks, since the AmZ8001 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.

SEGMENT TRAP AND ACKNOWLEDGE

The AmZ8010 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 AmZ8001 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 aborted. (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 and Program Counter onto the system stack and loads another Program Status and Program Counter 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 context 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 context 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.

DMA OPERATION

Direct memory access operations may occur between AmZ8001 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, segment attributes are checked. 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, 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_0-SN_6. When the DMA logical memory address is valid, the DMASYNC line must be HIGH on 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.

VIRTUAL MEMORY

Several features of the MMU can be used in conjunction with external circuitry to support virtual memory for the AmZ8001. 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. Status Registers can be used in recovering from virtual memory access faults.

MULTIPLE MMUs

MMU architecture directly supports two methods for implementing 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 AmZ8001 can address. This scheme uses the URS flag in the Mode Register in connection with the high-order bit of the segment number (SN_6).

The second approach uses several MMUs to implement multiple translation tables. Multiple tables can reduce the time required to switch tasks by assigning separate tables to each task. Multiple translation tables for multitask 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 16M bytes 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/$\overline{S}$ line to select the MMU that contains the appropriate table.

MMU ARCHITECTURE

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 AmZ8010 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 4). 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 5). 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.

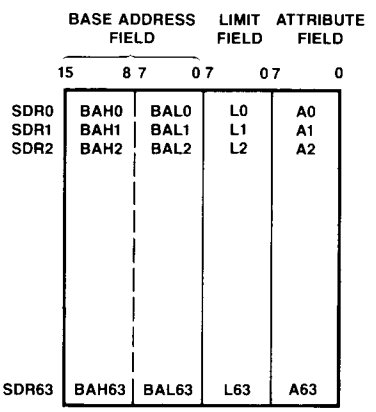


Figure 4. Segment Descriptor Registers

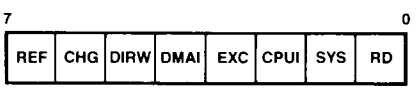


Figure 5. Attribute Field in Segment Descriptor Register

Execute-Only (EXC)
When this flag is set, the segment can be accessed only during an instruction fetch 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.

CONTROL REGISTERS

The three user-accessible 8-bit control registers in the MMU direct the functioning of the MMU (Figure 6). 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.

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 control bits.

Multiple Segment Table (MST)
This flag indicates whether multiple segment tables present in the hardware configuration are to be selected using the N/$\overline{S}$ line. When this flag is set, more than one table is present and the N/$\overline{S}$ 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/$\overline{S}$ line is High or Low. If the MST flag is set, the N/$\overline{S}$ line must match the NMS flag for the MMU to translate segment addresses, otherwise the MMU Address lines remain 3-stated.

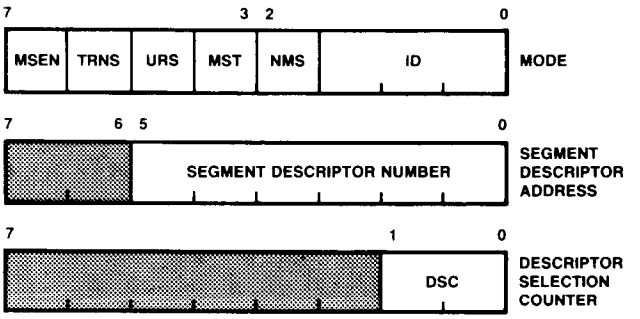


Figure 6. Control Registers

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 nontranslation 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. DSC can also be autoincremented to allow MMU block load facilities.

STATUS REGISTERS

Six 8-bit registers contain information useful in recovering from memory access violations (Figure 7). 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/$\overline{W}$ line is Low.

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

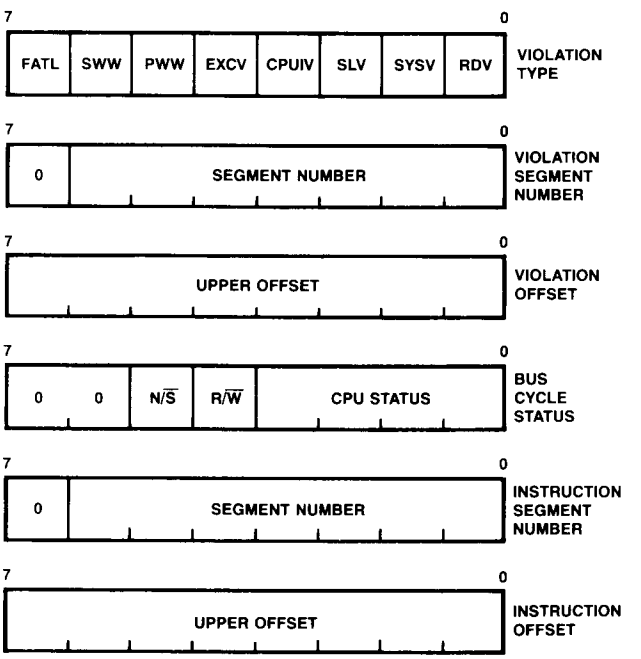


Figure 7. Status Registers

6

MMU COMMANDS

The various registers in the MMU can be read and written using AmZ8001 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_8-AD_{15}. If this command indicates that data is to be written into one of the MMU registers, the data is read from AD_8-AD_{15} 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_8-AD_{15} 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 autoincrementing 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 AmZ8010 translates addresses and checks for access violations by stepping through sequences of basic clock cycles corresponding to the cycle structure of the AmZ8001 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_0-ST_3 inputs. During a memory read cycle (Figure 8) the 7-bit segment number is input on SN_0-SN_6 one clock period earlier than the address offset; a HIGH on DMASYNC during T_3 indicates that the segment offset data is valid. The most significant eight bits of the address offset are placed on the AD_0-AD_{15} inputs early in the 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_3. Segment Trap Request and Suppress are asserted in T_2. Segment Trap Request remains LOW until Segment Trap Acknowledge is received. Suppress is asserted during the current machine cycle and terminates during T_3. Suppress is repeatedly asserted during CPU instruction execution cycles until the current instruction has terminated.

MMU COMMAND CYCLE

During the command cycle of the MMU (Figure 10), commands are placed on the Address/Data lines during T_1. 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_2. Data read from the MMU is placed on the Address/Data lines late in the T_{WA} cycle.

Input/Output and Refresh

Input/Output and Refresh operations are indicated by the status lines ST_0-ST_3. During these operations, the MMU refrains from any address translation or protection checking. The address lines A_8-A_{15} remain 3-stated.

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 an AmZ8001 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_8-AD_{15} and A_8-A_{23} lines are 3-stated. The $\overline{SUP}$ and $\overline{SEGT}$ 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 AmZ8010 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 and to flag special data to be returned on a read access. The Segment Trap remains LOW until a Trap Acknowledge signal is received. If a violation oc-

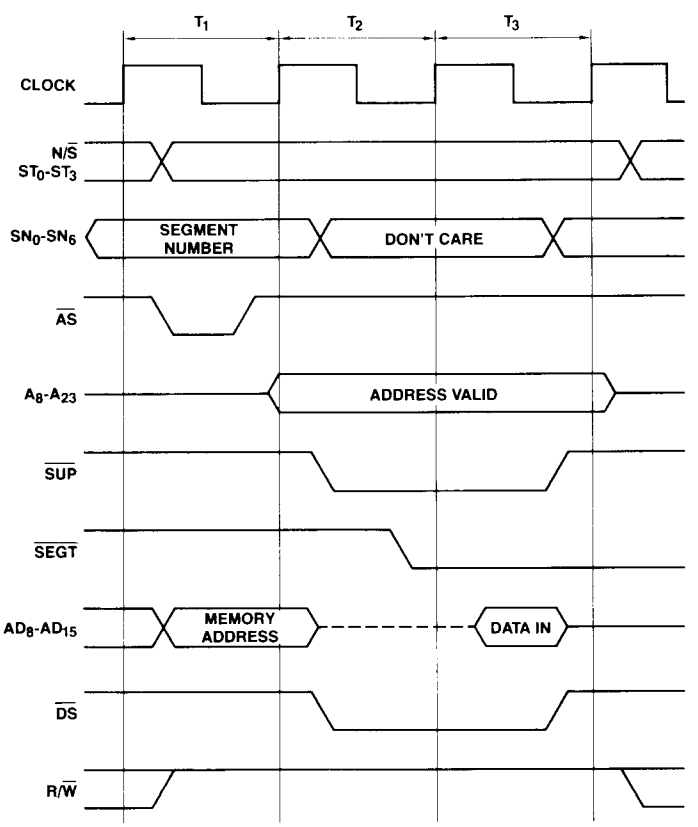


Figure 8. Memory Read Timing

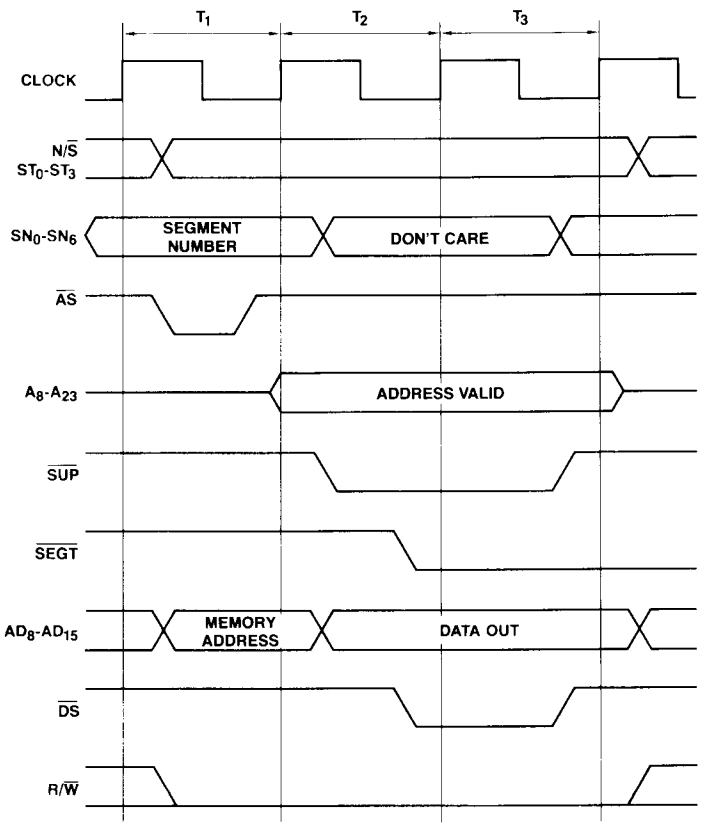


Figure 9. Memory Write Timing

8

curs, 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 AmZ8001 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.

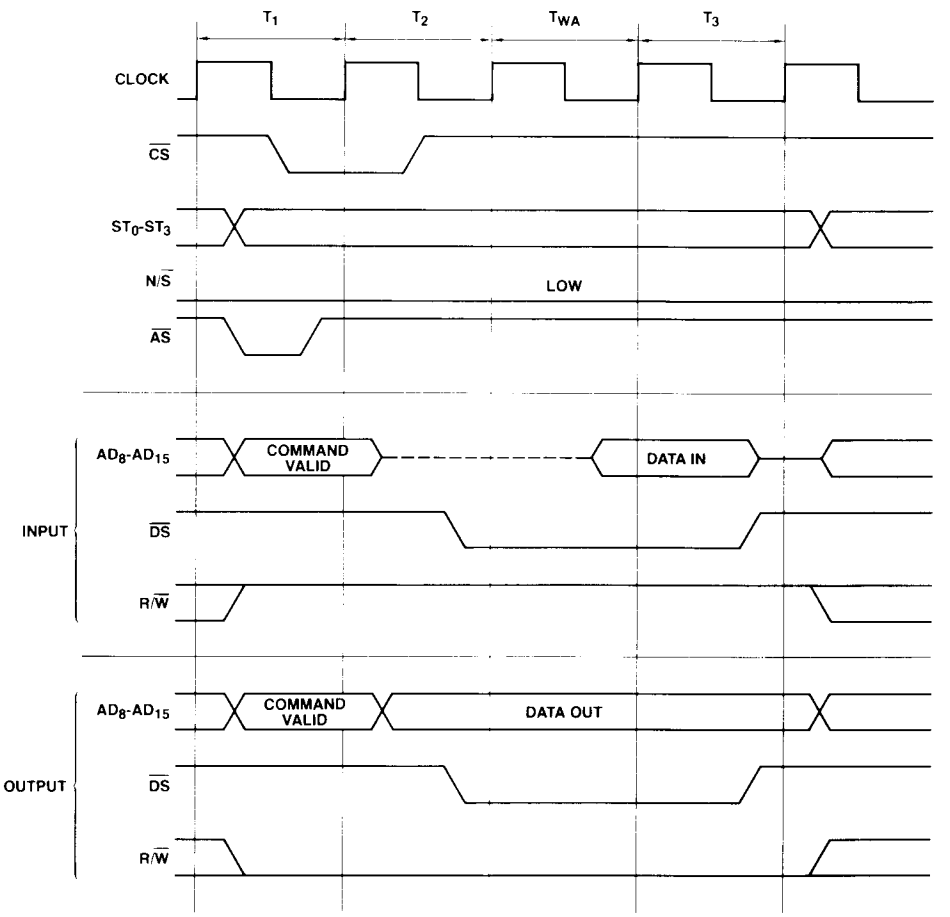


Figure 10. I/O Command Timing

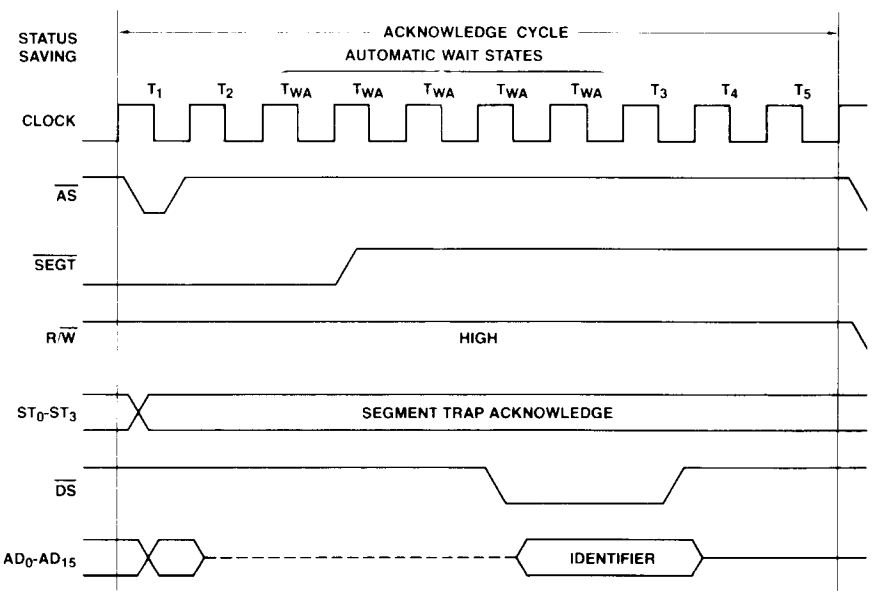


Figure 11. Segment Trap and Acknowledge Timing

INTERFACE SIGNAL DESCRIPTION

Address Bus (A_8-A_{23})

Outputs, active HIGH, 3-state. These address lines are the 16 most-significant bits of the physical memory location.

Address/Data Bus (AD_8-AD_{15})

Inputs/outputs, active HIGH, 3-state. These multiplexed address and data lines are used for commands, data and for logical addresses intended for translation.

Address Strobe ($\overline{AS}$)

Input, active LOW. The rising edge of $\overline{AS}$ indicates that AD_8-AD_{15}, ST_0-ST_3, R/$\overline{W}$ and N/$\overline{S}$ are valid.

System Clock (CLK)

Input. CLK is the 5V single-phase time-base input used for both the CPU and MMU.

Chip Select ($\overline{CS}$)

Input, active LOW. This line selects an MMU for a control command. It is not used during translation.

Decouple

Output from on-chip negative substrate-bias generator.

DMZ/Segment Number Synchronization Strobe (DMASYNC)

Input, active HIGH. A LOW on this line indicates a DMA access is occuring; a HIGH indicates the segment number is valid. It must always be HIGH during CPU cycles.

Data Strobe ($\overline{DS}$)

Input, active LOW. This line provides timing for the data transfer between the MMU and the AmZ8001 CPU.

Normal/System Mode (N/$\overline{S}$)

Input, LOW = System Mode. N/S indicates the AmZ8001 CPU or AmZ8016 DMA is in the Normal or System Mode. The signal can also be used to switch between MMUs during different phases of an instruction.

Reset ($\overline{RESET}$)

Input, active LOW. A LOW on this line resets the MMU.

Read/Write (R/$\overline{W}$)

Input, LOW = write. R/$\overline{W}$ indicates the AmZ8001 CPU or AmZ8016 DMA is reading from or writing to memory or the MMU.

Segment Trap Request ($\overline{SEGT}$)

Output, active LOW, open drain. The MMU interrupts the AmZ8001 CPU with a LOW on this line when the MMU detects an access violation or write warning.

Segment Number (SN_0-SN_6)

Inputs, active HIGH. The SN_0-SN_5 lines are used to address one of 64 segments in the MMU; SN_6 is used to selectively enable the MMU.

Status (ST_0-ST_3)

Inputs, active HIGH. These lines specify the AmZ8001 CPU or AmZ8016 DTC status.

ST_3-ST_0	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	Non-maskable interrupt acknowledge
0 1 1 0	Non-vectored 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	Extended processor transfer
1 1 1 1	Reserved

Suppress ($\overline{SUP}$)

Output, active LOW, open drain. This signal is asserted during the current bus cycle when any access violation except write warning occurs.

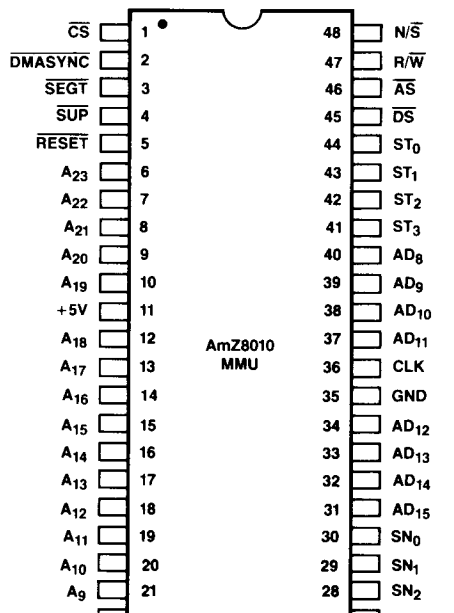
+5

+5 volt power supply.

GND

Ground.

CONNECTION DIAGRAM
Top View



Note: Pin 1 is marked for orientation.

Figure 12.

OPERATING RANGES

The following ranges apply to DC, switching and functional specifications.

Range	Part Number Suffix	Temperature	V_{CC}
C	DC	T_A = 0 to +70°C	4.75 to 5.25V

MAXIMUM RATINGS above which useful life may be impaired

Voltages on all inputs and outputs with respect to GND	−0.3 to +7.0V
Ambient Temperature under bias	0 to 70°C
Storage Temperature	−65 to +150°C

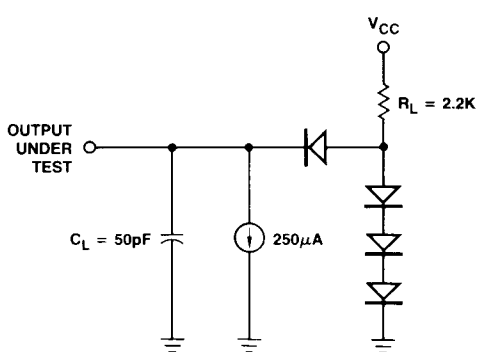
The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

ORDERING INFORMATION

Order Number	Speed	Package Type (Note 1)	Screening (Note 3)	Operating Range (Note 2)
AmZ8010DC	4MHz	D-48	C-1	C
AmZ8010DCB	4MHz	D-48	C-3	C

Notes:
1. P = Molded DIP, D = Hermetic DIP, F = Flat Pack. Number following letter is number of leads. See Appendix B of Data Book for detailed outline. Where Appendix B contains several dash numbers, any of the variations of the package may be used unless otherwise specified.
2. C = 0 to 70°C, V_{CC} = 4.75V to 5.25V, M = −55 to +125°C, V_{CC} = 4.50V to 5.50V.
3. See Appendix A of Data Book for details of screening. Levels C-1 and C-3 conform to MIL-STD-883, Class C. Level B-3 conforms to MIL-STD-883, Class B.

SWITCHING LOAD CIRCUIT



DC CHARACTERISTICS

All characteristics apply over the operating range unless otherwise noted

Parameter	Description	Test Conditions	AmZ8010DC Min	AmZ8010DC Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	$V_{CC}-0.4$	$V_{CC}+0.3$	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	$V_{CC}+0.3$	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	$I_{OH} = -250\mu A$	2.4		Volts
V_{OL}	Output Low Voltage	$I_{OL} = +2.0mA$		0.4	Volts
I_{IL}	Input Leakage	$0.4 \leq V_{IN} \leq +2.4V$		±10	μA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4V$		±10	μA
I_{CC}	V_{CC} Supply Current			300	mA

Note 1: Typical values are for $T_A = 25°C$, nominal supply voltages and nominal processing parameters.

AC CHARACTERISTICS

Number	Parameter	Description	Min (ns)	Max (ns)
1	TcC	Clock Cycle Time	250	
2	TwCh	Clock Width (High)	105	
3	TwCl	Clock Width (Low)	105	
4	TfC	Clock Fall Time		25
5	TrC	Clock Rise Time		25
6	TdDSA(RDv)	$\overline{DS}\downarrow$ (Acknowledge) to Read Data Valid Delay (Note 1)		100
7	TdDSA(RDf)	$\overline{DS}\uparrow$ (Acknowledge) to Read Data Float Delay (Note 1)	20	75
8	TdDSR(RDv)	$\overline{DS}\downarrow$ (Read) to AD Output Driven Delay (Note 1)		100
9	TdDSR(RDf)	$\overline{DS}\uparrow$ (Read) to Read Data Float Delay (Note 1)	20	75
10	TdC(WDv)	CLK↑ to Write Data Valid Delay		160
11	ThC(WDn)	CLK↓ to Write Data Not Valid Hold Time	30	
12	TwAS	Address Strobe Width	60	
13	TsOFF(AS)	Offset Valid to $\overline{AS}\uparrow$ Setup Time	60	
14	ThAS(OFFn)	$\overline{AS}\uparrow$ to Offset Not Valid Hold Time	60	
15	TdAS(C)	$\overline{AS}\downarrow$ to CLK↑ Delay	110	
16	TdDS(AS)	$\overline{DS}\uparrow$ to $\overline{AS}\downarrow$ Delay	50	
17	TdAS(DS)	$\overline{AS}\uparrow$ $\overline{DS}\downarrow$ Delay	50	
18	TsSN(C)	SN Data Valid to CLK↑ Setup Time	120	
19	ThC(SNn)	CLK↑ to SN Data Not Valid Hold Time	0	
20	TdDMAS(C)	DMASYNC Valid to CLK↑ Delay	120	
21	TdSTNR(AS)	Status (ST_0-ST_3, N/$\overline{S}$, R/$\overline{W}$) Valid to $\overline{AS}\uparrow$ Delay	60	
22	TdC(DMA)	CLK↑ to DYMASYNC↓ Delay	140	
23	TdST(C)	Status (ST_0-ST_3) Valid to CLK↓ Delay	140	
24	TdDS(STn)	$\overline{DS}\uparrow$ to Status Not Valid Delay	0	
25	TdOFF(Av)	Offset Valid to Address Output Valid Delay (Notes 1, 3 and 5)		175
26	TdST(Ad)	Status Valid to Address Output Driven Delay (Notes 1, 3 and 4)		155
27	TdDS(Af)	$\overline{DS}\uparrow$ to Address Output Float Delay (Note 1)	30	160
28	TdAS(Ad)	$\overline{AS}\downarrow$ to Address Output Driven Delay (Notes 1 and 3)		145
29	TdC(Av)	CLK↑ to Address Output Valid Delay (Notes 1 and 3)		255
30	TdAS(SEGT)	$\overline{AS}\uparrow$ to $\overline{SEGT}\downarrow$ Delay (Notes 1 and 2)		160
31	TdC(SEGT)	CLK↑ to $\overline{SEGT}\uparrow$ Delay (Notes 1 and 2)		300
32	TdAS(SUP)	$\overline{AS}\uparrow$ to $\overline{SUP}\downarrow$ Delay (Notes 1 and 2)		150
33	TdDS(SUP)	$\overline{DS}\uparrow$ to $\overline{SUP}\uparrow$ Delay (Notes 1 and 2)	30	155
34	TdCS(AS)	Chip Select Input Valid to $\overline{AS}\uparrow$ Setup Time	10	
35	ThAS(CSn)	$\overline{AS}\uparrow$ to Chip Select Input Not Valid Hold Time	80	
36	TdAS(C)	$\overline{AS}\uparrow$ CLK↑ Delay	0	
37	TsCS(RST)	Chip Select Input Valid to $\overline{RESET}\uparrow$ Setup Time	150	
38	ThRST(CSn)	$\overline{RESET}\uparrow$ to Chip Select Input Not Valid Hold Time	0	
39	TwRST	$\overline{RESET}$ Width (Low)	2TcC	
40	TdC(RDv)	CLK↑ to Read Data Valid Delay (Note 1)		460
41	TdDS(C)	$\overline{DS}\uparrow$ to CLK↑ Delay	30	
42	TdC(DS)	CLK↓ to $\overline{DS}\uparrow$ Delay	0	110

Notes:
1. 50pF Load.
2. 2K Pull-up.
3. These values apply to the AmZ8010-3 version only. An AmZ8010-2 version available soon will improve these values by 40ns.
4. Clock↑ to Address Valid in the AmZ8001/AmZ8002 CPU is specified at 100ns with 100pF load. With a load of 50pF, this delay is reduced to 80ns.
5. Clock↑ to Status Valid Delay in the AmZ8001/AmZ8002 CPU is specified at 110ns with a 100pF load. With a 50pF load, this delay is reduced to 100ns.

AC TIMING DIAGRAM