Z8015 Z8000® PMMU Paged Memory Management Unit

Product Specification

April 1985

FEATURES

- PMMU architecture supports multiprogramming systems and virtual memory implementations.
- Dynamic page 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.
- 64 pages, each 2048 bytes in length, can be mapped into a total physical address space of 16 megabytes; all 64 pages are randomly accessible.
- Pages larger or smaller than 2048 bytes can be easily implemented.
- The number of accessible pages can be increased by using multiple PMMUs.

GENERAL DESCRIPTION

The Z8015 Paged Memory Management Unit (PMMU), a new member of Zilog's Z8000 Family, is designed to support a paged virtual memory system for the Z8003 Virtual Memory Processor Unit (VMPU). Although designed primarily for the Z8003, the PMMU can also be used to support other CPUs in the Z8000 Family. The sophisticated memory management features of the PMMU include access validation for memory protection, a write-warning that gives advance warning of possible stack overflow, and the generation of instruction aborts for accesses to pages not in main memory. Each PMMU can manage a basic memory area of sixty-four 2048-byte, fixed-size pages. The VMPU's 8M byte logical address space is translated by the PMMU into a 16M byte physical address space. Page size can be easily changed and multiple PMMUs can be combined to support more pages.

FUNCTIONAL DESCRIPTION

The Z8015 Paged Memory Management Unit (PMMU) manages the 8M byte addressing spaces of the Z8003 VMPU. The PMMU provides dynamic page relocation as well as numerous memory protection features.

Dynamic page relocation makes user software addresses independent of the physical memory addresses, thereby freeing the user from specifying where information is located in the physical memory. It also provides a flexible, efficient method for supporting multiprogramming systems.

The PMMU uses a content-addressable translation table to transform each 23-bit logical address output from the VMPU into a 24-bit address for the physical memory. (Only logical memory addresses go to a PMMU for translation; I/O addresses and data bypass this component.) The translation table consists of 64 page descriptors; each descriptor contains address translation, status, and access information for one memory page. Each PMMU can then manage up to 64 pages of memory.

Multiple PMMUs can be used to support more than 64 pages within a given address space. In addition, PMMUs can be used to accommodate separate translation tables for System and Normal operating modes.

The PMMU is designed to support a memory page 2048 bytes in length. This basic page length can be increased or decreased using a minimal amount of external circuitry.

The PMMU is specially designed to operate with a Z8003 VMPU to implement a paged virtual memory system. If the current PMMU instruction addresses a page not in main memory (a page fault), the PMMU initiates an Instruction Abort operation in the VMPU. During an abort, the PMMU aborts the execution of the current instruction, then saves the information needed to restart the aborted instruction. On completion of the abort, the PMMU initiates a trap in the

VMPU to a routine that brings the addressed page into main memory, updates the descriptor table of the PMMU to allow address translation to the new page, and restarts the execution of the interrupted instruction.

The logical address that caused the page fault is available in three violation address registers of the PMMU. This information can be used to fetch the required instruction or data page into main memory and/or to create a page descriptor entry so that the executing program can access those instructions or data. The logical address of the instruction generating the page fault is available in three instruction address registers of the PMMU. This information can be used to reset the Program Counter to restart the instruction. The instruction to be restarted must also be examined to determine if adjustments must be made to any VMPU registers to ensure correct execution. Finally, the Read/Write Data Count register can be accessed so that certain instructions, such as Load Multiple, can be restarted correctly.

As an aid in implementing efficient paging algorithms, the PMMU provides Changed and Referenced flags for each page descriptor register. The Changed flag indicates that a page has been altered and hence must be copied to secondary storage before that physical memory can be overwritten by another page. The Referenced flag can be used to determine which pages have not been accessed by an executing program—these are the pages that should first be removed from memory when room must be made to bring another page into memory.

PMMU memory protection features safeguard memory areas from unauthorized or unintended access by associating special access restrictions with each page. A page is assigned a number of attributes when its descriptor is initially entered into the PMMU. When a memory reference is made, these attributes are checked against the status information supplied by the VMPU. If a mismatch occurs, the instruction is aborted, a Trap Request signal is generated, and the VMPU is interrupted. The VMPU then checks the status registers of the PMMU to determine the cause of the abort.

Pages are protected by modes of permitted use, such as read only, system only, and execute only. A Valid flag indicates whether or not a descriptor has been initialized. Other page management features include a Write Warning flag useful for stack operations.

The PMMU is controlled by 20 Special I/O instructions, which can be issued from the VMPU in System mode only. With these instructions, system software can assign program pages to arbitrary memory locations, restrict the use of pages, and monitor whether pages have been read or written.

The PMMU has two operating modes: an Address Translation mode in which addresses are translated automatically as they are received, and a Command mode, during which specific registers in the PMMU are accessed using Special I/O commands. Figure 1 shows two simplified block diagrams that illustrate the internal organization and data/signal flow within the PMMU. The resources used in the Translation and the Command modes are shown, separately, in Figures 1a and 1b.

SEGMENTED ADDRESSING AND ADDRESS TRANSLATION

Compared with 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 23-bit addresses output by the VMPU divide an 8M byte addressing space into 128 segments of up to 64K bytes each. A 23-bit segmented address consists of a 7-bit segment number and a 16-bit offset used to address any byte relative to the beginning of the segment. The two parts of the segmented address (segment number and offset) can be manipulated separately.

The PMMU divides physical memory into 2048-byte pages. Pages are assumed to be allocated in memory on 2048-byte boundaries so that the 11 low-order bits of the starting location of each page are always equal to zero. Segments in a virtual memory system can consist of pages that need not be in physical storage. Those segment pages in main memory need not be contiguous. Segments can have a variable number of pages. Certain pages can be designated so that writes into the last 128 bytes generate a warning trap without causing an instruction abort. If such a page is used as the last page of the system stack, the warning trap can be used to initiate the allocation of another page to the stack segment to prevent a stack overflow error.

The addresses manipulated by the programmer, used by instructions, and output by the VMPU are called logical addresses. The PMMU translates logical addresses into the physical addresses required for accessing memory; this process is called relocation.

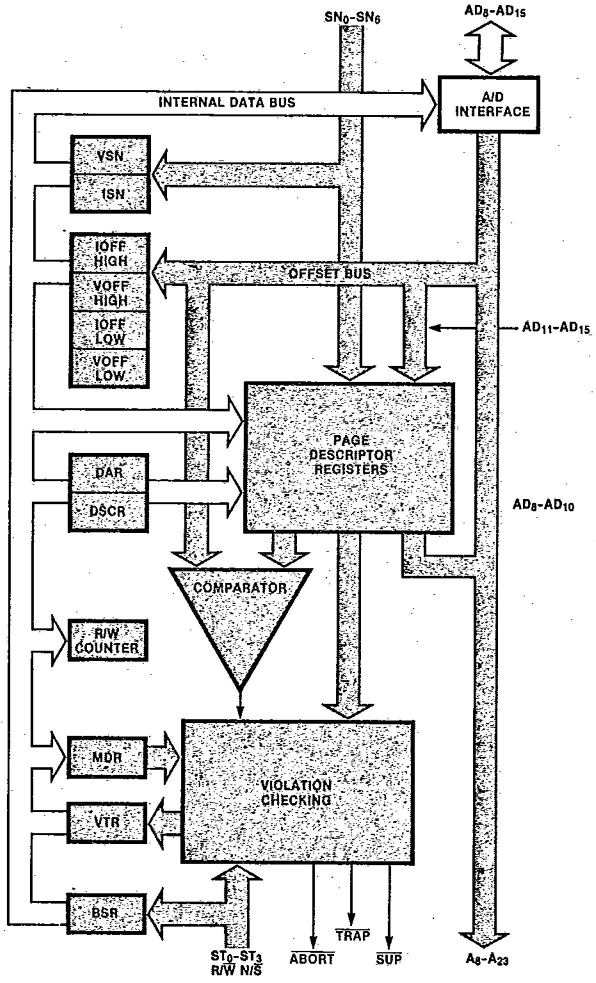


Figure 1a. Address Translation Mode PMMU Operating Modes, Simplified Flow Diagram

The translation activity in the PMMU that provides address relocation is controlled by four internal control flags and six input lines. The four flags are:

Master Enable (MSEN) Flag. This flag controls when the PMMU outputs physical addresses on its Address bus (A) lines. When this bit is clear, the A lines remain 3-stated and no checking is performed.

Translate (TRNS) Flag. This flag determines whether the output on the A lines is the logical address as input (with most significant bit at 0) or a translated address. When translation is not performed, no checking is done.

Multiple Page Table (MPT) Flag. This flag indicates whether separate PMMUs are to be used for System and Normal pages.

Normal Mode Select (NMS) Flag. When the Multiple Page Table flag is set, this flag indicates whether the PMMU contains System or Normal page descriptors.

The six input lines used in the control of the PMMU are:

N/$\overline{S}$ Line. This line is used by the PMMU to distinguish System mode accesses from Normal mode accesses. When the Multiple Page Table flag is set, the N/$\overline{S}$ line acts as a chip-select mechanism.

Chip Enable ($\overline{CE}$) Line. This line acts as a master enable control line: it must be asserted for any address translation to occur or for any address to be output by the PMMU.

Status Lines (ST_0-ST_3). These four lines are used by the PMMU to determine the type of transaction in progress.

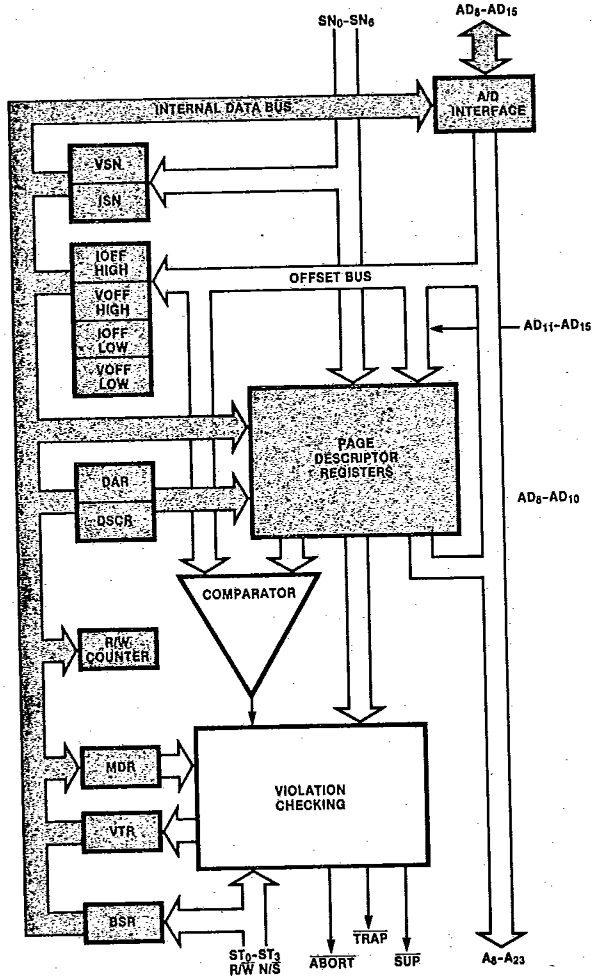


Figure 1b. Command Mode PMMU Operating Modes, Simplified Flow Diagram

Access violation checking, write warning checking, and page fault monitoring functions occur only when the PMMU is enabled for address translation. For example, if Chip Enable is not asserted, the PMMU does not generate an Abort Request even if none of its descriptors match the logical address.

The address translation process is transparent to user software; a simplified flow diagram of this process is shown in Figure 2. A content-addressable translation table in the PMMU compares the 7-bit segment number and five most-significant bits of the offset with the logical address field of each descriptor. If a match occurs and that descriptor's Valid flag is set, the physical address field of that descriptor is accessed. The 11-bit logical address within the page is concatenated to this 13-bit physical base address to obtain the actual physical memory location. Because the base address of a page always has the low-order 11 bits equal to zero, only the high-order 13 bits are stored in the PMMU and used in the translation. The PMMU outputs the 16 most significant bits of the translated address.

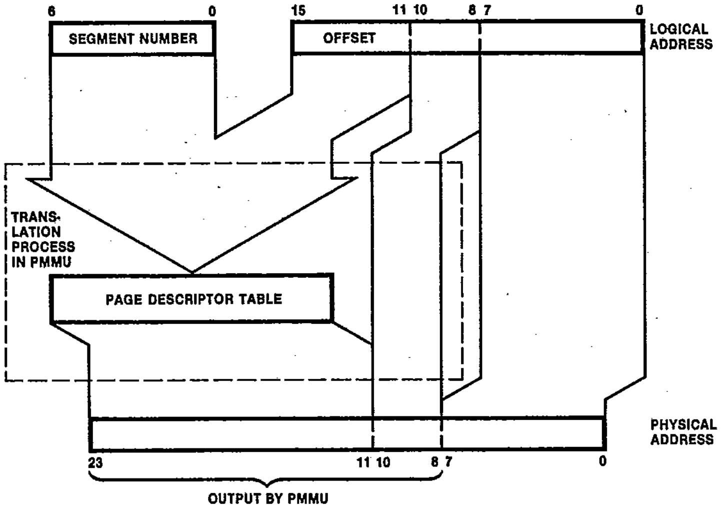


Figure 2. Logical-to-Physical Address Translation

MEMORY PROTECTION

Each memory page is assigned several attributes that are used to provide memory access protection. A memory request from the VMPU is accompanied by status information that indicates the attributes of the memory request. The PMMU compares the memory request attributes with the page attributes and generates Instruction Abort Request, Suppress, and Trap Request signals whenever it detects an attribute violation.

An Abort Request is used to generate the Abort and Wait inputs to the VMPU that cause the current instruction to be aborted. The Suppress input is used by the VMPU to inhibit stores into the memory and thus protect the contents of the memory from erroneous changes. A Trap Request informs the VMPU and the system control program of the violation so that appropriate action can be taken for recovery.

Three attributes: read only, execute only, and system access only, can be associated with each page. When an attempted access violates any one of the page attributes, Abort Request, Trap Request, and Suppress signals are generated by the PMMU.

Each descriptor register has a Valid flag in the attribute field. When set to 1, this flag indicates that the descriptor contains valid translation information and its logical address field is to be used in the associative match process. If Chip Enable is asserted and no match is found, the PMMU, if enabled, generates Abort Request, Trap Request, and Suppress signals. The PMMU is enabled under either of the following conditions: the MSEN and TRNS flags are both 1 and the MPT flag is 0; or the MSEN and TRNS flags are both 1, the MPT flag is 1, and both input N/S and the NMS flag have the same value.

Normally, the legal range of offsets within a segment goes from 0 to 65,535 bytes. A stack segment, however, has legal word offsets ranging downward from 65,534 to 0 bytes; the stack manipulation instructions cause stacks to grow toward lower memory locations. When a stack grows to the limit of its allocated segment, additional memory can be added to the segment. As an aid in maintaining stacks, the PMMU detects when a write is performed to the lowest-allocated 128 bytes of a stack page and generates a Trap Request if the DIRW attribute flag is set in the page descriptor. Since neither a Suppress nor Abort Request signal is generated, the write is allowed to proceed. This write warning can then be used to indicate that more memory (that is, another page) needs to be allocated to the segment.

PMMU REGISTER ORGANIZATION

The PMMU contains a set of 64 page descriptor registers that supply the information needed to map logical memory addresses to physical memory locations. The PMMU also contains three 8-bit control registers for programming the PMMU, and nine 8-bit status registers to record information in the event of an access violation.

Page Descriptor Registers

The segment number and five most-significant bits of a logical address determine, by associative lookup, which page descriptor register is used in address translation. Each register also contains the necessary information to enable checking to ensure that the type of reference made is permitted. An indication that the segment has been previously read or written is also contained in the register.

Each of the 64 page descriptor registers contains a 12-bit logical address field, a 13-bit physical address field, and a 7-bit attribute field (Figure 3).

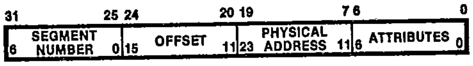


Figure 3. Page Descriptor Register Format

The logical address field is used during the associative search phase of address translation; a match of this field with the most-significant bits of the logical address indicates that the descriptor is to be used during physical address generation. The physical address field supplies the most-significant bits of the generated physical address.

The attribute field contains seven flags (Figure 4). Three flags protect the page against certain types of access, one indicates the special structure of the page, and two encode the types of accesses that have been made to the page. The seventh flag is used to indicate whether or not the information in the descriptor is valid. A flag is set when its value is 1. During a write to only the attribute field, bit 7 of the byte is ignored. When an attribute field is read, bit 7 is undefined. When an entire descriptor is accessed, bit 7 will be part of the physical address field. The following descriptions explain how these flags are used.

Valid (VALID). When this flag is set, the descriptor contains valid page information for the currently executing process. When this bit is clear, the logical address generated by the VMPU is not compared against the contents of the logical address field, so the descriptor is not used for address translation. Only descriptors that have this flag set are used during the associative search.

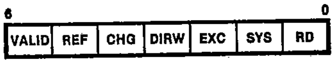


Figure 4. Attribute Field of Page Descriptor Register

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

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

Execute-Only (EXC). When this flag is set, the page can be accessed only during an Instruction Fetch cycle and is thus protected against access during other cycles.

Direction and Warning (DIRW). When this flag is set, the page's memory locations are considered to be organized in descending order and each write to the page is checked for access to the lowest 128 bytes. Such an access generates a Trap Request signal to warn of potential stack overflow, but neither an Abort Request nor a Suppress signal is generated.

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

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

The byte format that is required to write into or read from an attribute field is shown in Figure 5.

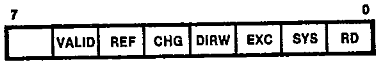


Figure 5. Format of Byte for Reading or Writing a Descriptor's Attribute

Control Registers

The three user-accessible, 8-bit control registers in the PMMU direct the functioning of the PMMU (Figure 6). The Mode register provides a sophisticated method for selectively enabling PMMUs in multiple-PMMU configurations. The Descriptor Address (DAR) register selects the particular page descriptor register to be accessed during a control operation. The Descriptor Selection Count (DSC) register points to the byte in the Page Descriptor register to be accessed during a control operation.

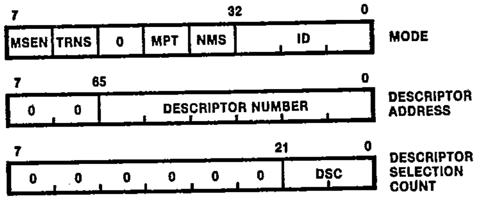


Figure 6. Control Registers

2081-004, 005, 006

Mode Register

The Mode register contains a 3-bit identification (ID) field that can distinguish up to eight enabled PMMUs in a multiple-PMMU configuration. This field is used during the Trap Request acknowledge sequence. In addition, the Mode register contains the following four flags:

Multiple Page Table (MPT). This flag indicates whether more than one page table is present in the hardware configuration. When this flag is set, more than one table is present and the N/S line is used to determine whether the PMMU contains the appropriate table.

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

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

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

Descriptor Address Register (DAR)

This register points to one of the 64 page descriptor registers. Control commands to the PMMU that access page descriptors implicitly use this pointer to select one of the page descriptor registers. The DAR has auto-incrementing capability so that multiple descriptors can be accessed in a block read/write fashion.

Descriptor Selection Count (DSC) Register

This register holds a 2-bit counter that indicates which byte in the descriptor is being accessed during Read or Write operations. A zero in this counter indicates that the highest-order byte of the descriptor is to be accessed (most-significant byte of the logical address field), a one indicates the next byte of the descriptor, a two indicates the third byte, and a three indicates the least-significant byte (containing the attribute field).

Status Registers

The following nine 8-bit status registers contain information useful in recovering from memory access violations (Figure 7):

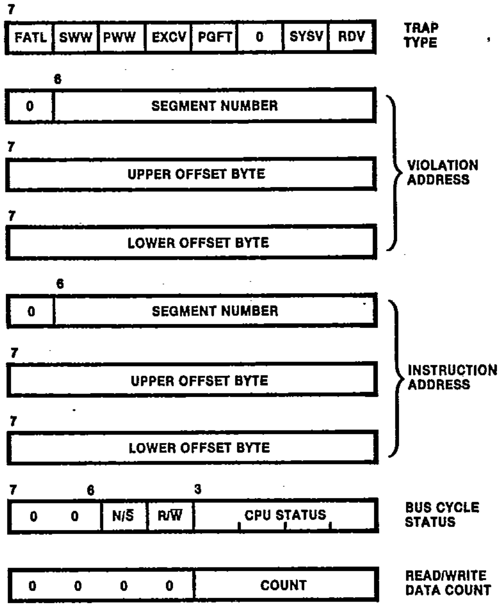


Figure 7. Status Registers

Trap Type Register. This register describes the conditions that generate a Trap Request signal.

Violation Segment Number and Violation Offset Registers. These three registers record the logical address that caused a Trap Request.

Instruction Address Registers. These three registers record the logical address of the last instruction fetched before the first warning, access violation, or page fault.

Bus Cycle Status Register. This register records the bus cycle status (status code, Read/Write operation and Normal/System mode).

Read/Write Data Count Register. This register contains a 4-bit counter that counts the number of read and write data transactions whose addresses have been translated by the PMMU since the last instruction fetch cycle. This count is locked when an Abort Request is generated, and indicates the number of successful data transactions performed by the aborted instruction.

Violation Type Register (VTR) Flags

The VTR is used by the PMMU to determine the cause of a Trap Request. The PMMU generates a Trap Request when: it detects an access violation, such as an attempt to write into a read-only page; it detects a warning condition, which is a write into the lowest 128 bytes of a page with the DIRW flag set; or no entry matches the logical address (a page fault). The following seven flags are contained by the Violation Type register (VTR):

Read-Only Violation (RDV). This flag is set when the VMPU attempts 1to access a read-only page and the R/W line is Low.

System Violation (SYSV). This flag is set when the VMPU accesses a system-only page and the N/S line is High.

Execute-Only Violation (EXCV). This flag is set when the VMPU attempts to access an execute-only page other than during an instruction fetch cycle.

Page Fault (PGFT). This flag is set when no logical address field of the valid descriptors in the PMMU matches the upper 12 bits of the logical address.

Primary Write Warning (PWW). This flag is set when an access is made to the lowest 128 bytes of a page with the DIRW flag set.

Secondary Write Warning (SWW). This flag is set when the VMPU writes data into the last 128 bytes of the system stack and EXCV, SYSV, PGFT, RDV, or PWW is set. With SWW set, subsequent write warnings for accessing the system stack do not generate a Trap Request.

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

ABORT, TRAP REQUEST, AND ACKNOWLEDGE

The PMMU generates a four-clock-cycle Abort Request ($\overline{\text{ABORT}}$) when it detects an access violation or a page fault. This signal on the $\overline{\text{ABORT}}$ (pin 33) and $\overline{\text{WAIT}}$ (pin 28) inputs of the VMPU inserts a five-cycle abort sequence that causes the VMPU to terminate instruction execution. A Trap Request is generated when the PMMU detects an access violation, page fault, or write warning. This signal on the translation trap line of the VMPU (SAT, pin 14) causes the VMPU to generate a trap acknowledge after the instruction abortion (for an access violation or page fault) or after the execution of the instruction (for a write warning). In the case of an access violation or page fault, the PMMU also activates Suppress ($\overline{\text{SUP}}$), which can be used by the memory to inhibit memory writes.

Trap Request remains Low until a trap acknowledge is received (status = 0100). If a VMPU-generated violation occurs, Suppress is asserted for that memory reference. (If a Z8001 or Z8002 CPU generates the violation, any subsequent CPU memory reference also causes Suppress to be asserted until the end of the instruction.) Intervening DMA accesses are not suppressed, however, unless they generate a violation. Violations detected during DMA accesses cause Suppress to be asserted for that access only; no Trap or Abort Requests are generated during DMA accesses.

Trap Requests to the VMPU are handled similarly to interrupts. To service a PMMU trap, the VMPU issues a trap acknowledge. The acknowledge is usually preceded by a dummy instruction fetch that is not used by the VMPU (the PMMU has been designed to ignore this dummy fetch). During the identifier fetch of the acknowledge cycle, all enabled PMMUs use the Address/Data (AD) lines to indicate their status. A PMMU that has generated a Trap Request outputs a 1 on the AD line associated with the number in its ID field; a PMMU that has not generated a trap request outputs a 0 on its associated AD line. AD lines with no associated PMMU remain 3-stated. During a trap acknowledge, a PMMU uses AD line 8 + i if its ID field is i.

Following the Acknowledge cycle, the VMPU automatically pushes the program status onto the system stack and loads another program status from the trap vector at location 20_H in the program status area. The PMMU's trap line is reset during the trap acknowledge. Suppress is not generated during the stack push. If the push operation creates a write warning condition, a Trap Request is generated and serviced at the end of the context swap. The SWW flag is also set. Servicing this second Trap Request also creates a write warning condition, but because the SWW flag is set, no Trap Request is generated. If a violation or page fault rather than a write warning occurs during the context swap, the FATL flag is set rather than the SWW flag. Subsequent violations or faults cause Suppress but not Trap Request to be asserted. Without the SWW and FATL flags, trap processing routines that generate memory violations or faults would repeatedly be interrupted and called to process the trap they created.

The VMPU 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 Trap Type register reset.

MULTIPLE PMMUs

Although only one PMMU should be actively translating addresses at a given time, PMMU architecture directly supports various methods for multiple PMMU configurations. The following four examples illustrate different ways that PMMUs can be used to implement memory management systems capable of handling more than 64 pages.

Example 1. The first approach extends the capability of one PMMU for handling 64 pages to a multiple-PMMU configuration that manages more than 64 pages. The Chip Enable line is used to select a particular PMMU to translate a page address. For example, if one PMMU is assigned only pages for logical addresses whose bit number 11 is a 0 and another PMMU is assigned those whose bit number 11 is a 1, then the state of output line AD_{11} can be used to select the appropriate PMMU to translate a logical address.

Example 2. Another way of using Chip Enable separates program pages from data pages. One PMMU is associated with translating addresses generated during instruction fetches (status codes 1100 and 1101) and the other with addresses generated during data fetches (status codes 1000, 1001, 1010, 1011, and 1111). Chip Enable for each PMMU is obtained from the status code (i.e., status lines ST_0–ST_3).

Example 3. Several PMMUs can be used to implement multiple translation tables. Multiple tables 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 PMMUs through software.

Example 4. A final method uses two translation tables to separate system from normal memory. The MPT and NMS flags in the Mode register can be used in conjunction with the $N/\overline{S}$ line to select the PMMU that contains the appropriate table.

CHANGING PAGE SIZE

The PMMU directly supports pages of 2048 bytes in length. However, the addition of external circuitry enables the PMMU to support systems with larger or smaller pages. The following examples illustrate the technique for changing the supported page size.

Example 1. To implement 4096-byte pages, address bit AD_{11} is not used in the translation process but is used directly as the most significant bit of the address location within a selected 4096-byte page. This can be achieved by doing the following: (1) set the AD_{11} input to be equal to the logical OR of the ST_3 and AD_{11} output lines of the VMPU, (2) require that the least significant bit in the logical address field of each descriptor register be set to 1, and (3) use the logical address bit AD_{11} output by the VMPU instead of the physical address A_{11} output by the PMMU. The AD_{11} input must be 1 during address translation but must equal the AD_{11} output by the CPU during command cycles to the PMMU; ST_3 is used to distinguish the two types of transactions.

Example 2. To implement 1024-byte pages an additional address bit must be translated. Two PMMUs are used. The $\overline{CE}$ input is driven by AD_{10} for one PMMU and its complement, $\overline{AD}_{10}$, for the other.

DMA OPERATION

At the start of a DMA cycle, DMASYNC must go Low whether or not the PMMU is used to translate DMA addresses, to indicate the beginning of a DMA cycle. When DMASYNC has been Low for two cycles, the PMMU assumes that a DMA device has control of the bus. A Low on DMASYNC inhibits the PMMU 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 the clock and then be Low by the next falling clock edge. The PMMU then performs its address translation and access protection functions during the clock period begun in this DMASYNC pulse. Upon the release of the bus at the termination of the DMA cycle, the DMASYNC line must go High. After two clock cycles of DMASYNC High, the PMMU assumes that the VMPU has control of the bus and that subsequent memory references are VMPU accesses. The first memory reference occurs at least two cycles after the VMPU regains control of the bus. During VMPU cycles, DMASYNC should remain High. Refer to the paragraph "Memory Read and Write" and Figure 8 for further information.

Direct memory access (DMA) operations can occur between instruction cycles and can be handled through the PMMU. The PMMU permits DMA in either the System or Normal mode of operation. For each memory access, the page attributes are checked, and if a violation is detected, Suppress is activated. 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. Neither a Trap Request nor an Abort Request is ever generated during a DMA operation, therefore warning conditions are not signaled.

PMMU COMMANDS

The various registers in the PMMU can be read and written using VMPU Special I/O commands. The machine cycles of these commands cause the status lines to indicate that a special input/output operation is in progress. During these machine cycles, the PMMU enters the command mode. In this mode, the rising edge of $\overline{AS}$ indicates that a command is present on lines AD_8–AD_{15}. If this command indicates that data is to be written into one of the PMMU registers, the data is read from lines AD_8–AD_{15} while $\overline{DS}$ is Low. If the command indicates that data is to be read from one of the PMMU registers, the data is placed on lines AD_8–AD_{15} while $\overline{DS}$ is Low.

There are five commands that read or write various fields in the page descriptor register. The status of the read/write line indicates whether the command is a read or a write.

The autoincrementing feature of the Descriptor Address Register (DAR) can be used to block load page descriptors using the repeat forms of the Special I/O instructions. The DAR is autoincremented at the end of the field. The command accessing the entire page descriptor register references the fields in the order of logical address, physical address, and attribute; four bytes are written in succession.

Table 1 gives the five commands that are used to write data into descriptor fields.

Table 1. Descriptor Field Write Commands

Opcode (Hex)	Instruction
OA	Read/Write Attribute field
OB	Read/Write Descriptor (all fields)
OE	Read/Write Attribute field; increment DAR
OF	Read/Write Descriptor (all fields); increment DAR
15	Reset all Valid Attribute flags

Table 2 gives the three commands that are used to read and write the control registers.

Table 2. Control Registers' Read/Write Commands

Opcode (Hex)	Instruction
00	Read/Write Mode register
01	Read/Write Descriptor Address register
20	Read/Write Descriptor Selector Count register

The status registers are read-only registers, although the Trap Type Register (TTR) can be reset. Twelve instructions, shown in Table 3, access these registers.

Table 3. Status Registers' Access Instructions

Opcode (Hex)	Instruction
02	Read Trap 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 Trap Type register
13	Reset SWW flag in VTR
14	Reset FATL flag in VTR
21	Read Violation Offset (low-byte) register
22	Read Read/Write Data Counter register
23	Read Instruction Offset (low-byte) register

USE OF THE PMMU WITH OTHER Z8000 CPUs

The PMMU is designed to operate in conjunction with the Z8003 VMPU; however, it can also be used with other CPUs in the Z8000 Family. The following examples suggest simple system configurations; more sophisticated arrangements are possible.

The Z8004 VMPU generates nonsegmented 16-bit addresses only. The PMMU can be used to implement a paged virtual memory by tying the segment number inputs of the PMMU to 0 and requiring the most significant seven bits of the logical address field to be 0. Since the Z8004 VMPU lacks a translation trap request input pin, the nonmaskable (or other interrupt request) pin should be used instead.

The PMMU extends the physical addressing capability of the Z8004 without using the segmentation mechanism of the Z8003. This use is similar to the way in which 16-bit minicomputers extend their addressing capability.

The Z8001 and Z8002 CPUs do not support the instruction abort mechanism. A page fault for one of these CPUs is, in general, non-recoverable, since during an interrupt, the current instruction runs to completion, possibly overwriting CPU registers. For the nonsegmented Z8002, this means that all pages that the CPU can access must be in physical memory and appropriate information must be in the PMMU page descriptor registers. For the segmented Z8001, this means that programs must explicitly request segments before accessing them and must free segments after use. It also means that segments are allocated in units of the page size and that limit protection is performed with this granularity. Use of the PMMU with the Z8001 and Z8002 CPUs permits a paged allocation of main memory and extends the physical address capability of the Z8002.

PMMU TIMING

The PMMU translates addresses and checks for access violations by stepping through sequences of basic clock cycles corresponding to the cycle structure of the VMPU. Timing diagrams that show the relative timing relationships of PMMU signals during the basic operations of memory read/write and PMMU control commands are given in this section.

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 before the address offset; a High on DMASYNC during T3 indicates that the segment offset data is valid. The address offsets 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 $\overline{AS}$. Status, mode, and chip enable information becomes valid early in the memory access cycle and must remain stable throughout. The most significant 16 bits of the address (physical memory location) remain valid until the end of T3. Abort Request, Trap Request, and Suppress are asserted in T2 (Figure 9). Abort Request is asserted for four clock cycles. Trap Request remains Low until trap acknowledge (status = 0100) is received. Suppress is asserted during the current machine cycle and terminates during T3.

PMMU Command Cycle

During the command cycle of the PMMU (Figure 11), commands are placed on lines AD_8–AD_{15} during T1. The status lines indicate that a Special I/O instruction is in progress, and the $\overline{CS}$ line enables the appropriate PMMU for that command. Data to be written to a register in the PMMU must be valid on lines AD_8–AD_{15} late in T2. Data read from the PMMU is placed on lines AD_8–AD_{15} late in the T_{WA} cycle.

Input/Output and Refresh

Input/output and refresh operations are indicated by codes on status lines ST_0–ST_3. During these operations, the PMMU refrains from any address translation or protection checking. Lines A_8–A_{23} remain 3-stated during these operations.

Reset

The PMMU 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 VMPU special I/O command. A hardware reset clears the Mode register, Trap Type Register, and Descriptor Selection Count register. If the $\overline{CS}$ line is Low, the Master Enable flag in the Mode register is set to 1. All other registers are undefined. After reset, lines AD_0–AD_{15} and A_8–A_{23} are 3-stated. The $\overline{SUP}$ and $\overline{ABORT}$ open-drain outputs are not driven. If the Master Enable flag is not set during reset, the PMMU does not respond to subsequent addresses on its AD lines. To enable a PMMU after a hardware reset, a PMMU command must be used in conjunction with $\overline{CS}$.

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

Abort, Trap Request, and Acknowledge

The PMMU generates a Trap Request whenever it fails to find a page entry corresponding to the logical address (that is, a page fault), detects an access violation, or detects a write into the lowest 128-byte block of a page with the DIRW flag set (Figure 12). In the case of an access violation or page fault, the PMMU also activates Suppress and Abort Request. The Suppress signal is used by memory to inhibit memory writes. The Trap Request remains Low until a trap acknowledge signal (status = 0100) is received. Violations detected during DMA cycles cause Suppress to be asserted during that cycle only, but no Trap Request is generated.

When the PMMU issues a Trap Request, it awaits the indication of a trap acknowledge. Subsequent violations occurring before the trap acknowledge indication is received are detected and appropriately processed. During a Trap Acknowledge cycle, the PMMU drives one of its Address/Data lines; the selection of the line is a function of the identification field of the Mode register. After the Trap Request has been acknowledged by the VMPU, the Violation Status register should be read by a special I/O command in order to determine the cause of the trap. The Trap Type register should be reset so that subsequent traps are recorded correctly.

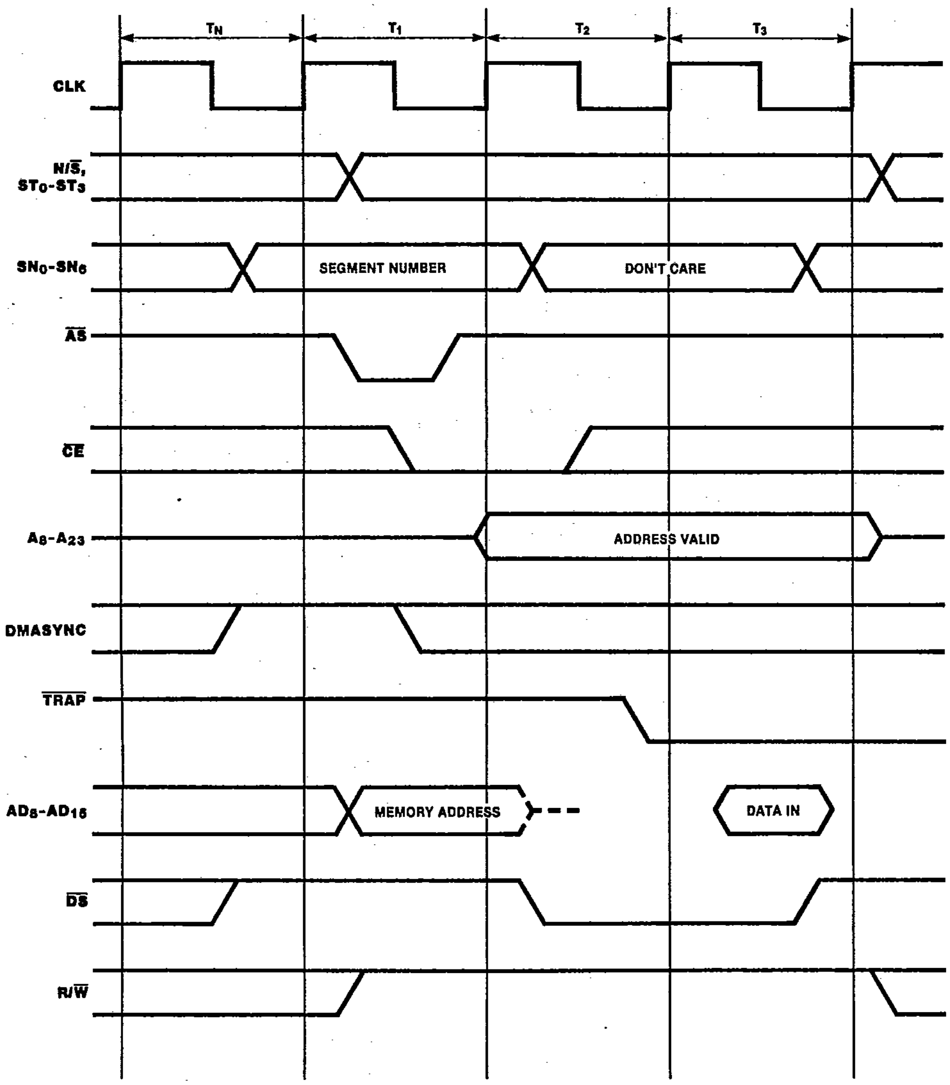


Figure 8. Memory Read Timing

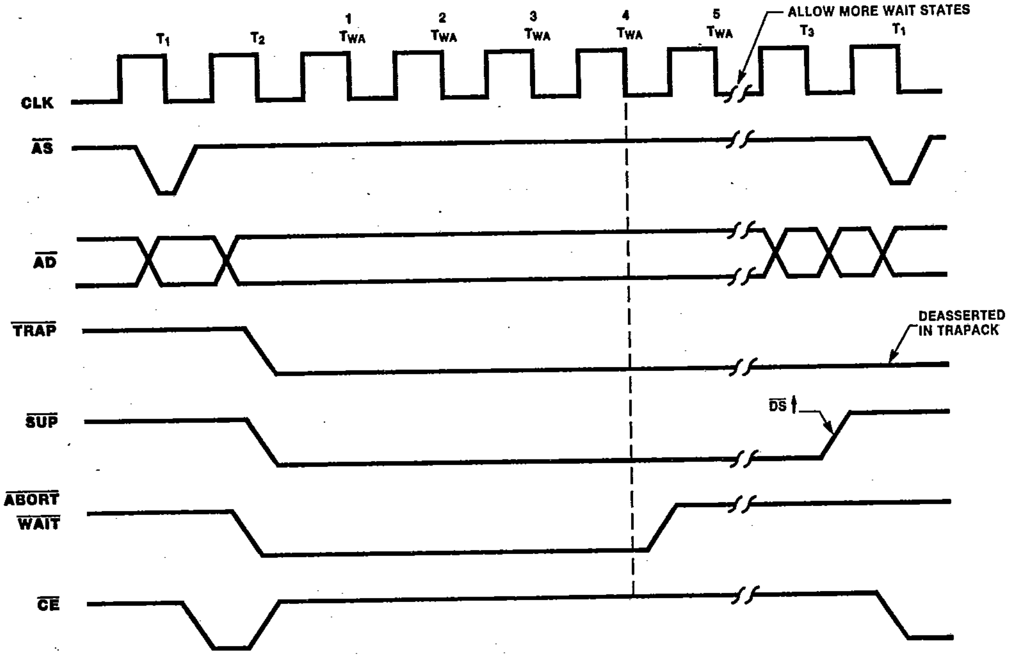


Figure 9. Abort and Trap Request Timing

Z8015 PMMU

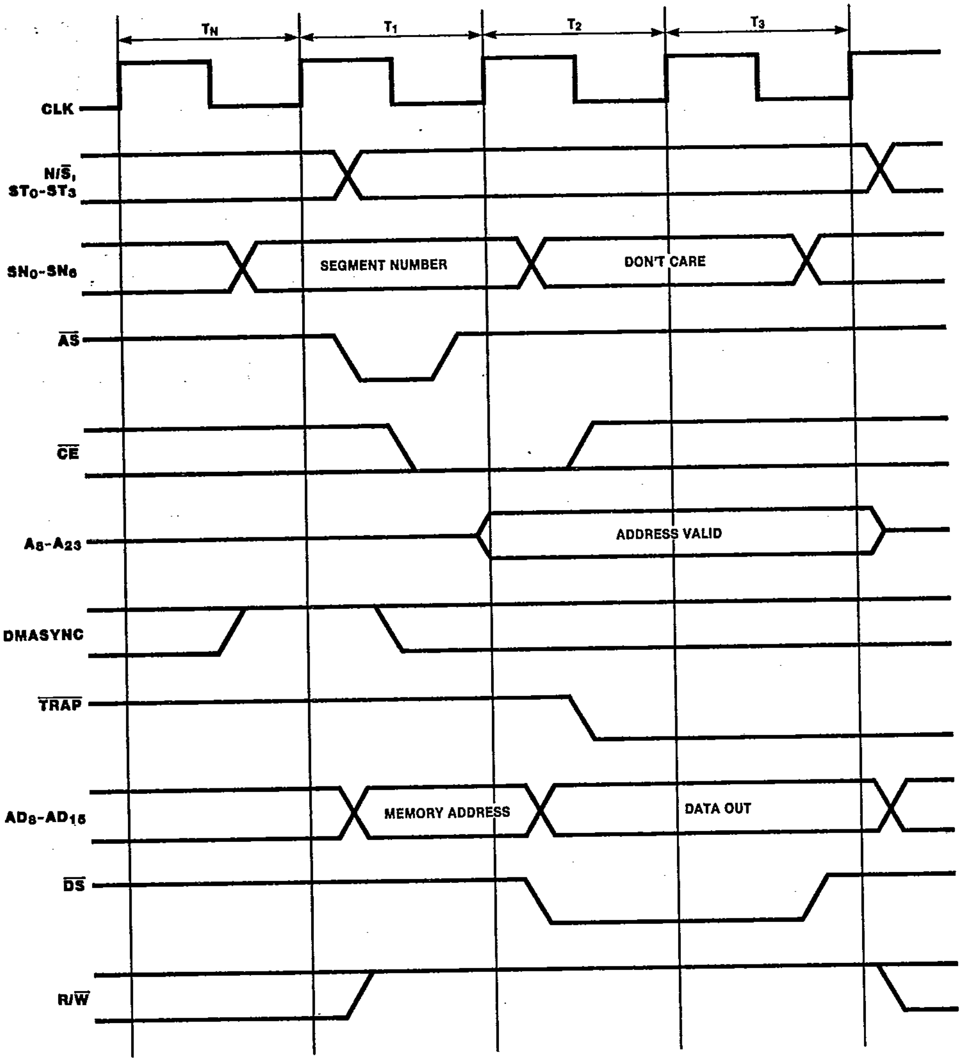


Figure 10. Memory Write Timing

2081-011

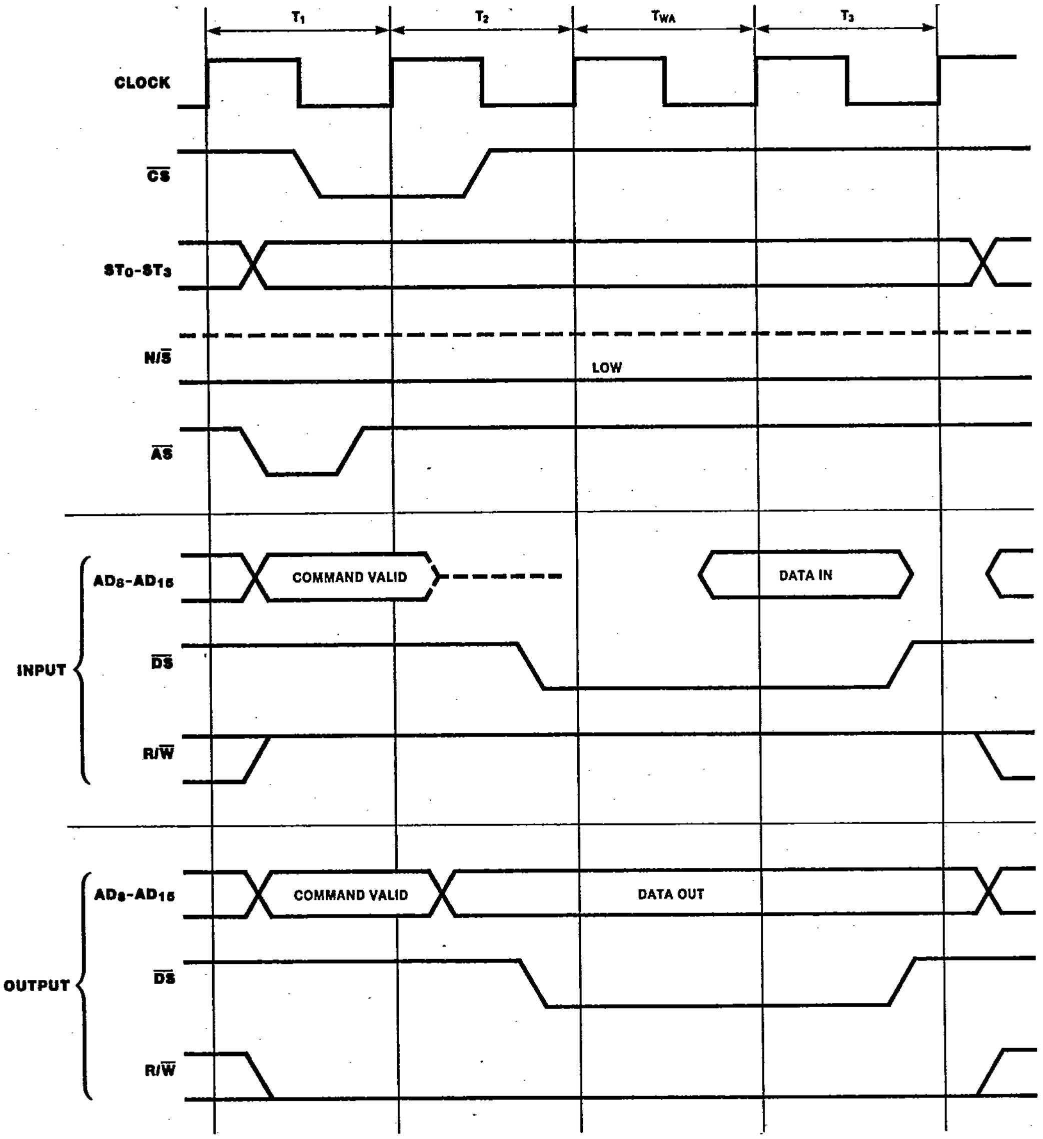


Figure 11. I/O Command Timing

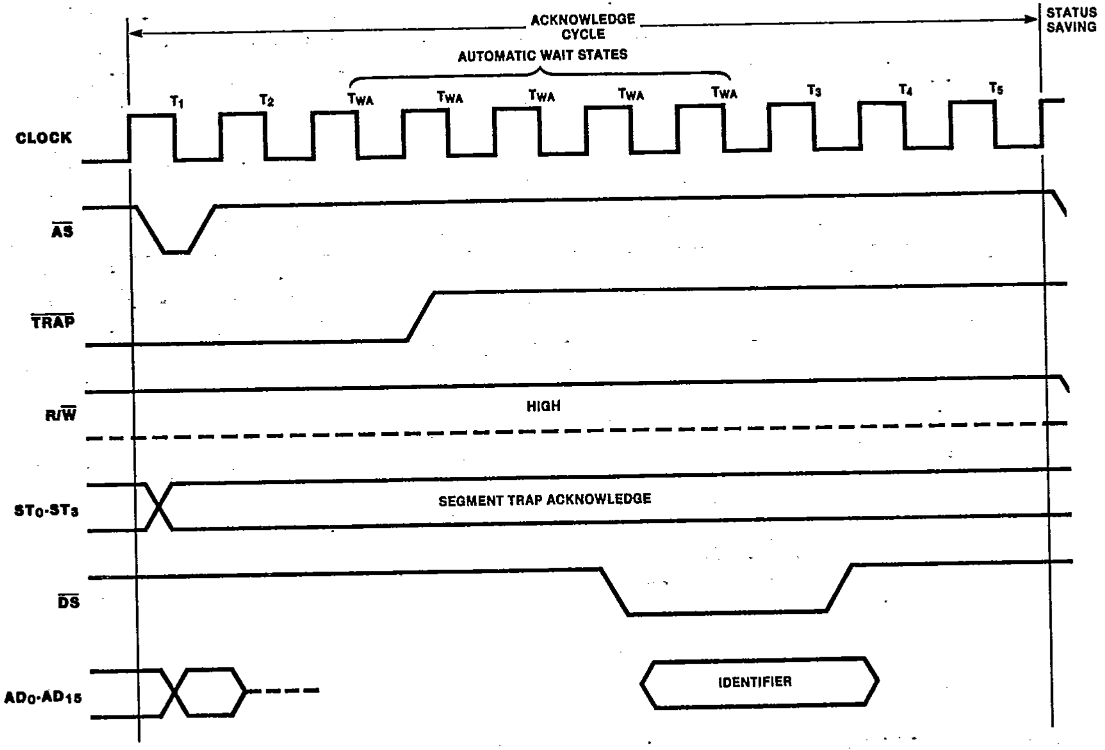


Figure 12. Trap Request and Acknowledge Timing

SIGNAL DESCRIPTIONS

The Z8015 is produced in 64-pin and 68-pin packages; the functions performed by the device's input and/or output pins are shown in Figure 13. Pin/signal name assignments are shown in Figure 14.

A_8–A_{23}. *Address Bus* (outputs, active High, 3-state). These address lines are the 16 most significant bits of the physical memory location.

$\overline{\text{ABORT}}$. *Abort Request* (output, active Low, open drain). A Low on this line indicates MMU requests for an instruction abort. This line is enabled when a page fault or access violation is detected.

AD_0–AD_{15}. *Address/Data Bus* (inputs/outputs, active High, 3-state). AD_0–AD_7 are used for addresses and inputs only. They carry the low-order byte in the offset of logical addresses intended for translation. AD_8–AD_{15} are multiplexed address and data lines that are used both for the eight most significant bits of the logical address and for commands.

$\overline{\text{AS}}$. *Address Strobe* (input, active Low). The rising edge of $\overline{\text{AS}}$ indicates that lines AD_0–AD_{15}, ST_0–ST_3, $R/\overline{W}$, $\overline{CE}$, and $N/\overline{S}$ are valid.

$\overline{\text{CE}}$. *Chip Enable* (input, active Low). This line selects a PMMU to translate a logical address.

CLK. *System Clock* (input). CLK is a +5V single-phase, time-base input used for both the VMPU and PMMU.

$\overline{\text{CS}}$. *Chip Select* (input, active Low). This line selects a PMMU for a control command.

DMASYNC. *DMA/Segment Number Synchronization Strobe* (input, active High). A Low on this line indicates a DMA access is occurring; a High indicates the segment number is valid. It must be High during VMPU cycles and Low when SN lines are 3-stated.

$\overline{\text{DS}}$. *Data Strobe* (input, active Low). This line provides timing for the data transfer between the PMMU and the Z8003 VMPU.

$N/\overline{S}$. *Normal/System Mode* (input, Low = System mode). $N/\overline{S}$ indicates to the PMMU that the VMPU or DMA is in the Normal or System mode.

$\overline{RESET}$. *Reset* (input, active Low). A Low on this line resets the PMMU.

$R/\overline{W}$. *Read/Write* (input, Low = write). $R/\overline{W}$ indicates whether the VMPU is reading from or writing to either memory or the PMMU.

SN_0–SN_6. *Segment Number* (inputs, active High). These lines provide the 7-bit segment number of a logical address.

ST_0–ST_3. *Status* (inputs, active High). These lines (Table 4) specify the status of the associated VMPU.

$\overline{SUP}$. *Suppress* (output, active Low, open-drain). This signal is asserted during the current bus cycle when a page fault or any access violation, except write warning, occurs.

$\overline{TRAP}$. *Trap Request* (output, active Low, open-drain). The PMMU interrupts the VMPU with a Low on this line when the PMMU detects a page fault, access violation, or write warning.

Table 4. Status Lines

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 (for example, to a PMMU)
0 1 0 0	Translation 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 (External Processing Architecture)
1 0 1 1	Stack memory request (External Processing Architecture)
1 1 0 0	Instruction space access
1 1 0 1	Instruction fetch, first word
1 1 1 0	External Processing Unit-CPU transfer
1 1 1 1	Bus lock, data memory request

Z8015 PMMU

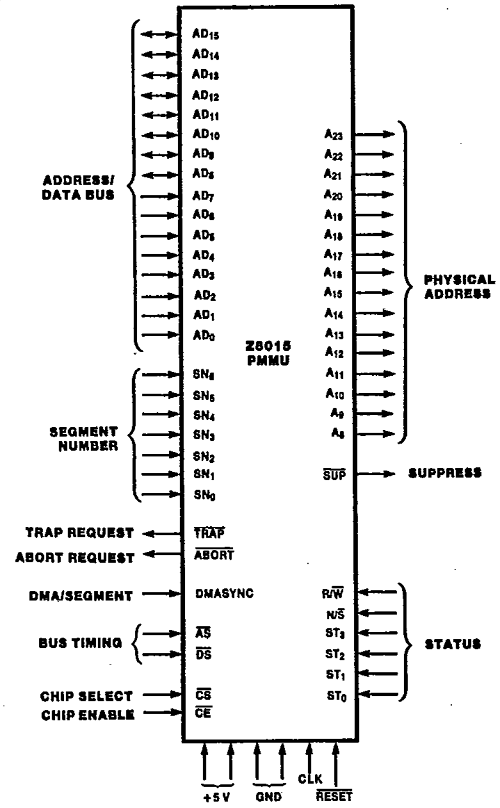


Figure 13. Pin Functions

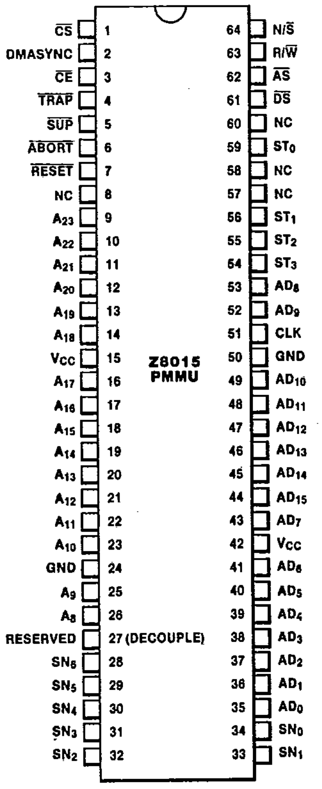


Figure 14a. 64-pin Dual-In-Line (DIP), Pin Assignments

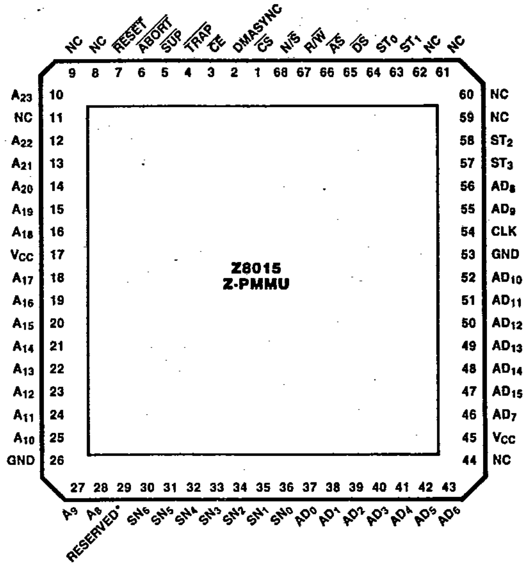


Figure 14b. 68-pin Chip Carrier, Pin Assignments

AC CHARACTERISTICS†

Number	Symbol	Parameters	Min (4 MHz)	Max (4 MHz)	Notes
1	TcC	Clock Cycle Time	250		
2	TwCh	Clock Width (High)	105		
3	TwCl	Clock Width (Low)	105		
4	TfC	Clock Fall Time		20	
5	TrC	Clock Rise Time		20	
6	TdDSA(RDv)	$\overline{\text{DS}}$ ↓ (Acknowledge) to Read Data Valid Delay		100	1
7	TdDSA(RDf)	$\overline{\text{DS}}$ ↑ (Acknowledge) to Read Data Float Delay	20	75	1
8	TdDSR(RDv)	$\overline{\text{DS}}$ ↓ (Read) to AD Output Driven Delay		100	1
9	TdDSR(RDf)	$\overline{\text{DS}}$ ↑ (Read) to Read Data Float Delay	20	75	1
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 AS ↑ Setup Time	45		
14	ThAS(OFFn)	$\overline{\text{AS}}$ ↑ to Offset Not Valid Hold Time	60		
15	TdAS(C)	$\overline{\text{AS}}$ ↓ to CLK ↑ Delay	110		
16	TdDS(AS)	$\overline{\text{DS}}$ ↑ to $\overline{\text{AS}}$ ↓Delay	50		
17	TdAS(DS)	$\overline{\text{AS}}$ ↑ to $\overline{\text{DS}}$ ↑ Delay	50		
18	TsSN(C)	SN Data Valid to CLK ↑ Setup Time	120		

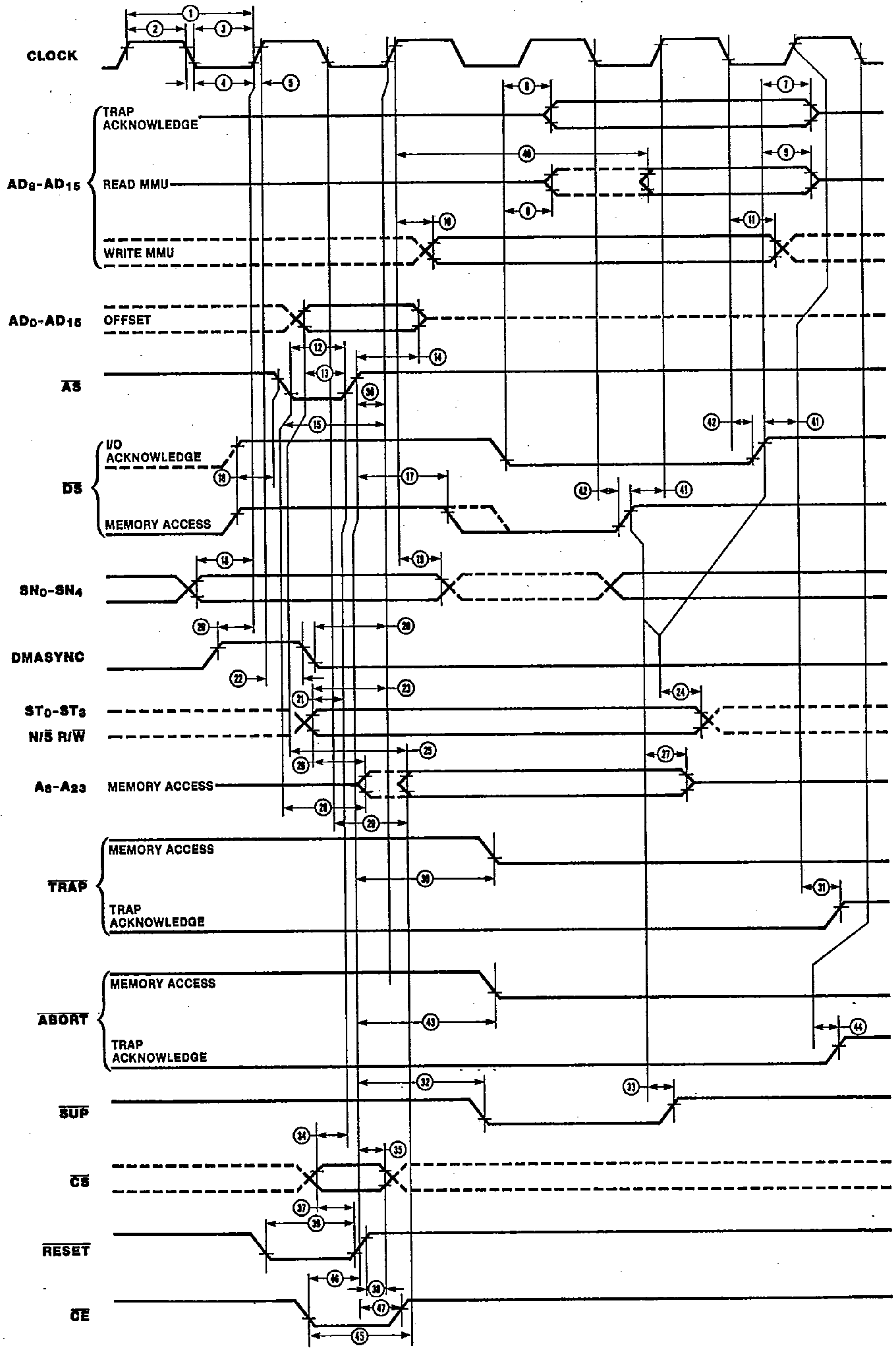
NOTES:
† All times given in nanoseconds (ns).
1. 50 pf load.
2. 2.2K pull-up.

AC CHARACTERISTICS† (Continued)

Number	Symbol	Parameters	Min (4 MHz)	Max (4 MHz)	Notes
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/$\overline{\text{S}}$, R/$\overline{\text{W}}$) Valid to $\overline{\text{AS}}$ ↑ Delay	60		
22	TdC(DMA)	CLK ↑ Delay	20		
23	TdST(C)	Status (ST$_0$–ST$_3$) Valid to CLK ↑ Setup Time	140		
24	TdDS(STn)	$\overline{\text{DS}}$ ↑ to Status Not Valid Delay	0		
25	TdOFF(Av)	Offset Valid to Address Output Valid Delay		200	1
26	TdST(Ad)	Status Valid to Address Output Driven Delay		180	1
27	TdDS(Af)	$\overline{\text{DS}}$ ↑ to Address Output Float Delay	30	160	1
28	TdAS(Ad)	$\overline{\text{AS}}$ ↓ to Address Output Driven Delay		170	1
29	TdC(Av)	CLK ↓ to Address Output Valid Delay		155	1
30	TdAS(TRAP)	$\overline{\text{AS}}$ ↑ to $\overline{\text{TRAP}}$ ↓ Delay		110	1,2
31	TdC(TRAP)	CLK ↑ to $\overline{\text{TRAP}}$ ↑ Delay		300	1,2
32	TdAS(SUP)	$\overline{\text{AS}}$ ↑ to $\overline{\text{SUP}}$ ↓ Delay		115	1,2
33	TdDS(SUP)	$\overline{\text{DS}}$ ↑ to $\overline{\text{SUP}}$ ↑ Delay	30	155	1,2
34	TsCS(AS)	Chip Select Input Valid to $\overline{\text{AS}}$ ↑ Setup Time	10		
35	ThAS(CSn)	$\overline{\text{AS}}$ ↑ to Chip Select Input Not Valid Hold Time	80		
36	TdAS(C)	$\overline{\text{AS}}$ ↑ to CLK ↑ Delay	0		
37	TsCS(RST)	Chip Select Input Valid to $\overline{\text{RESET}}$ ↑ Setup Time	150		
38	ThRST(CSn)	$\overline{\text{RESET}}$ ↑ to Chip Select Input Not Valid Hold Time	0		
39	TwRSTl	$\overline{\text{RESET}}$ Width (Low)	2TcC		
40	TdC(RDv)	CLK ↑ to Read Data Valid Delay		460	
41	TdDS(C)	$\overline{\text{DS}}$ ↑ to CLK ↑ Delay	30		
42	TdC(DS)	CLK ↓ to $\overline{\text{DS}}$ ↑ Delay	0	110	
43	TdAS(ABORT)	$\overline{\text{AS}}$ ↑ to $\overline{\text{ABORT}}$ Delay		110	
44	TdC(ABORT)	CLK ↓ to $\overline{\text{ABORT}}$ Delay	30	155	
45	TdCE(Av)	$\overline{\text{CE}}$ ↓ to Address Output Valid Delay		235	
46	TsCE(AS)	$\overline{\text{CE}}$ ↓ to $\overline{\text{AS}}$ ↑ Setup Time	0		
47	ThAS(CEn)	$\overline{\text{AS}}$ ↑ to Chip Enable Input Not Valid Hold Time	60		

NOTES:
† All times given in nanoseconds (ns).
1. 50 pf load.
2. 2.2K pull-up.

AC TIMING DIAGRAM



2081-017

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.

The Ordering Information section lists package temperature ranges and product numbers. Package drawings are in the Package Information section. Refer to the Literature List for additional documentation.

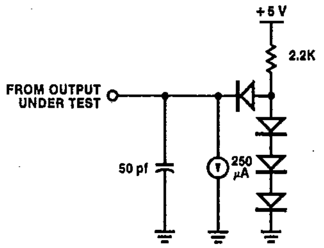
STANDARD TEST CONDITIONS

The following dc characteristics apply for the given standard test conditions unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin.

Standard conditions are as follows:

- $+4.75V \leq V_{CC} \leq +5.25V$
- GND = 0V
- $0°C \leq TA \leq +70°C$

The type of test circuit used is as follows:



DC CHARACTERISTICS

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

ORDERING INFORMATION

Z8015 Z-PMMU, 4.0 MHz		Z8015A Z-PMMU, 6.0 MHz	
64-pin DIP	**68-pin PCC**	**64-pin DIP**	**68-pin PCC**
Z8015 CS	Z8015 VS†	Z8015A CS	Z8015A VS†
Z8015 CE	Z8015 VE†	Z8015A CE	Z8015A VE†

Codes

First letter is for package; second letter is for temperature.

C = Ceramic DIP
P = Plastic DIP
L = Ceramic LCC
V = Plastic PCC

R = Protopack
T = Low Profile Protopack
DIP = Dual-In-Line Package
LCC = Leadless Chip Carrier
PCC = Plastic Chip Carrier (Leaded)

TEMPERATURE
S = 0°C to +70°C
E = −40°C to +85°C
M*= −55°C to +125°C

FLOW
B = 883 Class B

Example: PS is a plastic DIP, 0°C to +70°C.

†Available soon.
*For Military Orders, contact your local Zilog Sales Office for Military Electrical Specifications.