Z8010 MMU

LH8010

Memory Management Unit

Features

- Dynamic segment relocation makes software addresses independent of physical memory addresses.
- Sophisticated access validation protects memory areas from unauthorized or unintentional access.
- MMU architecture supports multiprogramming systems.
- Sixty-four variable-sized segments from 256 to 64K bytes can be managed within a total physical address space of 16M bytes; all 64 segments are randomly accessible.
- Multiple MMUs can support several translation tables for each of the six Z8001 address spaces.

Description

Declining memory costs coupled with the increasing power of microprocessors has accelerated the use of high-level languages, sophisticated operating systems, complex programs and large data bases in microcomputer systems. The Z8001 microprocessor CPU supports these trends with an eight megabyte direct address space as well as a rich and powerful instruction set. The Z8010 Memory Management Unit (MMU) provides flexible and efficient support for this large address space by offering dynamic segment relocation as well as numerous memory-protection features.

The primary memory of a computer is one of its major resources. As such, the management of this resource becomes a major concern as demands on it increase. These demands arise from multiple users (or multiple tasks within a dedicated application), the need to increase system integrity by limiting access

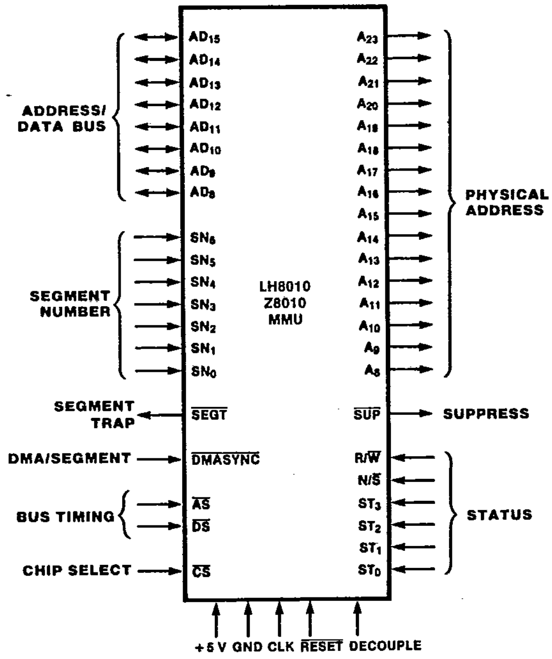

Figure 1. Pin Functions

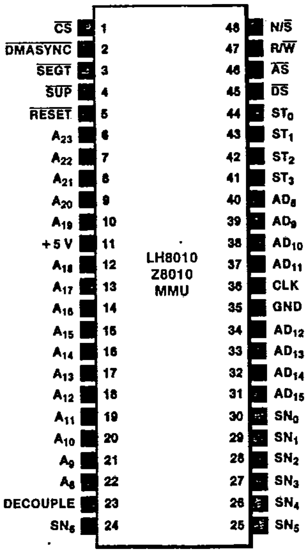

Figure 2. Pin Assignments

Description (Continued)

to various portions of the memory, and from the need to structure large, complex programs and systems.

Multiple tasks (or users) of a system that can reside anywhere in memory are called *relocatable*. Generally, systems in which all tasks are relocatable offer far greater flexibility in responding to changing system environments. Another aspect of multiple-task environments is sharing: separate tasks can execute the same program on different data, or several tasks may execute different programs using the same data.

Unfortunately, a problem that arises in multiple-task systems is that of system integrity. Tasks must be protected from unwanted interactions with other tasks; user tasks must be prohibited from performing operating system functions; and user tasks must also be protected from themselves so they cannot overflow the areas allotted to them.

In addition to these considerations, support for the design and implementation of large, complex programs and systems is itself an important consideration. Modern trends are toward the partitioning of a complex task into small, simple, self-contained subtasks that have well-defined interfaces. Because these subtasks interact with each other, communication between them must be carefully controlled. Memory-management systems can offer effective solutions for implementing large systems modularly designed.

The Z8010 Memory Management Unit supports multiple-process and large modular software systems with dynamic segment relocation. Futhermore, it enhances system integrity with a powerful set of memory protection features.

Relocation. 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 and providing a flexible, efficient method for supporting multi-programming systems.

The Z-MMU uses a translation table to transform the 23-bit logical addresses from the Z8001 CPU into 24-bit addresses for the physical memory. Memory segments are variable in size from 256 bytes to 64K, in increments of 256 bytes. Pairs of Z-MMUs support the 128 segment numbers available for the various Z8001 CPU address spaces. Within an address space, any number of Z-MMUs can be used to accommodate multiple translation tables for system and normal operating modes, or to support more sophisticated memory-management systems.

System Integrity. Z-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 "personality" consisting of several attributes when it is initially entered into the Z-MMU. When a memory reference is made, these attributes are checked against the status information supplied by the Z8001 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 and take appropriate action to correct the problem.

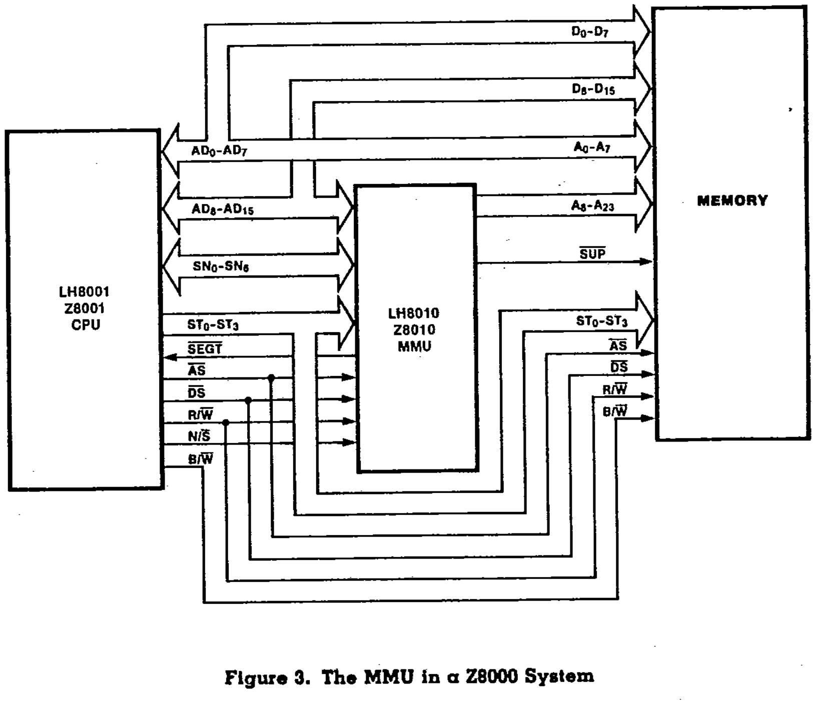

Figure 3. The MMU in a Z8000 System

Pin Description

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.

AD_8-AD_{15}. *Address/Data Bus* (inputs/outputs, active High, 3-state). These multiplexed address and data lines are used both for commands and for logical addresses intended for translation.

$\overline{AS}$. *Address Strobe* (input, active Low). The rising edge of $\overline{AS}$ indicates that AD_8-AD_{15}, ST_0-ST_3, CS, R/W and N/S are valid.

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

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

DMASYNC. *DMA/Segment Number Synchronization Strobe* (input, active High). A Low on this line indicates that a DMA access is occurring; a High indicates that the segment number is valid. It must always be High during CPU cycles.

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

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

Reserved. Do not connect.

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

R/$\overline{W}$. *Read/Write* (input, Low = write). R/$\overline{W}$ indicates the Z8001 CPU or Z8016 DMA is reading from or writing to memory or the MMU.

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

SN_0-SN_6. *Segment Number* (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.

ST_0-ST_3. *Status* (inputs, active High). These lines specify the Z8001 CPU 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	Status memory request (EPU)
1 1 0 0	Instruction space access
1 1 0 1	Instruction fetch, first word
1 1 1 0	Extension processor transfer
1 1 1 1	Reserved

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