MICROCOMPUTERS

68000-based supermicro uses distributed architecture

RON DeJOUNG, Onyx Systems Inc.

Intelligent I/O and memory management boost multi-user performance

Microcomputers with 16-bit architectures that operate with sophisticated multi-user operating systems such as UNIX need an internal design that can handle faster processing, rapid switching between tasks and strenuous memory-access requirements without degrading performance. An I/O channel-intensive architecture is one approach to these requirements, offloading system housekeeping and I/O manipulation from the main CPU.

One such "turbocharged" architecture is based on intelligent I/O channels and memory management. Onyx Systems Inc.'s 68000 system improves system performance by strengthening CPU-to-memory, CPU-to-I/O and CPU-to-disk throughput.

CPU to memory

CPU-to-memory data transfers in the Onyx 68000 system (Fig. 1) are aided by both the cache memory, which makes frequently requested data more accessible to the processor, and the memory-management unit, which maps data into and out of memory locations and divides main memory among simultaneous users.

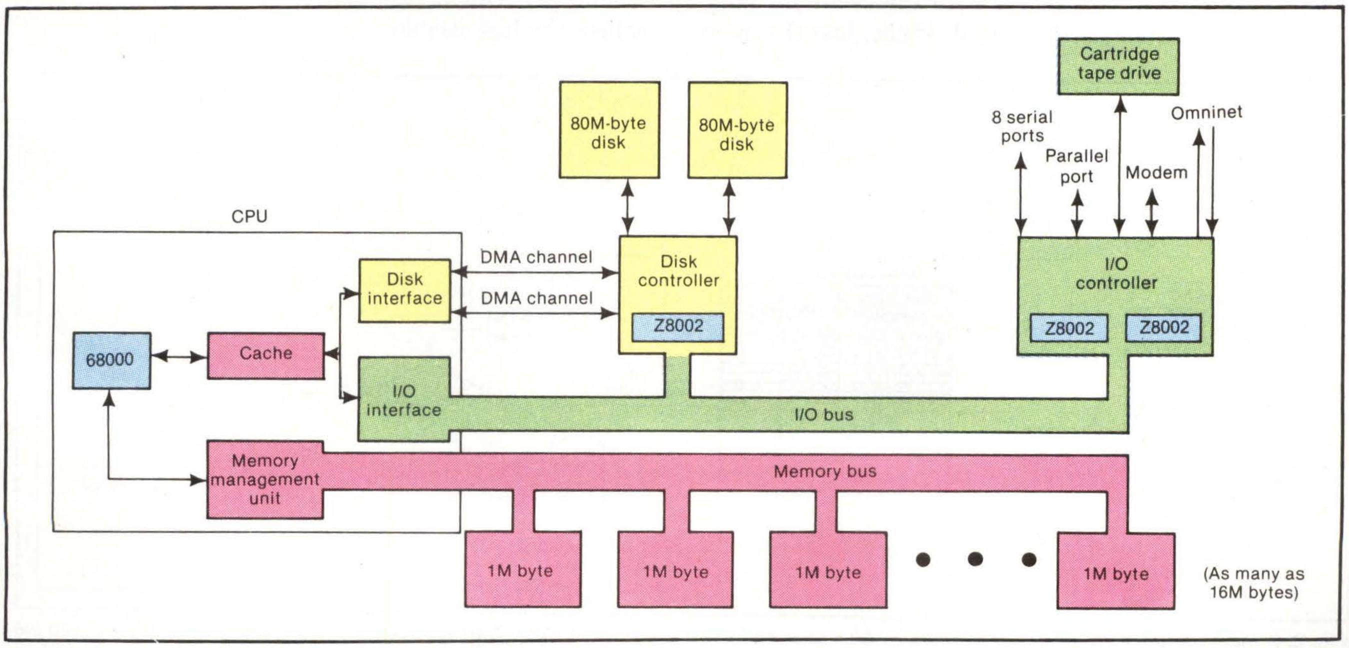

Fig. 1. Channel-intensive design. *The main CPU board contains a 12.5-MHz 68000 processor, a memory-management unit that governs memory accesses to cache and main memory, two intelligent disk channels with DMA and an I/O bus interface. The memory bus interface connects as much as 16M bytes of main memory with error-correcting code. Each of as many as four 8-in. Winchester disk drives has a dedicated 16-bit Z8002 control processor to optimize throughput. The I/O processor card houses dual Z8002A CPUs for support of as many as eight serial ports, a modem port, a parallel port and a cartridge-tape drive with as much as 12M bytes of storage.*

Cache. The Onyx 68000 system speeds memory access by using high-speed static RAM in an on-board cache memory residing between the MMU and the main memory. Main memory uses less costly dynamic RAM.

The cache memory is organized as a 1K- × 48-bit array. For each entry in cache, there are three separate fields: address, data and valid bit. When the processor or DMA master controller references main memory, it outputs a 24-bit virtual address and a 4-bit function code that specifies what type of memory access is desired, such as read, write, interrupt or supervisory (Fig. 3). The MMU compares bits 1 through 12 of the address against the address field contained in the selected cache entry. If they match, and the valid bit is set, a cache "hit" occurs, and the appropriate bytes from the data field of the cache are sent to the CPU. This cycle can be completed in less than 160 nsec.

If the desired data are not in the cache memory, a main memory cycle is initiated. When the data are ready, the cache address, data and valid bit are updated. In addition to the referenced word being loaded into cache, the 16-bit word with the next highest address is also loaded. This prefetch of the next word is useful because, during most sequential operations, the next program instruction is located in the next sequential address. This scheme considerably improves the hit ratio of the cache.

The cache is designed to operate as a write-through cache. The CPU checks the cache when writing to main memory; if a cache hit occurs, both the cache memory and the main memory are updated. If not, just the main memory is updated.

Memory management. The MMU maps the 1M byte to 16M bytes of main memory into as many as 16 segments. Each segment can be dedicated to one or more user processes and can be swapped quickly back and forth between memory and the disk. Main memory segments can be partitioned among as many as 31 concurrent user processes. Segments are allocated contiguous virtual address space even if contiguous physical memory is not available, and can be moved about the virtual memory space in 4K-word blocks.

To access main memory, the MMU looks at the first 4 bits of the 24-bit virtual address (Fig. 2). These 4 bits are used by a file called the "map RAM" to indicate which of the 16 segments are to be accessed. The second 8 bits in the virtual address are used by a "base address table" to specify one of as many as 256 pages comprising each segment. The last 12 bits of the virtual address select the appropriate word out of the 4K words in each page.

An 8K-byte block of the I/O processor memory can be selected to be in the 68000 address space.

If a segment were as large as its address space permitted, it would occupy 1M byte of physical memory. This would mean that even the smallest systems would require 4M bytes of memory to support a single user because the user and the system each require data and code segments. To avoid this excessive memory requirement, segment size can be limited by a file that specifies the last usable page in each segment.

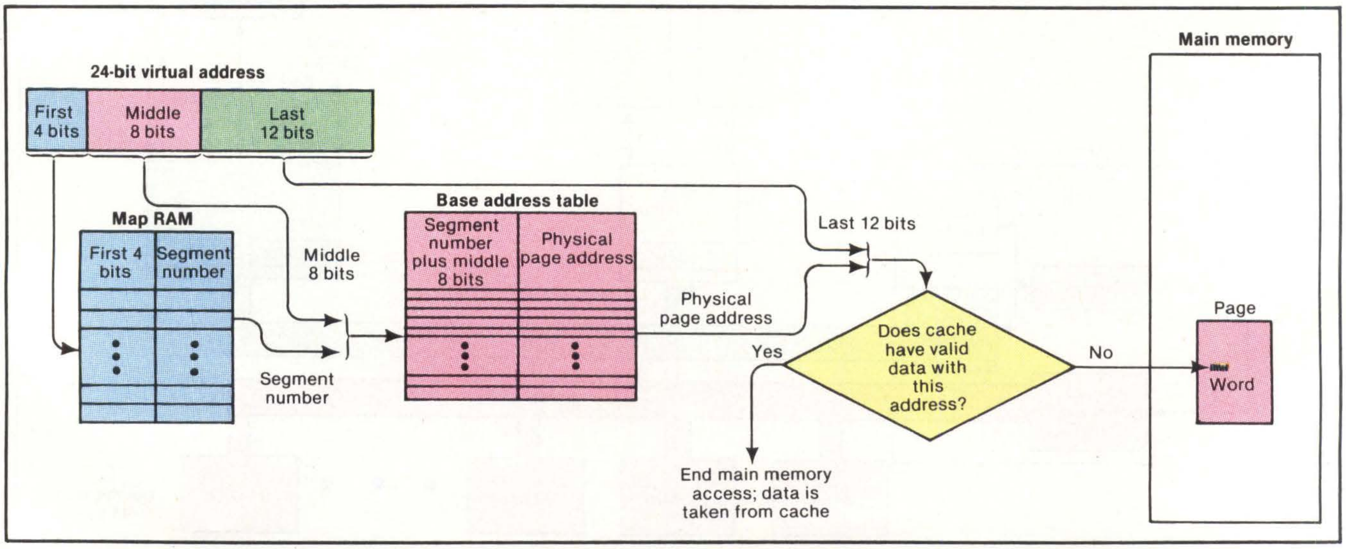

Fig. 2. A main-memory access *begins with a 24-bit virtual address issued by the CPU. The first 4 bits specifies a location in a file called the map RAM, which identifies one of 16 memory segments. This segment number is used with the next 8 bits of the virtual address to specify a location in a file called the base address table. The base address table returns the physical memory address of a 4K-word page within the selected segment (each segment comprises as many as 256 pages). The memory-management unit then checks to see if the required data are in cache. If they are not, then the last 12 bits of virtual address are combined with the page address to locate the required word in main memory. If a cache hit occurs, the data are taken from cache, and there is no main-memory access.*

CPU to I/O

The I/O interface (Fig. 3) is controlled by two Z8002 CPUs and associated control circuitry. The interface is mated to the main 68000 bus via control, address and data lines. The I/O bus interface is a time-multiplexed bus consisting of 17 address lines and 16 bidirectional data lines. A subset of 68000 control lines is included so that the main 68000 CPU can selectively access as much as 128K bytes of I/O processor shared memory. A DMA channel for I/O transfers results in a memory bus usage of 23.4 percent and a data rate of 900K bytes per sec.

One of the Z8002s (CPU/A) controls the serial and parallel ports, and the other (CPU/B) controls the network and tape backup. Each Z8002 runs at 6 MHz and has 8K bytes of EPROM, 4K bytes of static RAM, 128K bytes of dynamic RAM and the I/O bus interface. A mapper allows each Z8002 to address more than the 64K bytes of RAM it normally can address. The bottom 32K bytes of the address space is the same as the physical memory. The mapper selects one of four 32K-byte segments when addressing the top portion of memory.

An external bank-select register and a communication register permit inter-CPU communications and control of the I/O memory. Both can be written and read by the 68000 or either Z8002. Using the bank-select register, an 8K-byte block of the I/O processor memory can be selected to be in the 68000 address space. The communications register has control bits that can be set to allow either processor to interrupt the other, to allow the 68000 to reset either Z8002 and to enable or disable parity checking on 68000 memory reads.

Memory segments can be dedicated to one or more user processes, and can be swapped quickly back and forth between memory and disk.

The two control registers are addressed as memory-mapped I/O ports in the 68000 and as I/O address space in the Z8002 I/O space. Rather than use two addresses out of the 8K-byte shared memory space, the hardware registers occupy part of another 8K-byte page. Thus, each Z8002 processor uses 16K bytes of the 128K-byte

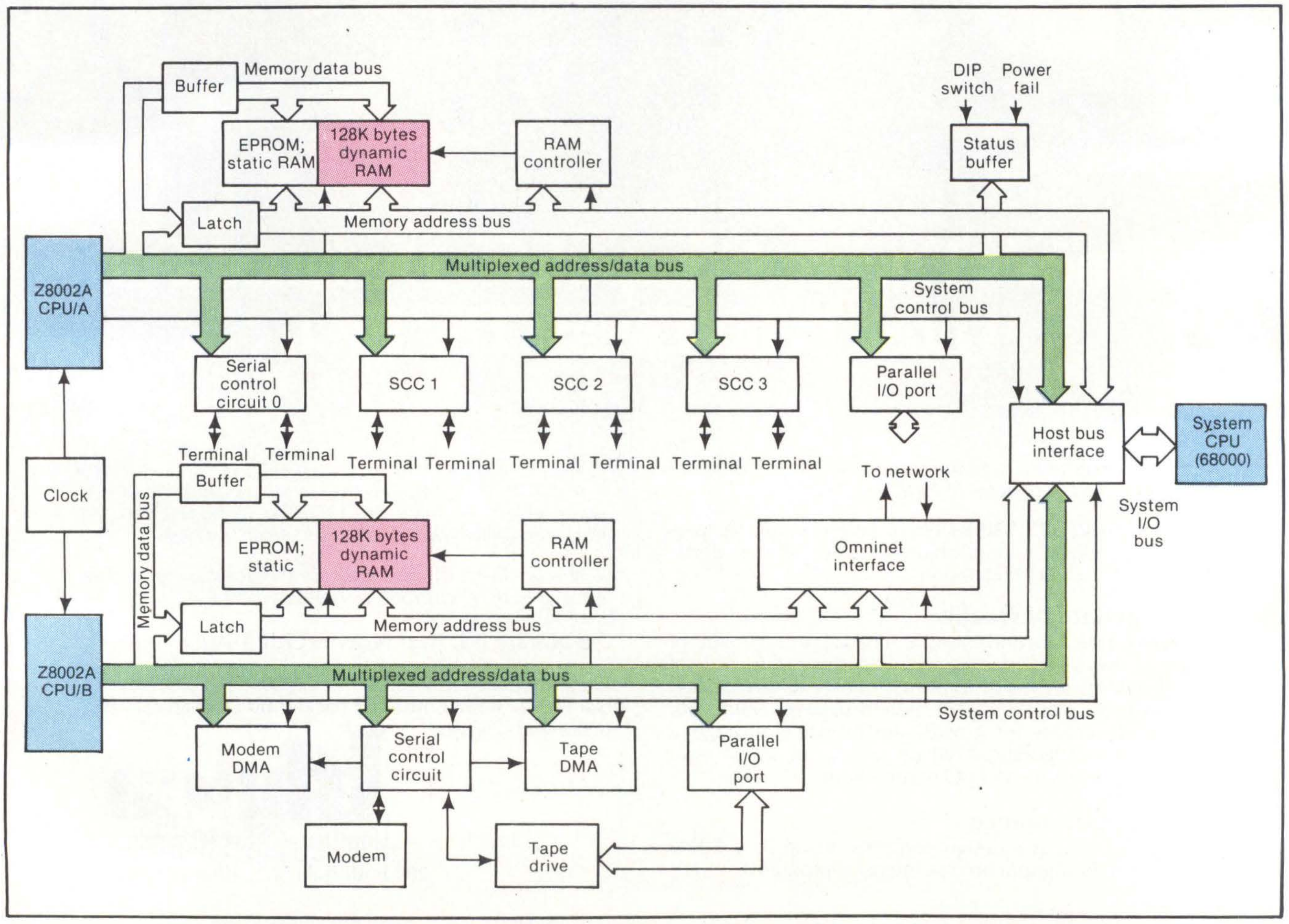

Fig. 3. The I/O interface *is based on two Z8002 CPUs, one (CPU/A) controlling eight serial ports and one parallel port, and the other (CPU/B) controlling modem and network interfaces. Each CPU has 128K bytes of local RAM, which is also accessible by the system CPU.*

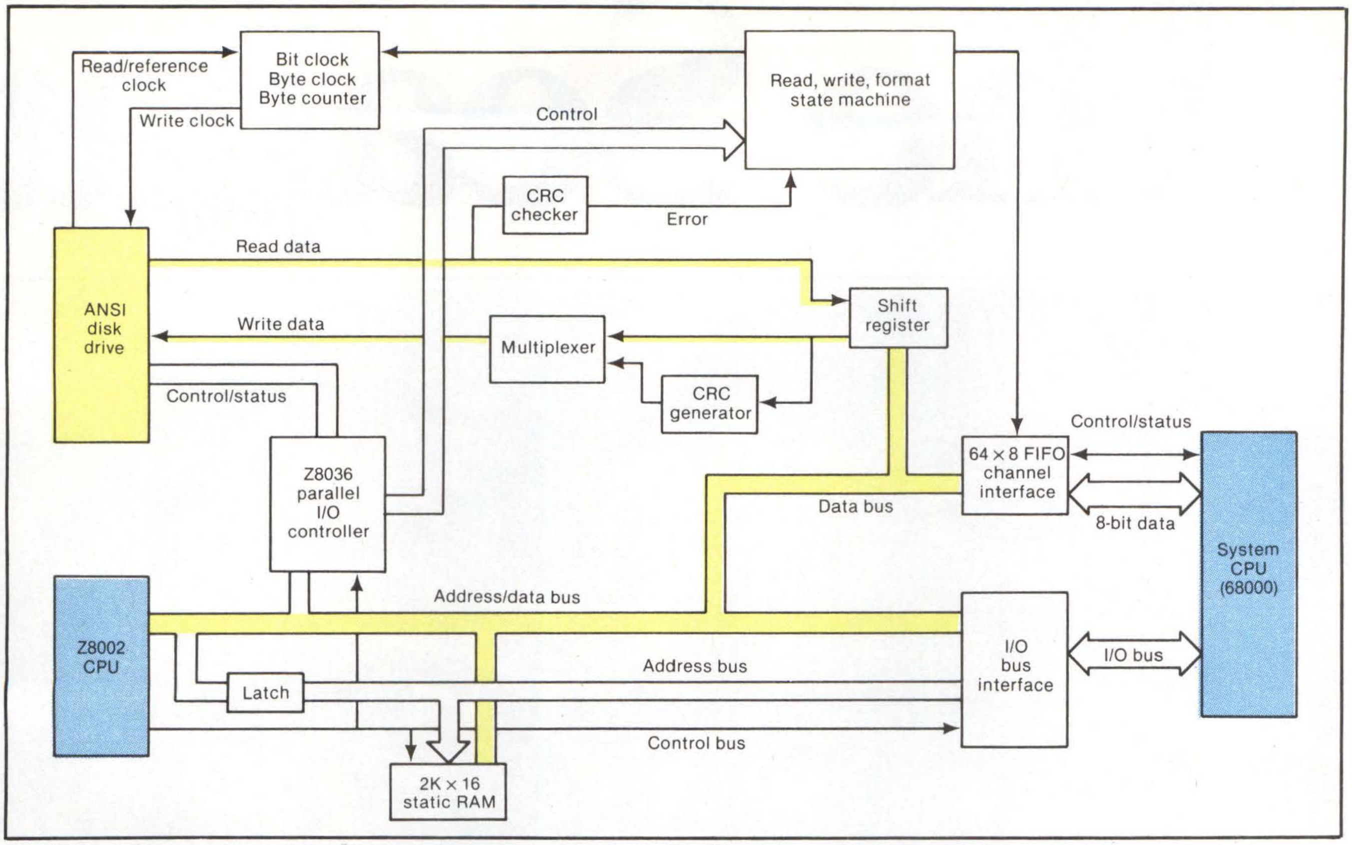

Fig. 4. The disk interface *incorporates a Z8002 CPU that receives commands and operating software from the system CPU. Data are transferred from the disk to the FIFO channel interface buffer in bytes, and then sent to the memory in a 32-bit DMA transfer controlled by the system CPU.*

I/O addressing range of the 68000. A DIP switch on the I/O board determines where the 16K-byte page will be located in the 68000 I/O space.

When the 68000 accesses the I/O bus, the Z8002 that is being addressed is bus requested. Once it has relinquished the bus, the 68000 will be allowed to read or write memory or the hardware registers.

CPU/A has four dual serial communications controllers, counter/timer circuits, parallel I/O and a calendar with battery backup. Each of the eight serial ports has its own programmable baud-rate generator. CPU/B has one dual serial communications controller, counter/timer circuits, two DMA chips for controlling the cartridge-tape drive and the modem port and a local network interface.

Interface

Like the I/O controllers, the disk controller (Fig. 4) uses a Z8002. The disk interface board also contains 4K bytes of static RAM, an I/O bus interface, a CPU DMA channel interface and the ANSI X3T9 disk interface.

Operating software and commands are sent to the Z8002 from the 68000 via the I/O bus. When a data transfer is to occur, the 68000 sets up its DMA channel and then signals the Z8002. The Z8002 then activates the disk control logic, and data are transferred through the DMA channel.

The disk I/O bus interface is similar to that of the I/O processor boards in that the 68000 CPU can access the local 4K-byte memory and all of the I/O ports. The Z8002 is held in a reset state while the 68000 is down-loading or controlling the hardware registers. These registers are memory mapped so that the 68000 can easily access them.

At a disk transfer rate of 10 MHz, data arrive at the FIFO buffer at a rate of 1.25M bytes per sec.

The CPU channel interface transfers data from the disk to the CPU. As data are read from the disk, the data are first stored in a 64- × 8-bit first-in-first-out register file. The output of the FIFO is connected to an 8-bit data bus controlled by the disk interface logic on the CPU board. After 4 bytes have been transferred to the CPU board, the DMA controller transfers them into the main memory. At a disk transfer rate of 10 MHz, data arrive at the output of the FIFO at a rate of 1.25M bytes per sec. This sequence continues until all of the requested sectors are transferred. □

Ron DeJoung is product marketing specialist at Onyx Systems Inc., San Jose, Calif.