Chapter 2

Hardware Description

This chapter provides an overview of the hardware of Multimax systems: their physical packaging; their modular processor, memory, and input/output components; their peripherals; and their hardware options.

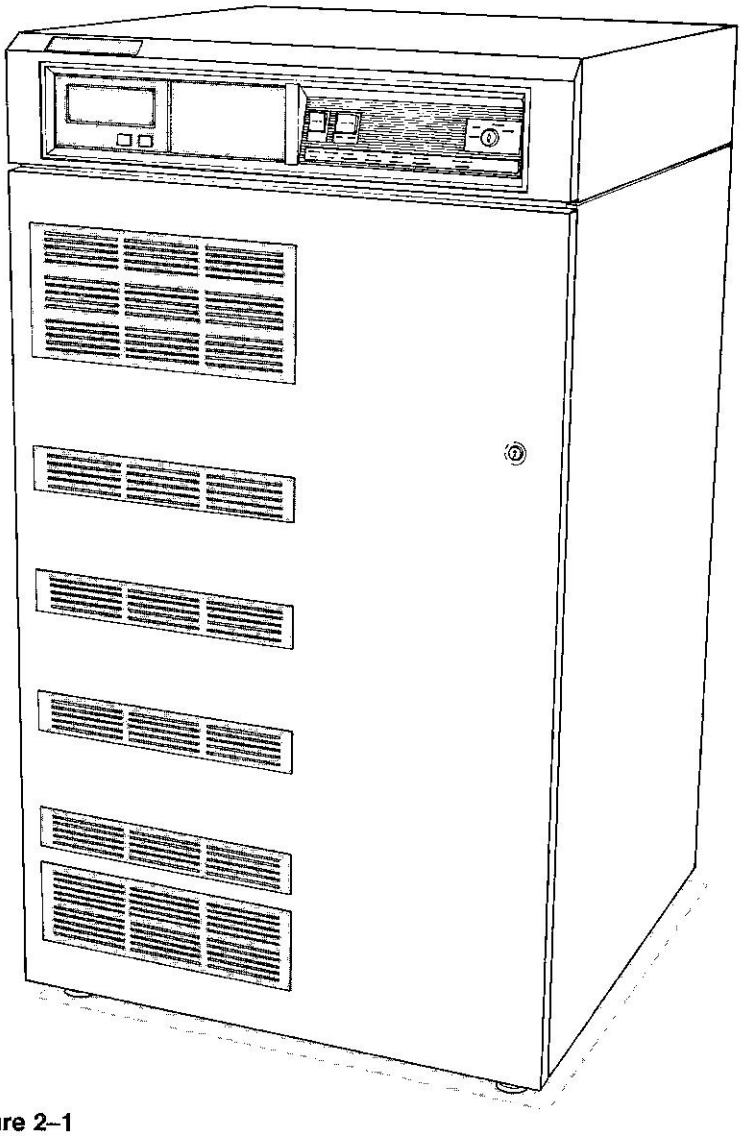
MULTIMAX SYSTEM PACKAGING

Encore currently offers four system models: the Multimax 310 and 320 and the Multimax 510 and 520. These models are described in the following paragraphs.

The Multimax 310/510

Multimax 310/510 systems (see Figure 1–1) are housed in one low cabinet that contains all system components except the system console, modem, any Ethernet-based Annex II terminal servers, and any Annex X.25 gateways. Each system provides the following internal resources:

- System power supplies
- Cooling fans
- One 11-slot Nanobus backplane with at least a minimal set of Nanobus cards
- Control panel and display
- I/O panels for mass storage, Ethernet, and system consoles
- From one to 16 Winchester disk drives
- One or two cartridge tape drives



Height:	51 in. (130 cm)
Width:	28 in (71 cm)
Depth:	28 in (71 cm)

Figure 2–1
The Multimax 310/510

The Multimax 320/520

A Multimax 320/520 system consists of two or more cabinets: the System Cabinet and one or more Peripheral Cabinets (normally attached to the System Cabinet or to one another).

The System Cabinet

This cabinet (see Figure 2–2) houses all Multimax 320/520 central system components. Mass storage devices are housed in Peripheral Cabinets, and all serial and parallel communications functions (except for the system console and modem) are accommodated by Ethernet-based Annex II terminal servers or Annex-X.25 gateways. The

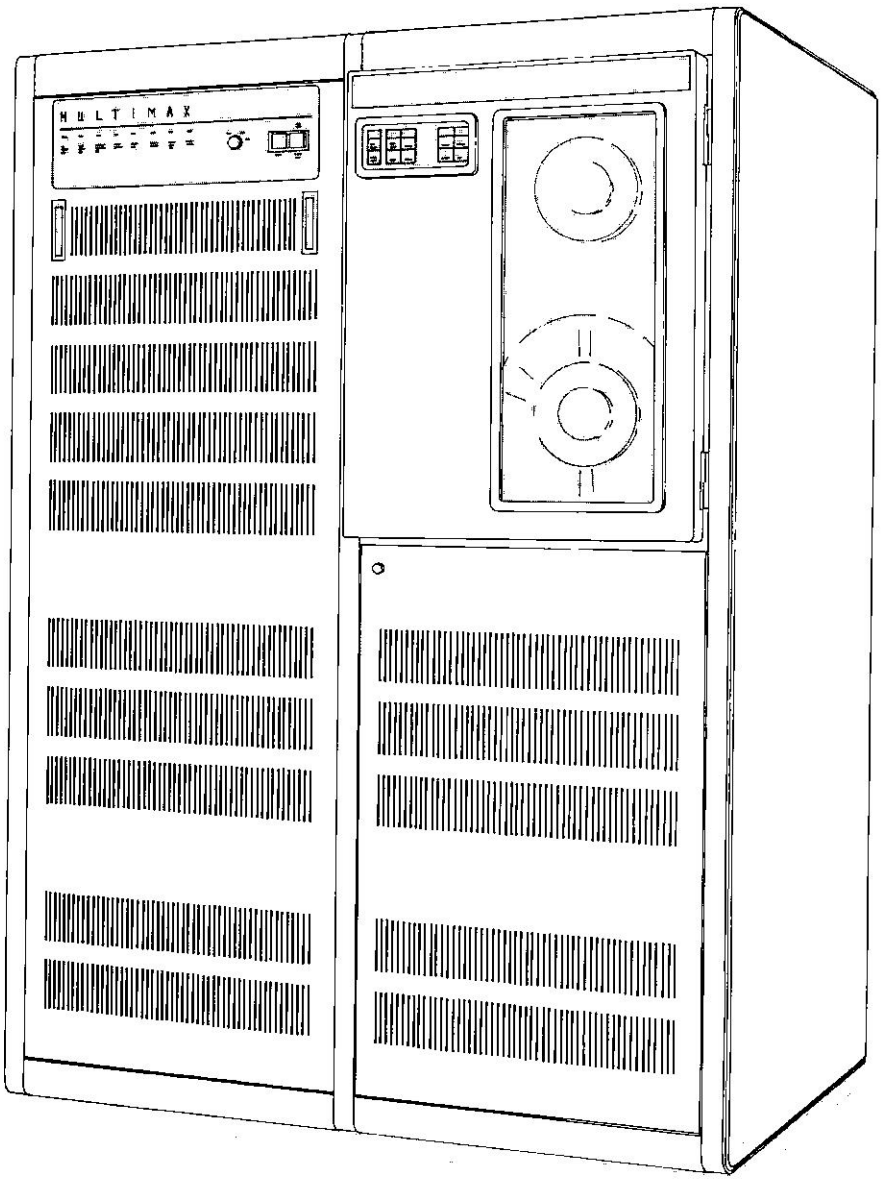
System Cabinet front panel contains status indicators as well as power and control switches. Each System Cabinet provides the following resources:

- System power supplies
- Cooling fan
- One 20-slot Nanobus backplane
- Control panel and display
- I/O Disk and tape drive controllers as required
- I/O connection panel

The I/O connection panel at the bottom rear of the cabinet provides all cabling connections to Peripheral Cabinets and Ethernet(s). It also provides a LOCAL port for the system console and a REMOTE port designed to permit console communications via modem.

The Multimax Peripheral Cabinet

The Peripheral Cabinet (see Figure 2–2) houses mass storage devices, and can contain one 6250 bpi half-inch tape drive and from one to six fixed-disk drives. Available as options are auxiliary peripheral cabinets with room for up to eight fixed-disk drives or one tape drive and up to six fixed-disk drives.



Height:	60.5 in. (154 cm)
Width:	41.75 in (106 cm)
Depth:	35 in (89 cm)

Figure 2–2
Multimax 320/520 System Cabinet

MULTIMAX FUNCTIONAL OVERVIEW

Diagrammed in Figures 2–3 and 2–4 are the functional components of the Multimax 310/510 and the Multimax 320/520. These components are the main system bus; system control, processor, memory, and input/output cards; mass storage peripherals; system console and modem; and Annex gateways and/or terminal servers for terminals,

modems, and line printers. Items rendered by solid lines are required on all Multimax machines; items rendered by dotted lines are optional.

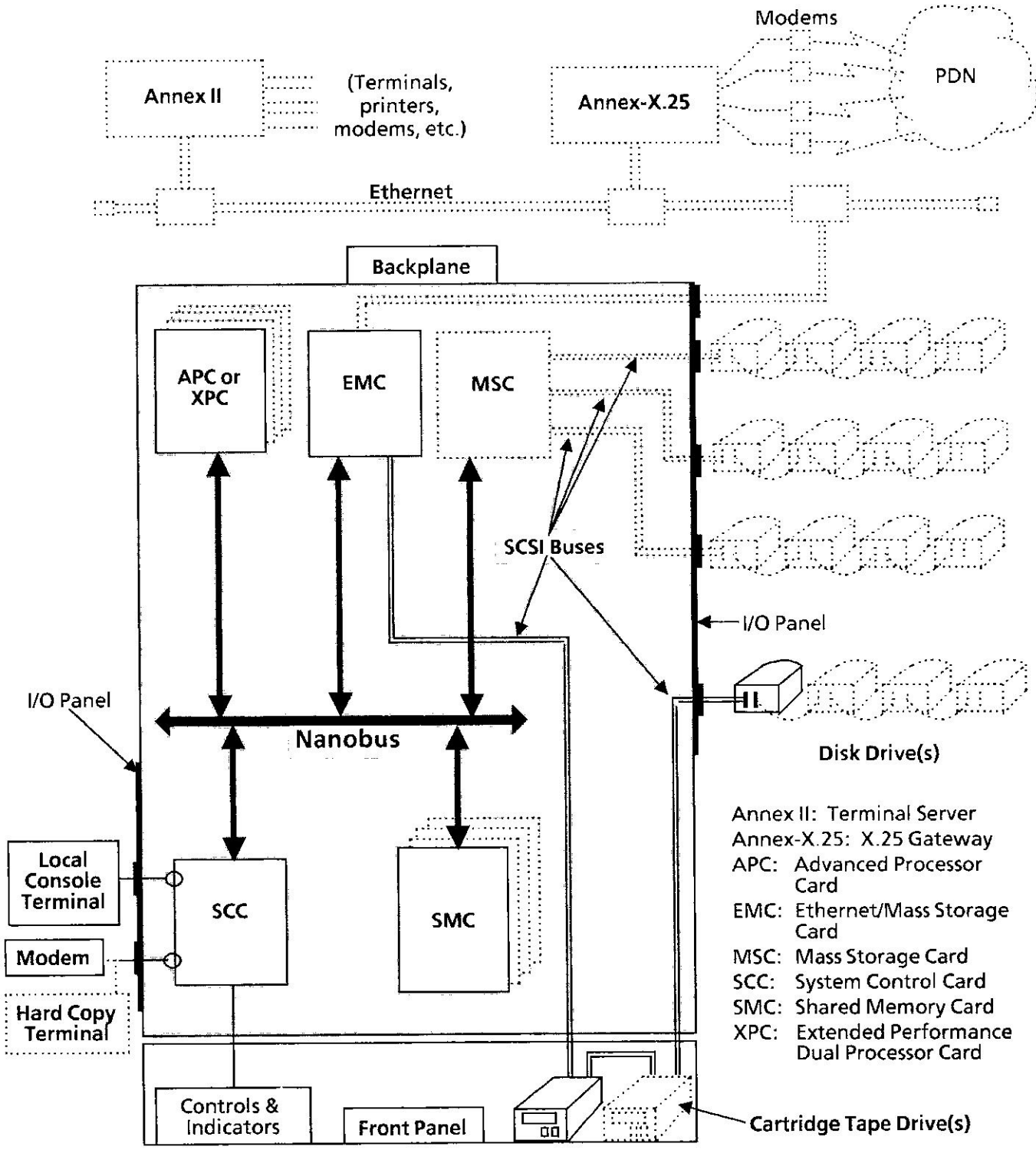


Figure 2–3
Multimax 310/510 System Functional Diagram

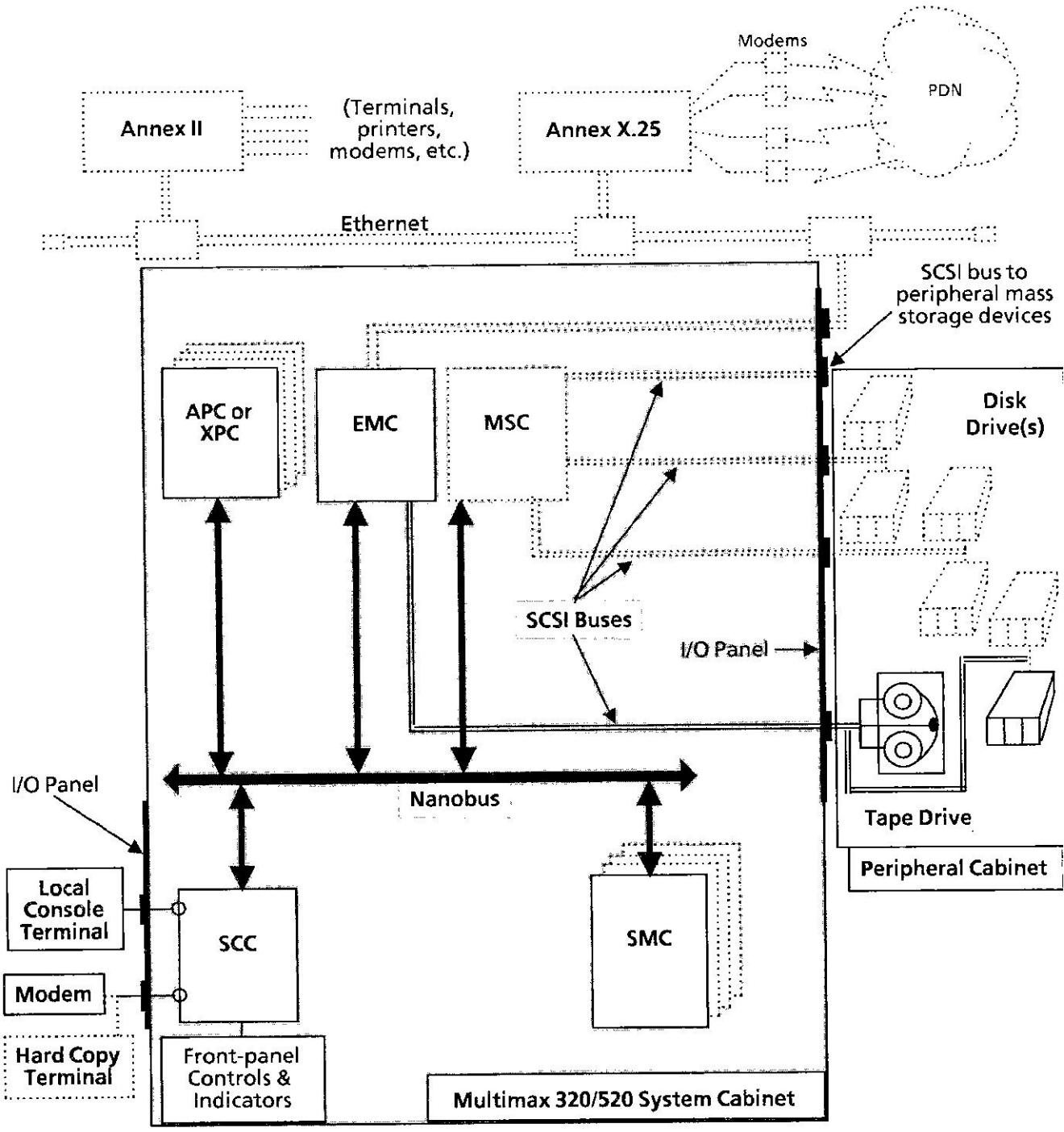


Figure 2–4
Multimax 320/520 System Functional Diagram

MULTIMAX 310/510 SUMMARY SPECIFICATIONS

- Size: 51" H x 28" W x 28" D (130 cm x 71 cm x 71 cm)
- Required front/rear clearance: 37" (100 cm)
- Weight: 400 lbs (182 kg) minimum
- Power requirements:
  - 240 VAC, single-phase, 20A (NEMA L6-20 connector)
- Heat dissipation:
  - Minimum (4 Nanobus cards, 1 fixed-disk drive, 1 cartridge tape drive): 2040 BTU/hour
  - Typical (6 Nanobus cards, 4 fixed-disk drives, 1 cartridge tape drive): 5000 BTU/hour
  - Maximum (11 Nanobus cards, 16 fixed-disk drives, 2 cartridge tape drives): 7434 BTU/hour.

Note: Multimax 310/510 systems with mass storage exceeding the above maximum limits can be configured with the aid of Multimax 320/520 Peripheral Cabinets. See the next section for add-on Peripheral Cabinet specifications.

MULTIMAX 320/520 SUMMARY SPECIFICATIONS (TWO-CABINET SYSTEM)

- Size: 60.5" H x 4.5" W x 36" D (154 cm x 103 cm x 91.5 cm)
- Required front/rear clearance: 37" (100 cm)
- Weight: 1000 lbs (455 kg) minimum
- Power requirements:
  - System Cabinet: 240 VAC, single-phase, 30A (NEMA L6-30 connector)
  - Peripheral Cabinet: 240 VAC, single-phase, 20A (NEMA L6-20 connector)
- Heat dissipation (two-cabinet system):
  - Minimum (4 Nanobus cards, 1 fixed-disk drive, 1 open-reel tape drive): 6720 BTU/hour
  - Typical (9 Nanobus cards, 4 fixed-disk drives, 1 open-reel tape drive): 15,000 BTU/hour
  - Maximum (20 Nanobus cards, 6 fixed-disk drives 1 tape drive: 30,000 BTU/hour

- Heat dissipation (add-on Peripheral Cabinet):
  - Add 350 BTU/hour for each disk drive (eight maximum for a Peripheral Cabinet without tape drive, six maximum for a Peripheral Cabinet with one tape drive) and 1500 BTU/hour for each tape drive (one maximum for each Peripheral Cabinet).

THE NANOBUS

The Nanobus was so named because, in its earliest implementation (maintained on the Multimax 320/520), it was one foot long – approximately the distance that light travels in one nanosecond. The narrower bus of the Multimax 310/510 is approximately 25% shorter but is otherwise identical to its wider precursor. The Nanobus is an extremely fast bipolar bus, providing a true data throughput of 100 Mbytes per second. It can transfer interrupts at high speed and supports hardware load levelling of interrupts among processors. Facilitating cache coherency and incorporating high reliability and fair arbitration features, the Nanobus supports internal data rates high enough to accommodate a large variety of future system enhancements.

The 320/520 Nanobus Backplane

The Nanobus backplane found in the Multimax 320/520 provides 20 card slots and accommodates three types of Nanobus cards:

- System Control Card (SCC): 1 slot
- Shared Memory Cards (SMCs): 8 slots (plus the two special slots, 11 and 13, in the the bus requester section)
- Bus requester* cards: 11 slots

Electrical and mechanical support for a fully loaded 20-card Nanobus comes standard with the Multimax 320/520 systems, guaranteeing easy expansion.

---

* Some Nanobus cards (the processor and I/O channel cards) can request use of the address bus but do not respond to requests for data. These cards are generically referred to in the following discussion as *requesters*. Some cards (SMCs) do not issue address bus requests, but do respond to requests for data from requester cards. These cards are referred to as *responders*. One card, however, the SCC, shares both characteristics and is therefore called a requester/responder.

Two slots in the requester section of the backplane have been specially modified to accomodate responder (memory) as well as requester cards, enabling memory expansion of up to 160 Mbytes.

The 310/510 Nanobus Backplane

The Nanobus backplane found in the Multimax 310/510 provides 11 card slots. Unlike the 320/520 backplane, it does not force a separation between memory and bus requestor cards, but allows any card (except the SCC, which has its own dedicated slot) to be installed in any slot in the backplane.

Maximizing Bus Performance

The Nanobus, diagrammed in Figure 2–5, possesses high-performance throughput and response time characteristics that enable system components to exchange large amounts of data with minimum communication delays.

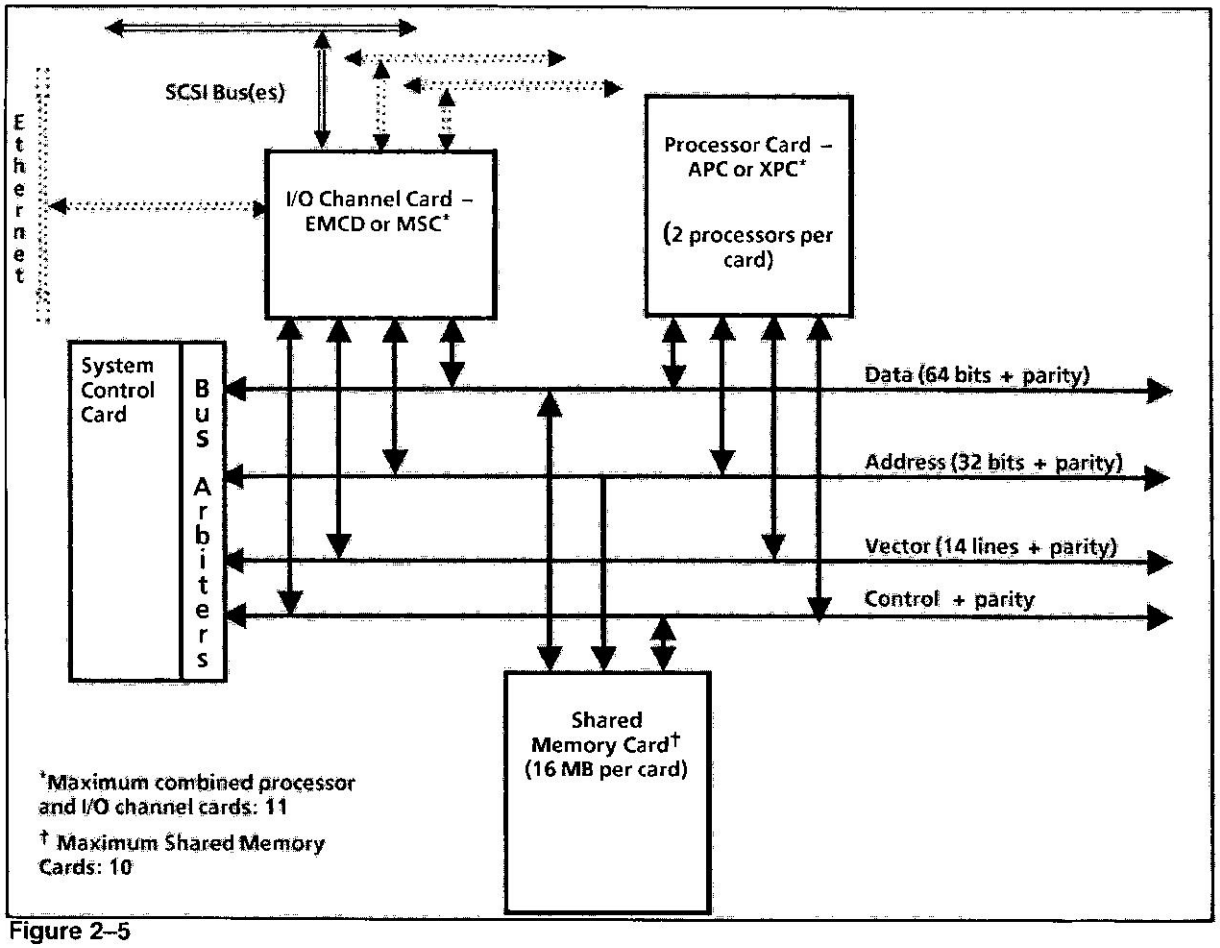


Figure 2–5
The Nanobus

The Nanobus incorporates the following features:

- High-speed synchronous operation. Provides up to 12.5 million bus "transactions" per second (all data transfers are synchronized with a 12.5 MHz bus clock).

- Separate parity-protected address and data buses. To ensure maximum bandwidth, addresses and data are not multiplexed on the same bus lines. The address bus is 32 bits wide, plus 4 parity bits; the data bus is 64 bits wide, plus 8 parity bits.
- Separate vector bus (14 bits wide). Provides a path for interrupt vector distribution throughout the Multimax. All requesters on the Nanobus can generate interrupts that must be fielded by other requesters. Since these interrupts move across the independent vector bus, they do not interfere with data and address transmissions.
- Separate, parity-protected control bus. This bus consists of many miscellaneous control lines carrying reset, power fail, and clock signals, as well as signals supplying auxiliary information about every data transfer.
- Pended "deferred response" bus operation. Fullest use of Nanobus cycles is ensured by allowing many transactions to occur simultaneously across the bus. After a bus requester asks for data, other unrelated requests and responses can occur before the bus responder returns that data. Bus protocols facilitate pended operation by tagging bus requests with the requester's identity (using control lines).
- Pipelined bus interfaces. Bus interfaces can pipeline multiple bus transaction requests by buffering them at different stages of processing. A memory card, for example, can send data to a requester, while simultaneously accepting an unrelated request for data from another requester. This pipelining is essential if the full bandwidth of bus interfaces is to be effectively utilized.
- Processor-memory interlocked operations. The Nanobus ensures rapid synchronization among processors through atomic test-and-set protocols among bus requesters and responders. Interlocked read-modify-write bus cycles can overlap one another and other bus transactions without compromising atomicity, and other system activity is not delayed while interlocked operations are taking place.
- Support of write-deferred cache protocols. The Nanobus was originally designed with inter-cache communication lines for write-deferred protocols. Although these lines were unused for the first two processor generations, their existence greatly simplified the development of the cache coherency protocols used on the XPC.

SYSTEM CONTROL CARD

The System Control Card (SCC) functions as the communications clearinghouse, bus arbiter, and diagnostic center for the Multimax. This card:

- Supervises hardware fault diagnosis
- Performs environmental monitoring
- Provides interface to front panel switches and indicators
- Provides local and remote console terminal interface

- Provides bus arbitration
- Generates bus timing signals
- Provides interval timing and time-of-year clock
- Controls system start-up, builds a configuration map of existing system resources, sizes memory, and assigns optimum memory interleaving characteristics

Figure 2–6 shows the basic functional blocks of the SCC. These blocks are discussed in the following paragraphs.

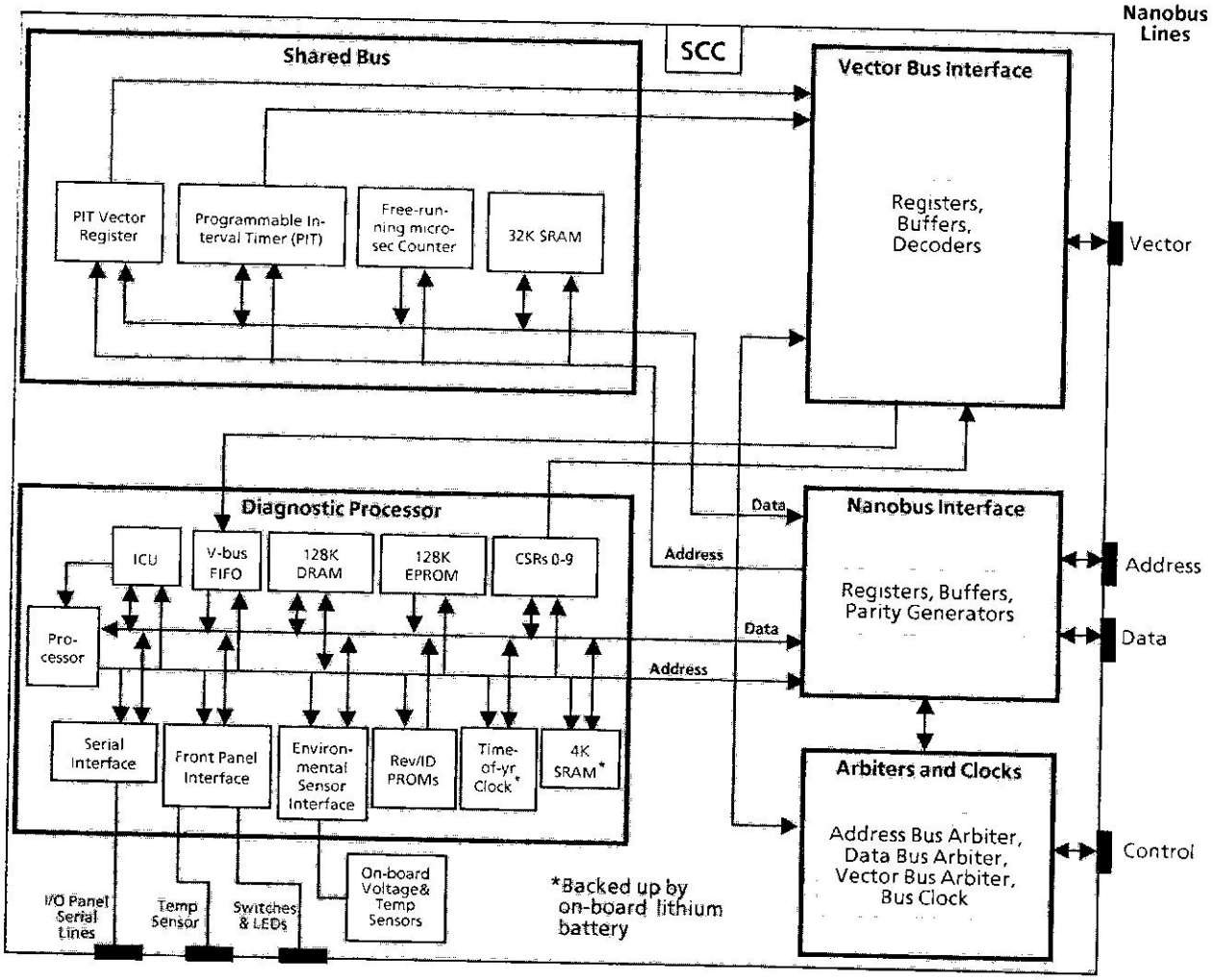


Figure 2–6
System Control Card Block Diagram

Diagnostic Processor

Based on a National Semiconductor 32-bit microprocessor, the SCC Diagnostic Processor is provided with 128K bytes of EPROM, 128K bytes of on-board dynamic RAM, and 4K bytes of non-volatile RAM (backed up by an on-board battery and sometimes called BBRAM).

The diagnostic processor performs system tests and initialization after power-up, provides a time-of-year clock, and supervises the system control panel. It also supervises the system console port and a second port designed to connect to a user terminal or a remote system console. If Nanobus card failures are detected during startup, the SCC denies that card access to the Nanobus and informs the operating system that the card is inactive.

Structures inside the Diagnostic Processor block are accessible to other cards on the Nanobus only via SCC instructions, not directly.

Environmental Monitors

The environmental sensor interface permits the SCC to monitor all system power supplies as well as system temperature. The SCC monitors three temperature sensors, one near the top of the card, a second near the bottom of the card, and a third on the front panel printed circuit board. By comparing the output of these three sensors, the SCC can provide early warning of problems ranging from excessive ambient temperature to blocked air passages or failed fans within the Multimax.

Environmental problems are reported to the operating system, which can log minor variations on the system console or, for severe problems, initiate an orderly system shutdown. This shutdown can extend all the way to removing system power since the SCC has control of the system AC circuit breaker and can cut AC power to the system.

Front Panel and Serial Interfaces

The SCC provides the interface to the front panel switches and indicators. It also provides four serial ports that are brought to the Multimax I/O panel. Of these four, two serve debugging purposes exclusively. Of the remaining two, one serves as the system console port and the other can be used either as a standard login port or (when the Multimax power mode switch is set to REMOTE) as a remote console port operating in parallel with the main console port.

Shared Bus

The shared bus section of the SCC can be accessed by all active requester modules on the Nanobus. This facility contains a programmable interval timer (PIT) for timed interrupts. Of particular note is the free-running counter, a 32-bit counter that

increments every microsecond and provides a mechanism for precise interval measurement by software. This logic also has 32K bytes of static RAM that is used to exchange commands and acknowledgements between the SCC and active Nanobus modules.

Nanobus Interface

This interface permits the SCC's diagnostic processor to access the Nanobus and allows 2-way traffic between the Nanobus and the SCC shared bus.

Arbiters and Clocks

Address, data, and vector bus arbitration are accomplished interactively between arbiters on the SCC and the individual Nanobus cards.

Address Bus Arbiter

The Address Bus Arbiter decides priority on a round-robin basis: once a requester module is granted access to the address bus, it becomes the lowest priority module, and the next logical module is assigned the highest priority.

Data Bus Arbiter

The Data Bus Arbiter uses a fixed priority algorithm to control responder access to the data bus. This algorithm gives the SCC the highest priority, gives any requester/responders in the last two slots (9 and 13) of the Nanobus the next highest priority, and gives Shared Memory Cards the lowest priority.

Vector Bus Arbiter

Vector bus arbitration occurs on two levels. The first is implemented by the SCC, which synchronizes all activity on the vector bus and thereby controls access to the vector bus itself. The second is a function of vector bus architecture and protocol. This level of arbitration involves determining which processor sharing the vector bus will accept a "classed" vector.

In general, the operating system dynamically assigns processors to one of four interrupt classes. Interrupts designate the appropriate class, and interrupt vector arbitration hardware directs the interrupt to the processor with the lowest current interrupt load. The result is that bursts of interrupt activity are evenly spread across the available processors in the class.

Bus Clock

The 12.5 MHz master system clock for the Multimax system is distributed from the SCC. All bus clock lines on the Nanobus are driven by this SCC logic.

ADVANCED DUAL PROCESSOR CARD

The Advanced Dual Processor Card (APC), diagrammed in Figure 2–7, represents the second generation of processor cards designed for Encore Multimax systems. The first generation, the Dual Processor Card (DPC), used two indpendent 10 MHz National Semiconductor 32032 processor chips; it employed a 32K-byte shared cache, and was capable of approximately 0.75 MIPS per processor.

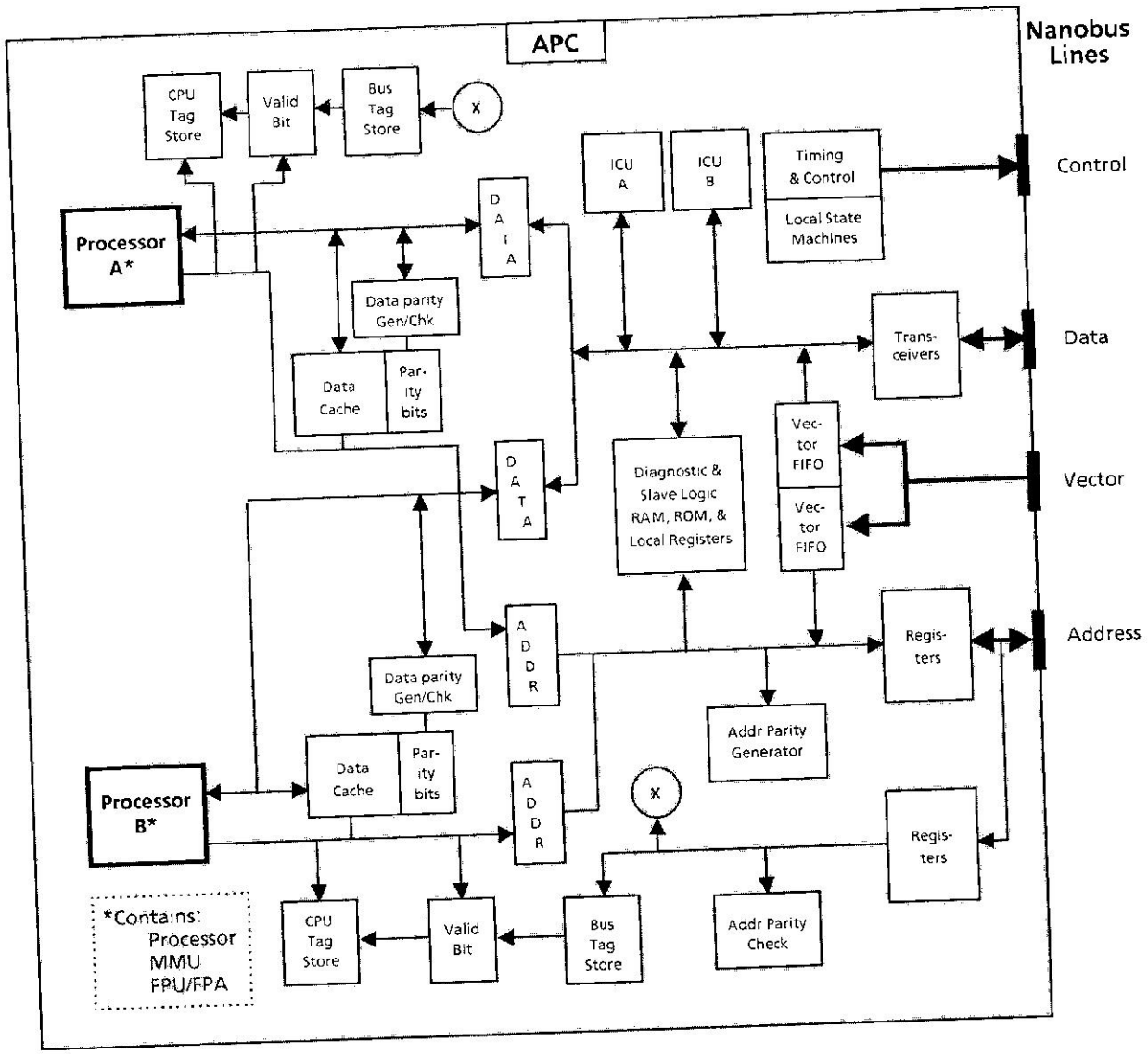


Figure 2–7
Advanced Dual Processor Card Block Diagram

The Advanced Dual Processor Card provides two independent 15MHz NS32332 processors, each with its own private 64K-byte cache memory. Each processor is associated with an NS32382 15MHz Memory Management Unit (MMU) that enables the generation of 32-bit physical addresses.

32-Bit Processor

Each National Semiconductor 32332 on the APC is a 32-bit processing unit with a full 32-bit data bus to memory. Executing approximately 2 million instructions per second (MIPS), the processor has a compactly encoded instruction set that provides efficient and economical support for high-level languages. Its fully integrated floating point instruction set and 13 addressing modes (designed for the kinds of accesses compilers generate) make the NS32332 a powerful and efficient processing engine for the Multimax. Some of its features are:

- 15 MHz clock
- Internal 20-byte prefetch queue
- Burst mode support that permits each processor to increase its throughput by reading additional bytes of memory on instruction prefetch cycles (has no effect on operand and slave cycles)
- Fast slave cycles that allow 32-bit data transfers to and from the NS32382 MMU and the optional floating point accelerator (FPA)
- Bus retries that allow automatic retry of instructions or data read with uncorrectable errors

Cache Memory

Multimax processors spend most of their time reading instructions and data from memory, executing those instructions, and (much less often) writing new or transformed data back to memory. In a shared memory multiprocessor environment, processors also communicate with one another, mostly through shared memory.

The Multimax decreases memory access time and bus loading by storing frequently referenced instructions and data in a 64K-byte cache of fast (25 ns) static RAM for each of the two processors on the APC. Memory data is stored in this cache whenever either of the processors on an APC reads or writes main memory locations, and an index of the addresses of the locations thus stored is kept in the cache tag memory array. Thereafter, any reference to those locations will access the cache, not main memory. Cache accesses do not incur the processor wait states required for main memory access, nor do they impose any traffic on the Nanobus.

The APC uses a *write-through* mechanism for updating cache entries: when a processor writes to one of its cache entries, it simultaneously writes to the associated location in memory, thus keeping main memory abreast of changes occuring in local cache.

The APC cache is kept current with changes in main memory (generated by writes from other processors or I/O devices) by means of the bus tag logic, which continuously scans the Nanobus for memory writes involving locally-cached addresses. When such writes are detected, the "valid" bit for that cache address is switched to its invalid state. Later, when the associated processor needs data from that cache address, the processor will recognize that the associated cache entry is now invalid and go to main memory rather than cache for the data.

Because the bus tag logic is independent of the CPU tag logic and replicates its data, maintaining cache concurrency through bus monitoring occurs without impacting speed of access to the cache by the processors.

Memory Management

Associated with each processor is a memory management unit (MMU) that provides hardware support for demand-paged virtual memory management (4K-byte page size). The MMU, a National Semiconductor NS32382, supports a full 32 bits (4 Gbytes) of virtual addressing per process, allowing application processes to grow well beyond the 16-Mbyte limit imposed by earlier generations of National Semiconductor processors.

Time Slice End Interrupt Control

Each processor is provided with private Time Slice End (TSE) logic (implemented by the National Semiconductor 32202), providing a way for a processor to terminate a compute-bound process after a software-selected amount of execution time. The TSE logic can generate an interrupt after from 1 to 2^{32} 100-nanosecond clock ticks (that is, 100 nanoseconds to approximately 430 seconds). TSE interrupts act directly on the processor and do not pass through the Vector Bus FIFO.

Floating Point Unit (FPU)

The APC's default mechanism for handling floating point operations is the NS32081 FPU. With this unit, the APC can perform 32-bit single-precision IEEE floating point operations at about 895K Whetstones per second and 64-bit double-precision operations at about 637K Whetstones. This IC is socket mounted and can be removed when the optional FPA is installed.

Optional Floating Point Accelerator (FPA)

Customers requiring maximum speed floating point calculations may wish to acquire the FPA option (see Figure 2–8) which approximately triples the speed of standard floating point calculations. The FPA also provides transcendental functions not supported by the standard FPU.

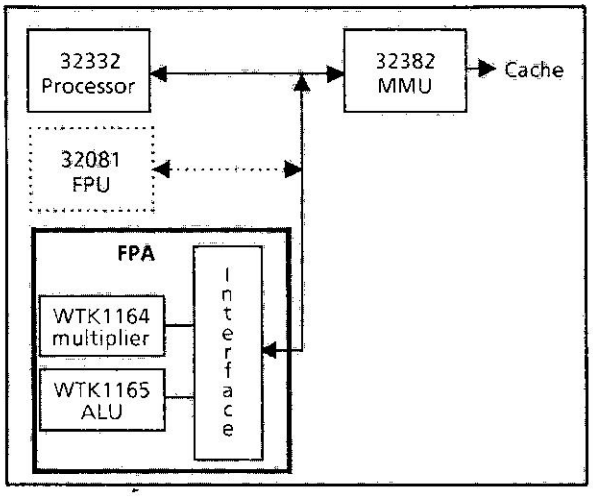


Figure 2–8
APC Processor with Floating Point Accelerator

Each FPA consists of three integrated circuits (ICs) that mount in one of two sets of three reserved sockets on the APC. One of these ICs is a Weitek WTK1164 multiplier; one is a WTK1165 Arithmetic Logic Unit (ALU); and one is a proprietary Encore chip carrying the necessary interface to allow the two Weitek chips to communicate with the processor. The proprietary chip has the following features:

- Encore-developed, full custom design to maximize Weitek performance with NS32332 processors.
- Support for all 32081 instructions plus transcendental functions. Provides early-completion detection for speed and maximum CPU/FPA overlap.

EXTENDED PERFORMANCE DUAL PROCESSOR CARD

The Extended Performance Dual Processor Card (XPC) represents the third generation of processor cards designed for Encore Multimax systems.

The XPC provides two independent 30 MHz NS32532 processors, each with its own private 256K-byte cache memory. Since virtual-to-physical memory mapping is built into the NS32532, no external Memory Management Unit (MMU) is required. The XPC is diagrammed in Figure 2–9.

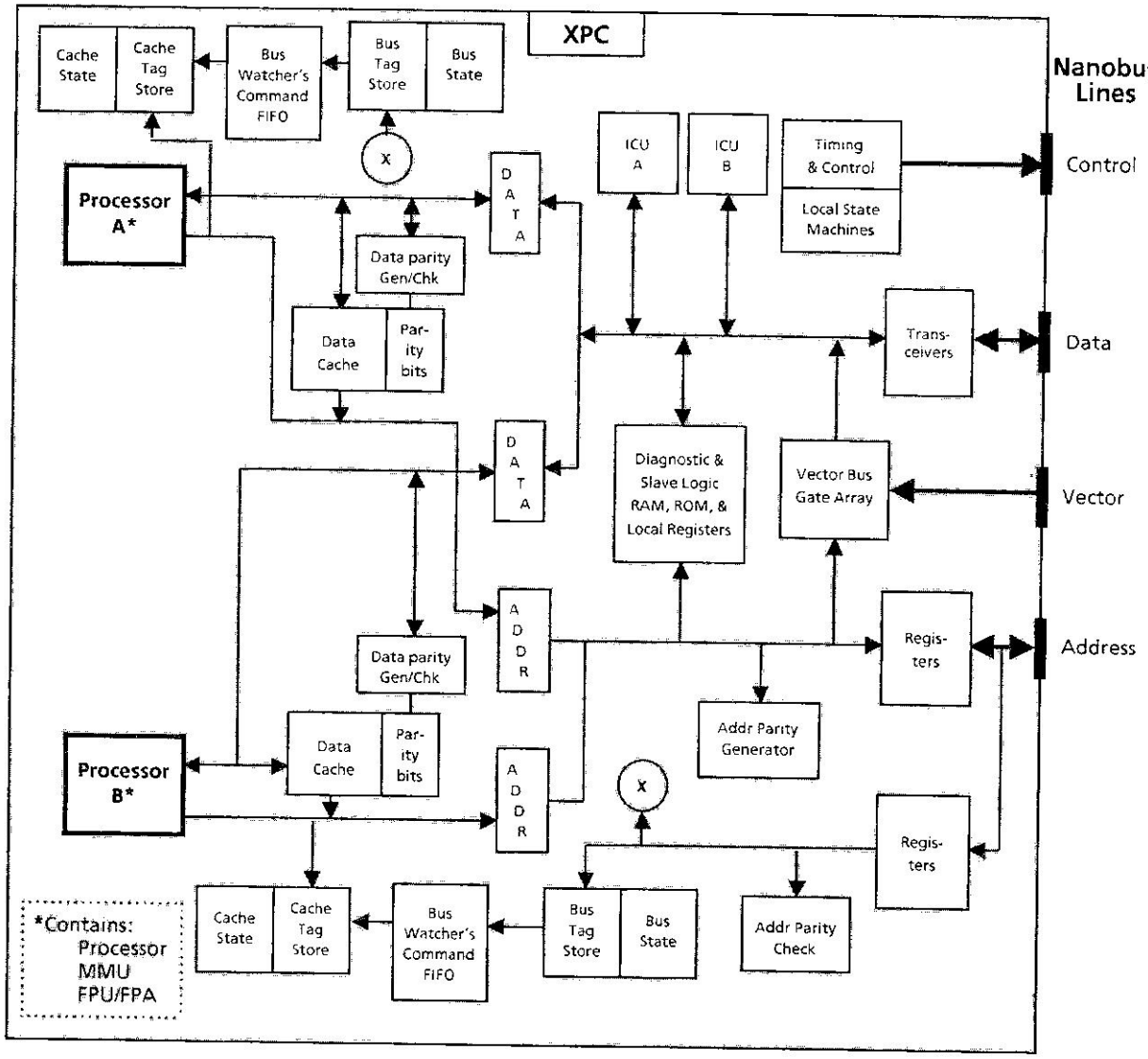


Figure 2–9
Extended Performance Dual Processor Block Diagram

Each NS32532 on the XPC is a 32-bit processing unit with a full 32-bit data bus to memory. Executing approximately 8.5 million instructions per second (MIPS) at the system level, the processor has a compactly encoded instruction set that provides efficient and economical support for high-level languages. Its fully integrated floating-point instruction set, and 13 addressing modes (designed for the kinds of accesses compilers generate) make the NS32532 a powerful and efficient processing engine for the Multimax. Some of its features include:

- 30 MHz clock
- Integrated Memory Management Unit (MMU)

- Integrated 512-byte instruction cache
- Integrated 1K-byte two-way set associative data cache
- Monitor logic to maintain cache coherency
- Out-of-sequence data referencing
- Fast slave cycles that allow 32-bit data transfers to and from the NS32381 floating point unit (FPU).
- Bus retries that allow automatic retry of instructions read with uncorrectable errors
- Full NS32000-family instruction set compatibility
- Substantially reduced clock-per-instruction times due to architectural enhancements and the addition of on-board caches
- Customized VLSI vector bus gate array that facilitates sending and receiving interrupts within the system

Cache Memory

The Multimax decreases memory access time and bus loading by storing the most recently referenced instructions and data in a 256K-byte cache of fast (25 ns) static RAM for each of the two processors on the XPC. Memory data is kept in cache whenever one of the processors on an XPC reads from or prepares to write to system memory, and an index of the addresses of the locations thus stored is kept in the cache tag memory array (CTAG). Thereafter, any reference to such locations will access the cache, not main memory. Cache accesses do not incur the processor wait states required for main memory access, nor do they incur traffic on the Nanobus.

The XPC uses a write-deferred protocol for maintaining coherency between the various processor caches. Each XPC cache is kept current with relevant changes in other processor caches by means of the bus tag (BTAG) logic. This logic monitors all bus traffic for attempts to access locations currently held in its own CTAG. It dedicates additional status bits to the identification of four possible states for each of its cache entries: **owned**, **shared**, **dirty**, and **invalid**. These states occur as described below.

- When a processor needs to read (but not write) data that is not contained in cache, the cache logic issues a **read public** command for the required address. If no other cache contains the data (that is, no other BTAG acknowledges the read request), the data is taken from system memory, put into local cache, and marked **owned**. If another cache contains the data, that cache acknowledges the request and marks its location as **shared**. The requesting cache accesses the data from system memory, also marking the location as **shared**.

- When a processor needs to write data, it must first have ownership of the location. If the location is not already owned, the cache issues a **read private** command, which causes any other caches containing this location to mark their entries **invalid**. Then the processor writes new data into its cache and marks the written location **dirty**. The new data remains in this cache location until one of the following events occurs:
  - The location is rewritten by the owning processor.
  - Another processor issues a **read private** request (in which case, the cache passes the data to the requesting processor and marks the location **invalid**).
  - The location is reallocated for use by new cache data (in which case it is written back to memory).

The foregoing description somewhat oversimplifies the complex transactions that can occur in maintaining cache coherency on the XPC. The most important feature of this architecture is that because the bus tag logic is independent of the CPU tag logic and replicates its data, maintaining cache concurrency through bus monitoring occurs without impacting speed of access to the cache by the processors. Some bus traffic is involved with the write-deferred scheme, but much less than is involved with traditional write-through protocols.

Memory Management

Built in to each processor is a Memory Management Unit (MMU) that provides hardware support for demand-paged virtual memory management. This MMU supports a full 32 bits (4 Gbytes) of virtual addressing.

Time Slice End Interrupt Control

Each processor is provided with private Time Slice End (TSE) logic (implemented by the National Semiconductor 32202), providing a way for a processor to terminate a compute-bound process after a software-selected amount of execution time. The TSE logic can generate an interrupt after from 1 to 2^{16} 1.08 microsecond clock ticks (that is, 1.08 microsecond to approximately 70 milliseconds). TSE interrupts act directly on the processor and do not pass through the Vector Bus FIFO.

Floating Point Unit (FPU)

The XPC's default mechanism for handling floating point operations is the NS32381 FPU. With this unit, the XPC can perform 32-bit single-precision IEEE floating point operations at about 2750K Whetstones per second and 64-bit double-precision operations at about 2090K Whetstones.

SHARED MEMORY CARD

Each Shared Memory Card (SMC – see Figure 2–10) provides 16 Mbytes of random access memory (RAM) in two independent banks of 1 Mbit MOS RAM chips. Each card supports 2-way interleaving between banks and 4-way interleaving between boards – permitting 8-way interleaving on systems that have at least four SMCs. The base address and interleaving characteristics of each SMC are set under software control at system startup.

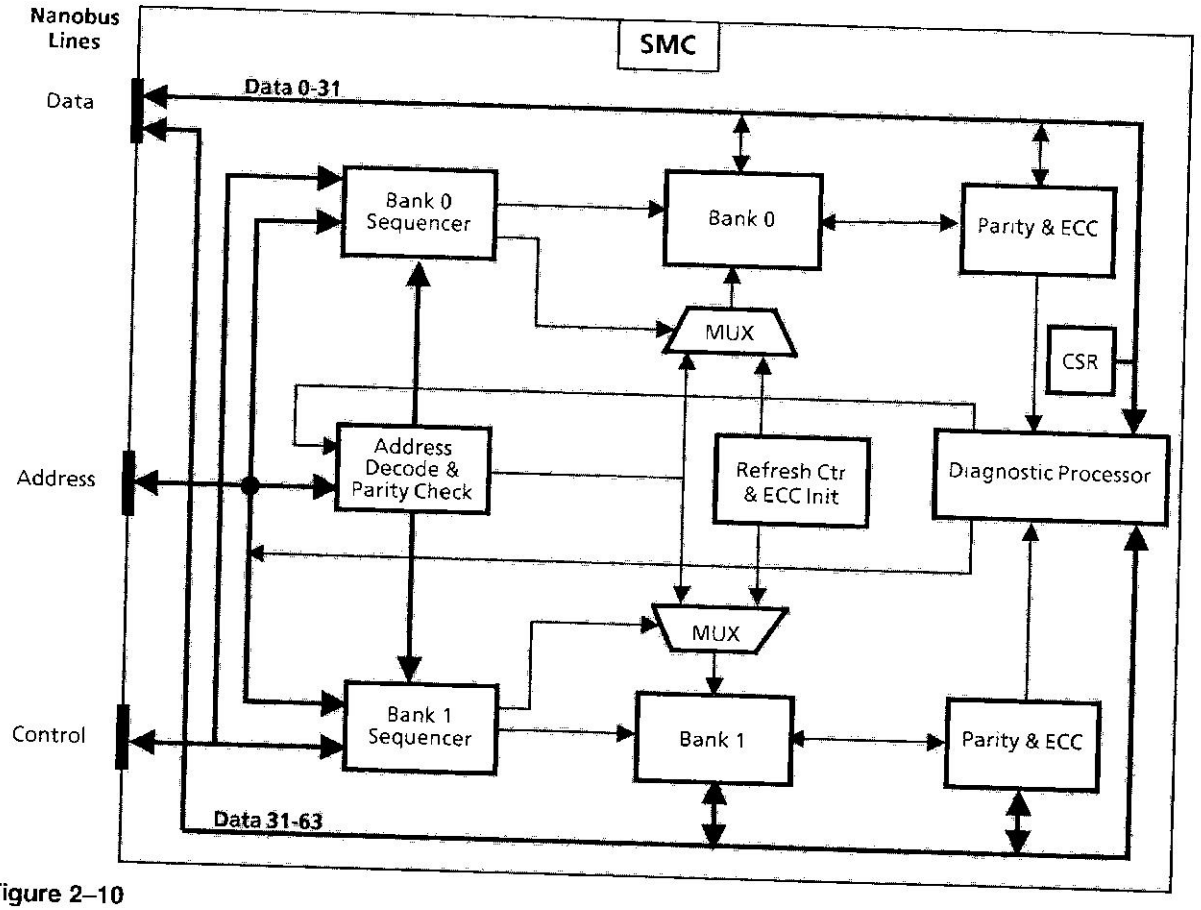


Figure 2–10
Shared Memory Card Block Diagram

The memory cycle time of each SMC is four Nanobus cycles, or 320 nanoseconds. During this interval, a given SMC can compose up to eight bytes of data for transfer to the Nanobus or accept and store up to eight bytes received from the Nanobus. Since the bus architecture allows another interleaved board to begin a new 8-byte memory transfer with each successive bus clock cycle, the four boards involved in 8-way interleaving can transfer double longwords of data (64 bits or 8 bytes) at an aggregate rate of 100 Mbytes/second.

Since each of the two memory banks on the SMC is provided with its own controller and request queue, the ability to deal with memory transactions grows with the

amount of memory to be accessed. On a Multimax fully loaded with memory cards, for example, there are 16 controllers and request queues. Consequently, large amounts of memory traffic can be handled with minimum contention for memory controller resources.

A key feature of the SMC is that any byte in its memory can be used as a multiprocessor "lock," and these locks can be set or reset across the Nanobus using atomic read-modify-write bus cycles. Other processors, when testing the state of a lock, will first read the byte's contents from the SMC into their cache, and subsequently read from the cache until the value of the lock changes. The result is that no load is imposed on the Nanobus or the SMC during the time spent waiting for the lock to change state.

All data is stored with an Error Correcting Code (ECC). Single bit-errors in each 32-bit longword are detected and corrected with each access; double-bit errors are detected and reported. In addition, the SMC "sweeps" the entire memory array during refresh and corrects any single-bit errors that it encounters. Since a full refresh sweep on each bank occurs approximately once every eight seconds, the likelihood of an uncorrectable double-bit error is dramatically reduced.

Each SMC carries a diagnostic processor that checks both of its memory banks at power-up and whenever directed to do so by the system diagnostic processor on the System Control Card. In addition, each memory bank maintains a Control/Status Register through which it reports single- or double-bit errors, or bus parity errors, to the requesting processor. The Parity logic identifies parity errors in written addresses and data so that the requesting processor can automatically retry the write operation.

INPUT/OUTPUT CHANNEL CARDS

Two Nanobus cards address Multimax mass storage and Ethernet requirements: the Ethernet/Mass Storage Card (EMC) and the Mass Storage Card (MSC). The former provides one channel for the Ethernet and a separate channel for mass storage I/O. The latter provides three high-performance mass storage I/O channels. Any Multimax supporting an Ethernet must contain at least one EMC. The EMC's mass storage channel accommodates disk and tape controllers. One or more EMCs provide an appropriate I/O capacity for small- and medium-sized systems. For large configurations, or very I/O intensive applications, a single EMC plus one or more MSCs combine to provide the appropriate solution.

Ethernet/Mass Storage Card

The EMC allocates one I/O channel each for the Ethernet and for an asynchronous Small Computer Systems Interconnect (SCSI) bus. Ethernet is an industry-standard 10 Mbit/second Local Area Network (LAN) used to connect the Multimax to Annex terminal servers and to other computer systems. Processing nodes with their own

Ethernet interfaces are connected by transceiver cables to transceivers, which in turn are connected to an Ethernet coaxial cable that can be as long as 500 meters. The Ethernet interface on the EMC attaches to an Ethernet port on the Multimax I/O panel.

The SCSI interface on the EMC supports up to six disk disk drives and one tape drive. If more storage or faster throughput is required, additional EMCs or one or more MSCs (described in the next section) can be installed in the Nanobus to support additional mass storage units or to lighten the SCSI bus load.

Encore provides two types of EMCs, identical except for their SCSI interface. One (the EMC) provides a SCSI interface appropropriate to the 1.5 Mbyte/second, single-ended, asynchronous controllers associated with mass storage devices on older Multimaxes; the other (the EMCD) provides an interface appropriate to the 4 Mbyte/second, differential, synchronous controllers integral with the mass storage devices that Encore currently provides. The former supports SCSI bus structures less than 20 feet long; the latter supports bus structures up to 50 feet long.

The EMC consists primarily of three functional blocks, shown in Figure 2–11. The Nanobus interface logic on this card is shared by the EMC processor, the SCSI control interface, and the LAN control interface. Any of these three functional blocks can become temporary master of the Nanobus interface and thereby gain access to the entire Nanobus memory space.

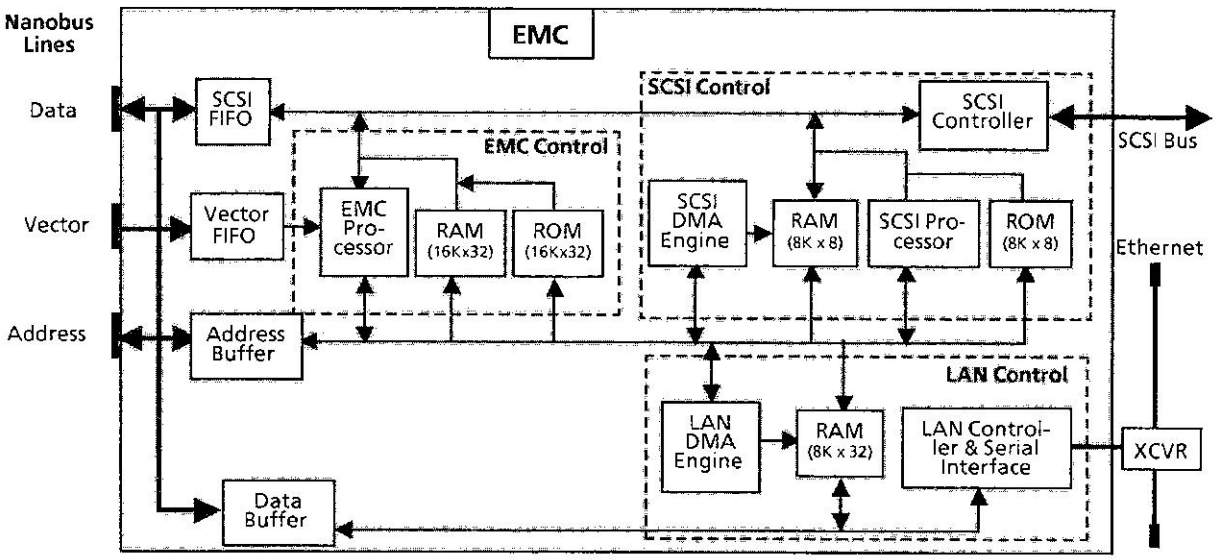


Figure 2–11
Ethernet/Mass Storage Card Block Diagram

EMC Control Logic

This portion of the EMC consists of an NS32032 processor equipped with local ROM for program storage, 64K bytes of local RAM for program and data storage, local control and status registers, vectored interrupts, and two windows into main memory.

The EMC processor has three principal tasks. It accepts LAN and SCSI control information from the system; it uses this information to initiate LAN and SCSI interface direct memory access (DMA) data transfers; and it monitors the operation of both the SCSI and the LAN control blocks.

SCSI Control Logic

The SCSI control interface consists of a SCSI bus controller, a 512-byte data FIFO, a microprocessor, and a dedicated SCSI DMA engine. The SCSI bus controller transfers data between the SCSI bus and the SCSI data FIFO under control of the SCSI processor. The contents of the SCSI data FIFO can be read from or written to Multimax main memory by the SCSI DMA engine.

LAN Control Logic

The LAN control interface consists of an Ethernet controller, a separate dedicated DMA engine, and 32K bytes of local memory. This memory is used to store transmitted and received data, command and status information, network management statistics, and diagnostic information. Any portion of the LAN memory can be filled from or transferred to Multimax main memory by the LAN DMA engine.

The LAN transceiver interface is compatible with Ethernet Revision 2 and IEEE 802.3 Ethernet specifications. Any transceiver that complies with either of these specifications can be used with the Multimax.

Mass Storage Card

The MSC, diagrammed in Figure 2–12, provides an intelligent interface between the Multimax system and mass storage devices. Usable in the same backplane with one or more EMCs, each MSC provides a high-speed differential interface to three synchronous or asynchronous SCSI buses, each with data transfer capability as high as 1.5 Mbytes/second in asynchronous mode or 4 Mbytes/second in synchronous mode.

Up to seven SCSI target devices can be attached to each SCSI bus. A target device can be a disk or a tape drive. The theoretical maximum fanout from a single MSC is 84 drives; the minimum fanout (one drive with integrated SCSI controller connected to each MSC SCSI port) is three drives. Optimum performance is obtained with three or four drives per channel. Larger configurations provide additional storage but not significant performance gains.

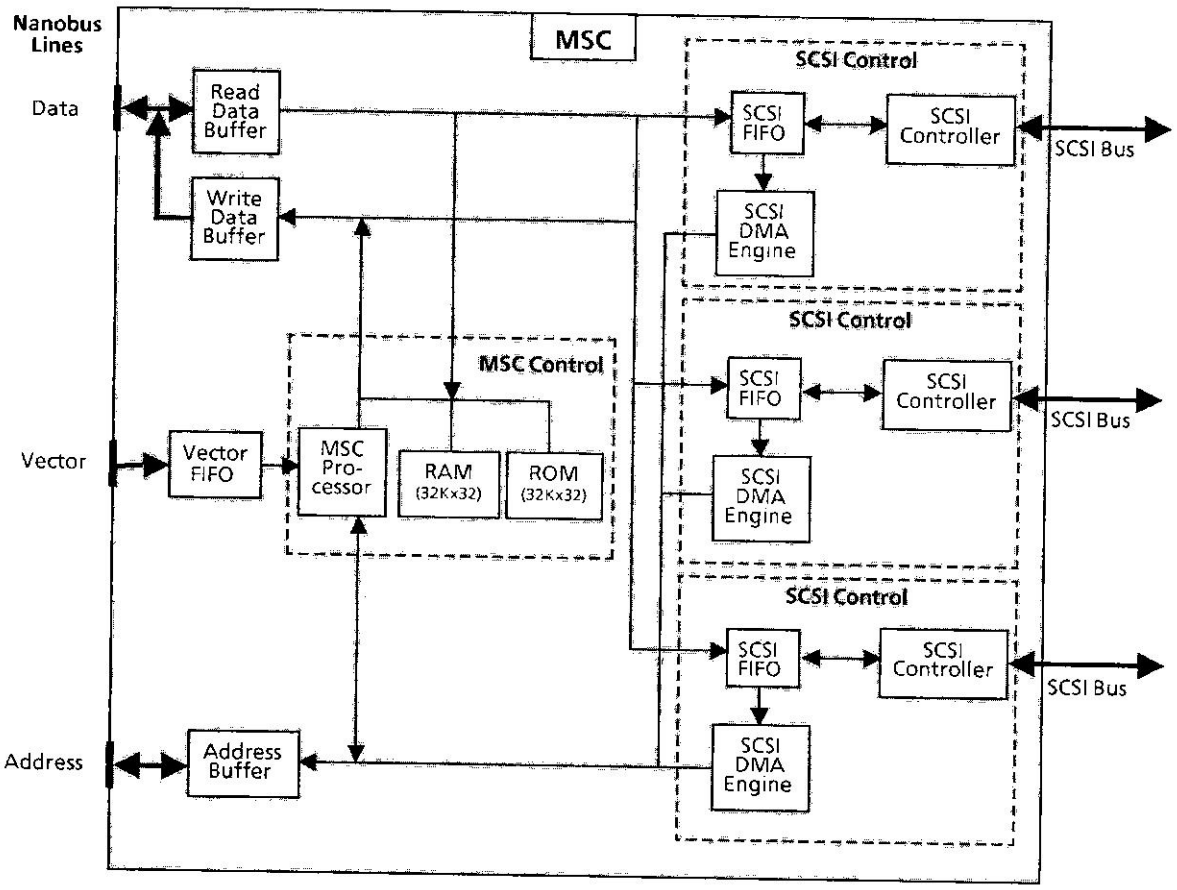


Figure 2–12
Mass Storage Card Block Diagram

MSC Control Logic

This portion of the MSC consists of an NS32332 processor equipped with 128K bytes of local ROM for program storage, 128K bytes of local RAM for program and data storage, local control and status registers, vectored interrupts, and two 1-Gbyte windows into main memory. A dual-threaded Nanobus hardware interface allows multiple outstanding Nanobus requests and up to 20 Mbytes/second throughput.

MSC SCSI Control

Each of the three MSC SCSI control interfaces consists of a SCSI bus controller, a 128-byte Read Data FIFO, a 64-byte Write Data FIFO, a microprocessor, and a dedicated SCSI DMA engine. The SCSI bus controller transfers data between the SCSI bus and the SCSI data FIFO under control of the MSC processor. The contents of the SCSI data FIFO can be read from or written to Multimax main memory by the SCSI DMA engine.

MULTIMAX HARDWARE OPTIONS

In addition to the standard Nanobus cards, a large variety of mass storage and networking options is available for Multimax systems. These options fall into the following categories.

Mass Storage Devices

Encore offers a full complement of high-performance disk and tape drive options for both the Multimax 310/510 and the Multimax 320/520. These options change with the available technology and are therefore not listed in this *Technical Summary*. Contact your Encore local sales representative for current descriptive literature.

Annex II Terminal Server

Annex II Terminal Servers provide the primary means of connecting terminals and printers to Multimaxes. Annex IIs connect up to 32 asynchronous devices and one parallel device to Ethernet LANs and have the primary purpose of supporting terminal/host communication in TCP/IP Ethernet environments. Hosts can be Multimaxes or any other machines that meet the TCP/IP Ethernet requirement.

The Annex II is unique among terminal servers in its ability to function as a front-end processor and thus to offload host machines. The editing front-end feature, for example, allows the Annex II to offload GNU Emacs overhead from generic TCP/IP hosts. When used with the Multimax, the Annex II executes the tty driver locally, thereby relieving the Multimax of most of the character interrupt and line editing burden.

The Annex II provides a powerful hardware platform with the following features:

- A National Semiconductor 32-bit microprocessor
- One to three Mbytes of local memory
- One Ethernet transceiver port
- 16 or 32 serial ports – for connecting to terminals, serial printers, or modems
- One printer port – for connecting to Centronics-type parallel printer

Annex IIs are small, quiet, free-standing, user-installable units that can operate in ordinary office environments using normal office power facilities. Terminals, modems, and printers connect to the Annex II, which in turn connects to the Ethernet. The Annex II thus provides access to hosts on the Ethernet as well as to remote hosts that can be reached through network gateways such as the Annex-X.25. From the point of view of wiring, the advantages of this approach include longer distances from terminal to host than is economical by other means, easier addition of terminals to a network by adding Annex IIs, and a simpler and more maintainable wiring plan.

Any Annex II-based terminal or modem can communicate with any TCP/IP host on the Ethernet; users can initiate multiple sessions on different hosts and switch conveniently among them.

Annex IIs permit printers (both low- and high-speed) to be located where they are needed along the Ethernet, obviating the need for long, dedicated printer cables and making it easy to situate printers close to users.

Customers who employ Annex IIs for terminal support can add more terminals by simply adding more Annex IIs, a nearly unlimited number of which can be distributed wherever required along an Ethernet.

Primary Applications

The primary application of the Annex II is to support terminal-to-host access in TCP/IP Ethernet environments. Figure 2–13 depicts a possible network configuration in which Annex IIs are used for such access. This figure shows three Annex IIs and four UNIX-based computers that support TCP/IP communications protocols. Terminals and modems can easily be connected to the same Annex II in any arbitrary arrangement. Users, whether local or remote, can connect to any of the hosts attached to the LAN with equal ease.

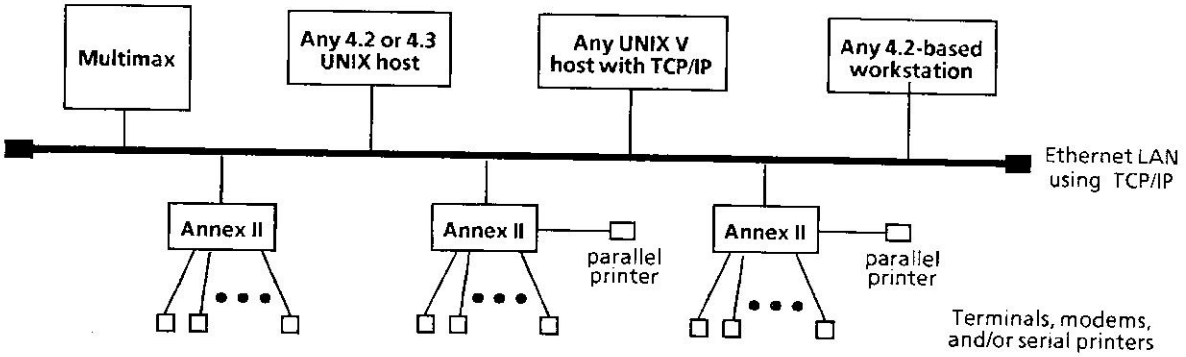


Figure 2–13
Example: Annex II Network Configuration

Summary of Annex II Features

When the Annex II is running its operational code (down loaded from a cooperating host either via the Ethernet or a serial port), it provides the following features and benefits.

- Supports both **rlogin** and **telnet** protocols directly.
- Provides full support of 4.3BSD TCP/IP – including subnets.
- Incorporates a complete implementation of IP routing and supports ICMP redirects and the 4.3BSD *route* daemon. Automatically handles large network environments with gateways.
- Provides modem support and automatic baud rate detection on all ports, transparently giving dial-up access to all systems on the LAN.
- Provides a powerful UNIX-style configurable command interface with a built-in **help** facility.
- Permits up to 128 sessions on the 16-port unit and up to 256 sessions on the 32-port unit; every user can access multiple hosts simultaneously.
- Allows line speeds of up to 38.4 Kbaud on each port, repainting a 24 x 80 terminal screen in 0.5 seconds or less.
- Provides a **telnet** daemon that allows modems and other communication devices to be accessed from any system on the network.
- Permits assigning an Internet address to port rotaries.
- Configurable to allow use of either RWHOD broadcasts or network name servers (IEN 116 or BIND) for host name-to-address translations.
- Supports the Berkeley UNIX print spooler, allowing one Annex II to handle a high-speed parallel printer and up to 32 serial printers.
- Provides a flexible security system that allows password protection of Annex II ports and resources.
- Relies on a centralized, host-based network management utility that provides convenience and flexibility.
- Implements a comprehensive version of the Serial Line Internet Protocol SLIP, providing an alternate method for operating code loading and Internet access. Any number of Annex II ports can be configured as SLIP interfaces.

Annex II Products

Two products compose the Annex II family, the Annex II-UX and the AnnexII-MX. The Annex II-UX is an Ethernet terminal server for TCP/IP environments and works with any system that supports TCP/IP (there must be one UNIX host on the network to download the Annex II's operating code). The Annex II-UX is a stand-alone product that operates independently of the Encore Multimax computer.

The Annex II-MX is a specialized form of the Annex II for use with Encore's Multimax computers. It behaves as does the Annex II-UX, but also functions as a

front-end processor for the Multimax, running the UMAX 4.2 or UMAX V tty drivers. (The UMAX kernels have been modified to permit this outboard operation of the tty drivers.) Running the tty driver in the Annex II provides a substantial offloading of work from the Multimax, since the Multimax in many instances receives input on a line-by-line rather than character-by-character basis. The Annex II-MX functions as the terminal server for Multimax computers.

Hardware Description

The Annex II (see Figure 2–14) is a powerful stand-alone computer containing a 32-bit microprocessor, 1 or 3 Mbytes of shared, dual-ported, parity-protected memory, and hardware-assisted character output. The architecture allows two bus masters: the microprocessor and the Ethernet interface.

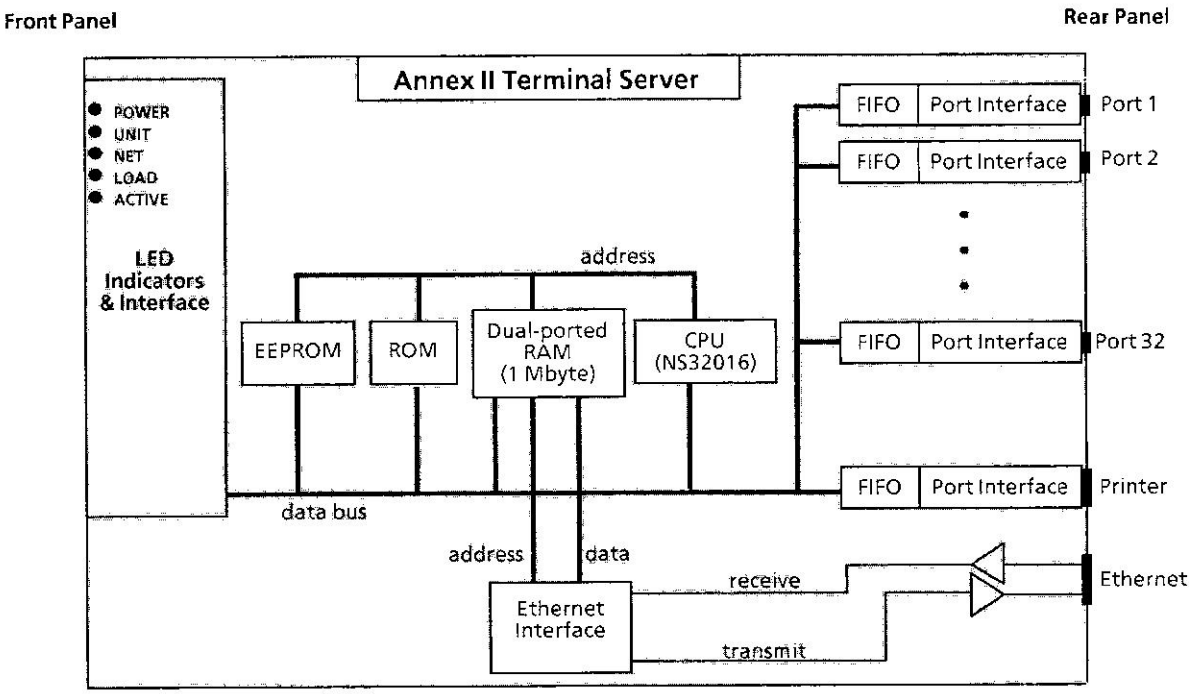


Figure 2–14
Annex II Block Diagram

The Annex II provides one Centronics-type parallel printer port and 16 or 32 serial ports that comply with the RS232 and RS432 standards. Each port permits line speeds ranging from 75 to 38,400 bits per second and each supports an auto-answer modem or hardware flow control (but not both simultaneously). Each port is provided with a 256-character hardware FIFO that greatly reduces interrupt overhead.

The Annex II has no disk. Runtime software is automatically down-loaded at power-up or reset over the Ethernet (or over one of the Annex II's serial lines) from a UNIX host computer running a file server utility program provided with the Annex II. To

minimize the effect of any software-generated downtime, the Annex II contains a watchdog timer that must be reset periodically by software. If the Annex II locks up and ceases running its control code properly, this timer times out, resetting the Annex II. Reset transfers control to the Annex II ROM code which automatically produces a crash dump over the network and then requests RAM code from a cooperating host and restarts the Annex II. This diskless, self-governing operation makes the Annex II capable of fully unattended operation.

The Annex II's Ethernet interface works with IEEE 802.3 or Ethernet Revision 2.0 transceivers.

Annex-X.25 Gateway

The Annex-X.25 Gateway is the hardware platform of a combined hardware/software product that enables communication between an X.25 Public Data Network (PDN) and computer systems that implement TCP/IP over a standard Ethernet local area network (LAN). The Annex-X.25 product supports a client/server type of interaction with the Ethernet hosts and can deal with multiple hosts simultaneously. It exchanges TCP/IP messages with the hosts and X.25 packets with the PDN, assuming responsibility for all protocol conversions.

In addition to the foregoing characteristics, the X.25 gateway software enables a Multimax system to function as an X.25 host and to support remote users connected to the PDN.

Annex-X.25 Gateway Hardware

The Annex-X.25 is a a powerful stand-alone computer containing a 32-bit microprocessor, memory, and hardware-assisted I/O. The Annex-X.25 contains read-only memory (ROM) containing firmware that performs internal and Ethernet diagnostics and down loads the server's operating code over the Ethernet from a cooperating host. The server also contains electrically-erasable programmable read-only memory (EEPROM) that provides non-volatile storage for configuration parameters.

The Annex-X.25 hardware platform is diskless. At boot time, the Annex-X.25 operating code is down-loaded, at the hardware's request, from a cooperating host on the network. A watchdog timer, reset at regular intervals by properly running operating code, will reboot the hardware in the unlikely event that code execution should hang and be unable to perform the timer reset operation. The result is that the Annex-X.25 can run without human intervention of any sort for very long periods of time.

X.25 Gateway Software

Software provided with the Annex-X.25 product includes:

- An X.29 server (Annex-X.25-resident) for remote Packet Assembler/Disassembler (PAD) users.
- An asynchronous PAD (Annex-X.25-resident) that implements CCITT Recommendation X.3 for interactive terminals.
- A programmatic interface (host-resident) for host processes that require an interface to X.25 level 3. This interface consists of a set of library routines that can be implemented as applications providing client/server interaction between the host and the X.29 server in the Annex-X.25.