Design of the AlphaServer Multiprocessor Server Systems

by

Fidelma M. Hayes

ABSTRACT

Digital's AlphaServer multiprocessor systems are high-performance servers that combine multiprocessing technology with PC-style I/O subsystems. The system architecture allows four processing nodes, four memory nodes (up to a maximum of 2 GB), and two I/O nodes. All nodes communicate through a system bus. The system bus was designed to support multiple generations of Alpha processor technology. The architecture can be implemented in different ways, depending on the size of the system packaging.

INTRODUCTION

The AlphaServer 2100 (large pedestal) and the AlphaServer 2000 (small pedestal) servers from Digital combine multiprocessing Alpha technology with an I/O subsystem traditionally associated with personal computers (PCs). The I/O subsystem in the AlphaServer systems is based on the Peripheral Component Interconnect (PCI) and the Extended Industry Standard Architecture (EISA) buses. All AlphaServer products, including the AlphaServer 2100 cabinet version, share common technology and support at least three generations of the Alpha processor. In addition, the servers support three operating systems: Microsoft's Windows NT version 3.5, and Digital's DEC OSF/1 version 3.0 (and higher) and OpenVMS version 6.1 (and higher).

The AlphaServer systems are designed to be general-purpose servers for PC local area network (LAN) and database applications. All models of the system use a common multiprocessing bus interconnect that supports different numbers of nodes, depending on the system configuration. The systems share a common CPU, memory, and I/O architecture. The number of CPUs, the amount of memory, the number of I/O slots, and the amount of internal storage vary depending on the mechanical packaging. The flexibility of the architecture allows the quick development of new and enhanced systems.

This paper discusses the transformation of a set of requirements into high-performance, cost-effective product implementations. The following section describes the evolution of the AlphaServer design from an advanced development project into a design project. The paper then describes the CPU module, the multiprocessor system bus, and the memory module. Subsequent sections discuss module and silicon technology and the high-availability features incorporated into the design. The paper ends with a performance summary and conclusions about the project.

CONCEPT DEVELOPMENT

The engineering investigations of a client-server system originated from a business need that Digital perceived when it introduced the first systems to incorporate the Alpha technology in late 1992. Among Digital's first products in the server market were the DEC 4000 high-performance departmental system, the DEC 3000 deskside workstation/server, and the EISA-based Alpha PC. The lack of an explicitly identified, general-purpose system for the mid-range system market generated many requests from Digital's MicroVAX II system customers. Requests from these customers propelled the AlphaServer product development effort.

From the beginning of the project, two major constraints were evident: The schedule required a product by mid-1994, and the budget was limited. Accordingly, the product team was required to leverage other developments or to find newer, less costly ways of achieving the product goals. Work on the AlphaServer systems started as a joint effort between an advanced development team and a business planning team. The business team developed market profiles and a list of features without which the system would not be competitive. The business team followed a market-driven pricing model. The profit expected from the system dictated the product cost for the system. This cost is referred to as "transfer cost." The business team's cost requirement was critical: if it could not be met, the project would be canceled. Furthermore, the entry-level system was required to

1. Support at least two CPUs, with performance for a single CPU to yield 120 SPECmarks and 100+ transactions per second (TPS) on the TPC-A benchmark.

2. Support at least 1 gigabyte (GB) of memory.

3. Support multiple I/O buses with at least six option slots supported on the base system.

4. Provide high-availability features such as redundant power supplies, redundant array of inexpensive disks (RAID), "warm swap" of drives, and clustering.

5. Provide a number of critical system connectivity options, including Ethernet, fiber distributed data interface (FDDI), and synchronous controllers.

6. Support the Windows NT, the DEC OSF/1, and the OpenVMS operating systems.

Given these criteria, the engineering team decided to base the development of the new server on concepts taken from two Digital products and combine them with the enclosures, power supplies, and options commonly associated with PCs. The DEC 4000 server is a multiprocessor system with a Futurebus+ I/O subsystem; it provided the basis for the multiprocessor bus design.[1] The DECpc 150 PC is a uniprocessor system with an EISA I/O subsystem; it provided a model for designing an I/O subsystem capable of running the Windows NT operating system. The engineering team chose PC-style peripherals because of their low cost.

A strategic decision was made to incorporate the emerging PCI bus into the product in addition to the EISA bus. Major PC vendors had expressed high interest in its development, and they believed the PCI bus would gain acceptance by the PC community. The PCI bus provides a high-performance, low-cost I/O channel that allows connections to many options such as small computer systems interface (SCSI) adapters and other common PC peripherals.

After the initial design had been completed, changing market and competitive environments imposed additional requirements on the design team.

1. The initial transfer cost goal was reduced by approximately 13 percent.

2. Support for a maximum of four processor modules was necessary.

To meet these new requirements, the design team had to modify the system design during the product development phase.

SYSTEM OVERVIEW

The base architecture developed for Digital's AlphaServer multiprocessor systems allows four processing nodes, four memory nodes (up to a maximum of 2 GB), and two I/O nodes. All nodes communicate through a system bus. The system bus was designed to support multiple generations of Alpha processor technology. The architecture can be implemented in different ways, depending on the size of the system packaging. It is flexible enough to meet a variety of market needs. Two implementations of the architecture are the AlphaServer 2100 and the AlphaServer 2000 products. Figure 1 is a block diagram of the AlphaServer 2100 implementation of the architecture.

[Figure 1 (Block Diagram of the AlphaServer 2100 System Architecture) is not available in ASCII format.]

In the AlphaServer 2100 large pedestal server, the system bus supports eight nodes. It is implemented on a backplane that has seven slots. The seven slots can be configured to support up to four processors. Due to the number of slots available, the server supports only 1 GB of memory when four processors are installed. It supports the full 2 GB of memory with three processors or less. The eighth node, which is the system bus-to-PCI bridge, is resident on the backplane. This provides a 32-bit PCI bus that operates at 33 megahertz (MHz). It is referred to as the primary PCI bus on the system.

A second I/O bridge can be installed in one of the system bus slots. This option, which will be available in 1995, will provide a 64-bit PCI bus for the system. A 64-bit PCI is an extension of a 32-bit PCI bus with a wider data bus. It operates at 33 MHz and is completely interoperable with the 32-bit PCI specification.[2] Options designed for the 32-bit PCI bus will also work in a 64-bit PCI slot.

EISA slots are supported through a bridge on the primary PCI bus on the system. Only one EISA bus can be supported in the system since many of the addresses used by EISA options are fixed.[3] Support of a single EISA bus is not perceived as an issue given the migration from the EISA bus to the much higher performing PCI bus. The maximum supported bandwidth on an EISA bus is 33 megabytes per second (MB/s) versus the maximum bandwidth on a 32-bit PCI bus of 132 MB/s. The EISA bus is used in the system for support of older adapters that have not migrated to PCI.

The AlphaServer 2000 small pedestal system supports five nodes on the system bus. The backplane provides four system bus slots, allowing a maximum configuration of two processor modules and two memory modules. The system bus-to-PCI bridge resides on the backplane and is the fifth node. A system bus slot can also be used to support the optional second I/O bridge.

The AlphaServer 2100 cabinet system is a rackmountable version of the large pedestal AlphaServer 2100 system. The rackmountable unit provides a highly available configuration of the pedestal system. It incorporates two separate backplanes. One backplane supports eight system bus nodes that are implemented as seven system bus slots. The eighth node (the system bus-to-PCI bridge) resides on the backplane. The second backplane provides the I/O slots. The number and configuration of I/O slots are identical to the AlphaServer 2100 pedestal system. The rackmount unit provides minimal storage capacity. Additional storage is supported in the cabinet version through StorageWorks shelves. These storage shelves can be powered independently of the base system unit, providing a highly available configuration.

Table 1 gives the specifications for the AlphaServer 2100 and the AlphaServer 2000 pedestal systems. Information on the cabinet version is not included because its characteristics are similar to the AlphaServer 2100 large pedestal version. All multiprocessing members of the AlphaServer family use the same processor and memory modules and differ only in system packaging and backplane implementations. This illustrates the flexibility of the architecture developed for the system and decreases the development time for new models.

Table 1 AlphaServer System Specifications

Specifications	Large Pedestal AlphaServer 2100 System	Small Pedestal AlphaServer 2000 System	Comments
Height, inches	27.6	23.8	
Width, inches	16.9	16.9	
Depth, inches	31.9	25.6	
Maximum DC power output, watts per supply	600	400	Two possible per system in either redundant or current shared mode
Number of system slots	7	4	
Number of processors supported	4	2	
Minimum memory	32 MB	32 MB	
Maximum memory	2 GB	640 MB	
Embedded I/O controllers supported	1	1	
Optional I/O controllers supported	1	1	
32-bit PCI slots	3	3	
64-bit PCI slots (on separate I/O controller module)*	2	2	
EISA slots	8	7	
Serial ports	2	2	
Parallel port	1	1	
Ethernet ports (AUI and 10Base-T)	1	Not integral to system	Up to 18 total network ports supported on system via PCI and EISA

			options
SCSI II controller	1	1	
Removable media bays	3	2	
Internal warm-swap drive slots	16	8	

* Future option

CPU MODULE

The CPU module contains an Alpha processor, a secondary cache, and bus interface application specific integrated circuits (ASICs). As previously mentioned, the system architecture allows multiple processor generations. Multiple variations of the processor module are available for the system, but different variations cannot be used in the same system. Software has timing loops that depend on the speed of the processor and cannot guarantee synchronization between processors of different speeds. The CPU modules provide a range of performance and cost options for the system owner.

The cost-focused processor module uses the Alpha 21064 processor operating at 190 MHz. This chip was designed with Digital's fourth-generation complementary metal-oxide semiconductor (CMOS) technology. It has separate on-chip caches for instruction and data. The instruction cache holds 8 kilobytes (KB) of memory, and the data cache holds 8 KB. The 1-MB second-level data cache is implemented in 15-nanosecond (ns) static random-access memory (SRAM) devices. It is a write-back, direct-mapped cache. The access time to the second-level cache is a multiple of the CPU clock cycle. The use of 15-ns SRAMs resulted in a read-and-write cycle time of 26.3 ns to the second-level cache. This is a five-times multiple of the CPU cycle time. The additional 11.3 ns is needed for round-trip etch delay and address buffer delay. The use of 12-ns SRAMs was considered, but the read-and-write cycle time would have to decrease to 21 ns to improve performance. The reduction of 3 ns was not sufficient to meet the timing requirements of the module; therefore, the less costly 15-ns SRAMs were used.

Higher performance processor modules are also available for the system. These modules are based on the Alpha 21064A processor, which was designed using fifth-generation CMOS technology. The Alpha 21064A processor module operates at 275 MHz. The processor has separate on-chip instruction and data caches. The 16-KB instruction cache is direct mapped, and the 16-KB data cache is a 2-way, set-associative cache. The backup cache holds 4 MB of memory. The combination of higher processor speed, larger internal on-chip caches, and a large second-level cache reduces the number of accesses to main memory and processes data at a higher rate. As a result, the performance of the system is increased by approximately 20 percent.

MULTIPROCESSOR SYSTEM BUS

The technology developed for the system bus in the DEC 4000 departmental server provided the basis for the multiprocessor bus designed for the AlphaServer system.[1] The system bus in the DEC 4000 product has the following features:

1. The 128-bit multiplexed address and data bus operates at a 24-ns cycle time. The bus runs synchronously.

2. The bus supports two CPU nodes, four memory nodes, and a single I/O node.

3. The bus supports addressing for block transfer only. A block is 32 bytes of data.

4. I/O is treated as either primary or secondary. Primary I/O refers to devices that could respond without stalling the system bus. This designation is restricted mainly to control and status registers (CSRs) that exist on system bus nodes, e.g., the I/O bridge.

5. All I/O on remote buses is referred to as secondary I/O and is accessed via a mailbox protocol. Mailboxes were invented to hide slow accesses to CSRs on remote I/O buses.

   A CSR read could potentially take 1 to 10 microseconds, which is very slow relative to the processor cycle time. The bus is "nonpended," which means it would stall during a slow access. When a bus stalls, all accesses to CPUs and memories have to wait until the CSR access is complete. This could cause data to back up and potentially overflow. To avoid this state, either the system bus or the software device driver has to be pended.

   A mailbox is a software mechanism that accomplishes "device driver pending." The processor builds a structure in main memory called the mailbox data structure. It describes the operation to be performed, e.g., CSR read of a byte. The processor then writes a pointer to this structure into a mailbox pointer register. The I/O node on the system bus reads the mailbox data structure, performs the operation specified, and returns status and any data to the structure in memory. The processor then retrieves the data from this structure and the transaction is complete. In this way, the mailbox protocol allows software pending of CSR reads; it also allows the software to pass byte information that is not available from the Alpha 21064A processor.[4,5]

Changes to the System Bus

Although the DEC 4000 system bus provided many features desirable in a multiprocessor interconnect, it did not meet the system requirements defined during the concept phase of the AlphaServer project. Two major hurdles existed. One was the lack of support for four CPUs and multiple I/O nodes. A second, more important issue was the incompatibility of the mailbox I/O structure with the Windows NT operating system.

The initial port of the Windows NT operating system to the DECpc 150 PC assumed direct-mapped I/O. With direct mapping the I/O is physically mapped into the processor's memory map, and all reads/writes to I/O space are handled as uncached memory accesses. Clearly, this was incompatible with the nonpended bus, which assumes the use of mailboxes. Consequently, the designers studied the advantages and disadvantages of using mailboxes to determine if they should be supported in the Windows NT operating system. They found that the software overhead of manipulating the mailbox structure made CSR accesses approximately three times slower than direct accesses by the hardware. Thus the CPU performing the I/O access waits longer to complete. For this reason, the designers chose not to use mailboxes.

The designers also had to ensure that the system bus would be available for use by other processors while the I/O transaction was completing. To satisfy this requirement, they added a retry mechanism to the system bus. The retry support was very simple and was layered on top of existing bus signals. A retry condition exists when the CPU initiates a cycle to the I/O that cannot be completed in one system bus transaction by the I/O bridge. The CPU involved in the transaction is notified of the retry condition. The CPU then "backs off" the multiprocessor bus and generates that transaction some period of time later. Other processor modules can access memory during the slow I/O transaction. The retry procedure continues until the I/O bridge has the requested data. At that stage, the data is returned to the requesting CPU.

Byte Addressing.

Byte granularity had been handled in the mailbox data structure. After the direct-mapped I/O scheme was adopted, the designers had to overcome the lack of byte addressability in the Alpha architecture. Therefore, the designers participated in a collaborative effort across Digital to define a mechanism for adding byte addressability in the Alpha architecture. The new scheme required the use of the four lower available Alpha Ad:[08:05] address bits to encode byte masks and lower order address bits for the PCI and EISA buses. For more details, see the paper on the AlphaServer 2100 I/O subsystem in this issue.[6]

The designers required a redefinition of the address map. All I/O devices are now memory mapped. The Alpha 21064A processor has a 34-bit address field that yields an address space of 16 GB. This 16-GB address region may be subdivided into 4-GB quadrants. Each quadrant can be individually marked as cacheable or noncacheable memory. The DEC 4000 system architecture split the 16-GB region in half: 8 GB was allocated as cacheable memory space and the remaining 8 GB as noncacheable space. Memory-mapped I/O devices are mapped into noncacheable space. The decision to support multiple I/O buses in the new systems together with the decision to memory map all I/O (i.e., no mailbox accesses) yielded a noncacheable memory requirement in excess of the 8 GB allocated in the DEC 4000 system. Therefore the designers of the AlphaServer systems changed the address map and allocated a single quadrant (4 GB) of memory as cacheable space and the remaining 12 GB as noncacheable. These 12 GB are used to memory map the I/O.

Arbitration. The bus used in the DEC 4000 system supports two CPU nodes and a single I/O node. To achieve the AlphaServer product goals of multiple I/O bridges and multiple CPU nodes, the designers changed the address map to accommodate CSR space for these extra nodes and designed a new arbiter for the system. The arbiter includes enhanced functionality to increase the performance of future generations of processors. Some key features of the arbiter are listed below.

1. The arbiter is implemented as a separate chip on all processor modules. The logic was partitioned into a separate chip to accommodate a flexible architecture and to allow additional arbitrating nodes in the future. As many as four arbiters can exist in the system. Only one arbiter is enabled in the system. It is on the processor installed in slot 2 of the system bus.

2. I/O node arbitration is interleaved with CPU node arbitration. The arbitration is round robin and leads to an ordering scheme of CPU 0, I/O, CPU 1, I/O, CPU 2, I/O, CPU 3, I/O. This scheme attempts to minimize I/O latency by ensuring many arbitration slots for I/O devices. Processors still have more than adequate access to the system bus due to the nature of I/O traffic (generally bursts of data in short periods of time). On an idle bus, the arbiter reverts to a first-come, first-served scheme.

3. The arbiter implements an exclusive access cycle. This allows an arbitrating node to retain the use of the system bus for consecutive cycles. This cycle is used by the I/O bridge in response to a PCI lock cycle. A PCI lock cycle may be generated by a device that requires an atomic operation, which may take multiple transactions to complete. For example, the AlphaServer 2100 and AlphaServer 2000 systems use a PCI-to-EISA bridge chip set (Intel 82430 chip set).[7] This chip set requests a

lock cycle on PCI when an EISA device requires an atomic read-modify-write operation.

The use of atomic read-modify-write operations is common in older I/O adapter designs. The I/O bridge on the system bus requests an exclusive access cycle from the arbiter. When it is granted, all buffers in the path to memory are flushed and the device has exclusive use of the PCI and the system bus until its transaction is completed. The use of this mode is not recommended for new adapter designs due to the unfair nature of its tenure on the system bus. It was implemented in the AlphaServer product design to support older EISA devices.

MEMORY MODULE

Main memory is accessed over the system bus either by processors (after missing in their on-board caches) or by I/O nodes performing direct memory access (DMA) transactions. They are called commanders.

The memory controller incorporates a number of performance-enhancing features that reduce latency in accessing the dynamic RAM (DRAM) array. One concept used is called a stream buffer. Stream buffers reduce the read latency to main memory. Reads to main memory normally require 9 to 10 cycles on the system bus, depending on the speed of DRAMs in the array. The use of stream buffers reduces this time to 7 cycles. The stream buffers provide a facility to load data fetched from the DRAM array prior to the receipt of a read request for that data.

A stream is detected by monitoring the read addresses from each commander on the system bus. The logic simply keeps a record of the memory addresses of the previous eight read transactions from each commander and compares each subsequent read address to see if the new address is contiguous to any of the recorded addresses. If a new address is determined to be contiguous to any of the previous eight addresses, a new stream is declared. As a result, one of the stream buffers is allocated to a new stream.

A stream buffer is implemented as a four-deep, first-in, first-out (FIFO) buffer. Each entry in the FIFO buffer is 32 bytes, which is equivalent to the system bus line size. Each memory module contains four stream buffers that can be allocated to different commanders. A least recently used (LRU) algorithm is used to allocate stream buffers. When a new stream is detected, or an existing stream is empty, the stream buffer fills from the DRAM array by using successive addresses from the head of the stream. After a buffer has been allocated and some amount of data has been placed in the FIFO buffer, "hit" logic compares incoming read addresses from the system bus to the stream address. If a comparison of these two addresses is successful, read data is delivered from the memory module without incurring the latency of accessing the DRAM array.

An invalidation scheme is used to ensure that the stream buffers stay coherent. Write cycle addresses are checked to see if they coincide with a stream buffer address. If the write address is equal to any address currently in the stream buffer, that entire stream buffer is declared invalid. Once it is invalidated, it can be reallocated to the next detected stream.

Writes to main memory complete on the system bus in six cycles, which is achieved using write buffers in the memory controller. The write transactions are essentially "dump and run." The total write buffering available in each memory module is 64 bytes, which is large enough to ensure that the system bus never has to stall during a write transaction.

The implementation of the memory module differs from the AlphaServer 2100 to the AlphaServer 2000 system. Both memory modules contain the same memory controller ASICs, but the implementation of the DRAM array is different. Due to space constraints on the AlphaServer 2100, the DRAM array was implemented as a flat, two-sided surface-mount module. On the AlphaServer 2000, single in-line memory modules (SIMMs) were used for the DRAM array. Memory module capacities vary from 32 MB to 512 MB. The AlphaServer 2100 system provides four system bus slots that can be populated with memory modules. The maximum supported configuration is 2 GB with four memory modules. This limits the maximum system configuration to three processors since one of the processor slots must be used as a memory slot. The AlphaServer 2000 system provides two system bus slots that can be populated with memory. The maximum memory supported in this system is 640 MB. This configuration consists of one 512-MB module and one 128-MB module. The maximum memory constraint is dictated by the power and cooling available within this system package. The AlphaServer 2000 still supports two processor modules when configured with maximum memory. Figure 2 shows a block diagram of the AlphaServer 2000 memory module.

[Figure 2 (Block Diagram of the AlphaServer 2000 Memory Module) is not available in ASCII format.]

TECHNOLOGY CHOICES

This section briefly discusses some of the decisions and trade-offs made concerning module and silicon technology used in the systems.

Module Technology

The designers partitioned the logic into modules for two reasons: (1) Removable processor and memory modules allow for installation of additional memory and processors and (2) They also allow for easy upgrade to faster processor speeds. Since modularity adds cost to a system, the designers decided that the I/O subsystem

(EISA and PCI logic) should reside on the backplane. They deviated from this strategy for the AlphaServer 2100 system design because the PCI-to-EISA bridge was a new, unfamiliar design. Fixing any problems with this chip set or any of the supporting logic would have required a backplane upgrade, which is a time-consuming effort. For this reason, the engineers chose to build an I/O module for the AlphaServer 2100 system that contained the PCI-to-EISA bridge; associated control logic; controllers for mouse, keyboard, printer, and floppy drive; and the integral Ethernet and SCSI controllers. This module was eliminated in the AlphaServer 2000 system due to the design stability of the I/O module.

The Metral connector specified by the Futurebus+ specification was chosen for the system bus implementation on the DEC 4000 product. This choice was consistent with the design of the DEC 4000 server, which is a Futurebus+ system. Cost studies undertaken during the initial design of the AlphaServer 2100 system showed that the cost per pin of the Metral connector was high and added a significant cost to the system. The team decided to investigate the use of either the PCI or the EISA connector for the system bus, since both connectors are used widely in the system. The PCI connector is actually a variant of the MicroChannel Architecture (MCA) connector used in microchannel systems. SPICE simulations showed that it performed better than the Metral connector on the Futurebus+.[8] The team chose a 240-pin version of the connector for implementation because it met the system requirements and had a low cost.

Due to the choice of the MCA connector, the board thickness was limited to a maximum of 0.062 inches. An 8-layer layup was chosen for the module technology. The processor modules had a requirement for both a 5.0-V supply and a 3.0-V supply. The designers chose a split plane to distribute the power rather than two separate power planes for each voltage. Routing high-speed signals across the split was minimized to reduce any emissions that might arise from using a split plane. Testing later validated this approach as emissions from this area were minimal.

Silicon Technology

The system partitioning required the design of four ASICs. These were the CPU bus interface ASIC, the memory bus interface ASIC, the system arbiter, and the system bus-to-PCI bridge. The DEC 4000 implementation of the Futurebus+ used an externally supplied gate-array process that was customized to meet the performance needs of the bus and the performance goals of the first Alpha systems. Gate-array costs are determined by the number of chips that are produced on the chosen gate-array process. The volume of chips produced by the gate-array process for the DEC 4000 system was low because the process was specially adjusted for that system application. As a result, the volume of chips was directly proportional to the volume of the DEC 4000 systems built.

Therefore, the cost per component produced by this process was relatively high.

If they had used this customized gate-array process, the designers of the AlphaServer product could not have met their cost goals. They needed a more generic process that could produce chips that many system vendors could use. This would ensure that the line utilization was high and that the cost per component was low. Therefore, they changed the technology to one that is standard in the industry. Gate-array process technology had evolved since the DEC 4000 design, and a standard technology that was capable of meeting the system timing requirements was available. Extensive SPICE simulations verified the process capability. ASICs that were implemented with this process had no difficulty meeting the bus timing.[8]

Another interesting feature of the analog design on the AlphaServer 2100 system involves the support of 11 loads on the PCI. The PCI specification recommends 10 loads as the "cookbook" design.[2] The system requirement on the AlphaServer 2100 was to support three PCI slots, the integral PCI-Ethernet chip, the NCR810 (PCI-to-fast-SCSI controller), and the PCI-to-EISA bridge. Each PCI connector has been modeled to be equivalent to two electrical loads. Taking account of the system bus-to-PCI bridge and the additional load contributed by the I/O module connector yielded a PCI bus with 11 electrical loads. Extensive SPICE simulations of the bus and careful routing to ensure a short bus guaranteed that the new design would meet the electrical specifications of the PCI bus.[8]

SYSTEM START-UP

The design team incorporated many availability features into the AlphaServer 2100 and AlphaServer 2000 servers. These included support of "hot-swap" storage devices that can be removed or installed while the system is operating, error correction code (ECC)-protected memory, redundant power supplies, and CPU recovery. Perhaps the most interesting part of the design for availability was the emphasis on ensuring that the system had enough built-in recovery and redundancy to allow it to remain in a usable or diagnosable state. Large systems sometimes have complicated paths in which to access the initial start-up code, and a system failure in that path can leave the owner with no visible failure indication. Moreover, in a multiprocessor system with more than one CPU installed, it is highly desirable to initialize the resident firmware and the operating system even if all CPUs are not in working order. The AlphaServer 2100 and 2000 systems employ two schemes to help achieve this goal.

The start-up code for the AlphaServer 2100 and AlphaServer 2000 systems is located in flash read-only memory (ROM), which resides on a peripheral bus behind the PCI-to-EISA bridge. In starting up a multiprocessing operating system, only one processor is designated to access the start-up code and initialize the operating system. This is referred to as the primary processor. Accessing the start-up code requires the processor, system bus, memory, and most of the I/O subsystem to be functional.

The AlphaServer systems have a number of features that help make the start-up process more robust. Each processor module contains a separate maintenance processor implemented as a simple microcontroller that connects to a serial bus on the system. The serial bus is a two-wire bus that has a data line and a clock line. On power-up the processor module performs a number of diagnostic tests and logs the results in an electrically erasable programmable read-only memory (EEPROM) on the module. This EEPROM resides on the serial bus. If a CPU fails one of its power-up tests or if it has an error logged in its EEPROM, then it is not allowed to be the primary processor. Assume that four CPUs are installed in the system; if only CPU 0 fails, then CPU 1 is the primary processor. If CPU 0 and CPU 1 fail, then CPU 2 is the primary processor. If CPU 0, CPU 1, and CPU 2 fail, then CPU 3 is the primary processor. If all four CPUs fail, then CPU 0 is the primary processor. If any one of the CPUs fails, a message is displayed on the operator control panel to inform the user that there is a problem. Any secondary CPU that has failed is disabled and will not be seen by the firmware console or the operating system. The primary processor then uses the system bus to access the start-up code in the flash ROM.

The flash ROM may contain incorrect data. The flash ROMs on many systems have a program update, and errors from a power spike or surge can be introduced into the ROM code during the update procedure. User error is another common way to introduce data error; for example, a user can accidentally press a key while the update program is running. Flash ROMs can also fail from intrinsic manufacturing faults such as current leakage, which will eventually convert a stored "1" into a stored "0," thus corrupting the program stored in the flash ROMs. Many techniques in the industry partially solve the problem of corrupted flash ROM data. One well-known technique uses a checksum and reports an error to the user if the data is not correct. Another technique provides a second set of flash ROMs and a switch that the user manipulates to transfer control to the new set in the event of a failure. The designers studied many previously used methods, but rejected them since they required intervention by the user.

In the AlphaServer 2100 and the AlphaServer 2000 system design, the design team implemented a scheme that did not require user intervention in the event of flash ROM corruption. The system has 1 MB of flash ROM of which the first 512 KB contain the system initialization code. This code is loaded into main memory, and many data integrity tests are performed. These include single and multiple bit parity checks, various data correction code checking, and a checksum calculation. The processor detects an error if the checksum calculation fails, i.e., if the calculated value is not equal to the stored value. The processor then writes a value to a register on the I/O module, which automatically changes the address pointing to the flash ROM to a second bank of flash ROM. This combination of hardware and software support provides a way for the AlphaServer 2100 system user to overcome any flash ROM corruption.

DESIGN CONSIDERATIONS FOR THE AlphaServer 2000 SYSTEM

The design of the AlphaServer 2000 small pedestal system followed the AlphaServer 2100 system. Market pressures dictated the need for a smaller system with a lower entry-level cost. The introduction of the smaller server was scheduled to coincide with the release of the Windows NT version 3.5 operating system.

An examination of the AlphaServer 2100 development schedule revealed the following interesting points: (1) System power on occurred nine months after the team was formed; (2) Initial system shipments occurred eight months later; (3) The eight-month time period was spent mainly in porting and qualifying operating system software.

Based on these facts, the system designers believed that the key to reducing the time-to-market of the AlphaServer 2000 system was to eliminate the dependency on synchronizing the design schedule with an operating system release. Consequently, the new system could not require any software changes at the operating system level. Any changes would have to be transparent to software. To achieve this, the designers took advantage of a new feature in the DEC OSF/1 and the OpenVMS operating systems called dynamic system recognition (DSR).

A DSR machine is defined as a machine that requires no new software development. Operating systems, however, require licensing; this information is dependent upon the system model number. There are two components to building a DSR machine.

1. A programmer's view of the machine must be a subset of an already supported machine. In the case of the AlphaServer 2000, the designers decided to make it a subset of the AlphaServer 2100. A clear understanding of how the operating systems initialized the AlphaServer 2100 system was critical to understanding what changes could be made. A joint team of hardware and software engineers examined various pieces of the code to identify the areas of the system design that could be changed. Investigations revealed that the system bus configuration code for the AlphaServer 2100 is somewhat generic. It assumes a maximum of eight nodes, which is the AlphaServer 2100 implementation. The I/O node to the primary PCI bus is expected to be present. The presence of additional processors and memories is detected by reading the CSR space of each module. A module that is present gives a positive acknowledgment. The design team could therefore

reduce the number of system bus slots from seven to four. This had no effect on the software since nonexistent slots would merely be recognized as modules not installed in the system.

The physical packaging of the AlphaServer 2000 also dictated that the number of I/O slots be reduced from 11 (8 EISA and 3 PCI) to 10. Given the industry trend toward PCI, the desirable mix would have been 6 EISA slots and 4 PCI slots. The PCI bus configuration code searched for as many as 32 PCI slots, which is the number allowed by the PCI specification.[2] After careful consideration, the designers determined that the addition of another PCI slot would involve a change in interrupt tables to accommodate the additional interrupts and vectors required by the additional slot. Therefore, the team decided to implement 3 PCI and 7 EISA slots.

2. The other component to building a DSR machine is to provide the system model number to the operating system so that licensing information can be determined. The system resident code that runs at start-up is referred to as the console. The console and the operating systems communicate via a data structure known as the hardware parameter block (HWRPB). The HWRPB is used to communicate the model number to the operating system, which uses this number to provide the correct licensing information.

The AlphaServer 2000 system was completed in approximately nine months. Qualification was not dependent on the operating system schedules. By building a DSR machine, the design team met the project's time-to-market requirements.

PERFORMANCE SUMMARY

Table 2 summarizes the performance of the systems described in this paper. The numbers are heavily influenced by the processor speed, cache, memory, and I/O subsystems. The systems exceeded the performance goals specified at the beginning of the project. In some cases the important benchmarks that had been relevant in the industry changed during the course of system development. In the transaction processing measurement, for example, the TPC-A benchmark was superseded by the TPC-C benchmark.

The AlphaServer 2100 server was the price-performance leader in the industry at the time of its introduction in April 1994. Successive improvements in processor and I/O subsystems should help the AlphaServer 2100 and 2000 products maintain that position in the industry.

Table 2 System Performance

	AlphaServer	AlphaServer

	2100 4/275 System	2000 4/200 System
SPECint92*	200.1	131.8
SPECfp92*	291.1	161.0
AIM III+		
Number of AIMs	227.5	177.5
User loads	1941.2	1516.0
Estimated TPS++	850	660

Notes:
* Single-processor system only
+ Dual-processor system only
++ TPS is an abbreviation for transactions per second. These numbers are estimated for a quad-processor system using OpenVMS version 6.1 running Rdb.

CONCLUSIONS

The design team exceeded all the product requirements set at the beginning of the AlphaServer project. The transfer cost of the final product was 10 percent better than the goal. The reduced cost was achieved despite the erratic price levels for DRAMs, which were much higher in 1994 than predicted in late 1992.

Separate cost targets were established for each portion of the system, and each design engineer was responsible for meeting a particular goal. Constant cost reviews ensured that variances could be quickly addressed. The requirement to run three operating systems quickly expanded the size and scope of the project. The operating system developers became an integral part of the design team. Multiple reviews and open communication between the hardware development team and the software groups were essential to managing this work. The hardware team performed system-level testing on all three operating systems. This proved invaluable in tracking down bugs quickly and resolving them in either hardware or software.

The project team delivered the expected performance and functionality on schedule. Development time was allocated for new power and packaging subsystems (using third-party design companies), new modules, new ASICs, new system firmware, and porting of three operating systems. To attain the schedule, development tasks were frozen at the beginning of the project. The tasks were also categorized into three classes: mandatory, nonessential, and disposable. Consequently, engineers were able to make trade-offs when required and maintain the integrity of the product. Another key factor to meeting the schedule was the use of knowledge and technology developed for previous products. This yielded many benefits: less design time, fewer resources required, known simulation environment, and less time to a working prototype.

ACKNOWLEDGMENTS

Many people contributed to the success of this project. They spent long weekends and nights working to a schedule and a set of requirements that many thought were unachievable. The inspired dedication of team members made this project a reality. Although not complete, the following list credits those who were the most deeply involved: Vince Asbridge, Kate Baumgartner, Rachael Berman, Jack Boucher, John Bridge, Bob Brower, Harold Buckingham, Don Caley, Steve Campbell, Dave Carlson, Mike Chautin, Marco Ciaffi, Doug Field, Rich Freiss, Nitin Godiwala, Judy Gold, Paul Goodwin, Tom Hunt, Paul Jacobi, Steve Jenness, Jeff Kerrigan, Will Kerchner, Jeff Metzger, John Nerl, Mike O'Neill, Kevin Peterson, Bryan Porter, Ali Rafiemeyer, Lee Ridlon, Andy Russo, Stephen Shirron, Andrei Shishov, Gene Smith, Kurt Thaller, Frank Touserkani, Vicky Triolo, and Ralph Ware. Special thanks also to our manufacturing team in Massachusetts, Canada, and Scotland.

REFERENCES AND NOTE

1. B. Maskas, S. Shirron, and N. Warchol, "Design and Performance of the DEC 4000 AXP Departmental Server Computing Systems," Digital Technical Journal, vol. 4, no. 4 (Special Issue, 1992): 82-99.

2. PCI Local Bus Specification, Revision 2.0 (Hillsboro, OR: PCI Special Interest Group, Order No. 281446-001, April 1993).

3. E. Solari, ISA and EISA, Theory and Operation (San Diego, CA: Annabooks, 1992).

4. R. Sites, ed., Alpha Architecture Reference Manual (Burlington, MA: Digital Press, Order No. EY-L520E-DP, 1992).

5. DECchip 21064 Microprocessor Hardware Reference Manual (Maynard, MA: Digital Equipment Corporation, Order No. EC-N0079-72, 1992).

6. A. Russo, "The AlphaServer 2100 I/O Subsystem," Digital Technical Journal, vol. 6, no. 3 (Summer 1994): 20-28.

7. 82420/82430 PCIset ISA and EISA Bridges (Santa Clara, CA: Intel Corporation, 1993).

8. SPICE is a general-purpose circuit simulator program developed by Lawrence Nagel and Ellis Cohen of the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.

BIOGRAPHY

Fidelma M. Hayes As an engineering manager in the Server Group, Fidelma Hayes led the development of the AlphaServer 2100 and AlphaServer 2000 systems. Prior to this work, she led the design of the DECsystem 5100. She has contributed as a member of the development team for several projects, including the DECsystem 5800 CPU, the PRISM system design, and the MicroVAX 3100. Fidelma joined Digital in 1984 after receiving a bachelor's degree in electrical engineering from University College Cork, Ireland. She is currently working toward a master's degree in computer science at Boston University.

TRADEMARKS

Alpha, AlphaServer, DEC 4000, DEC OSF/1, DECpc, Digital, MicroVAX, OpenVMS, and StorageWorks are trademarks of Digital Equipment Corporation.

Intel is a trademark of Intel Corporation.

Microsoft is a registered trademark and Windows NT is a trademark of Microsoft Corporation.

SPECfp, SPECint, and SPECmark are registered trademarks of the Standard Performance Evaluation Council.

SPICE is a trademark of the University of California at Berkeley.

TPC-A and TPC-C are trademarks of the Transaction Processing Performance Council.