

Design and Implementation of the AlphaServer 4100 CPU and Memory Architecture

The DIGITAL AlphaServer 4100 system is Digital Equipment Corporation's newest four-processor midrange server product. The server design is based on the Alpha 21164 CPU, DIGITAL's latest 64-bit microprocessor, operating at speeds of up to 400 megahertz and beyond. The memory architecture was designed to interconnect up to four Alpha 21164 CPU chips and up to four 64-bit PCI bus bridges (the AlphaServer 4100 supports up to two buses) to as much as 8 gigabytes of main memory. The performance goal for the AlphaServer 4100 memory interconnect was to deliver a four-multiprocessor server with the lowest memory latency and highest memory bandwidth in the industry by the end of June 1996. These goals were met by the time the AlphaServer 4100 system was introduced in May 1996. The memory interconnect design enables the server system to achieve a minimum memory latency of 120 nanoseconds and a maximum memory bandwidth of 1 gigabyte per second by using off-the-shelf data path and address components and programmable logic between the CPU and the main memory, which is based on the new synchronous dynamic random-access memory technology.

The DIGITAL AlphaServer 4100 system is a symmetric multiprocessing (SMP) midrange server that supports up to four Alpha 21164 microprocessors. A single Alpha 21164 CPU chip may simultaneously issue multiple external accesses to main memory. The AlphaServer 4100 memory interconnect was designed to maximize this multiple-issue feature of the Alpha 21164 CPU chip and to take advantage of the performance benefits of the new family of memory chips called synchronous dynamic random-access memories (SDRAMs). To meet the best-in-industry latency and bandwidth performance goals, DIGITAL developed a simple memory interconnect architecture that combines the existing Alpha 21164 CPU memory interface with the industry-standard SDRAM interface.

Throughout this paper the term latency refers to the time required to return data from the memory chips to the CPU chips—the lower the latency, the better the performance. The AlphaServer 4100 system achieves a minimum latency of 120 nanoseconds (ns) from the time the address appears at the pins of the Alpha 21164 CPU to the time the CPU internally receives the corresponding data from any address in main memory. The term bandwidth refers to the amount of data, i.e., the number of bytes, transferred between the memory chips and the CPU chips per unit of time—the higher the bandwidth, the better the performance. The AlphaServer 4100 delivers a maximum memory bandwidth of 1 gigabyte per second (GB/s).

Before introducing the DIGITAL AlphaServer 4100 product in May 1996, the development team conducted an extensive performance comparison of the top servers in the industry. The benchmark tests showed that the AlphaServer 4100 delivered the lowest memory latency and the highest McCalpin memory bandwidth of all the two- to four-processor systems in the industry. A companion paper in this issue of the *Journal*, "AlphaServer 4100 Performance Characterization," contains the comparative information.¹

This paper focuses on the architecture and design of the three core modules that were developed concurrently to optimize the performance of the entire

memory architecture. These three modules—the motherboard, the synchronous memory module, and the no-external-cache processor module—are shown in Figure 1.

Motherboard

The motherboard contains connectors for up to four processor modules, up to four memory module pairs, up to two I/O interface modules (four peripheral component interconnect [PCI] bus bridge chips total), memory address multiplexers/drivers, and logic for memory control and arbitration.² All control logic on the motherboard is implemented using simple 5-ns 28-pin programmable array logic (PAL) devices and more complex 90-megahertz (MHz) 44-pin programmable logic devices (PLDs) clocked at 66 MHz. Several motherboards have been produced to support various numbers of processor modules, memory modules, and I/O interface modules. The AlphaServer 4100 supports one to four processor modules, one to four memory module pairs (8-GB maximum memory), and one I/O interface module (up to two PCI buses).³

Synchronous Memory Module

The synchronous memory modules are custom-designed, 72-bit-wide plug-in cards installed in pairs to cover the full width of the 144-bit memory data bus. Synchronous memory modules that provide 32 megabytes (MB) to 256 MB per pair were designed using 16-megabit (Mb) SDRAM chips. These memory modules contain nine, eighteen, thirty-six, or seventy-two 100-MHz SDRAM chips clocked at 66 MHz, four 18-bit clocked data transceivers, address fan-out buffers, and control provided by 5-ns 28-pin PALs. To increase the maximum amount of memory in the system, a family of plug-in compatible memory modules was designed, providing up to 2 GB per pair using 64-Mb extended data out dynamic random-access memory (EDO DRAM) chips. These modules contain 72 or 144 EDO DRAM chips controlled by two custom application-specific integrated circuits (ASICs) providing data multiplexing and control, four 18-bit clocked data transceivers, and address fan-out buffers. Consequently, the AlphaServer 4100 memory architecture provides main memory capacities of 32 MB to 8 GB with a minimum latency of 120 ns to any address. This paper concentrates on the implementation of the synchronous memory modules, although the EDO memory modules are functionally compatible. The reconfigurability description later in this paper provides more details of the implementation of the EDO memory modules.

No-External-Cache Processor Module

The no-external-cache processor module is a plug-in card with a 144-bit memory interface that contains one Alpha 21164 CPU chip, eight 18-bit clocked data transceivers, four 12-bit bidirectional address latches, and control provided by 5-ns 28-pin PALs and 90-MHz 44-pin PLDs clocked at 66 MHz. The Alpha 21164 CPU chip is programmed to operate at a synchronous memory interface cycle time of 66 MHz (15 ns) to match the speed of the SDRAM chips on the memory modules. Although there are no external cache random-access memory (RAM) chips on the module, the Alpha 21164 itself contains two levels of on-chip caches: a primary 8-kilobyte (KB) data cache and a primary 8-KB instruction cache, and a second-level 96-KB three-way set-associative data and instruction cache. The no-external-cache processor module was designed to take advantage of the multiple-issue feature of the Alpha 21164 CPU. By keeping the latency to main memory low and by issuing multiple references from the Alpha 21164 CPU to main memory at the same time to increase memory bandwidth, the performance of many applications actually exceeds the performance of a processor module with a third-level external cache.¹ Numerous applications perform better, however, with a large on-board cache. For this reason, the AlphaServer 4100 offers several variants of plug-in compatible processor modules containing a 2-MB, 4-MB, or greater module-level cache. The paper “The AlphaServer 4100 Cashed Processor Module Architecture and Design,” which appears in this issue of the *Journal*, contains more related information.⁴

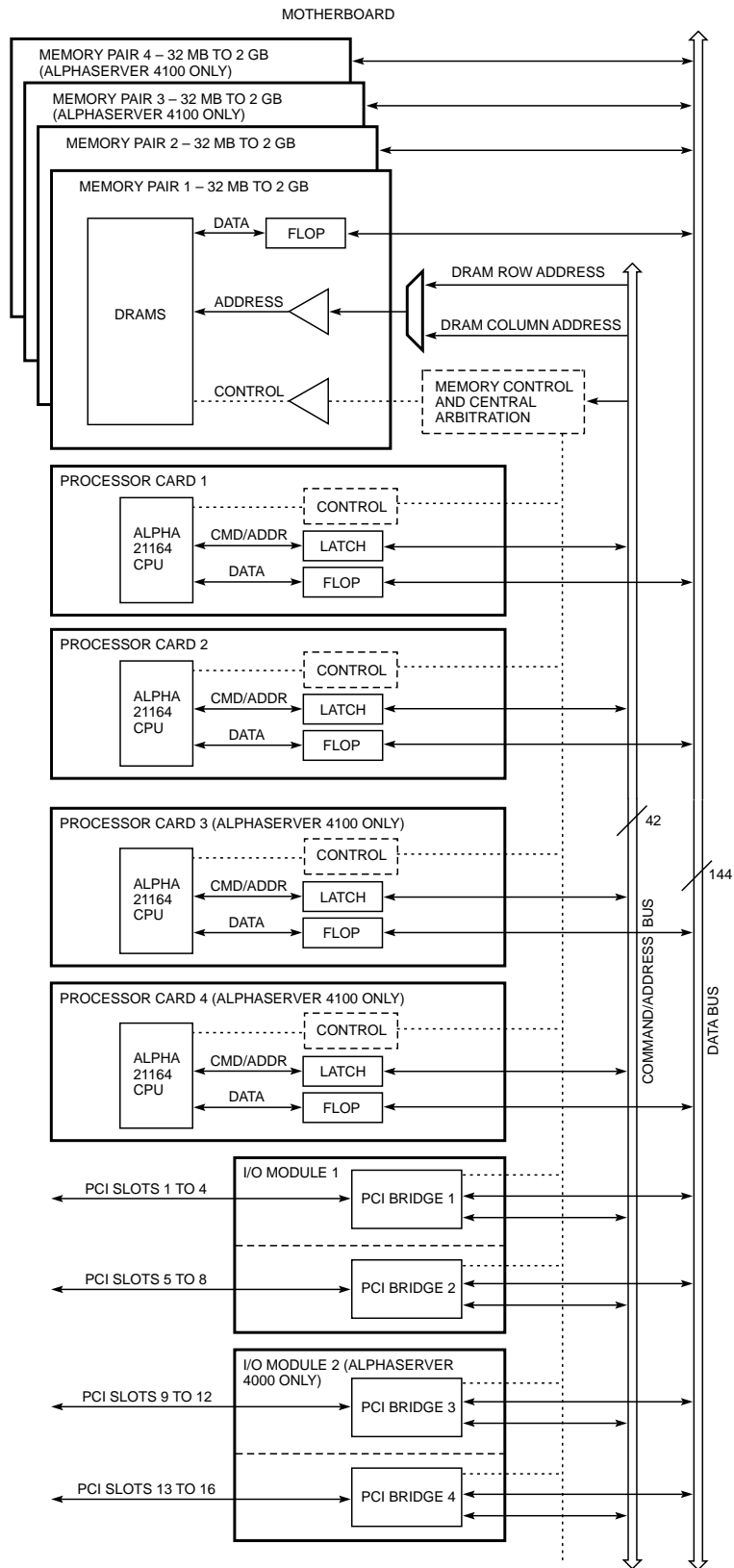
The three components of the core module set were designed concurrently to address five issues:

1. Simple design
2. Quick design time
3. Low memory latency
4. High memory bandwidth
5. Reconfigurability

Simple Design

The Alpha 21164 CPU chip is based on a reduced instruction set computing (RISC) architecture, which has a small, simple set of instructions operating as fast as possible. AlphaServer 4100 designers set the same goal of simplicity for the rest of the server system.

The AlphaServer 4100 interconnect between the CPU and main memory was optimized for the Alpha 21164 chip and the SDRAM chip. To keep the design simple, only off-the-shelf data path and address components and reprogrammable control logic devices were placed between the Alpha 21164 and SDRAM



Note that the AlphaServer 4000 system contains the same CPU-to-memory interface as the AlphaServer 4100 but supports half the number of processors and memory modules and twice the number of PCI bridges. The AlphaServer 4000 motherboard was designed at the same time as the AlphaServer 4100 motherboard but was not produced until after the AlphaServer 4100 motherboard was available.

Figure 1
AlphaServer 4100 Memory Interconnect

chips. The designers removed excess logic and hardware features, minimized the “glue” logic between the CPU chip and main memory, reduced memory latencies as much as possible, and used custom ASICs only when necessary.

Data Path between the CPU and Memory

The external interface of the Alpha 21164 chip provides 128 bits of data plus 16 bits of error-correcting code (ECC), thus enabling single-bit error correction and multiple-bit error detection over the full width of the data path, which is shown in Figure 2. These 144 signals are connected to eight 18-bit bidirectional transceivers on the processor module. As illustrated in Figure 1, the motherboard connects up to four processor modules and up to four memory module pairs. Each memory module contains 72 bits of information; therefore, a pair of memory modules is required to provide the necessary 144 data signals. Each pair of memory modules contains eight additional 18-bit bidirectional transceivers that are connected directly to a number of SDRAM chips. The data transceiver used on the processor module and on the memory module is the 56-pin Philips ALVC162601 in a 14-millimeter (mm)-long package with 0.5-mm pitch pins. Error detection and correction using the 16 ECC bits is performed inside the Alpha 21164 chip on all read transactions. Data path errors are checked by the PCI bridge chips on all transactions, including read and write transactions between each CPU and memory, and any errors are reported to the operating system.

The data path is clocked at each stage by a copy of a single-phase clock. The clock is provided by a low-skew clock distribution system built from the 52-pin CDC586 phase-locked loop clock driver.⁵ The clock cycle is controlled by an oscillator on the processor module and runs as fast as 66 MHz (15-ns minimum cycle time) while delivering less than a 2-ns worst-case skew (i.e., the difference in the rising edge of the clock) between any two components, including the Alpha 21164, SDRAMs, and any transceiver on any module.

Read transaction data is returned from the pins of the SDRAMs to the pins of the Alpha 21164 in two clock cycles (30 ns), as shown in Table 1. The no-external-cache processor has no module-level data cache, so data is clocked directly into the Alpha 21164 from the transceiver. In Table 1, read data that corresponds to transactions Rd1 and Rd2 is returned from the same set of SDRAM chips in consecutive cycles. Read data that corresponds to transaction Rd3 is returned from a different set of SDRAM chips with a one-cycle gap to allow the data path drivers from transaction Rd2 to be turned off before the data path drivers for transaction Rd3 can be turned on. This process prevents tri-state overlap. As a result, consecutive read transactions have address bus commands either four or five cycles apart. Note that the Alpha 21164 data, command, and address signals are shown for only one processor (CPU1), which issues transaction Rd1. The other transactions are issued by other processors.

Write transaction data is also transferred from the pins of the Alpha 21164 CPU to the pins of the SDRAMs in two clock cycles (see Table 2). Write data

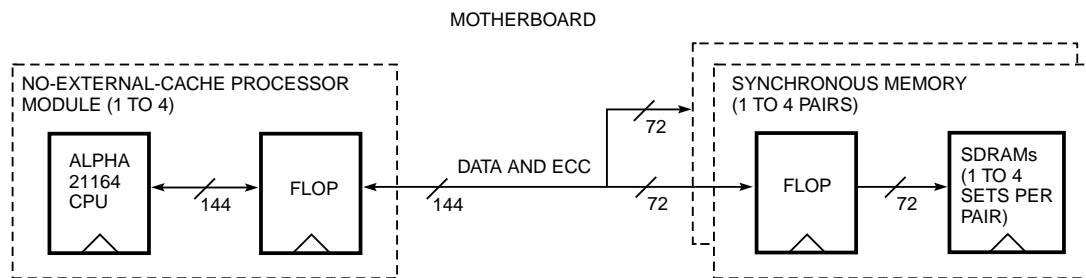


Figure 2
Data Path between the CPU and Memory

Table 1
CPU Read Memory Data Timing

Cycle (15 ns)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Address Bus Command		Rd1				Rd2				Rd3					Rd4			
SDRAM Data						1	1	1	1	2	2	2	2		3	3	3	3
Motherboard Data							1	1	1	1	2	2	2	2		3	3	3
CPU1: Alpha 21164 Data								1	1	1	1							
CPU1: Alpha 21164 Command	Rd1																	
CPU1: Alpha 21164 Address	Addr1																	

Table 2
CPU Write Memory Data Timing

Cycle (15 ns)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Address Bus Command	Wr1				Wr2					Wr3					Wr4			
SDRAM Data			1	1	1	1		2	2	2	2		3	3	3	3		4
Motherboard Data		1	1	1	1		2	2	2	2		3	3	3	3		4	4
Alpha 21164 Data	1	1	1	1		2	2	2	2		3	3	3	3		4	4	4

always incurs a one-cycle gap between transactions. As a result, all but the first two consecutive write transactions have address bus commands five cycles apart.

Since the AlphaServer 4100 interconnect between the CPU and main memory was optimized for the SDRAM memory chip, the transaction timing, as shown in Tables 1 and 2, was designed to provide data in the correct cycles for the SDRAMs without the need for custom ASICs to buffer the data between the motherboard and SDRAM chips. This design works well for an infinite stream of all reads or all writes because of the SDRAM pipelined interface; however, when a write transaction immediately follows a read transaction, a gap or “bubble” must be inserted in the data stream to account for the fact that read data is returned later in the transaction than write data. As a result, every write transaction that immediately follows a read transaction produces a five-cycle gap in the command pipeline. Table 3 shows the read/write transaction timing.

Address Path between the CPU and Memory

The Alpha 21164 provides 36 address signals (byte address <39:4>, i.e., bits 4 through 39), 5 command bits, and 1 bit of parity protection. These 42 signals are connected directly to four 12-bit bidirectional latched transceivers on the processor module, as illustrated in

Figure 3. The motherboard latches the full address and drives first the row and then the column portion of the address to the memory modules. Each synchronous memory module buffers the row/column address and fans out a copy to each of the SDRAM chips using four 24-bit buffers. Similar to traditional dynamic random-access memory (DRAM) chips, SDRAM chips use the row address on their pins to access the page in their memory arrays and the column address that appears later on the same pins to read or write the desired location within the page. Consequently, there is no need to provide the entire 36-bit-wide address to the memory modules. All address components used for transceivers, latches, multiplexers, and drivers on the no-external-cache processor module, the motherboard, and the synchronous memory module consist of the 56-pin ALVC16260 or the ALVC162260, which is the same part with internal output resistors. Address parity is checked by the PCI bridge chips on all transactions, and any errors are reported to the operating system.

The address path uses flow-through latches for the first half of the address transfer (i.e., the row address) from the Alpha 21164 to the SDRAMs. When the address appears at the pins of the Alpha 21164, the latched transceiver on the processor module, the multiplexed row address driver on the motherboard,

Table 3
CPU Read/Write Memory Data Timing

Cycle (15 ns)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Address Bus Command	Rd1				Wr2									Wr3				
SDRAM Data					1	1	1	1				2	2	2	2		3	3
Motherboard Data						1	1	1	1		2	2	2	2		3	3	3

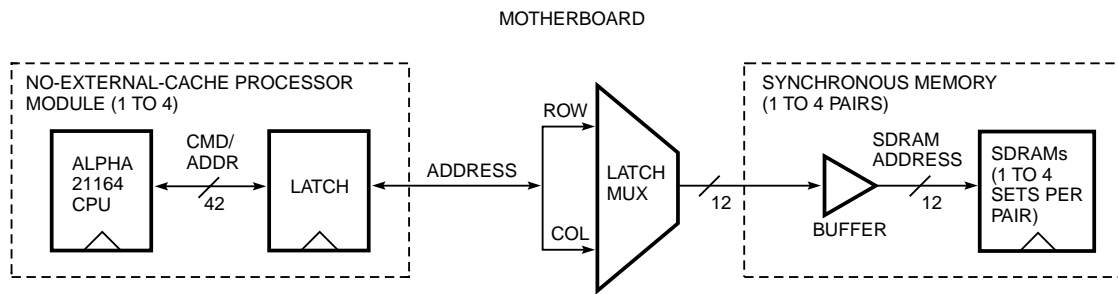


Figure 3
Address Path between the CPU and Memory

and the fan-out buffers on the memory modules are all open and turned on, enabling the address information to propagate directly from the Alpha 21164 pins to the SDRAM pins in two cycles. The motherboard then switches the multiplexer and drives the column address to the memory modules to complete the transaction (see Table 4). Back-to-back memory transactions are pipelined to deliver a new address to the SDRAM chips every four cycles. The full memory address is driven to the motherboard in two cycles (cycles 0–1, 4–5, 8–9), whereas additional information about the corresponding transaction (which is used only by the processor and the I/O modules) follows in a third cycle (cycles 2, 6, 10). To avoid tri-state overlap, the fourth cycle is allocated as a dead cycle, which allows the address drivers of the current transaction to be turned off before the address drivers for the next transaction can be turned on (cycles 3, 7, 11). These four cycles constitute the address transfer that is repeated every four or five cycles for consecutive transactions. Note that the one-cycle gap inserted between transactions Rd3 and Rd4 for reasons indicated earlier in the read data timing description causes the row address for transaction Rd4 to appear at the pins of the SDRAMs for three cycles instead of two.

Control Path between the CPU and Memory

The Alpha 21164 provides five command bits (four Alpha 21164 CMD signals plus the Alpha 21164 Victim_Pending signal) that indicate the operation being requested by the Alpha 21164 external interface.⁶ These five command bits are included in the 42 command/address (CA) signals indicated in Figure 3

and are driven directly and unmodified through the latched address transceivers on the processor module to become the motherboard command/address. Since the AlphaServer 4100 interconnect between the CPU and main memory was optimized for the Alpha 21164 CPU chip, the Alpha 21164 external CMD signals map directly into the 6-bit encoding of the memory interconnect command used on the motherboard, thus avoiding the need for custom ASICs to manipulate the commands between the CPU and motherboard.

Prudently chosen encodings of the Alpha 21164 external CMD signals resulted in only two command bits (to determine a read or a write transaction) and one address bit (to determine the memory bank) being used by a 5-ns PAL on the processor module to directly assert a Request signal to the motherboard to use the memory interconnect. Figure 4 shows the control path between the CPU and memory. If the central arbiter is ready to allow a new transaction by the processor module asserting a Request signal (i.e., if the memory interconnect is not in use), then a 5-ns PAL on the motherboard asserts the control signal Row_CS to each of the memory modules in the following cycle. At the same time, another 5-ns PAL on the motherboard decodes 7 bits of the address and drives the Sel<1:0> signal to all memory modules to indicate which of the four memory module pairs is being selected by the transaction. Each synchronous memory module uses another 5-ns PAL to immediately send the corresponding chip select (CS) signal to the requested SDRAM chips on one of the CS<3:0> signals when the Row_CS control signal is asserted if selected by the value encoded on Sel<1:0>, as shown in Figure 4.

Table 4
CPU Read Memory Address Timing

Cycle (15 ns)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Address Bus Command		Rd1				Rd2				Rd3					Rd4		
SDRAM Address	Row Addr1	Col Addr1	Row Addr2	Col Addr2	Row Addr3	Col Addr3	...	Row Addr4	Col Addr4								
Motherboard Address	Mem Addr1	Info1		Mem Addr2	Info2		Mem Addr3	Info3		...	Mem Addr4	Info4					
Alpha 21164 Address	Addr1	Addr2		Addr3			Addr4				Addr5						

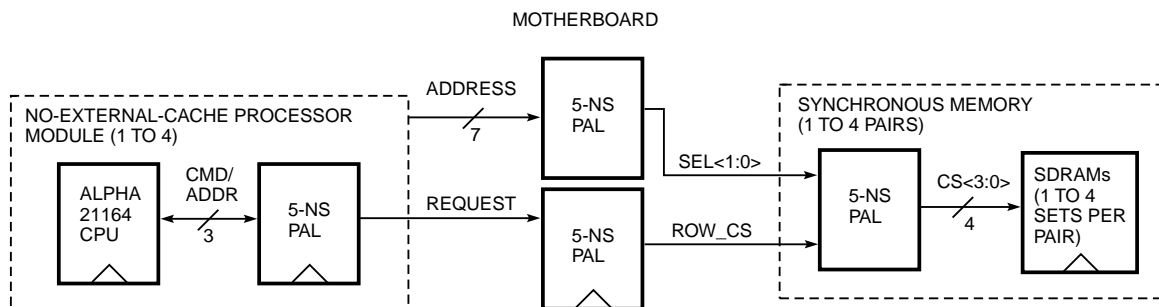


Figure 4
Control Path between the CPU and Memory

Table 5 shows the control signals between the processor modules, the memory modules, and the central arbiter on the motherboard for multiple processor modules issuing single read transactions. The central arbiter receives one or more Request<n> signals from the processor modules and asserts a unique Grant<n> signal to the processor module that currently owns the bus. The arbiter then drives a copy of the CA signal to every processor module along with the identical Row_CS signal to every memory module to mark cycle 1 of a new transaction. Note that the cycle counter begins at cycle 1 with each new CA/Row_CS assertion and may stall for one or more cycles when gaps appear on the memory interconnect. Two transactions may be pipelined at the same time. For simplicity of implementation in programmable logic devices, the cycle counter of each transaction is always exactly four cycles from the other.

Table 6 shows a single processor module issuing two consecutive read transactions (dual-issue) followed by a third read transaction at a later time. Normally, the node issuing the transaction on the bus deasserts the Request signal in cycle 2. If a node continues to assert the Request signal, the central arbiter continues to assert the Grant signal to that node to allow guaranteed back-to-back transactions to occur. Note that the first CA cycle occurs three cycles after the assertion of the Request signal because of the delay within the central arbiter to switch the Grant signal

between processors. The third CA cycle occurs only one cycle after the node asserts the Request signal, however, because of bus parking. Bus parking is an arbitration feature that causes the central arbiter to assert the Grant signal to the last node to use the bus when the bus is idle (following cycle 7 of transaction Rd2). Consequently, if the same processor wishes to use the bus again, the assertion of CA and Row_CS signals occurs two cycles earlier than it would without the bus parking feature.

Data Transfers between Two CPU Chips (Dirty Read Data)

The Alpha 21164 CPU chips contain internal write-back caches. When a CPU writes to a block of data, the modified data is held locally in the write-back cache until it is written back to main memory at a later time. The modified (dirty) copy of the block of data must be returned in place of the unmodified (stale) copy from main memory when another CPU issues a read transaction on the memory interconnect. The memory modules return the stale data at the normal time on the memory interconnect, and the dirty data is returned by the processor module containing the modified copy in the cycles that follow. The processor module issuing the read transaction ignores the stale data from memory.

Therefore, to maintain cache coherency between the write-back caches contained in multiple Alpha

Table 5
Multiple CPU Read Memory Control Timing

Cycle Counter (15-ns cycle)			1	2	3	4	5	6	7	-	1	2	(3)	3	4	5	6	7
Request<n>		1234	1234	24	24	24	24	3	3	3	3	4	4	4	4	4		
Grant<n>	...	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	4
CA, Row_CS (New transaction)			X				X				X					X		
Address/Command Bus		Addr/Rd1	Info1			Addr/Rd2	Info2			Addr/Rd3	Info3		...	Addr/Rd4	Info4			
SDRAM CMD (RAS,CAS,WE)		ACT 1	Read 1			ACT 2	Read 2			ACT 3	...	Read 3		ACT 4	Read 4			
SDRAM CS			X		X		X		X		X			X		X		X

Table 6
Single CPU Read Memory Control Timing

Cycle Counter (15-ns cycle)				1	2	3	4	5	6	7	-	5	6	7	-		1	2
Request<n>	1	1	1	1	1	1	1	1									1	1
Grant<n>	2	2	1	1	1	1	1	1	1	1						1	1	1
CA, Row_CS (New transaction)				X				X									X	
Address/Command Bus			Addr/Rd1	Info1			Addr/Rd2	Info2									Addr/Rd3	Info3
CPU1: Alpha 21164 Data										1	1	1	1	2	2		2	2

21164 CPU chips, each read transaction that appears on the memory interconnect causes a cache probe (snoop) to occur at all other CPU chips to determine if a modified (dirty) copy of the requested data is found in one of the internal caches of another Alpha 21164 CPU chip. If it is, then the appropriate processor module asserts the signal Dirty_Enable<n> for a minimum of five cycles to allow the memory module to finish driving the old data. The processor module deasserts the signal when the dirty data has been fetched from one of the internal caches and is ready to be driven onto the motherboard data bus. Table 7 shows read data corresponding to transaction Rd1 being returned from CPU2 to CPU1 five cycles later than the data from memory, which is ignored by CPU1. Note the one-cycle gap in cycles 10 and 15 to avoid tri-state overlap between the memory module and processor module data path drivers.

As discussed earlier in this section, the AlphaServer 4100 system implements memory address decoding and memory control without using custom ASICs on the motherboard, synchronous memory, or no-external-cache processor modules. Using PALs allows the address decode function and the fan-out buffering to the large number of SDRAMs to be performed at the same time, thus reducing the component count and the access time to main memory. All the necessary glue logic between the Alpha 21164 CPU and the SDRAMs, including the central arbiter on the motherboard, was implemented using 5-ns 28-pin programmable PALs or 90-MHz 44-pin ispLSI 1016 in-circuit reprogrammable PLDs produced by Lattice Semiconductor. These devices can be reprogrammed directly on the module using the parallel port of a laptop personal computer. Each no-external-cache processor module uses five PALs and four PLDs; the mother-

board (arbiter and memory control) uses eight PALs and three PLDs; and each synchronous memory module uses three PALs.

As shown in Table 1, the minimum memory read latency (read data access time) is eight cycles (120 ns) from the time a new command and address arrive at the pins of the Alpha 21164 chip to the time the first data arrives back at the pins. The SDRAMs are programmed for a burst of four data cycles, so data is returned in four consecutive 15-ns cycles. Two transactions at a time are interleaved on the memory interconnect (one to each of the two memory banks), which allows data to be continuously driven in every bus cycle. This results in the maximum memory read bandwidth of 1 GB/s.

Trade-offs Made to Reduce Complexity

The Alpha 21164 external interface contains many commands required exclusively to support an external cache. By not including a module-level cache on the no-external-cache processor module, only Read, Write, and Fetch commands are generated by the Alpha 21164 external interface; the Lock, MB, SetDirty, WriteBlockLock, BCacheVictim, and ReadMissModSTC commands are not used.^{6,7} This design allows the logic on the processor module that is asserting the Request signal to the central arbiter to be implemented simply in a small 28-pin PAL because only two of the Alpha 21164 CMD signals are required to encode a Read or a Write command. Similarly, allowing a maximum of two memory banks in the system, independent of the number of memory modules installed, enables the Request logic to the central arbiter to be implemented in the 28-pin PAL, since only one address bit (byte address <6>) is required to determine the memory bank.

Table 7
Dirty Read Data Timing

Cycle (15 ns)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Address Bus Command		Rd1				Rd2										Rd3		
SDRAM CS		X		X		X								X		X		X
SDRAM CMD (RAS,CAS,WE)	ACT 1		Read 1		ACT 2		Read 2		ACT 3		Read 3	
SDRAM Data						1	1	1	1							2	2	2
Motherboard Data							1	1	1	1		Dirty1	Dirty1	Dirty1	Dirty1		2	2
CPU1: Alpha 21164 Command	Rd1		Rd3				Snp2									Rd5		
CPU1: Alpha 21164 Address	Addr1		Addr3				Addr2									Addr5		
CPU1: Alpha 21164 Response											Miss2							
CPU1: Alpha 21164 Data								(1)	(1)	(1)	(1)		Dirty1	Dirty1	Dirty1	Dirty1		
CPU2: Alpha 21164 Command		Rd2	Snp1		Rd4													Snp3
CPU2: Alpha 21164 Address		Addr2	Addr1		Addr4													Addr3
CPU2: Alpha 21164 Response						Dirty1												
CPU2: Alpha 21164 Data												Dirty1	Dirty1	Dirty1	Dirty1			2
Dirty_Enable<n>							Dirty	Dirty	Dirty	Dirty	Dirty							

To decode memory addresses in 28-pin PALs, the AlphaServer 4100 system uses the concept of memory holes. The memory interconnect architecture and console code support seven different sizes of memory modules and up to four pairs of memory modules per system for a total system memory capacity of 32 MB to 8 GB. Any mix of memory module pairs is supported as long as the largest memory pair is placed in the lowest-numbered memory slot. The physical memory address range for each of the four memory slots is assigned as if all four memory module pairs are the same size. Consequently, if two additional memory pairs that are smaller than the pair in the lowest-numbered slot are installed in the upper memory slots, there will be a gap or “hole” in the physical memory space between the two smaller memory pairs (see Table 8). Rather than require each memory module to compare the full memory address to a base address and size register to determine if it should respond to the memory transaction, the 28-pin PAL driving Sel<1:0> on the motherboard (see Figure 4) uses the seven address bits Addr<32:26> and the size of the memory module in the lowest-numbered slot to encode the memory slot number of the selected memory module pair. Console code detects any memory holes at power-up and tells the operating systems that these are unusable physical memory addresses.

Another simplification that the AlphaServer 4100 system uses is to remove I/O space registers from the data path of the processor and memory modules. Because there are no custom ASICs on these modules, reading and writing control registers would have required additional data path components. Since all the error checking is performed by either the 21164 CPU chip or the PCI bridge chips and since there are no address decoding control registers required on the memory modules, there was no need for more than a few bits of control information to be accessed by software on the processor or memory modules. The I²C bus (slow serial bus) already present in the I/O subsystem was used for transferring this small amount of information.

Furthermore, in the process of removing the I/O space data path from the motherboard and processor modules, the firmware (i.e., the console code, Alpha

21164 PAL code, and diagnostic software), which is often placed in read-only memories (ROMs) on the processor module or motherboard, was moved to the I/O subsystem. Only a small 8-KB single-bit serial ROM (SRAM) was placed on each processor module that would initialize the Alpha 21164 chip on power-up and instruct the Alpha 21164 to access the rest of the firmware code from the I/O subsystem.

Quick Design Time

To provide stable CPU and memory hardware for I/O subsystem hardware debug and operating system software debug and thus allow the DIGITAL AlphaServer 4100 to be introduced on schedule in May 1996, the core module set was designed and powered on in less than six months. This primary goal of the AlphaServer 4100 project was achieved by keeping the design team small, by using only programmable logic and existing data path components, and by keeping the amount of documentation of design interfaces to a minimum.

The design team for the motherboard, no-external-cache processor module, and synchronous memory module consisted of one design engineer, one schematic/layout assistant, one signal integrity engineer, and two simulation engineers. The team also enlisted the help of members of the other AlphaServer 4100 design teams.

The architecture and actual final logic design of the core module set were developed at the same time. By using programmable logic and off-the-shelf address and data path components, the logic was written in ABL code (a language used to describe the logic functions of programmable devices) and compiled immediately into the PALs and PLDs while the architecture was being specified. If the desired functionality did not fit into the programmable devices, the architecture was modified until the logic did fit. All three modules were designed by the same engineer at the same time, so there was no need for interface specifications to be written for each module. Furthermore, modifications and enhancements could be made in parallel to each design to optimize performance and reduce complexity across all three modules.

Table 8
Memory Hole Example

Memory Slot 1	2-GB Module Pair	00000000 – 07FFFFFFF
Memory Slot 2	2-GB Module Pair	08000000 – 0FFFFFFF
Memory Slot 3	1-GB Module Pair	10000000 – 13FFFFFFF
	Memory Hole	14000000 – 17FFFFFFF
Memory Slot 4	1-GB Module Pair	18000000 – 1BFFFFFFF
	Unused Memory	1C000000 – 1FFFFFFF

Because the design did not incorporate any custom ASICs, the core system was powered on as soon as the modules were built. Any last-minute logic changes required to fix problems identified by simulation could be made directly to the reprogrammable logic devices installed on the modules in the laboratory. In particular, the reset and power sequencing logic on the motherboard was not even simulated before power-on and was developed directly on actual hardware.

Since the I/O subsystem was not available when the core module set was first powered on, the software that ran on the core hardware was loaded from the serial port of a laptop personal computer and through the Alpha 21164 serial port, and then written directly into main memory. Diagnostic programs that had been developed for simulation were loaded into the memory of actual hardware and run to test a four-processor, fully loaded memory configuration. This testing enabled signal integrity fixes to be made on the hardware at full speed before the I/O subsystem was available. When the I/O subsystem was powered on, the core module set was operating bug free at full speed, allowing the AlphaServer 4100 to ship in volume six months later.

As mentioned in the section Simple Design, the central arbiter logic on the motherboard was implemented in programmable logic. Consequently, by quickly changing to the reprogrammable logic on the motherboard instead of performing a lengthy redesign of a custom ASIC, designers were able to avoid several logic design bugs that were found later in the custom ASICs of other AlphaServer 4100 processor and memory modules.

Low Memory Latency

Minimizing the access time of data being returned to the CPU on a read transaction was a major design goal for the core module set. The core module set design was optimized to deliver the Addr and CS signals to the SDRAMs in two cycles (30 ns) from the pins of the Alpha 21164 CPU and to return the data from the SDRAMs to the Alpha 21164 pins in another two cycles (30 ns). With the SDRAMs operating at a two-cycle internal row access and a two-cycle internal column access to the first data (60 ns total internal SDRAM access time), the main memory latency is 120 ns.

The low latency was accomplished in four ways:

1. By removing custom ASICs and error checking from the data path between the pins of the Alpha 21164 CPU chip and main memory
2. By combining the SDRAM row/column address multiplexer with address fan-out buffering on the motherboard
3. By simplifying the memory address decode and memory interconnect request logic
4. By using bus parking

Many multiprocessor servers share a common command/address bus by issuing a request to use the bus in one cycle, by either waiting for a grant to be returned from a central arbiter or performing local arbitration in the next cycle, and by driving the command/address on the bus in the cycle that follows. This sequence occurs for all transactions, even when the memory bus is not being used by other nodes. The AlphaServer 4100 memory interconnect implements bus parking, which allows a module to turn on its address drivers even though it is not currently using the bus. If the Alpha 21164 on that module initiates a new transaction, the command/address flows directly to memory in two less cycles than it would take to perform a costly arbitration sequence. Transaction Rd3 in Table 6 shows an example of the effects of bus parking.

High Memory Bandwidth

One of the most important features of the SDRAM chip is that a single chip can provide or consume data in every cycle for long burst lengths. The AlphaServer 4100 operates the SDRAMs with a burst length of four cycles for both reads and writes. Each SDRAM chip contains two banks determined by Addr<6>, which selects consecutive memory blocks. If accesses are made to alternating banks, then a single SDRAM can continuously drive read data in every cycle. The arbitration of the AlphaServer 4100 memory interconnect supports only two memory banks, so the smallest memory module, which consists of one set of SDRAMs, can provide the same 1-GB/s maximum read bandwidth as a fully populated memory configuration, i.e., a system configured with the minimum amount of memory can perform as well as a fully configured system.

To increase the single-processor memory bandwidth, the arbitration allows two simultaneous read transactions to be issued from a single processor module. As long as the arbitration memory bank restrictions and arbitration fairness restrictions are obeyed, it is possible to issue back-to-back read transactions to memory from a single CPU with read data being returned to the Alpha 21164 CPU in eight consecutive cycles instead of the usual four (see Tables 1 and 6). This dual-issue feature and the other low memory latency and high memory bandwidth features of the AlphaServer 4100 architecture enabled the AlphaServer 4100 system to meet the best-in-industry performance goals for McCalpin memory bandwidth.¹

As discussed in the section Simple Design and illustrated in Figure 3, to avoid tri-state overlap, whenever read data is returned by a different set of SDRAMs (on the same memory module or on a different memory module), a dead cycle is placed between bursts of four data cycles to allow one driver to turn off

before the next driver turns on. By keeping the lower-order address bits connected to all SDRAMs, i.e., by not interleaving additional banks of memory chips on low-order address bits, consecutive accesses to alternating memory banks such as large direct memory access (DMA) sequences can potentially achieve the full 1-GB/s read bandwidth of the data bus. With the dead cycle inserted, the read bandwidth of the memory interconnect is reduced by 20 percent.

The data bus connecting the processor, memory, and I/O modules was implemented as a traditional shared 3.3-volt tri-state bus with a single-phase synchronous clock at all modules. As a result, the bus becomes saturated as more processors are added and bus traffic increases. To keep the design time as short as possible, the AlphaServer 4100 designers chose not to explore the concept of a switched bus, on which more than one private transfer may occur at a time between multiple pairs of nodes. Clearly, the AlphaServer 4100 system has reached the practical upper limit of bus bandwidth using the traditional tri-state bus approach.

Reconfigurability

The AlphaServer 4100 hardware modules were designed to allow enhancements to be made in the future without having to redesign every element in the system.

Motherboard Options

The AlphaServer 4100 motherboard contains four dedicated processor slots, eight dedicated memory slots (four memory pairs), and one slot for an I/O module with two PCI bus bridges. Designed at the same time but not produced until after the AlphaServer 4100 motherboard was available, the AlphaServer 4000 motherboard contains only two processor slots, four memory slots (two memory pairs), and slots for two I/O modules allowing four PCI bus bridges. Since module hardware verification in the laboratory is a lengthy process, the AlphaServer 4000 motherboard was designed to use the same logic as the AlphaServer 4100 except for the programmable arbitration logic, which had a different algorithm because of the extra I/O module. When the signals on the AlphaServer 4000 motherboard were routed, all nets were kept shorter than the corresponding nets on the AlphaServer 4100 motherboard so that every signal did not need to be reexamined. Only those signals that were uniquely different were subject to the full signal integrity verification process.

Memory Options

The synchronous memory modules available for the AlphaServer 4100 are all based on the 16-Mb SDRAM.

Using this size chip allowed designers to build synchronous memory modules that contain 9, 18, 36, and 72 SDRAMs and provide, respectively, 32 MB, 64 MB, 128 MB, and 256 MB of main memory per pair. The memory architecture supports synchronous memory modules that contain up to 1 GB of main memory per pair (up to 4 GB per system) by using the 64-Mb SDRAMs; however, when the AlphaServer 4100 system was introduced, the pricing and availability of the 64-Mb SDRAM did not allow these larger capacity synchronous memory modules to be built.

At the same time the synchronous memory modules were being designed, a family of plug-in compatible memory modules built with EDO DRAMs was designed and built. The memory architecture supports EDO memory modules containing up to 2 GB of main memory per pair (up to 8 GB per system) by using the 64-Mb EDO DRAM. When the AlphaServer 4100 system was introduced, the 64-Mb EDO DRAM was available and EDO memory modules containing 72 or 144 EDO DRAMs were built providing 1 GB and 2 GB of main memory per pair. To round out the range of memory capacities and to provide an alternative to the synchronous memory modules in case there was a cost or design problem with the new 16-Mb SDRAM chips, a family of EDO memory modules was also built using 16-Mb and 4-Mb EDO DRAMs, providing 64 MB, 256 MB, and 512 MB of main memory per pair.

Although EDO DRAMs can provide data at a higher bandwidth than standard DRAMs, a single EDO DRAM cannot return data in four consecutive 15-ns cycles like the single SDRAM used on the synchronous memory modules. Therefore, a custom ASIC was used on the EDO memory module to access 288 bits of data every 30 ns from the EDO DRAMs and multiplex the data onto the 144-bit memory interconnect every 15 ns. To imitate the two-bank feature of a single SDRAM, a second bank of EDO DRAMs is required. Consequently, the minimum number of memory chips per EDO memory module is 72 four-bit-wide EDO DRAM chips, whereas the minimum number of memory chips per synchronous memory module is only 18 four-bit-wide SDRAM chips or as few as 9 eight-bit-wide SDRAM chips.

When the AlphaServer 4100 system was introduced, the fastest EDO DRAM available that met the pricing requirements was the 60-ns version. When this chip is used on the EDO memory module, data cannot be returned to the motherboard as fast as data can be returned from the synchronous memory modules. To support the 60-ns EDO DRAMs, a one-cycle (15 ns) increase in the access time to main memory is required. Support for this extra cycle of latency was designed into the memory interconnect by placing a one-cycle gap between cycles 2 and 3 (see Table 1) of any read transaction accessing a 60-ns EDO memory module. Consequently, the read memory latency is one cycle longer

and the maximum read bandwidth is 20 percent less when using EDO memory modules built with 60-ns EDO DRAMs. Note that it is possible to have a mixture of EDO memory modules and synchronous memory modules in the same system. In such a case, only the memory read transactions to the 60-ns EDO memory module would result in a loss of performance.

New versions of the EDO memory modules that contain 50-ns EDO DRAMs providing up to 8 GB of total system memory are scheduled to be introduced within a year after the introduction of the AlphaServer 4100. These modules will not require the additional cycle of latency, and as a result they will have identical performance to the synchronous memory modules.

Processor Options

The no-external-cache processor module was designed to support either a 300-MHz Alpha 21164 CPU chip with a 60-MHz (16.6-ns) synchronous memory interconnect or a 400-MHz Alpha 21164 CPU chip with a 66 MHz (15-ns) synchronous memory interconnect. As previously mentioned, the Alpha 21164 itself contains a primary 8-KB data cache, a primary 8-KB instruction cache, and a second-level 96-KB three-way set-associative data and instruction cache. The no-external-cache processor module contains no third-level cache, but by keeping the latency to main memory low and by issuing multiple references from the same Alpha 21164 to main memory at the same time to increase memory bandwidth, the performance of many applications is better than that of a processor module containing a third-level external cache.¹

Applications that are small enough to fit in a large third-level cache perform better with an external cache, however, so the AlphaServer 4100 offers several variants of plug-in compatible processor modules containing a 2-MB, 4-MB, or greater module-level cache. In addition, cached processor modules are being designed to support Alpha 21164 CPU chips that run faster than 400 MHz while still maintaining the maximum 66-MHz synchronous memory interconnect. The architecture of the cached processor module was developed in parallel with the core module set, and several enhancements were made to the CPU and memory architecture to support the module-level cache. See the companion paper "The AlphaServer 4100 Cached Processor Module Architecture and Design" for more information.⁴

Versions of the Alpha 21164 chip that operate at 400 MHz and faster require 2-volt power, while slower versions of the Alpha 21164 require only 3.3 volts. The AlphaServer 4100 motherboard does not provide 2 volts of power to the processor module connectors; consequently, a 3.3-to-2-volt converter card is used on the higher-speed processor modules to provide this unique voltage. Each new version of

processor module is plug-in compatible, and systems can be upgraded without changing the motherboard. This is true even if the frequency of the synchronous memory interconnect changes, although all processor modules in the system must be configured to operate at the same speed. The oscillators for both the high-speed internal CPU clock and the memory interconnect bus clock are located on the processor modules to allow processor upgrades to be made without modifying the motherboard.

Summary

The high-performance DIGITAL AlphaServer 4100 SMP server, which supports up to four Alpha 21164 CPUs, was designed simply and quickly using off-the-shelf components and programmable logic. When the AlphaServer 4100 system was introduced in May 1996, the memory interconnect design enabled the server to achieve a minimum memory latency of 120 nanoseconds and a maximum memory bandwidth of 1 gigabyte per second. This industry-leading performance was achieved by using off-the-shelf data path and address components and programmable logic between the CPU and the SDRAM-based main memory. The motherboard, the synchronous memory module, and the no-external-cache processor module were developed concurrently to optimize the performance of the memory architecture. These core modules were operating successfully within six months of the start of the design. The AlphaServer 4100 hardware modules were designed to allow future enhancements without redesigning the system.

Acknowledgments

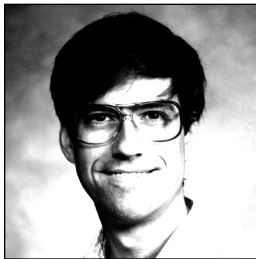
Bruce Alford from Revenue Systems Engineering assisted with the schematic entry, module layout, manufacturing issues, and power-up logic design, and succeeded in smoothly transitioning the core module set to his long-term engineering support organization. Roger Dame handled signal integrity and timing analysis, while Dale Keck and Arina Finkelstein worked on simulation. Don Smelser and Darrel Donaldson provided technical guidance and moral support.

References and Notes

1. Z. Cvetanovic and D. Donaldson, "AlphaServer 4100 Performance Characterization," *Digital Technical Journal*, vol. 8, no. 4 (1996, this issue): 3-20.
2. S. Duncan, C. Keefer, and T. McLaughlin, "High Performance I/O Design in the AlphaServer 4100 Symmetric Multiprocessing System," *Digital Technical Journal*, vol. 8, no. 4 (1996, this issue): 61-75.

3. The AlphaServer 4000 system contains the same CPU-to-memory interface as the AlphaServer 4100 system but supports half the number of processors and memory modules and twice the number of PCI bridges. The AlphaServer 4000 motherboard was designed at the same time as the AlphaServer 4100 motherboard but was not produced until after the AlphaServer 4100 motherboard was available.
4. M. Steinman et al., "The AlphaServer 4100 Cached Processor Module Architecture and Design," *Digital Technical Journal*, vol. 8, no. 4 (1996, this issue): 21-37.
5. R. Dame, "The AlphaServer 4100 Low-cost Clock Distribution System," *Digital Technical Journal*, vol. 8, no. 4 (1996, this issue): 38-47.
6. *Alpha 21164 Microprocessor Hardware Reference Manual* (Maynard, Mass.: Digital Equipment Corporation, Order No. EC-QAEQA-TE, September 1994).
7. The Fetch command is not implemented on the AlphaServer 4100 system, but there is no mechanism to keep it from appearing on the CMD pins of the Alpha 21164 CPI chip. The Fetch command is simply terminated without any additional action.

Biography



Glenn A. Herdeg

Glenn Herdeg has been working on the design of computer modules since joining Digital in 1983. A principal hardware engineer in the AlphaServer Platform Development group, he was the project leader, architect, logic designer, and module designer for the AlphaServer 4100 motherboard, no-external-cache processor modules, and synchronous memory modules. He also led the design of the AlphaServer 4000 motherboard. In earlier work, Glenn served as the principal ASIC and module designer for several DEC 7000, VAX 7000, and VAX 6000 projects. He holds a B.A. in physics from Colby College and an M.S. in computer systems from Rensselaer Polytechnic Institute and has two patents. Glenn is currently involved in further Alpha-based server system development.