Design and Performance of the DEC 4000 AXP Departmental Server Computing Systems

1 Abstract

DEC 4000 AXP systems demonstrate the highest performance and functionality for Digital's 4000 series of departmental server systems. DEC 4000 AXP systems are based on Digital's Alpha AXP architecture and the IEEE's Futurebus+ profile B standard. They provide symmetric multiprocessing performance for OpenVMS AXP and DEC OSF/1 AXP operating systems in an office environment. The DEC 4000 AXP systems were designed to optimize the cost-performance ratio and to include upgradability and expandability. The systems combine the DECchip 21064 microprocessor, submicron CMOS sea-of-gates technology, CMOS memory and I/O peripherals technology, a high-performance multiprocessing backplane interconnect, and modular system design to supply the most advanced functionality for performance-driven applications.

The goal of the departmental server project was to establish Digital's 4000 family as the industry's most cost-effective and highest-performance departmental server computing systems. To achieve this goal, two design objectives were proposed for the DEC 4000 AXP server. First, migration was necessary from the VAX architecture, which is based on a complex instruction set computer (CISC), to the Alpha AXP architecture, which is based on a reduced instruction set computer (RISC). Second, for expansion I/O in an upgradable office environment enclosure, migration was necessary from the Q-bus to the Futurebus+ I/O bus.[1] In addition, the new system had to provide balance between processor performance and I/O performance. Maintaining customer investments in VAX and MIPS applications through support of OpenVMS AXP and DEC OSF/1 AXP operating systems was implicit in the architecture migration objective. Migration, porting, and upgrade paths of various applications were defined.

This paper focuses on the design of the DEC 4000 AXP hardware and firmware. It begins with a discussion of the system architecture and the selection of the system technology. The paper then details the CPU, I/O, memory and power subsystems. It concludes with a performance summary.

2 System Overview

The DEC 4000 AXP system provides supercomputer class performance at office system cost.[2] This combination was achieved through architecture and technology selections that provide optimized uniprocessor performance, low additional cost symmetric multiprocessing (SMP), and balanced I/O throughput. High I/O throughput was accomplished through a combination of integrated controllers and a bridge to Futurebus+ expansion I/O. The

system uses a modular, expandable, and portable enclosure, as shown in Figure 1. With current technologies, the system supports up to 2 gigabytes

(GB) of dynamic random-access memory (DRAM), 24GB of fixed mass storage, and 16GB of removable mass storage. The DEC 4000 AXP system is partitioned into the following modular subsystems:

- Enclosure (BA640 box)
- CPU module (DECchip 21064 processor)
- I/O module
- Memory modules
- Mass storage compartments and storage device assembly (brick)
- Futurebus+ Expansion I/O, Futurebus+ controller module (FBE)
- Power supply modules - universal line front-end unit (FEE)
  - Power system controller (PSC)
  - DC-DC converter unit 5.0 volt (V) (DC5)
  - DC-DC converter unit 2.1 V, 3.3 V, 12.0 V (DC3)
- Cooling subsystem
- Centerplane module
- Operator control panel (OCP)
- Digital storage systems interface (DSSI) and small computer systems interface (SCSI) termination voltage converter (VTERM)

Figure 2 shows these subsystems in a functional diagram. The subsystems are interconnected by a serial control bus, which is based on Signetic's I^2C bus.[3]

NOTE

Figure 1 (DEC 4000 AXP System Enclosure) is a photograph and is unavailable.

3 System Architecture

From the beginning of the project, it was apparent that the I/O subsystem had to be equal to the increased processing power provided by the DECchip

21064 CPU. Although processing power was taking a revolutionary jump in performance with no cost increase, disk and main memory technology were still on an evolutionary cost and performance curve. The metrics that had been used for VAX systems were difficult, if not impossible, to meet through linear scaling within a fixed cost bracket. These metrics were based on VAX-11/780 units of performance (VUPs); they give main memory capacity in megabytes (MB)/VUP, disk-queued I/O (QIO) completions in QIO/s /VUP, and disk data rate in MB/s/VUP. As an example, Table 1 gives the metrics for a VAX 4000 Model 300 scaled linearly to 125 VUPs and then nonlinearly scaled for the DEC 4000 AXP system implementation. Performance

modeling of the DECchip 21064 CPU suggested that 125 VUP was a reasonable goal for the DEC 4000 AXP.

Without an Alpha AXP architecture customer base, we did not know if these metrics would scale linearly with the processor performance. The DECchip 21064 processor technology has the potential for attracting new classes of compute-intensive applications that may make these metrics obsolete. We therefore chose a nonlinear extrapolation of the metrics for our initial implementation. By trading off disk and memory capacity for I/O throughput performance, we kept within established cost and performance goals. The implementation metrics were not limited by the architecture; further scaling up of metrics was planned. Of the four metrics, the disk capacity metric has the most growth potential.

To ensure compliance with both the Alpha AXP architecture and the Futurebus+ specifications, the system was partitioned as shown in Figure 2. The bridge between the CPU subsystem and the Futurebus+ subsystem afforded maximum design flexibility to accommodate specification changes, modularity, and upgradability. The I/O module was organized to balance the requirements between CPU performance and I/O throughput rates. The DEC 4000 AXP system implementation is based on open standards, with a six-slot Futurebus+ serving as the expansion I/O bus and the system bus serving to interconnect memory, CPUs, and the I/O module. The modularity of the system enables module swap upgrades and configurability of the I/O subsystem such that performance and functionality may be tailored to user requirements. The modularity aspects of the system design extend into the storage compartment where each brick has a dedicated controller and power converter. Support for DSSI, SCSI, and high-speed 10MB/s SCSI provides maximum flexibility in the storage compartment. The modular mass storage compartments enable user optimization for bulk storage, fast access, or both.

The cost of SMP was a key issue initially, since Digital's SMP systems were considered high-end systems. Pulling high-end functionality into lower-cost systems through architecture and technology selection was managed by evaluation of performance and cost through trial designs and software breadboarding. Several designs of a CPU module were proposed, including various organizations of one or two DECchip 21064 CPUs per module interfaced to I/O and memory subsystems. Optimization of complexity, parts cost, performance, and power density resulted in a CPU module with one processor that could operate in either of two CPU slots on the centerplane. Consequently, a system bus had to be developed that could be interfaced by processors, memory, and I/O subsystems in support of the shared-memory architecture.

As development of the DECchip 21064 processor progressed, hardware engineers and chip designers established a prioritized list of design goals

for the system bus as follows:

1. Provide a low-latency response to the CPU's second-level cache-miss transactions and I/O module read transactions without pending transactions.

2. Provide a low-cost shared-memory bus, based on the cache coherence protocol, that would facilitate upgrades to faster CPU modules. This provision implied a simple protocol, synchronous timing, and the use of transistor-transistor logic (TTL) levels rather than special electrical interfaces.

3. Provide I/O bandwidth enabling local I/O to operate at 25 megabytes per second (MB/s) and the Futurebus+ to operate at 100MB/s.

4. Provide scalable memory bandwidth, based on protocol timing of 25 nanoseconds (ns) per cycle, which scales with improvements in DRAM and static memory (SRAM) access times.

5. Use module and connector technology consistent with Futurebus+ specifications.

The cache coherence protocol of the system bus is designed to support the Alpha AXP architecture and provide each CPU and the I/O bus with a consistent view of shared memory. To satisfy the bandwidth and latency requirements of the processor's instruction issue rate, the processor's second-level cache size, 128-bit access width, and 32-byte block size were optimized to avoid bandwidth limits to performance. The block size and access width were made consistent with the system bus, which satisfied the I/O throughput metrics. Consideration was given to support of a 64-byte block on the 128-bit-wide bus. Such support would have resulted in a 17 percent larger miss penalty and higher average memory access time for the CPU and I/O, more storage and control complexity, and hence higher cost.

Simplicity of the bus protocol was achieved by limiting the number and variations of transactions to four types-read, write, exchange, and null. The exchange transaction enables the second-level cache of the CPU to exchange data, that is, to perform a victim write to memory at the same time as the replacement read transaction. This avoided the coherence complexity associated with a lingering victim block after the replacement read transaction completed.

To address the issue of bandwidth requirements over time as faster processors become available, an estimate of 40 percent bus utilization for each processor with a 1MB second-level cache was obtained from trace-based performance models. The utilization was shown to be reduced by using a 4MB second-level cache or by using larger caches on the DECchip 21064 chip. This approach was reserved as a means to support future CPU upgrades.

Figure 3 is a block diagram of the length-limited seven-slot synchronous system bus. To achieve tight module-to-module clock skew control for this single-phase clock scheme, clocks are radially distributed from the CPU 1 module to the seven slots. This avoided the added cost of a separate module dedicated for radial clock distribution, and enabled the bus arbitration circuitry to be integrated onto the CPU 1 module.

Arbitration of the two CPU modules and the I/O module for the system bus is centralized on the CPU 1 module. To satisfy the I/O module's latency requirements, the arbitration priority allows the I/O module to interleave with each CPU module. In the absence of other requests, a module may utilize the system bus continuously. Shared-memory state evaluations from the bus addresses during continuous bus utilization causes CPU "starvation" from the second-level cache. To avoid CPU starvation from the second-level cache, the arbitration controller creates one free cycle after three consecutive bus transactions.

4 Technology Selection

The primary force behind technology selection was to realize the full performance potential of the DECchip 21064 microprocessor with a balanced I/O subsystem, weighted by cost minimization, schedule goals, and operation in an office environment. SPICE analysis was used to evaluate various module and semiconductor technologies. A technology demonstration module was designed and fabricated to correlate the SPICE models and to validate possible technology. Based on demonstrations, the project proceeded with analytical data supported by empirical data.

The 25-watt DECchip 21064 CPU was designed in a 3.3-V, 0.75-micrometer complementary metal-oxide semiconductor (CMOS) technology and was packaged in a 431-pin grid array (PGA). The CPU was the only given technology in the system. The power supply, air cooling, and logical and electrical CPU chip interfacing aspects of the CPU module and system bus designs evolved from the DECchip 21064 specifications. System design attention focused on powering and cooling the CPU chip. Compliance with power and cooling specifications was determined to be achievable through conventional voltage regulation and decoupling technology and conventional fan technology.

To address system integrity and reliability requirements, all data transfer interconnects and storage devices had to be protected. The DECchip 21064 CPU's data bus and second-level cache are longword error detection and correction (EDC) protected. The system bus is longword parity protected. The memory subsystem has 280-bit-wide EDC-protected memory arrays. The Futurebus+ is longword parity protected.

System Bus Clocking

To establish the 25-ns bus cycle time, analog models of the interconnect were developed and analyzed for 5.0-V CMOS transceivers. Assuming an edge-to-edge data transfer scheme, the modelers evaluated the timing from a driver transition to its settled signal, including clock input to driver delay, receiver setup time, and module-to-module clock skew. The cycle time and the data transfer width were combined to determine compliance with low latency and bandwidth. Further analysis revealed that the second-

level cache access timing was critical for performing shared-memory state lookups from the bus. One solution to this problem was to store duplicate tag values of the second-level cache. This was evaluated and found to be too expensive to implement. However, the study did show that a duplicate

tag store of the CPU's primary data cache had a performance advantage and was affordable if implemented in the CPU module's bus interface unit (BIU) chips.

To evaluate second-level cache access timing, a survey of SRAM access times, density, availability, and cost was taken. Results showed that a 1MB cache using 12-ns access time SRAMs was optimal. With a 12-ns access time SRAM, the critical timing could be managed through the design of the BIU chips. The SRAM survey also showed that a 4MB second-level cache could be planned as a follow-on boost to performance, as SRAM prices declined. Trace-based performance simulations proved that these cache sizes satisfied performance goals of 125 VUP. This clock rate required a bus stall mechanism to accommodate current DRAM access times in the memory subsystem, which will enable future enhancements as access times are reduced.

The system bus clocks are distributed as positive emitter-coupled level (PECL) differential signals; four single-phase clocks are available to each slot. Each module receives, terminates, and capacitively couples the clock signals into noninverting and inverting PECL-to-CMOS level converters to provide four edges per 25-ns clock cycle. System bus handshake and data transfers occur from clock edge to clock edge and utilize one of two system bus clocks. A custom clock chip was implemented to provide process, voltage, temperature, and load (PVTL) regulation to the pair of application-specific integrated circuit (ASIC) chips that compose each BIU. The clock chip achieves module-to-module skews of less than 1 ns.

Our search for a clock repeater chip that could minimize module-to-module skew and chip-to-chip skew on a module, and yet directly drive high fan-out ASIC chips with CMOS-level clocks, led us to Digital's Semiconductor Operations Group. Such a chip was in design; however, it was tailored for use at the DEC 6000 system bus frequency. The Semiconductor Operations Group agreed to change the chip to accommodate the DEC 4000 AXP system bus frequency.

I/O Bus Technology

Because of technology obsolescence, I/O buses have a 21-year life cycle divided into 3 phases. During the first 7 years of acceptance, peripherals and applications are developed and supported. Sustained acceptance takes hold in the next 7 years as peripherals and applications are enhanced. In the last 7 years, a phase out or migration of peripherals and applications occurs. For the DEC 4000 AXP systems, our first priority was selection of an open expansion I/O bus in the first third of its life cycle. In addition, we wanted to select an open IEEE standard bus that would attract third-party developers to provide I/O solutions to customers. The following

prioritized criteria were established for the selection of a new I/O bus:

1. Open bus that is an accepted industry standard in the beginning third of its life cycle

2. Compatibility with Alpha AXP architecture

3. Minimum data rate of 100MB/s

4. Scalable features that are performance-extensible through architecture (e.g., bus width), and/or through technology improvements (e.g., semiconductor device performance and integration)

5. Minimum 64-bit data path

6. Support of bridges to other I/O buses

7. Minimal interoperability problems between devices from different vendors

After examination of several I/O buses that satisfied these criteria, the Futurebus+ was selected. At the time of our investigation, however, the Futurebus+ specification was in development by the IEEE and a wide range of interest was evident throughout the industry. By providing the right support to the Futurebus+ committee, Digital was in a position to help stabilize and bring the specification to completion.

A Digital team represented the project's interests on the IEEE P896.2 Specification Committee and proposed standards as the DEC 4000 AXP system design evolved. This team achieved its goal by helping the IEEE Committee define a profile that enabled the Futurebus+ to operate as a high-performance I/O expansion bus. To mitigate schedule impact due to instability of the Futurebus+ specifications, the I/O module's Futurebus+ interface was architected to accommodate changes through a more discrete, rather than a highly integrated implementation. Compliance with the Futurebus+ specifications influenced most mechanical aspects of the module compartment design, as is evident from the centerplane, card cage, module construction and size, and power supply voltage specifications and implementations.

Module Technology

Module technology was selected to maximize signal density within the fewest layers with minimal crosstalk and to provide a uniform signal distribution impedance for any module layer. Physical-to-electrical modeling tools were used to create SPICE models of connectors, chip packages, power planes, signal lines of various lengths and impedances (based on the module construction technology), and multiple signal lines. Because the placement of components affects signal performance and quality and system performance (e.g., in the second-level processor cache), module floor plans and trial layouts were completed. A module layout tool was used to ensure producibility compliance with manufacturing standards as well as signal routing constraints. The module layout process was iterative. As sections of the module routing were completed, SPICE models of the etch were extracted. These extracted models were connected to SPICE models of

chip drivers and run. Analysis was completed and required changes were implemented and analyzed again. The process continued until the optimal specification conformance was achieved for all signals.

Module size was estimated based on system functionality requirements and a study of the size and power requirements of that functionality. To simplify the enclosure design, module size specifications are consistent with the Futurebus+ module specifications. To achieve lower system costs, the processor, memory, and I/O modules are based on the same ten-layer controlled impedance construction.

Chip engineers avoided the specification of fine-pitch surface-mount chips when possible. Component choices and module layouts were completed with a view toward manufacturability. Cost analysis showed that mixed, double-sided surface-mount components and through-hole components had insignificant added cost when fused tin-lead module technology and wet-film solder-mask technology were used. The required layer construction and impedances of 45, 70, and 100 ohms could easily be achieved within cost goals through this technology. Solder-mask over bare copper technology was also evaluated to determine if fine-pitch surface-mount components achieved higher yield through the solder reflow process. This evaluation showed fused tin-lead technology was better suited, based on defect densities, for the manufacturing process. Consequently, all DEC 4000 AXP modules are implemented with fused tin-lead module technology and wet-film solder-mask technology.

Semiconductor Technology

As a result of a performance, cost, power, and module real estate study, CMOS technology was used extensively. The custom-designed PVTL clock chips were developed in 1.0-micrometer CMOS technology to supply CMOS-level signals for driving directly into the BIU chips. Each module's BIU used the same 0.8-micrometer ASIC technology and die size to closely manage clock skews. Each system bus module's BIU is implemented by two identical chips operated in an even and an odd slice mode. Chip designers invented a method for accepting 5.0-V signals to be driven into their 3.3-V biased DECchip 21064 CPU. Consequently, the selection and implementation of 5.0-V ASIC technology were easier. ASIC vendor selection was based on (1) performance of trial designs and timing analysis of parity and EDC trees, (2) SPICE analysis of I/O drivers with direct-drive input clock cells, and (3) a layout ability to support wide clock trunks and distributed clock buffering to effect low skews.

All memory chips on the CPU module, memory module, and I/O module were implemented in submicron CMOS or BiCMOS technology. All the I/O and power subsystem controller chips such as the SCSI and DSSI controllers, Ethernet controllers, serial line interfaces, and analog-to-digital converters were implemented in CMOS technology.

Speed or high drive is critical in radial clock distribution, Futurebus+

interfacing, or memory module address and control signal fan-out. In these special cases, 100K ECL operated in positive mode (PECL) or BIPOLAR technology was employed.

System Bus Protocol and Technology

The cache coherence protocol for the shared-memory system bus is based on a scheme in which each cache that has a copy of the data from memory also has a copy of the information about it. All cache controllers monitor or snoop on the bus to determine whether or not they have a copy of the shared block. Hence the system bus protocol is referred to as a snooping protocol, and the system bus is referred to as a snooping bus.[4]

The 128-bit-wide synchronous system bus provides a write update 5-state snooping protocol for write-back cache-coherent 32-byte block read and write transactions to system memory address space. Each module uses a 192-pin signal connector-the same connector used by Futurebus+ modules. Each module interfaces between the system bus and its back port with two 299-pin PGA packages containing CMOS ASIC chips, which implement the bus protocol. A total of 157 signals and 35 reference connections implement the system bus in the 192-pin connector (6 interrupt and error, 8 clock and initialization, 128 command and address or data, 4 parity, 11 protocol). All control/status registers (CSRs) are visible from the bus to simplify the data paths as well as to support SMP.

To simplify the snooping protocol, only full block transactions are supported; masking or subblock transactions occur in each module's BIU. Transactions are described from the perspectives of a commander, a responder, and a bystander. The address space is partitioned into CSR space that cannot be cached, memory space that can be cached, and secondary I/O space for the Futurebus+ and I/O module devices. Secondary I/O space is accessible through an I/O module mailbox transaction, which pends or retries the system bus when access to very slow I/O controller registers conflicts with direct memory access (DMA) traffic. This software-assisted procedure also provides masked byte read and write access to I/O devices as well as a standard software interface. The use of 32-bit peripheral DMA devices avoided the need to implement hardware address translators. The software drivers provide physical addresses; hence mapping registers are not necessary.

The I/O module drives two device-related interrupt signals that are received by both CPU modules due to SMP requirements. One interrupt is associated with the Futurebus+, and the other is associated with all the device controllers local to the I/O module. The I/O module provides a silo register of Futurebus+ interrupt pointers and a device request register of local device interrupt requests. CPU 1 or CPU 2 is the designated interrupt dispatcher module. Privileged architecture library software subroutines, known as PALcode, run on the primary CPU module and read the device interrupt register or Futurebus+ interrupt register to determine which local devices or which Futurebus+ device handlers are to be dispatched.

The enclosure, power, and cooling subsystems are capable of interrupting both processors when immediate attention is required. A CPU can obtain information from subsystems shown in Figure 2 through the serial control bus. The serial control bus enables highly reliable communications between field replaceable subsystems. During power-up, it is used to obtain

configuration information. It is also used as an error-logging channel and as a means to communicate between the CPU subsystem, power subsystem, and the OCP. The nonvolatile RAM (NVRAM) chip implemented on each module allowed the firmware to use software switches to configure the system. The software switches avoided the need for hardware switches and jumpers, field replaceable unit identification tags, and handwritten error logs. As a result, the hardware system is fully configured through firmware, and fault information travels with the field replaceable unit.

The five-state cache coherence protocol assumes that the processor's primary write-through cache is maintained as a subset of the second-level write-back cache. The BIU on the CPU module enforces this subset policy to simplify the simulation verification process. Without it, the number of verification cases would have been excessive, difficult to express, and difficult to simulate and check for correctness. The I/O module implements an invalidate-on-write policy, such that a block it has read from memory will be invalidated and then re-read if a CPU writes to the block. The I/O module participates in the coherency policy by signaling shared status to a CPU read of a block it has buffered. The five states of the cache coherence protocol are given in Table 2.

The cache coherence protocol ensures that only one CPU module can return a dirty response. The dirty response obligates the responding CPU module to supply the read data to the bus, since the memory copy is stale and the memory controller aborts the return of the read data. Bus writes always clear the dirty bit of the referenced cache block in both the commander module and the module that takes the update.

A CPU has two options when a bus transaction is a write and the block is found to be valid in its cache. A CPU either invalidates the block or accepts the block and updates its copy, keeping the block valid. This decision is based on the state of the primary cache's duplicate tag store and the state of the second-level cache tag store. Acceptance of the transaction into the second-level cache on a tag match is called conditional update. When the commander is the I/O module, the write is accepted by a CPU only if the block is valid. Depending on the state of the primary data cache duplicate tag store, two types of hit responses can be sent to an I/O commander-I/O update always and I/O conditional update. In the case of either I/O or CPU commander writes, if the valid block is in the primary data cache, the block is invalidated. The two acceptance modes of I/O writes by a CPU are programmable because accepting writes uses approximately 50 percent more second-level cache bandwidth than invalidating writes.

To implement the cache coherence protocol, the CPU module's second-level cache stores information as shown in Figure 4 for each 32-byte cache block.

Figure 5 shows the cycle timing and transaction sequences of the system bus. Write transactions occur in six clock cycles. Read, null, and exchange transactions occur in seven clock cycles. A null transaction enables a commander to nullify the active transaction request or to acquire the bus

and avoid resource contention, without modifying memory. The arbitration controller monitors the bus transaction type and follows the transactions, cycle by cycle, to know when to rearbitrate and signal a new address and command cycle. Additional cycles can be added by stalling in cycle 2 or cycle 4. Transactions begin when the arbitration controller grants the use of the CPU module's second-level caches to a commander module. The controller then signals the start of the address and command cycle 0 (CA). The commander drives a valid address, command, and parity (CAD) in cycle 1. A commander may stall in cycle 2 before supplying write data (WD) in cycles 2 and 3.

Read data (RD) is received in cycles 5 and 6. The addressed responder confirms the data cycles by asserting the acknowledge signal two cycles later. The commander checks for the acknowledgment and, regardless of the presence or absence, completes the number of cycles specified for the transaction. Snooping protocol results are made available half way through cycle 3. As shown in Figure 5, the protocol timing from valid address to response of two cycles is critical. A responder or bystander may stall any transaction in cycle 4 by asserting a stall signal in cycle 3. The bus stalls as long as the stall signal is asserted. Arbitration is overlapped with the last cycle of a transaction, such that tristate conflict is avoided.

A 29-bit lock address register provides a lock mechanism per cache block to assist with software synchronization operations. The lock address register is managed by each CPU as it executes load from memory to register locked longword or quadword (LDx_L) and store register to memory conditional longword or quadword (STx_C) instructions.[5] The lock address register holds an address and a valid bit, which are compared with all bus transaction addresses. The valid bit is cleared by bus writes to a matching address or by CPU execution of STx_C instructions. The register is loaded and validated by a CPU's LDx_L instruction. This hardware and software construct, as a means of memory synchronization, statistically avoids the known problems with exclusionary locking schemes. Exclusionary locking schemes create resource deadlocks, transaction ordering issues, and performance degradation as side effects of the exclusion. This construct allows a processor to continue program execution while hardware provides the branch conditions. The lock fails only when it loses the race on a write collision to the locked block.

A bus transaction address that hits on a valid lock address register must return a snooping protocol shared response, even if the block is not valid in the primary and second-level caches. The shared response forces writes to the block to be broadcast, and STx_C instructions to function correctly. The NULL transaction is issued when a STx_C write is aborted due to the failure of the lock to avoid system memory modification.

5 CPU Module Subsystems

Each CPU module consists of a number of subsystems as shown in Figure 3. The CPU module's subsystems are

1. DECchip 21064 processor

2. 1MB or 4MB physically addressed write-back second-level cache

3. BIU chips containing write merge buffers, a duplicate tag store of the processor's 8-kilobyte (KB) data cache for invalidate filtering and write update policy decisions, an arbitration controller, a system bus interface, an address lock register, and CSRs

4. System bus and processor clock generators, clock and voltage detectors, and clock distributors

5. System bus reset control

6. 8KB serial ROM for power-up software loading of the processor

7. Microcontroller (MC) with serial system bus interface and serial line unit for communication with the processor's serial line interface

8. NVRAM chip on the serial control bus

Since a CPU module has to operate in either CPU 1 or CPU 2 mode, the CPU 2 connector was designed to provide an identification code that enables or disables the clock drivers and configures the CSRs' address space and CPU identification code. As a result, arbitration and other slot-dependent functions are enabled or disabled when power is applied.

A reliability study of a parity-protected second-level cache showed that the SRAMs contributed 44.7 percent of the failure rate. By implementing EDC on the data SRAM portion of the second-level cache, a tenfold improvement in per processor mean time to failure was achieved. Consequently, six SRAM chips per processor were implemented to ensure high reliability.

The multiplexed interface to the second-level cache of the CPU module allows the processor chip and the system bus equal and shared access to the second-level cache. To achieve low-latency memory access, both the microprocessor and the system bus operate the second-level cache as fast as possible based on their clocks. Hence the second-level cache is multiplexed, and ownership defaults to the microprocessor. When the system bus requires access, ownership is transferred quickly with data SRAM parallelism while the tag SRAMs are monitored.

Many of the CPU module subsystems are found in the interface gate array called the C^3 chip. Two of these chips working in tandem implement the BIU and the second-level cache controller. Write merge buffers combine masked write data from the microprocessor with the cache block as part of an allocate-on-write policy. Since the microprocessor has write buffers that perform packing, full block write around the second-level cache was implemented as an exception to the allocate-on-write policy. To meet schedule and cost goals with few personnel, one complex gate array

was designed rather than several lower-complexity gate arrays. Hence the data path and the control functions were partitioned such that the microprocessor could operate as an even or odd member of a pair on the CPU 1 or the CPU 2 module.

The system bus clock design is somewhat independent of the processor clock, but the range is restricted due to the implementation of the snooping protocol timing, the multiplexed usage of the second-level cache, and the CPU interface handshake and data timing. Therefore, the system bus cycle time is optimized to provide the maximum performance regardless of the processor speed. Likewise, the processor's cycle time is optimized to provide maximum performance regardless of the bus speed. Considerable effort resulted in a second-level cache access time that enabled the CPU's read or write accesses to complete in four internal clock cycles, called the four-tick loop timing of the second-level cache. To realize both optimizations, the CPU's synchronous interface is supported by an asynchronous interface in the BIU. Various timing relationships between the processor and the system bus are controlled by programmable timing controls in the BIU chips.

To achieve the tight, four-tick timing of the second-level cache, double-sided surface-mount technology was used to place the SRAM chips physically close together. This minimized address wire length and the number of module vias; hence the driver was loaded effectively. This careful placement was combined with the design of a custom CMOS address fan-out buffer and multiplexer chip (CAB) to achieve fast propagation delays. The CAB chip was implemented in the same CMOS process as the DECchip 21064 CPU to obtain the desired throughput delay. Combined with 12-ns SRAM chips, the CAB chip enabled optimization of the CPU's second-level cache timing as well as the system bus snooping protocol response timing.

6 I/O Module, Mass Storage, and Expansion I/O Subsystems

The I/O module consists of a local I/O subsystem that interfaces to the common I/O core and a bridge to the Futurebus+ for I/O options. By incorporating modularity into the design, a broad range of applications could be supported. To satisfy the disk performance and bulk storage metrics given in Table 1, mass storage was configured based on applications requirements. Fast access times of 3.5-inch disks and multiple spindles were selected for applications with results in QIO/s. The density of 5.25-inch disks was selected for applications requiring more storage space. As indicated in Table 1, the metrics of greater than 4,000 QIO/s determined the performance requirements of the storage compartment. Each of the four disk storage compartments in the system enclosure can hold a full-size 5.25-inch disk if cost-effective bulk storage is needed. If the need is for the maximum number of I/Os per second, each compartment can hold up to four 3.5-inch disks in a mini array.

Configurations of 3.5-inch disks in a brick enable optimization of throughput through parallelism techniques such as stripe sets and redundant array of inexpensive disks (RAID) sets. The brick configuration also enables fault tolerance, at the expense of throughput, by using shadow sets. With this technique, each storage compartment is interfaced to the system through a separate built-in controller. The controller is capable of running in either DSSI mode for high availability storage in cluster connections with other OpenVMS AXP or VMS systems, or in SCSI mode for local disk storage available from many different vendors. For applications in which a disk volume is striped across multiple drives that are in different storage cavities, the benefit from the parallel seek operations of the drives combines with the parallel data transfers provided by the multiple bus interfaces. The main memory capacity of the system allows for disk caching or RAM disks to be created, and the processing power of the system can be applied to managing the multiple disk drives as a RAID array. With current technology, maximum fixed storage is 8GB with 5.25-inch disks and 24GB with 3.5-inch disks. If the built-in storage system is inadequate, connection to an external solution can occur through the Futurebus+.

The BIU is implemented by two 299-pin ASIC chips. The bridge to the Futurebus+ and the interface to the local I/O devices are provided with separate interfaces to the system bus. Each interface contains two buffers that can each contain a hexword of data. This allows for double buffering of I/O writes to memory for both interfaces and for the prefetching of read data by which the bridge improves throughput. These buffers also serve to merge byte and longword write transaction data into a full block for transfer over the system bus. In this case, the write to main memory is preceded by a read operation to merge modified and unmodified bytes within the block.

The Ethernet controllers and SCSI and DSSI controllers can handle block transfers for most operations, thus avoiding unnecessary merge transactions. As shown in Figure 3, the I/O module integrates the following:

1. Four storage controllers that support SCSI, high-speed SCSI, or DSSI for fixed disk drives and one SCSI controller for removable media drives

2. 128KB of SRAM for disk-controller-loadable microcode scripts

3. Two Ethernet controllers and their station address ROMs, with switch-selectable ThinWire or thick-wire interfaces

4. 512KB flash erase programmable ROM (FEPROM) for console firmware

5. Console serial line unit (SLU) interface

6. Auxiliary SLU interface with modem control support

7. Time-of-year (TOY) clock, with battery backup

8. 8KB of electrically erasable memory (EEROM) for console firmware support

9. Serial control bus controller and 2 kilobits of NVRAM

10.64-bit-wide Futurebus+ bridge

11.BIU, containing four hexwords of cache block buffering, two mailbox registers, and the system bus interface

The instability of the Futurebus+ specifications and the use of new, poorly specified controller chips presented a high design risk for a highly integrated implementation. Therefore the Futurebus+ bridge and local I/O control logic were implemented in programmable logic to isolate the high risk design areas from the ASIC development process.

7 Memory Subsystem

As shown in Figure 3, up to four memory modules can reside on the system bus. This modularity of the memory subsystem enabled maximum configuration flexibility. Based on the metrics listed in Table 1, 2GB of memory were expected to satisfy most applications requirements. Given this 2GB design goal, the available DRAM technology, and the module size, the total memory size was configured for various applications.

The memory connectors provide a unique slot identification code to each BIU, which is used to configure the CSRs' address space based on the slot position. Memory modules are synchronous to the system bus and provide high-bandwidth, low-latency dynamic storage. Each memory module uses 4-bit-wide, 1- and 4-megabit-deep DRAM technology in various configurations to provide 64MB, 128MB, 256MB, or 512MB of storage on each module.

To satisfy memory performance goals, each memory module is capable of operating alone or in one of numerous cache block interleaving configurations with other memory modules with a read-stream capability. A performance study of stream buffers revealed an increase in performance from memory-resident read-stream buffers. The stream buffers allow each memory module to reduce the average read latency of a CPU or I/O module. Thus more bandwidth is usable on a congested bus because the anticipated read data in a detected access sequence is prefetched. The stream buffer prefetch activity is statistically determined by bus activity.

Overall memory bandwidth is also improved through exchange transactions, which use victim writes with replacement read parallelism. Intelligent memory refresh is scheduled based on bus activity and anticipated opportunities. Write transactions are buffered from the bus before being written into the DRAMs to avoid stalling the bus.

Data integrity, memory reliability, and system availability are enhanced by the EDC circuitry. Each memory module consists of 2 or 4 banks with 70 DRAM chips each. This enables 256 data bits and 24 EDC bits to be accessed at once to provide low latency for the system bus. A cost-benefit analysis

showed a savings of DRAM chips when EDC is implemented on each memory module. The processor's 32-bit EDC requires 7 check bits as opposed to the 128-bit EDC, which requires 12 check bits and uses fewer chips per

bank. The selected EDC code also provides better error detection capability of 4-bit-wide chips than the processor's 32-bit EDC.

To improve performance, separate EDC logic is implemented on the write path and read path of each memory module's BIU. The EDC logic performs detection and correction of all single-bit errors and most 2-bit, 3-bit, and 4-bit errors in the DRAM array. The EDC's generate function can detect certain types of addressing failures associated with the DRAM row and column address bits, along with the bank's select address bits. Failures associated with these addressing fields can be detected, thus improving data integrity. Software errors can be scrubbed from memory by the CPU when requested through use of PALcode subroutines, which use the LDx_L and STx_C synchronization construct without having to suspend system operations.

8 Enclosure and Power Subsystems

The DEC 4000 AXP enclosure seen in Figure 1 is called the BA640 box and is 88.0 centimeters (cm) high, 50.6 cm wide, and 76.2 cm deep. It weighs 118 to 125 kilograms fully configured. The enclosure is designed to operate in an office environment from 10 to 35 degrees Celsius. The power cord can connect to a conventional wall outlet which supplies up to 20 amperes at either 120 V AC or 240 V AC.

The DEC 4000 AXP system is a portable unit that provides rear access and simplified installation and maintenance. The system is mounted on casters and fits easily into an open office environment. Modular design allowed compliance with standards, ease of manufacturing, and easy field servicing. Constructed of molded plastics, the chassis is sectioned into a card cage, a storage compartment, a base containing four 6-inch variable-speed DC fans and casters, an air plenum and baffle assembly, front and rear doors, and side panels. The mass storage compartment supports up to 16 fixed-storage devices and 4 removable storage devices. Expansion to storage enclosures is supported for applications that require specialized storage subsystems.

Feedback from field service engineers prompted us to omit useless light-emitting devices (LEDs) in each subsystem, since access to most electronics is from the rear. As a result, the OCP was made common to all subsystems through the serial control bus and made visible inside the front door of the enclosure. It provides DC on/off, halt, and restart switches, and eight LEDs, which indicate faults of CPU, I/O, memory, and Futurebus+ modules. The fault lights are controlled either by a microcontroller on either CPU module or by an interface on the I/O module.

Futurebus+ slot spacing was provided by the IEEE specification. The system bus slot spacing for each module was determined by functional requirements. For example, the CPU module requires 300 linear feet of air flow across the

DECchip 21064 microprocessor's 3-inch square heat sink, as seen in Figure 1, to ensure the 25-watt chip could be cooled reliably. Since VAX 4000 systems provide this same air flow across modules, cooling was not a major design obstacle. The module compartment's Futurebus+, system bus, and power subsystems can be seen in the enclosure back view of Figure 6.

NOTE

Figure 6 (DEC 4000 AXP Enclosure Rear View) is a photograph and is unavailable.

All electronics in the enclosure, as shown in Figure 7, are air cooled by four 6-inch fans in the base. Air is drawn into the enclosure grill at the top front, guided along a plenum and baffle assembly and down through the module compartment and power supply compartment to the base. Air is also drawn through front door louvers and across the storage compartments and down to the base. Electronics connected to the power subsystem monitor ambient and module compartment exhaust temperatures. Thus the fan speed can be regulated based on temperature measurements, reducing acoustic noise in an air-conditioned office environment.

NOTE

Figure 7 (DEC 4000 AXP Modular Electronics) is a photograph and is unavailable.

The centerplane assembly consists of a storage backplane, a module backplane, and an electromagnetic shield. This implementation avoids dependence on cable assemblies, which are unreliable and difficult to install and repair. Defined connectors and module sizes allowed the enclosure development to proceed unencumbered by module specification changes. The shielded module compartment provides effective attenuation of signals up to 5 gigahertz. There are six Futurebus+ slots, four memory slots, two CPU slots, one I/O slot, and four central power module slots, which include the FEU, PSC, DC5, and DC3 units.

The storage compartment contains six cavities, as seen in the enclosure front view of Figure 8. Two cavities are for removable media, and four are for fixed storage bricks. A storage brick consists of a base plate and mounting hardware, disk drives, local disk converter (LDC), front bezel assembly, and wiring harnesses. The LDC converts a distributed 48.0 V to 12.0-V and 5.0-V supplies and a 5.0-V termination reference for the brick to ensure compliance with voltage regulation specifications and termination voltage levels of current and future disks.

NOTE

Figure 8 (DEC 4000 AXP System Enclosure Front View) is a photograph and is unavailable.

The 20-ampere power subsystem can deliver 1,400 watts of DC power divided across 2.1 V, 3.3 V, 5.0 V, 12.0 V, and 48.0 V. The enclosure can cool 1,500 watts of power and can be configured as a master or a slave of AC

power application. Use of a universal FEU eliminates the need for selecting operating voltages of 120 V or 240 V AC. The FEU converts the input AC into 385 V DC, which is distributed to provide 48 V DC to two step-down DC-to-DC converters, which work in parallel to share the load current. The combined 48 V DC output from these converters is delivered to the rest of the system.

Control of distributed power electronics is difficult and expensive with dedicated electronics. A cost-effective alternative was use of a one-chip CMOS microcontroller, surrounded with an array of sensor inputs through CMOS analog-to-digital converters, to provide PSC intelligence. Decision-making ability in the power subsystem enabled compliance with voltage-sequencing specifications and fail-safe operation of the system. The microcontroller can control each LDC and communicate with the CPU and OCP over the serial control bus. It monitors over and under voltage, temperature, and energy storage conditions in the module and storage compartments. It also reports status and failure information either to the CPU or to a display on the PSC module, which is visible inside the enclosure back door.

9 Firmware

The primary goal of the console interface is to bootstrap the operating system through a process called boot-block booting. The console interface runs a minimal I/O device handler routine (boot primitive) to read a boot block from a device that has descriptors. The descriptors point to the logical block numbers where the primary bootstrap program can be found, and the console interface loads it into system memory. To accomplish this task, the firmware must configure and test the whole system to ensure the boot process can complete without failures. To minimize the bootstrap time, a fast memory test executes in the time necessary to test the largest memory module, regardless of the number of memory modules. If failures are detected after configuration is completed, the firmware must provide a means for diagnosis, error isolation, and error logging to facilitate the repair process. The DEC 4000 AXP system provides a new console command interface as well as integrated diagnostic exercisers in the loadable firmware.

The firmware resides on two separate entities, a block of serial ROM on the CPU module and a block of FEPROM on the I/O module. The serial ROM contains software that is automatically loaded into the processor on power-up or reset. This software is responsible for initial configuration of the CPU module, testing minimal module functionality, initializing enough memory for the console, copying the contents of the FEPROM into this initialized console memory, and then transferring control to the console code.

The FEPROM firmware consists of halt dispatch, entry/exit, diagnostics, system restart, system bootstrap, and console services functional blocks.

PALcode subroutines provide a layer of software with common interfaces to upper levels of software. PALcode serves as a bridge between the hardware behavior and service requirements and the requirements of the operating system. The system takes advantage of PALcode for hardware-

level interrupt handling and return, security, implementation of special operating system kernel procedures such as queue management, dispatching to the operating system's special calls, exception handling, DECchip 21064 virtual instruction cache management, virtual memory management,

and secondary I/O operations. Through a combination of hardware- and software-dependent PALcode subroutines, OpenVMS AXP, DEC OSF/1 AXP, and future operating systems can execute on this hardware architecture.

10 Performance Summary

The DEC 4000 AXP Model 610 system's performance numbers as of November 10, 1992 are given in Table 3. Its performance will continue to improve.

11 Summary

DEC 4000 AXP systems demonstrate the highest performance and functionality for Digital's 4000 series of departmental server systems. Based on Digital's Alpha AXP architecture and the IEEE's Futurebus+ profile B standard, the systems provide symmetric multiprocessing performance for OpenVMS AXP and DEC OSF/1 AXP operating systems in an office environment. The DEC 4000 AXP systems were designed to optimize the cost-performance ratio and to include upgradability and expandability. The systems combine Digital's CMOS technology, I/O peripherals technology, a high-performance multiprocessing backplane interconnect, and modular system design to supply the most advanced functionality for performance-driven applications.

12 Acknowledgments

Development of a new system requires contributions from individuals throughout the corporation. The authors wish to acknowledge those who contributed to the key aspects of the DEC 4000 AXP system. Centerplanes: Henry Enman, Jim Padgett; CPU: Nitin Godiwala, George Harris, Jeff Metzger, Eugene Smith, Kurt Thaller; Firmware: Dave Baird, Harold Buckingham, Marco Ciaffi, John DeNisco, Charlie Devane, Paul LaRochelle, Keven Peterson; Futurebus Exerciser: Philippe Klein, Kevin Ludlam, Dave Maruska; Futurebus+: Barbara Archinger, Ernie Crocker, Jim Duval, Sam Duncan, Bill Samaras; I/O: Randy Hinrichs, Tom Hunt, Sub Pal, Prasad Paranjape, Chet Pawlowski, Paul Rotker, Russ Weaver; Management: Jesse Lipcon, Gary P. Lidington; Manufacturing: Mary Doddy, Al Lewis, Allan Lyall, Cher Nicholas; Marketing: Kami Ajgaonkar, Charles Monk, Pam Reid; Mechanical: Jeff Lewis, Dave Moore, Bryan Porter, Dave Simms; Memory: Paul Goodwin, Don Smelser, Dave Tatosian; Operations: Jeff Kerrigan; Operating Systems: AJ Beaverson, Peter Smith; Power: John Ardunio, Robert White; Simulation: Paul Kinzelman; Systems: Vince Asbridge, Mike Collins, Dave Conroy, Al Deluca, Roger Gagne, Tom Orr, Eric Piip; Thermal: Steve Cieluch, Sharad Shah.

13 References and Note

1. IEEE Standard for Futurebus+-Physical Layer and Profile Specification IEEE Standard P896.2-1991 (New York: The Institute of Electrical and Electronics Engineers, April 24, 1992).

2. Supercomputer performance as defined by the composite theoretical performance (CTP) rating of 397, with the DECchip 21064 operated at 6.25 ns, as established by the U.S. export regulations.

3. Inter-Integrated Circuit Serial Bus Specification (I^2C Bus Specification), (Sunnyvale, CA: Signetics Company, 1988).

4. J. Hennessy and D. Patterson, Computer Architecture: A Quantitative Approach (San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1990): 467-474.

5. R. Sites, ed., Alpha AXP System Reference Manual, Version 5.0 (Maynard: Digital Equipment Corporation, 1992).

14 Trademarks

15 Biographies

Barry A. Maskas Barry Maskas is the project leader responsible for architecture, semiconductor technology, and development of the DEC 4000 AXP system buses, processors, and memories. He is a consulting engineer with the Entry Systems Business Group. In previous work, he was responsible for the architecture and development of custom VLSI peripheral chips for VAX 4000 and MicroVAX systems. Prior to that work, he was a codesigner of the MicroVAX II CPU and memory modules. He joined Digital in 1979, after receiving a B.S.E.E. from Pennsylvania State University. He holds three patents and has eleven patent applications.

Stephen F. Shirron Stephen Shirron is a consulting software engineer in

the Entry Systems Business Group and is responsible for OpenVMS support of new systems. He contributed to many areas of the DEC 4000, including PALcode, console, and OpenVMS support. Stephen joined Digital in 1981 after completing B.S. and M.S. degrees (summa cum laude) at Catholic University. In previous work, he developed an interpreter for VAX/Smalltalk-80 and wrote the firmware for the RQDX3 disk controller. Stephen has two patent

applications and has written a chapter in Smalltalk-80: Bits of History, Words of Advice.

Nicholas A. Warchol Nick Warchol, a consulting engineer in the Entry Systems Business Group, is the project leader responsible for I/O architecture and I/O module development for the DEC 4000 AXP systems. In previous work, he contributed to the development of VAX 4000 systems. He was also a designer of the MicroVAX 3300 and 3400 processor modules and the RQDX3 disk controller. Nick joined Digital in 1977 after receiving a B.S.E.E. (cum laude) from the New Jersey Institute of Technology. In 1984 he received an M.S.E.E. from Worcester Polytechnic Institute. He has four patent applications.