The VAXstation 4000 Model 90

By Michael A. Callander, Sr., Lauren M. Carlson, Andrew R. Ladd, Mitchell O. Norcross

1 Abstract

The VAXstation 4000 Model 90 is the latest member of the VAXstation product line. Based on the NVAX CPU, the Model 90 was designed as a module upgrade to the VAXstation 4000 Model 60 system. The Model 90 has 2.7 times the CPU performance of the Model 60 and provides base-level, two-dimensional graphics performance of 266,000 vectors per second. It supports up to 128MB of memory, an SCSI-1 bus interface, a TURBOchannel option, a synchronous communication option, and several graphics options. The design team used only programmable devices to implement the new logic designed into the system. In addition, a breadboard provided the basis for logic and software verification.

2 Introduction

During the summer of 1991, Digital's Semiconductor Engineering Group began planning a new VAX workstation based on the NVAX CPU chip.[1] The development process had three main goals: to increase CPU performance, to maintain an aggressive time-to-market schedule, and to provide upgrade compatibility with the VAXstation Model 60.

The primary goal of the VAXstation 4000 Model 90 design was to implement a workstation with well over twice the CPU performance of its predecessor, the Model 60. The advent of high-performance workstations based on reduced instruction set computers (RISC) required any new VAX workstation to provide a significant performance increase over previous VAX workstations to be competitive in the marketplace. The Model 90 met this goal by achieving 2.7 times the performance of the VAXstation Model 60.

The second major goal of the project was to develop and ship the system as quickly as possible. This was mandated by competitive pressures in the workstation market. We proposed an aggressive best-case schedule which forecast a breadboard running within three months of the project proposal, prototypes running the VMS system within five months, and a customer ship date within eleven months. The development teams achieved almost every project milestone within a few weeks of the proposed schedule.

The final major goal of the project was to design the system such that it could be offered as a simple module upgrade to the VAXstation Model 60. There were two main reasons for designing the system as an upgrade. First, it protected the customer's investment in the Model 60 components. Second, by using as many components as possible from the Model 60 design, we could reduce the hardware and software engineering effort required to produce the new system. The Model 90 system module provides a direct upgrade from the Model 60. The only system component or option on the Model 60 that is not supported by the Model 90 is the entry-level graphics option.

This paper presents the design methodology we followed to meet our project goals. It discusses the four major components in the Model 90 system. It describes the physical design of the system board and the breadboard system we used for logic verification and debugging of software. The paper ends with a comparison of performance data for Digital's workstations.

3 Design Methodology

The design methodology used during the Model 90 project consisted of the following approaches:

1. Complex logic, software, and firmware from existing designs would be used whenever possible.

2. All new logic would be implemented using programmable technology.

3. A breadboard would be built as early as possible.

4. Logic would be simulated only if it could not be verified with the breadboard.

These approaches were influenced and shaped by our aggressive schedule, by the emergence of new programmable technologies, and by the availability of certain VAX system designs that included some of the subsystems that we planned to use. These influences are discussed in this section.

The strategy of using existing hardware and software components stemmed from our goal to deliver the Model 90 as quickly as possible. The project schedule did not allow time for the development of any major new pieces of hardware or software. Consequently, we used as much hardware and software from other VAX products as possible.

Our aggressive schedule also prompted us to explore different technologies and verification methods. On our previous projects, we used conventional gate array or standard cell technology, and typically we strove for exhaustive logic simulation and timing verification prior to releasing chip designs. Our goal was to have fully functional first-pass silicon. Unfortunately, the first pass of a gate array was rarely fully functional. This approach had two consequences on the project schedule: (1) First-pass hardware was usually delayed as much as possible to allow for more thorough logic simulation and timing verification, and (2) A second pass was needed if first-pass silicon was not fully functional, adding several months to the overall project schedule.

At the time of our design, several new programmable silicon technologies were emerging that promised performance, densities, and package sizes comparable to gate array technology. As the logical design of the system progressed through its early stages, we evaluated these new technologies and found that they had matured enough to be used in the design of the Model 90. We chose Xilinx field programmable gate arrays (FPGAs) and AMD MACH PALs to implement large-scale integration, and standard PALs to implement smaller logic functions. These programmable technologies allowed us to build first-pass hardware with the full expectation that we would need to make inevitable changes in response to logic bugs and timing problems. Fortunately, with the new technologies, bug fixes were made in a matter of minutes or days, instead of the weeks or months it would have taken using conventional gate arrays.

During the Model 90 project, we examined our previous notions about the roles of prototyping and simulation in product development. Because the core of the Model 90 was borrowed from the VAX 4000 Model 500, an opportunity arose for us to build a breadboard system consisting of the programmable I/O and graphics interface designs attached to a VAX 4000 Model 500.[2] Unlike a conventional prototype, the breadboard logic was expected to change; therefore we included reconfigurable connections to the FPGAs. Also, the breadboard system did not need to meet any of the physical constraints, such as size and layout, that are normally required of a conventional prototype. An early breadboard system provided the clear benefits of rapid testing and change of hardware and software.

Because we could change logic quickly and easily on the breadboard, the role of simulation on this project focused on verifying module interconnect, and not on exhaustive logic verification. We maintained a working system-simulation model, with a basic set of regression tests, as a reference for logic changes and as a tool for debugging. Logic verification was performed on the breadboard to an extent not possible using simulation.

4 Major System Components

Figure 1 is a block diagram showing the primary components in the Model 90. In this paper we focus on four distinct components in the system: the core, the memory subsystem, the I/O subsystem, and the graphics subsystem.

The core chip set is composed of a 74.4-megahertz (MHz) NVAX CPU, the NVAX memory controller (NMC), and the NVAX I/O adapter (NCA). The NVAX CPU also controls a 256-kilobyte (KB) write-back secondary cache that reduces memory read latency and decreases memory write traffic.

The memory subsystem supports a 64-bit data path to main memory that is composed totally of single in-line memory modules (SIMMs). Main memory sizes of up to 128 megabytes (MB) are supported by the Model 90.

The I/O subsystem comprises two independent 32-bit buses that communicate with the various I/O and graphics options of the Model 90. One bus interfaces to the optional TURBOchannel adapter,[3] the firmware read-only memory (ROM) chips used for console and diagnostics, and the various graphics options available with the Model 90. The other bus interfaces to the Ethernet and EDAL controllers. The EDAL is a general-purpose 16-bit I/O bus. The EDAL controller consists of a CDAL-to-EDAL chip (CEAC) and a small computer system interface (SCSI) quadword first-in first-out (FIFO) chip, known as SQWF. These two chips communicate over the EDAL bus with the system's remaining I/O devices.

Finally, the graphics subsystem provides support for three different graphics options. These options include one low-cost graphics option and two high-performance three-dimensional accelerators.

The majority of the components used in the Model 90 had been used in previous VAX systems. Table 1 lists the major Model 90 components and indicates the source of these components.

5 VAXstation 4000 Model 90 Core

The NVAX CPU, the NMC, the NCA, and the backup cache compose the core of the system module. This core architecture was taken directly from the VAX 4000 Model 500 system. This architecture was chosen because it would meet our performance goals; it provided simple interfaces to our memory, I/O, and graphics subsystems; and because the design was completed and stable. The NVAX CPU is a fully custom complementary metal-oxide semiconductor (CMOS) CPU fabricated in Digital's 0.75-micrometer CMOS-4 process. The NCA and NMC are also fully custom CMOS chips, but are fabricated using Digital's 1.0-micrometer CMOS-3 process. The three custom chips communicate with each other over a 64-bit bidirectional bus named the NDAL.

The NVAX CPU contains a 2KB virtual instruction cache, an 8KB write-through instruction/data primary cache, and, on the Model 90, interfaces to a 256KB write-back instruction/data secondary cache. It contains an on-chip floating-point unit and branch prediction logic. The NVAX CPU pipelines instruction execution at the macroinstruction level as well as the traditional microinstruction level.

The NCA provides direct memory access (DMA) and programmed I/O (PIO) support between the 64-bit NDAL bus and two 32-bit bidirectional CDAL buses named CP1 and CP2. In the Model 90 system, these buses run at an 80-ns cycle time and interface to all the graphics and I/O devices in the system. The NCA also contains the VAX standard interval timer register as well as many of the I/O control and status registers.

The NMC services NDAL memory requests using a 64-bit dynamic random-access memory (DRAM) bus called the NMI, which is protected with an error correction code. The NMC, as configured in the Model 90, supports up to 128MB of main memory. It also supports a directory-based broadcast coherence protocol to maintain coherency between the write-back cache of the NVAX CPU and the system's DMA devices.

6 Memory Subsystem

The memory subsystem of the Model 90 is based on the design of the VAX 4000 Model 500 system. In the memory subsystem, the NMC handles all NDAL memory references by transferring them over the 64-bit NMI. The NMC supports data transfer rates up to 58.5MB per second over the NMI when used with a 74.4-MHz NVAX CPU. Memory is configured in sets; each set contains two banks of interleaved 64-bit wide memory. External multiplexers and transceivers are required to perform interleaving. The NMC provides most of the memory control signals, and only simple bank selection logic is required externally.

The Model 90 memory subsystem implements two sets of memory using the same 36-bit wide SIMMs that are used in the Model 60 system. Four SIMMs are required for each set. By using either the 4MB or 16MB SIMMs, the Model 90 allows memory configuration sizes of 16MB, 32MB, 64MB, 80MB, or 128MB.

Due to module space constraints and cost concerns, we investigated alternatives to the four GMX memory data path chips used on the VAX 4000 Model 500 memory modules. We determined that two low-cost gate arrays designed for the VAXstation VLC could be used instead. These gate arrays provided the same multiplexer and transceiver functions found in the GMX chips. Because the NMI on the Model 90 consists of only two loads, the high-drive capability of the GMX chips was not required. We used a simple PAL to decode the bank selection signals from the NMC and to generate the control signals required for the gate arrays.

Because the Model 90 design uses the NMC, we received an additional benefit of having error correction code protection at no additional cost to the system. The NMC implements a single-bit error correct, double-bit error detect code (SEC/DED) across every 64-bit word of memory data. The eight bits of error correction code replaced the eight bits of parity used on the Model 60.

7 I/O Subsystem

Given that the Model 90 system was an upgrade to the Model 60, a requirement of the I/O subsystem design was to provide support for all I/O devices/options found on the Model 60. The Model 60 I/O design consisted of an interface to a 16-bit bus known as the EDAL where most of the system I/O devices resided. The Model 60 also supported a TURBOchannel adapter that connected to a 32-bit CDAL bus.

The main task of the Model 90 I/O subsystem design was to provide an interface between the two 32-bit CP buses provided by the NCA and the 16-bit EDAL bus and the 32-bit TURBOchannel adapter option offered on the Model 60. The design work necessary included a small PAL design for the TURBOchannel interface on the CP2 bus and the design of two programmable gate arrays for the interface between the 32-bit CP1 bus and the 16-bit EDAL. The following list describes the Model 90 I/O devices and options and explains why each was chosen.

- Ethernet-The Model 90 Ethernet interface is implemented with the second-generation Ethernet controller (SGEC), which provides an Ethernet connection through a ThinWire or thick-wire cable, selectable by a switch on the rear of the system box. The SGEC, which connects to the CP1 bus and was used on the VAX 4000 Model 500 system, facilitates scatter/gather mapping and dual internal FIFO buffering. We chose the VAX 4000 Model 500 design to implement an Ethernet controller because it required no new logic design.

- Small Computer System Interface-The SCSI bus interface is implemented using the NCR 53C94 SCSI controller chip that was used on the Model 60.[4] The NCR 53C94 device connects to the EDAL bus and performs DMA operations to and from main memory in concert with the two programmable gate arrays known as the CEAC and SQWF. DMA virtual-to-physical address translation is performed by the SQWF chip based on 8,192 mapping registers implemented in external static RAMs.

- Serial Lines-The DC7085 quad universal asynchronous receiver/transmitter (UART) chip was chosen to provide the Model 90 with four serial lines for the keyboard, mouse, modem, and printer/console ports. The DC7085 provides a 64-entry FIFO queue that is shared by all four receive lines and is implemented in a small external SRAM.

- Sound-The Model 90 sound functionality is implemented using the AMD 79C30 sound chip just as it was in the Model 60. The programmed I/O interface to this device allows both record and playback functions through a jack on the front panel, and provides voice-quality sound.

- TURBOchannel-The Model 90 provides a single slot into which any TURBOchannel option that is supported by the VMS operating system may be installed. On the Model 60, the TURBOchannel adapter was designed to interface to a CDAL that was not a complete implementation of the general-purpose CDAL bus. For the new design, a small amount of interface logic was necessary to adapt the TURBOchannel option to the CP2 bus.

- Synchronous Communications Option-The Model 90 supports the same multiple protocol communications option that is offered by the Model 60. This interface was implemented on the EDAL bus and allows use of synchronous wide-area network communication through protocols such as high-level data link control (HDLC) and synchronous data link control (SDLC).

- Miscellaneous EDAL Devices-The other devices and registers on the Model 90 16-bit EDAL are a 16-bit system configuration register, an 8-bit light-emitting diode register, an Ethernet identification ROM, and a watch chip. All of these devices also existed on the Model 60 EDAL bus and were accessed in a similar manner.

CEAC and SQWF Chip Designs

One of the major pieces of design work required for the Model 90 I/O subsystem was to interface the 32-bit CP1 bus to all the I/O devices that reside on the 16-bit EDAL bus. This interface was partitioned into two tightly coupled designs called the CEAC and SQWF. The CEAC chip is primarily responsible for handling control of I/O register read and write requests from the NCA to the various devices on the EDAL. The SQWF chip handles DMA transfers and buffering of data between the SCSI controller chip and the NCA.

The CEAC chip, which was first implemented in a Xilinx 3090 FPGA and later converted to a conventional gate array, is a 3,400-gate design and uses 119 I/O pins of a 160-pin plastic quad flat package (PQFP). It performs the CP1 bus arbitration between the SQWF for SCSI DMA, the SGEC for Ethernet DMA traffic, and the NCA for I/O register access. The CEAC responds to NCA I/O accesses that are directed at internal CEAC/SQWF registers and EDAL device registers. Its slave sequencer controls read, write, and chip-select signals that control EDAL devices. The CEAC has CP1 and EDAL multiplexing logic which selects between addresses and data and is controlled by the slave sequencer. The CEAC chip contains an interrupt controller which consists of interrupt request and mask registers, priority decoding logic, and interrupt vector generation logic. The CEAC also has a master sequencer that supports the SQWF during transfers of DMA data on the CP1 bus.

The SQWF chip, which was first implemented in a Xilinx 4005 FPGA and later converted to a conventional gate array, is a 3,900-gate design and uses 110 I/O pins of a 160-pin PQFP. The SQWF responds to requests from the NCR 53C94 SCSI controller chip to do DMA transfers. During SCSI DMA, the SQWF chip helps to optimize utilization of the CP1/NDAL/NMI buses by buffering up to eight bytes of data in either direction. The SQWF performs byte swapping to map the NCR 53C94's 16-bit transfers to arbitrary main memory byte boundaries. The SQWF contains a 22-bit main memory address byte counter and a direction bit which are accessible as registers in I/O space. The SQWF chip also performs DMA virtual-to-physical address translation by referencing an 8,192-page address map store based in external SRAM.

8 Graphics Subsystem

One of the keys to producing a workstation around the VAX 4000 Model 500 core was the ability to integrate graphics support into the system successfully. In addition, maintaining the high level of graphics performance found in the Model 60 was viewed as an important goal. The Model 60 offered three very good graphics options. The Model 60 low-cost graphics (LCG) option features an inexpensive frame buffer module and two-dimensional graphics acceleration logic contained within a large gate array on the system module. The other Model 60 graphics options, SPXg and SPXgt, are three-dimensional graphics accelerators. The SPXg is an 8-plane option, and the SPXgt is a 24-plane option. The three-dimensional graphics options simply replace the LCG frame buffer in the Model 60. We realized that the Model 90 system had to support a high-performance, entry-level, two-dimensional graphics option and the three-dimensional SPXg and SPXgt options. The first major task in the design of the Model 90 was defining the entry-level graphics option.

LCSPX Graphics Option

From the start of the Model 90 project, we knew we could not support LCG. The LCG control logic on the Model 60 was embedded within a very large gate array that also served as a memory and I/O controller. This part was not compatible with our core architecture. Redesigning the Model 60 LCG logic to fit our system would have been a major design task requiring a midsize gate array. This was well beyond our engineering schedule and resources.

To find a graphics option that would provide the desired performance and have a low hardware and software development cost, we met with a number of graphics hardware and software engineers. We found that a new X terminal, the VXT 2000, was being developed with graphics based on a cost-reduced version of the SPX graphics module originally used in the VAXstation 3100 system. This module was close to the Model 60 LCG in both cost and performance. In addition, it was designed to interface directly to a CDAL bus and was software compatible with the VAXstation 3100 SPX. As a result, the module could interface directly to our CP2 bus with a minimal number of changes to its supporting software.

To use the VXT 2000 SPX module in the Model 90, we needed to lay out the module again to fit the physical constraints of our system. This new module was named LCSPX (low-cost SPX). No logic design work was required on the LCSPX or on the system module to support it. A connector on the CP2 bus provides the interface to the LCSPX module.

Although the performance of the VXT 2000 SPX module was close to that of the LCG on the Model 60, we wanted to extract as much performance out of the LCSPX module as possible. To improve the performance of the LCSPX, we increased the clock speed of the module. A speed analysis of the module was performed to determine how much margin existed in the design. The original VXT 2000 SPX module ran at 20 MHz, and we determined that by upgrading a number of components, the LCSPX could run at 25 MHz. As a result of this 25 percent increase in speed, the performance of the LCSPX module exceeds the performance of the Model 60 LCG for almost all operations.

SPXg and SPXgt Graphics Options

On the Model 60, the SPXg and SPXgt graphics options plug into the LCG frame buffer port, and a subset of the LCG control logic provides access to these options. To support SPXg and SPXgt on the Model 90, a port that emulated the LCG frame buffer port was required. The Model 60 supports both a PIO and a DMA interface to the SPXg and SPXgt, but the Model 90 supports only a PIO interface.

We considered a DMA interface for the Model 90, but discarded the idea for several reasons. A DMA interface similar to the Model 60, which supports virtual DMA, requires more logic than would fit in the programmable technologies we were considering for the Model 90. A simpler DMA interface would not have been compatible with VMS graphics system software and would have required a large number of changes to the software. Finally, it appeared that processing on the SPXg and SPXgt modules, and not data bandwidth, was the limiting factor in performance in the Model 60 system. Based on this analysis, a high-speed PIO interface was built.

The SPXg/SPXgt interface on the Model 90 simply translates CP2 bus read and write commands into frame buffer port transactions. The interface is pipelined such that it can keep up with the peak transfer rate of the CP2 bus. We implemented the majority of SPXg/SPXgt interface logic using two AMD MACH PALs. One of these large PALs contains the control sequencer and generates all CP2, frame buffer port, and data path control signals. The other MACH PAL contains an address decoder and address data path. A few miscellaneous medium-scale integration components make up the remainder of the interface. Performance analysis of the SPXg and SPXgt modules shows that performance on the Model 90 is virtually the same as on the Model 60.

9 Physical Design

The physical design of the Model 90 system board presented many challenges. Being a module upgrade from the Model 60, the Model 90 used a system board that had many fixed-position obstacles for placement and routing, such as connectors and stand-off post holes. In addition to the seven connectors and the single switch along the back of the unit, seven more connectors scattered about the module had to retain their positions. Also, the Model 90 had to fit eight SIMMs in the same area that the Model 60 had six SIMMs. Furthermore, programmable technologies generally provide logic of less density than conventional gate arrays, and therefore require more module space. To meet these challenges, we eliminated on-board main memory (8MB were present on the Model 60) and reduced the size of the secondary cache from the originally planned 512KB to 256KB.

The Model 90 system board measures 16 inches by 10.5 inches, and has 8 layers of etch, approximately 100 surface-mount and through-hole components, 23 connectors, 5 oscillators, and over 300 discrete resistors and capacitors. All components are mounted on a single side. Figure 2 shows the Model 90 system module.

10 Model 90 Breadboard System

Our logic verification strategy depended on building a breadboard early in the design cycle. This breadboard allowed quicker and more accurate hardware verification than logic simulation. In addition, the breadboard allowed debugging of console and VMS software earlier than a conventional prototype.

The breadboard system was based on a VAX 4000 Model 500. Logically, the breadboard simply extended the CP1 and CP2 buses of a VAX 4000 Model 500 system to include the complete I/O and graphics subsystems of the Model 90. The Model 90 breadboard was an eight-layer etch module and included all the devices on the Model 90 CP1, CP2, and EDAL buses. The breadboard system used a VAX 4000 Model 500 test backplane that allowed complete physical access to both sides of the VAX 4000 Model 500 CPU module. A socket with pins that extended 1 inch through the back of the module was used on the NCA chip of the VAX 4000 Model 500 CPU module. The breadboard, which contained the holes for the NCA, was then attached to the VAX 4000 Model 500 CPU module by soldering it to the extended socket pins. CP bus clocks were not directly routed to the breadboard logic. To control clock skew, a phase lock loop (PLL) was used on the breadboard to regenerate the CP bus clocks. With this configuration, the breadboard system was able to run at full speed.

Once the breadboard system was assembled, we were able to execute console commands after a quick debugging of the system. At this time, very little of the breadboard logic was being used because the console program was using the VAX 4000 Model 500 I/O devices and not the Model 90 devices. The hardware team began debugging the breadboard logic piece by piece. Debugging was quick because a completely functional console and I/O system already existed. Simple functions, such as register reads and writes, were debugged using the console examine and deposit commands. More complex functions, such as reading and writing to an SCSI disk, were tested by writing test programs in VAX MACRO, downloading them into memory, and executing them using the console.

After some of the major pieces of functionality were verified by the hardware group, members of the VMS group began to use the breadboard. A modified version of the VMS operating system was used to debug VMS device drivers. Drivers for the serial lines, LCSPX, SPXg, SPXgt, and the SCSI port were debugged. In addition to software debugging, this effort provided the software to perform extended verification. The hardware group was able to use graphics test packages running under DECwindows software, disk exercisers, system exercisers, and other tools supported by the VMS operating system. This provided a verification environment we could never achieve with traditional simulation methods.

At this point, we were still using the VAX 4000 Model 500 console. The breadboard was then used to debug the Model 90 console code. We disabled the system support chip, which controls much of the console support hardware in the VAX 4000 Model 500, and began using the Model 90 console support hardware. A base console that included minimal power-up self-test, basic command support, and SCSI boot support was debugged by the Model 90 console team. Once the console was functional, the VMS group returned and debugged boot support for the Model 90 using the breadboard. When this was finished, the software was completely debugged and ready to be loaded onto the first Model 90 prototype.

As soon as we assembled the first Model 90 prototype system, we realized the benefits of all the work performed using the breadboard system. During the first day of debugging, we ran the console program and booted the VMS system with minimal effort. We also ran DECwindows software using the

LCSPX and the SPXg and SPXgt graphics options. This quick debugging allowed additional prototype systems to be built immediately and shipped to various development and verification groups throughout the company.

11 Performance

The VAXstation 4000 Model 90 represents the fastest VAX workstation ever produced. Its CPU performance surpasses previous VAX workstations and is comparable to Digital's RISC-based workstations. By utilizing the NVAX CPU chip, the Model 90 achieves 2.7 times the performance of the Model 60 when measured against the SPECmark benchmarks.[6] Table 2 gives the CPU performance of the VAXstation Model 90 compared to other Digital workstations.

NOTE

*SPECmark is a quantitative measurement of performance, determined by running a suite of ten benchmark programs.

LCSPX is the entry-level, two-dimensional graphics option offered on the Model 90. The performance of this option is better than the LCG option offered on the Model 60 for most graphics operations. Table 3 compares the LCSPX graphics performance to Digital workstations using standard two-dimensional metrics.

Note: *GPCmark is a quantitative measurement of performance, determined by dividing a normalizing constant by the elapsed time, in seconds, required to perform the test.

SPXg and SPXgt are high-performance, three-dimensional graphics accelerators offered on both the Model 60 and the Model 90. Table 4 compares the three-dimensional graphics performance of several of Digital's workstations using standard three-dimensional metrics. In addition, Table 5 gives three-dimensional performance using the picture-level benchmark (PLB) suite.

12 Summary

The NVAX CPU chip provides the high performance that makes the VAXstation 4000 Model 90 competitive in today's market. The design methodology used during the project allowed us to develop and ship the Model 90 quickly and to provide a simple upgrade path for existing VAXstation customers.

13 Acknowledgements

The authors would like to acknowledge the following people for their contributions to the VAXstation Model 90 development: Gregg Bouchard, Paul Campos, Jon Crowell, Terry Furman, Dave Ives, Bill Laprade, Jim Lundberg, Curt Miller, Jan Nordh, Jim Reilley, Brian Rost, Mike Sullivan, Pat Sullivan, Mike Warren, and Tom Wenners.

14 References

1. G. Uhler et al., "The NVAX and NVAX+ High-performance VAX Microprocessors," Digital Technical Journal, vol. 4, no. 3 (Summer 1992, this issue): 11-23.

2. J. Crowell et al., "Design of the VAX 4000 Model 400, 500, and 600 Systems," Digital Technical Journal, vol. 4, no. 3 (Summer 1992, this issue): 60-72.

3. TURBOchannel Hardware Specification (Palo Alto, CA: Digital Equipment Corporation, TRI/ADD Program, 1990).

4. Small Computer System Interface (SCSI) (New York: American National Standards Institute, ANSI X3.131-1986, 1986).

5. Digital's VAXstation 4000 Family Performance Summary, Version 3.0 (Maynard: Digital Equipment Corporation, 1992).

15 Biographies

Michael A. Callander, Sr. Michael Callander is a principal engineer in Digital's Semiconductor Engineering Group. Mike was the technical leader for the VAXstation 4000 Model 90 system. His previous experience with Digital includes design and architectural specification for various CPU modules and systems, including the VAX 8200 and the VAX 6000 Model 400 and Model 500. Mike received his B.S.E.E. from the University of Massachusetts in 1982 and joined Digital upon graduation. Mike has authored several technical papers and has a number of patent applications pending.

Lauren M. Carlson A senior hardware engineer in the Semiconductor Engineering Group, Lauren Carlson is currently working on the design of a peripheral chip set for a new microprocessor. Previously, she designed the VAXstation 4000 Model 90 CDAL-to-EDAL adapter chip (CEAC) gate array, which is part of the I/O subsystem. Lauren also contributed to the design of the VAXstation 4000 Model 90 system module and another VAX system CPU module. Prior to this, Lauren worked in the Advanced VAX Development Group. She received her B.S.E.E. from Worcester Polytechnic Institute in 1986 and joined Digital in 1987.

Andrew R. Ladd Andy Ladd is a principal engineer in the Semiconductor Engineering Group and is the project leader for the VAXstation 4000 Model 90 CPU and low-cost graphics module designs. Previously, he provided timing verification support for the DECchip 21064 and co-designed two bus interface chips for the VAX 6000 Model 400 CPU module. Andy joined Digital in 1986. He received his B.S. in computer engineering from the University of Illinois (1984) and his M.S. in computer science and engineering from the University of Michigan (1991). Andy is a member of the IEEE Computer Society, Tau Beta Pi, and Eta Kappa Nu.

Mitchell O. Norcross Senior engineer Mitch Norcross has been designing and analyzing digital subsystems since he joined Digital in 1986. As a member of the Semiconductor Engineering Group, Mitch designed a gate array for the VAXstation 4000 Model 90 and contributed to the design and analysis of several system and CPU modules. Prior to joining SEG, Mitch designed a gate array for Digital's first fault-tolerant VAX system, the VAXft 3000. He received a B.E. in electrical engineering (1985) and an M.S. in computer engineering (1987), both from Manhattan College. Mitch holds one patent on fault-tolerant system design.

16 Trademarks

The following are trademarks of Digital Equipment Corporation: DECstation, DECwindows, Digital, ThinWire, TURBOchannel, VAX, VAX 4000, VAX MACRO, VAXstation, VMS, and VXT 2000.

MACH is a trademark and PAL is a registered trademark of Advanced Micro Devices, Inc.

SPECmark is a registered trademark of the Standard Performance Evaluation Cooperative.