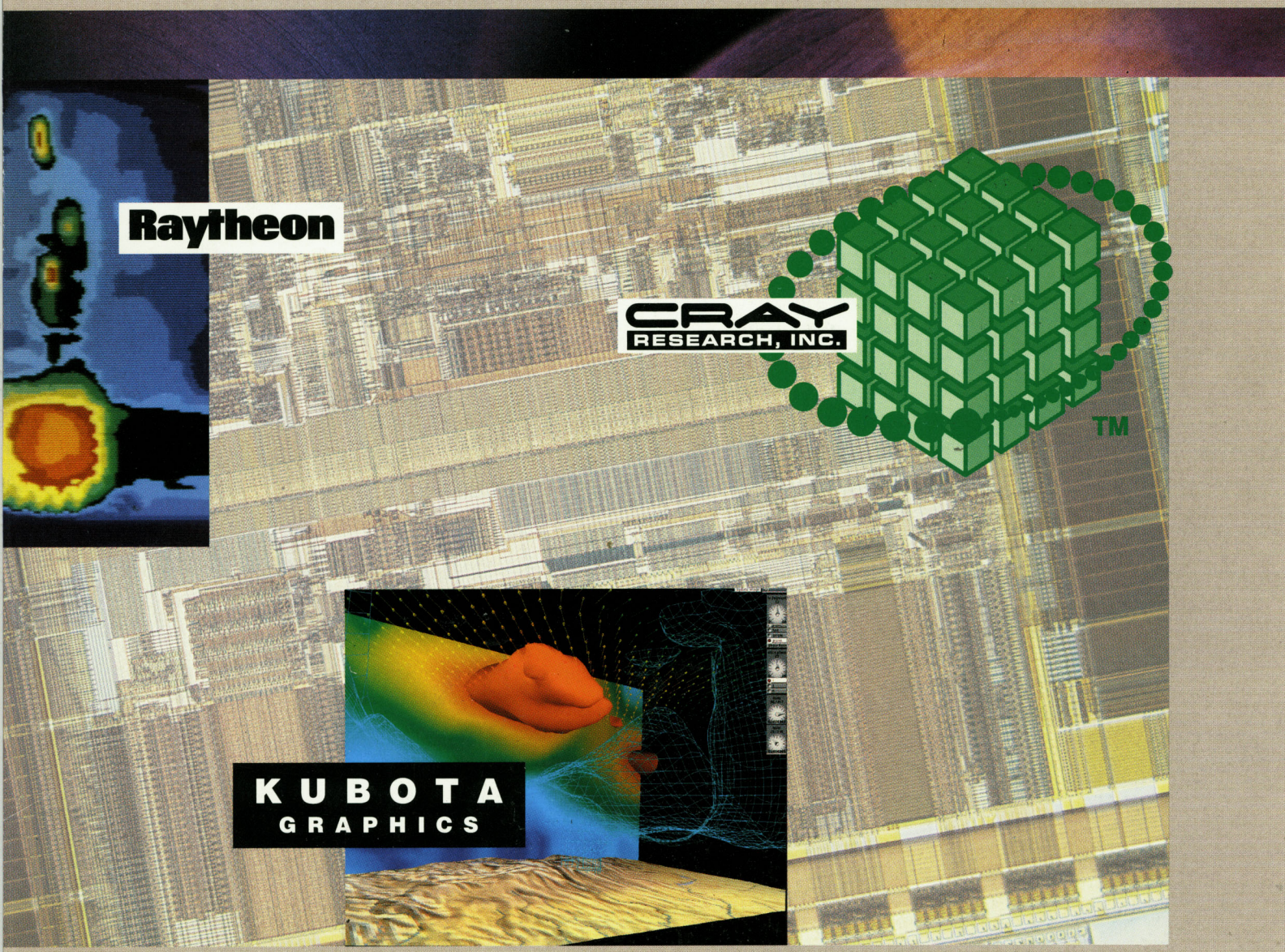
- *Alpha AXP Partners – Cray, Raytheon, Kubota*
- *DECchip 21071/21072 PCI Chip Sets*
- *DLT2000 Tape Drive*

Digital Technical Journal

Digital Equipment Corporation



Volume 6 Number 2
Spring 1994

The *Digital Technical Journal* is a refereed journal published quarterly by Digital Equipment Corporation, 30 Porter Road LJO2/D10, Littleton, Massachusetts 01460. Subscriptions to the *Journal* are $40.00 (non-U.S. $60) for four issues and $75.00 (non-U.S. $115) for eight issues and must be prepaid in U.S. funds. University and college professors and Ph.D. students in the electrical engineering and computer science fields receive complimentary subscriptions upon request. Orders, inquiries, and address changes should be sent to the *Digital Technical Journal* at the published-by address. Inquiries can also be sent electronically to DTJ@CRL.DEC.COM. Single copies and back issues are available for $16.00 each by calling DECdirect at 1-800-DIGITAL (1-800-344-4825). Recent back issues of the *Journal* are also available on the Internet at gatekeeper.dec.com in the directory /pub/DEC/DECinfo/DTJ.

Digital employees may order subscriptions through Readers Choice by entering VTX PROFILE at the system prompt.

Comments on the content of any paper are welcomed and may be sent to the managing editor at the published-by or network address.

ISSN 0898-901X Documentation Number EY-F947E-TJ

Book production was done by Quantic Communications, Inc.

Cover Design

Our cover displays the logos of three Digital Alpha AXP partners—Cray Research, Raytheon, and Kubota Graphics—who present papers in this issue. The graphic accompanying each logo represents an aspect of the technology described. Cray Research's 3-D torus interconnection network is designed as a cube with connected opposing faces; using three dimensions is optimum for systems with hundreds or thousands of processors and increases system resiliency and bandwidth. The image next to the Kubota logo was generated by Colin Sharp of Project Sequoia at the San Diego Supercomputer Center using Kubota's graphics accelerator and an Alpha AXP workstation. From data sets of wind vectors, temperature, and measurements of water content, a virtual 3-D world emerges in which a scientist can explore and test hypotheses. Raytheon's analysis of its militarized Alpha AXP computer is represented here by a thermal map; in contrast to commercial computers which operate in the range of 0 degrees C to 50 degrees C, computers designed for military use must operate in a range as wide as −54 degrees C to 85 degrees C. Underlying all these images is a photomicrograph of the Alpha AXP microprocessor used by each company to create high-performance systems.*

The cover was designed by Joe Pozerycki, Jr., of Digital's Design Group.

** Sequoia 2000 is a large, interdisciplinary research and development program to create the storage, database, visualization, and networking systems scientists need to study the complexities of global change. Project Sequoia 2000 is supported through a primary grant from Digital in partnership with funding from numerous industry, state, and federal government partners.*

Contents

Editor's Introduction

Jane C. Blake
Managing Editor

This issue of the *Digital Technical Journal* presents papers from three companies—Cray Research, Raytheon, and Kubota Graphics—that have developed high-performance systems based on the Alpha AXP 64-bit microprocessor. Also included here are papers about the Alpha AXP chip sets for building PCI-based systems and on the compression technique used in the DLT2000 tape product.

Cray Research, the parallel vector processor and supercomputing pioneer, has developed its first massively parallel processor (MPP) for customers who seek the price/performance advantages of an MPP design. As Kent Koeninger, Mark Furtney, and Martin Walker explain, Cray's MPP uses hundreds of fast commercial microprocessors, in this case Digital's DECchip 21064; whereas a parallel vector processor uses dozens of custom (more expensive) vector processors. Their paper focuses on the CRAY T3D system—an MPP designed to enable a wide range of applications to sustain performance levels higher than those attained on parallel vector processors. The authors review major system aspects, including the programming model, the 3-D torus interconnect, and the physically distributed, logically shared memory.

For the U.S. military, Raytheon has designed an extended environment, commercial off-the-shelf (E^2COTS) computer based on the DECchip 21066/68 AXPvme 64 board. Bob Couranz discusses the characteristics of the E^2COTS board that provide the military with cost and performance advantages. He describes how designers addressed the military's reliability requirements, one of which is computer operation in a wide temperature range of −54 degrees C to 85 degrees C. Packaging modifications made by Raytheon include reconfiguration of the module board for conduction cooling as opposed to the convection cooling of the commercial product.

Kubota Graphics' advanced 3D imaging and graphics accelerator is used in Digital's DEC 3000 Alpha AXP workstations and in Kubota's workstations. Ron Levine's paper interweaves a description of the Kubota accelerator product with a tutorial on imaging, graphics, and volume rendering. He begins by distinguishing between imaging and graphics technologies and their relationship to volume rendering methods. He then reviews application areas, such as medical imaging and seismic exploration, and expands on volume rendering techniques. The final section addresses the Kubota implementation, the first desktop-level system to provide interactive volume rendering.

Digital encourages broad industry application of the Alpha AXP family of microprocessors. Sam Nadkarni, Walker Anderson, Lauren Carlson, Dave Kravitz, Mitch Norcross, and Tom Wenners describe the chip sets—one cost focused and one performance focused—system designers can use to easily build PCI-based Alpha AXP 21064 systems. The authors also present an overview of the EB64+ evaluation kit. This companion to the chip sets gives designers sample designs and an evaluation platform which allows them to quickly evaluate the cost and performance implications of their design choices.

The state-of-the-art DLT2000 tape drive offers high data throughput, up to 3M bytes/s, and high data capacity, up to 30G bytes (compressed). David Cressman outlines the product issues that drove the DLT2000 development and then details the developers' investigation of the performance impact to the tape drive design of two different data compression algorithms, the Lempel-Ziv algorithm and the Improved Data Recording Capability (IDRC) algorithm. He reviews the tests conducted to measure compression efficiency and data throughput rates. The test results, unexpected by developers, reveal that the design using Lempel-Ziv compression generally achieves higher storage capacity and data throughput rates than the IDRC-based design.

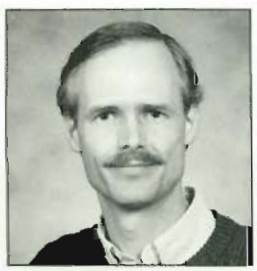
Jane Blake

Biographies

Walker Anderson A principal engineer in the Semiconductor Engineering Group, Walker Anderson is currently the manager of graphics and multimedia chip verification. He was the verification team leader for the NVAX chip and for the DECchip 21071/21072 chip sets as well as a co-leader of the verification team for a future Alpha AXP chip. Before joining Digital in 1988, Walker was a diagnostic and testability engineer in a CPU development group at Data General Corporation for eight years. He holds a B.S.E.E. (1980) from Cornell University and an M.B.A. (1985) from Boston University.

Lauren M. Carlson A senior hardware engineer in the Semiconductor Engineering Group, Lauren Carlson is currently working on the design of a core logic chip set for a new microprocessor. Prior to this, she worked on the design of the cache/memory controller of the DECchip 21071 chip set and completed the hardware functional verification of the chip set on the EB64+ evaluation board. Lauren has also contributed to the design of the I/O controller and system module of the VAXstation 4000 Model 90. Lauren holds a patent on gate array design. She has a B.S.E.E. from Worcester Polytechnic Institute (1986) and joined Digital in 1987.

Robert Couranz Robert Couranz received a B.S.E.E. and a D.Sc. in electrical engineering and computer science from Washington University and an M.S.E.E. in automatic control theory from the Air Force Institute of Technology. He was elected to Tau Beta Pi and Sigma Xi. He has served as a consultant on computer architecture to the Defense Science Board and the Department of Defense. He is presently the Technical Director of Raytheon's Computer Products Operation.



David C. Cressman A consulting software engineer in the Tapes and Solid State Disk Engineering Group, Dave Cressman is currently working on the development of digital linear tape (DLT) products. He developed the SCSI firmware for the TZ85 and TZ86 tape products and was responsible for the TMSCP firmware of the TF85 and TF86 tape products. Dave joined Digital in 1988 after seven years with Data General Corporation, where he developed a SCSI subsystem controller and operating system device drivers. He received B.S.C.S. and B.S.E.E. degrees (1981) from State University of New York (SUNY) at Stony Brook.

Mark Furtney Mark Furtney specializes in software for high-performance parallel systems. He has been employed by Cray Research in the Software Division since 1982, where he worked on CRAY-2 parallel software and led the development of Cray's Autotasking compiling system. He is now the group leader for Tools, Libraries, Commands, and MPP Software for various systems, including the CRAY T3D and follow-on systems. Mark holds a B.S. (1968) in mechanical engineering from Clarkson University, an M.S. (1970) in nuclear engineering from MIT, and a Ph.D. (1983) in computer science from the University of Virginia.

R. Kent Koeninger Kent Koeninger has been the MPP Software Program Manager for Cray Research since 1992. Prior to this, he was a supercomputer specialist for Apple Computer, where he modeled the Cray to Apple's unique interactive-graphics, mass-storage, and high-speed-networking requirements. Earlier, while at the NASA/Ames Research Center, he repeatedly upgraded the supercomputers to the fastest available. A notable event was the first field installation of the CRAY X-MP system. Kent has a B.S. (cum laude, 1977) in mathematics from California State University at Bakersfield and is a National Merit Scholar.

David Kravitz David Kravitz received a B.S.E.E. from the Massachusetts Institute of Technology. Upon joining Digital in 1985, he worked on the cache control and processor chips for the VAX 6000 Models 400 and 500 systems in Hudson, Massachusetts, and a Cluster Interconnect (CI) chip in Jerusalem, Israel. As a senior hardware engineer in the Semiconductor Engineering Group, David designed the data path chip for the DECchip 21071 and DECchip 21072 chip sets. He is currently working on a low-cost microprocessor.

Ronald D. Levine Based in Berkeley, California, Ron Levine is an independent consultant who specializes in 3-D graphics software and systems. His recent work for Digital includes developing the Graphics Boot Camp intensive training program and writing several technical overviews. For other clients, he consults on graphics algorithms for hardware implementation and on standard 3-D graphics device interfaces. Ron holds a Ph.D. in mathematics and A.B. and M.A. degrees in physics, all from the University of California. He is former Chairman of the Department of Mathematics and Computer Science at Humboldt State University.

Samyojita A. Nadkarni Sam Nadkarni is the program manager for CPU core logic chip sets in the Semiconductor Engineering Group. She was the leader of the DECchip 21071 development project. Prior to that, Sam led the development of the memory controller chip used in the VAX 4000 Models 400, 500, and 600 systems. She also worked on memory controller/bus adapter chips for the VAX 4000 Model 300 and MicroVAX 3500 systems. Sam joined Digital in 1985 and holds a Bachelor of Technology (1983) from the Indian Institute of Technology and an M.S. (1985) from Rensselaer Polytechnic Institute.

Mitchell O. Norcross Senior engineer Mitch Norcross is currently project leader for a second-generation core logic chip set for the DECchip 21064. Since joining Digital in 1986, Mitch has contributed to the design of several ASICs and systems, including the DECchip 21072 chip set, the VAXstation 4000 Model 90, and 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 two patents related to fault-tolerant system design.

Martin Walker Martin Walker directed all applications activity in support of Cray T3D development. He was co-creator and first director of Cray's Parallel Applications Technology Program. Presently, he is General Manager of APTOS, a European applications company created by Cray Research and Stern Computing Systems. Prior to joining Cray, following fifteen years of scientific research, he managed MPP development at Myrias Research Corporation. Martin has a B.Sc. from Carleton University, Ottawa, and a Ph.D. from the University of London, U.K.

Thomas M. Wenners Thomas Wenners is a principal hardware engineer in the Semiconductor Engineering Group. He is the project leader responsible for various high-performance mother boards for Alpha AXP PCs. In addition, he is involved with issues concerning high-speed clocking in Alpha AXP chips. Tom's previous work includes the module design of the VAX 6000 Model 600 and VAX 4000 Model 90, as well as module design and signal integrity support on ESB products. Tom joined Digital in 1985. He received a B.S.E.E. (cum laude, 1985) and an M.S.E.E. (1990) from Northeastern University.

Foreword

Scott A. Gordon
Manager
Strategic Programs,
Semiconductor Operations

Early in the development of the Alpha program, Digital's management put forward a strategic direction that would significantly shape the application and reach of Alpha AXP technology in the market. That direction was to make Alpha AXP technology "open." In making the technology open, Digital sought to provide a broader and richer set of products than the company could provide by itself and in so doing extend the range of Alpha AXP technology and the competitiveness of Alpha AXP products in the market. This represented a significant departure from the operating business model of Digital's successful VAX business, where the technology was proprietary to Digital. Accordingly, the Alpha program required significant changes to previous business practices. Ongoing interaction with customers and business partners helped shape and clarify these changes. The resulting initiative to make the Alpha AXP technology open consisted of three primary components:

1. Digital would sell Alpha AXP technology at all levels of integration—chip, module, system.
2. Digital would provide open licensing of Alpha AXP technology.
3. Digital would work closely with partners to extend the range of Alpha AXP technology and products in the market.

The first key element in opening the Alpha AXP technology was the decision to sell the technology at all levels of integration. With access to the technology at multiple levels of integration, customers and business partners can focus on their own development or application areas of expertise and extend Alpha AXP technology to new products or markets in ways that most effectively meet their own business needs. The three papers from Cray Research, Raytheon, and Kubota in this issue of the *Digital Technical Journal* are good examples of utilizing and extending the range of Alpha AXP technology from three different levels of integration.

The CRAY T3D massively parallel processor (MPP) system utilizes Alpha AXP technology at the chip level. Building on the performance leadership of the Alpha AXP microprocessor, Cray Research focused on key areas in the development of a leadership MPP system—communication and memory interconnect, packaging, and the programming model and tools.

Starting with Digital's AXPvme 64 module, Raytheon adapted it to meet the extended environmental and reliability requirements for defense application. By starting with an existing module design, Raytheon was able to maintain software compatibility with commercial Alpha AXP systems, thus providing a very cost-effective way of deploying advanced Alpha AXP computer technology in a military environment.

Lastly, starting from the system level, Kubota developed an advanced 3D imaging and graphics accelerator for Digital's DEC 3000 AXP workstation systems. Using the basic system capabilities of the workstation, Kubota's 3D imaging and graphics accelerator extends the range of the Alpha AXP technology to high-performance medical imaging, seismic, and computational science applications—even to the realm of virtual reality games.

The decision to sell at all levels of integration meant that Digital's Semiconductor Operations moved from being a captive supplier of microprocessor and peripheral support chips exclusively for Digital's systems business to being an open merchant supplier. Concurrently, it also meant an expansion of Digital's OEM business at the module and system level. Whereas the business infrastructure was already in place for Digital to expand the board and systems OEM business, some changes were required to meet the needs of external chip customers in ways different from those established with Digital's internal systems groups. Previously, technical support was provided informally, chip designer to system designer, while the development tools and supporting peripheral chips required for designing-in the microprocessor were often developed uniquely by the system group itself. Along with the marketing and application support resources required to support Digital's

Semiconductor Operations as a merchant supplier, a full range of hardware and software development tools and supporting peripheral chips needed to be developed to support the family of Alpha AXP microprocessors for external customers. The fourth paper in this issue describes part of this "whole product" solution developed for the DECchip 21064 microprocessor—the PCI core logic chip set and an evaluation board kit. Together, the chip set and the evaluation board kit (which includes OSF/1 or Windows NT software tools) provide customers the ability to develop Alpha AXP PCI systems with minimal design and engineering effort.

A second fundamental element in opening the Alpha AXP technology to the broad marketplace was to openly license the technology. A critical requirement of both chip customers and potential partners was that Alpha AXP microprocessors be available from a second source to (1) ensure their security of supply and (2) extend the range of chip implementations to broaden the markets served by the Alpha AXP technology. This is the basis for the Alpha AXP semiconductor partnership with Mitsubishi Electric Corporation announced in March 1993. Mitsubishi plans to begin supplying Alpha AXP microprocessors based on 0.5-micron technology to the open market by the end of 1994. In addition to licensing the chip and architecture, Digital also licenses other elements of the Alpha AXP technology to meet the needs of our customers and partners, including Digital's OSF/1 UNIX operating system.

With access at all levels of integration and through open licensing, Digital sought and established multiple partner and customer relationships to extend the range of Alpha AXP technology and products in the market. From portable computing to supercomputing, from embedded applications to complete system solutions, over seventy-five companies are currently using Alpha AXP technology in their products. This issue of the *Digital Technical Journal* provides a sampling of the ever-broadening set of Alpha AXP products and applications enabled through open access to the technology.

R. Kent Koeninger
Mark Furtney
Martin Walker

A Shared Memory MPP from Cray Research

The CRAY T3D system is the first massively parallel processor from Cray Research. The implementation entailed the design of system software, hardware, languages, and tools. A study of representative applications influenced these designs. The paper focuses on the programming model, the physically distributed, logically shared memory interconnect, and the integration of Digital's DECchip 21064 Alpha AXP microprocessor in this interconnect. Additional topics include latency-hiding and synchronization hardware, libraries, operating system, and tools.

Today's fastest scientific and engineering computers, namely supercomputers, fall into two basic categories: parallel vector processors (PVPs) and massively parallel processors (MPPs). Systems in both categories deliver tens to hundreds of billions of floating-point operations per second (GFLOPS) but have memory interconnects that differ significantly. After presenting a brief introduction on PVPs to provide a context for MPPs, this paper focuses on the design of MPPs from Cray Research.

PVPs have dominated supercomputing design since the commercial success of the CRAY-1 supercomputer in the 1970s. Modern PVPs, such as the CRAY C90 systems from Cray Research, continue to provide the highest sustained performance on a wide range of codes. As shown in Figure 1, PVPs use dozens of powerful custom vector processors on a high-bandwidth, low-latency, shared-memory interconnect. The vector processors are on one side of the interconnect with hundreds to thousands of memories on the other side. The interconnect has uniform memory access, i.e., the latency and bandwidth are uniform from all processors to any word of memory.

MPPs implement a memory architecture that is radically different from that of PVPs. MPPs can deliver peak performance an order of magnitude faster than PVP systems but often sustain performance lower than PVPs. A major challenge in MPP design is to enable a wide range of applications to sustain performance levels higher than on PVPs.

The work described in this paper was partially supported by the Advanced Research Projects Agency under Agreement No. MDA972-92-0002 dated January 21, 1992.

MPPs typically use hundreds to thousands of fast commercial microprocessors with the processors and memories paired into distributed processing elements (PEs). The MPP memory interconnects have tended to be slower than the high-end PVP memory interconnects. The MPP interconnects have nonuniform memory access, i.e., the access speed (latency and bandwidth) from a processor to its local memory tends to be faster than the access speed to remote memories.

The processing speed and memory bandwidth of each microprocessor are substantially lower than those of a vector processor. Even so, the sum of the speeds of hundreds or thousands of microprocessors can often exceed the aggregate speed of dozens of vector processors by an order of magnitude. Therefore, a goal for MPP design is to raise the efficiency of hundreds of microprocessors working in parallel to a point where they perform more useful work than can be performed on the traditional PVPs. Improving the microprocessor interconnection network will broaden the spectrum of MPP applications that have faster times-to-solution than on PVPs.

A key architectural feature of the CRAY T3D system is the use of physically distributed, logically shared memory (distributed-shared memory). The memory is physically distributed in that each PE contains a processor and a local dynamic random-access memory (DRAM); accesses to local memory are faster than accesses to remote memories. The memory is shared in that any processor can read or write any word in any of the remote PEs without the assistance or knowledge of the remote processors or the operating system. Cray Research provides a shell of circuitry around the processor that allows

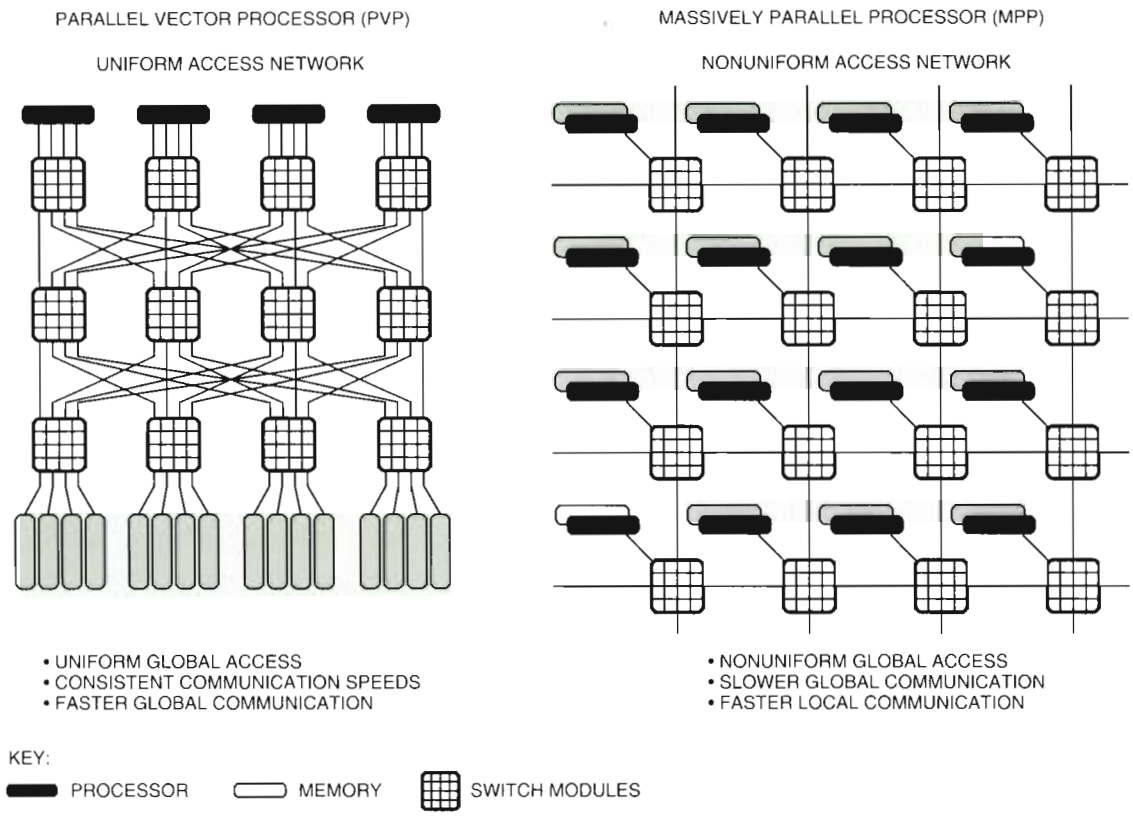


Figure 1 Memory Interconnection Architectures

the local processor to issue machine instructions to read remote memory locations. Distributed-shared memory is a significant advance in balancing the ratio between remote and local memory access speeds. This balance, in conjunction with new programming methods that exploit this new capability, will increase the number of applications that can run efficiently on MPPs and simplify the programming tasks.

The CRAY T3D design process followed a top-down flow. Initially, a small team of Cray Research applications specialists, software engineers, and hardware designers worked together to conduct a performance analysis of target applications. The team extracted key algorithmic performance traits and analyzed the performance sensitivity of MPP designs to these traits. This activity was accomplished with the invaluable assistance and advice of a select set of experienced MPP users, whose insights into the needs of high-performance computing profoundly affected the design. The analysis identified key fundamental operations and hardware/software features required to execute parallel programs with high performance. A series of discussions on engineering trade-offs, software reusability issues, interconnection design studies and simulations, programming model designs, and performance considerations led to the final design.

The resulting system architecture is a distributed memory, shared address space, multiple instruction, multiple data (MIMD) multiprocessor. Special latency-hiding and synchronization hardware facilitates communication and remote memory access over a fast, three-dimensional (3-D) torus interconnection network. The majority of the remote memory accesses complete in less than 1 microsecond, which is one to two orders of magnitude faster than on most other MPPs.[1,2,3]

A fundamental challenge for the CRAY T3D system (and for other MPP systems) is usability. By definition, an MPP with high usability would sustain higher performance than traditional PVP systems for a wide range of codes and would allow the programmer to achieve this high performance with a reasonable effort. Several elements in the CRAY T3D system combine to achieve this goal.

- The distributed-shared memory interconnect allows efficient, random, single-word access from any processor to any word of memory.
- Cray's distributed memory, Fortran programming model with implicit remote addressing is called CRAFT. It provides a standard, high-level interface to this hardware and reduces the effort needed to arrive at near-optimum performance for many problem domains.[4]
- The heterogeneous architecture allows problems to be distributed between an MPP and its PVP host, with the highly parallel portions on the MPP and the serial or moderately parallel portions on the PVP host. This heterogeneous capability greatly increases the range of algorithms that will work efficiently. It also enables stepwise MPP program development, which lets the programmer move code from the PVP to the MPP in stages.
- The CRAY T3D high-speed I/O capabilities provide a close coupling between the MPP and the PVP host. These capabilities sustain the thousands of megabytes per second of disk, tape, and network I/O that tend to accompany problems that run at GFLOPS.

The remainder of this paper is divided into four sections. The first section discusses the results of the applications analysis and its critical impact on the CRAY T3D design, including a summary of critical MPP functionality. The second section characterizes the system software. The software serves multiple purposes; it presents the MPP functionality to the programmer, maps the applications to the hardware, and serves as the interface to the scientist. In the third section, the hardware design is laid out in some detail, including microprocessor selection and the design issues for the Cray shell circuitry that surrounds the core microprocessor and implements the memory system, the interconnection network, and the synchronization capabilities. The fourth section presents benchmark results. A brief summary and references conclude the paper.

The Impact of Applications on Design

As computing power increases, computer simulations increasingly use complex and irregular geometries. These simulations can involve multiple materials with differing properties. A common trend is to improve verisimilitude, i.e., the semblance of reality, through increasingly accurate mathematical descriptions of natural laws.

Consequently, the resolution of models is improving. The use of smaller grid sizes and shorter time scales resolves detail. Models that use irregular and unstructured grids to accommodate geometries may be dynamically adapted by the computer programs as the simulation evolves. The algorithms increasingly use implicit time stepping.

A naïve single instruction, multiple data (SIMD) processor design cannot efficiently deal with the simulation trends and resulting model characteristics. Performing the same operation at each point of space in lockstep can be extremely wasteful. Dynamic methods are necessary to concentrate the computation where variables are changing rapidly and to minimize the computational complexity. The most general form of parallelism, MIMD, is needed. In a MIMD processor, multiple independent streams of instructions act on multiple independent data.

With these characteristics and trends in mind, the design team chose the kernels of a collection of applications to represent target applications for the CRAY T3D system. The algorithms and computational methods incorporated in these kernels were intended to span a broad set of applications, including applications that had not demonstrated good performance on existing MPPs. These kernels included seismic convolution, a partial multigrid method, matrix multiplication, transposition of multidimensional arrays, the free Lagrange method, an explicit two-dimensional Laplace solver, a conjugate gradient algorithm, and an integer sort. The design team exploited the parallelism intrinsic to these kernels by coding them in a variety of ways to reflect different demands on the underlying hardware and software. For example, the team generated different memory reference patterns ranging from local to nearest neighbor to global, with regular and irregular patterns, including hot spots. (Hot spots can occur when many processors attempt to reference a particular DRAM page simultaneously.)

To explore design trade-offs and to evaluate practical alternatives, the team ran different parallel implementations of the chosen kernel on a parameterized system-level simulator. The parameters characterized machine size, the nature of the processors, the memory system, messages and communication channels, and the communications network itself. The simulator measured rates and durations of events during execution of the kernel implementations. These measurements influenced the choices of the hardware and the programming model.

The results showed a clear relationship between the scalability of the applications and the speed of accessing the remote memories. For these algorithms to scale to run on hundreds or thousands of processors, a high-bandwidth, low-latency interprocessor interconnect was imperative. This finding led the designers to choose a distributed-shared memory, 3-D torus interconnect with very fast remote memory access speeds, as mentioned in the previous section.

The study also indicated that a special programming model would be necessary to avoid remote memory accesses when possible and to hide the memory latency for the remaining remote accesses. This finding led to the design of the CRAFT programming model, which uses hardware in the interconnect to asynchronously fetch and store data from and to remote PEs. This model helps programmers to distribute the data among the shared memories and to align the work with this distributed data. Thus, they can minimize remote references and exploit the locality of reference intrinsic to many applications.

The simulations also showed that the granularity of parallel work has a significant impact on both performance and the ease of programming. Performing work in parallel necessarily incurs a work-distribution overhead that must be amortized by the amount of work that gets done by each processor. Fine-grained parallelism eases the programming burden by allowing the programmer to avoid gathering the parallel work into large segments. As the amount of work per iteration decreases, however, the relative overhead of work distribution increases, which lowers the efficiency of doing the work in parallel. Balancing these constraints contributed to the decisions to include a variety of fast synchronization mechanisms, such as a separate synchronization network to minimize the overhead of fine-grained parallelism.

Software

Cray Research met several times a year with a group of experienced MPP users, who indicated that software on existing MPPs was unstable and difficult to use. The users believed that Cray Research needed to provide clear mechanisms for getting to the raw power of the underlying hardware while not diverging too far from existing programming practices. The users wished to port codes from workstations, PVPs, and other MPPs. They wanted to minimize the porting effort while maximizing the resulting performance. The group indicated a strong need for stability, similar to the stability of existing CRAY Y-MP systems. They emphasized the need to preserve their software investments across generations of hardware improvements.

Reusing Stable Software

To meet these goals, Cray Research decided to reuse its existing supercomputing software where possible, to acquire existing tools from other MPPs where appropriate, and to write new software when needed. The developers designed the operating system to reuse Cray's existing UNICOS operating system, which is a superset of the standard UNIX operating system. The bulk of the operating system runs on stable PVP hosts with only microkernels running on the MPP processors. This design enabled Cray Research to quickly bring the CRAY T3D system to market. The resulting system had a minimal number of software changes and retained the maximum stability and the rich functionality of the existing UNICOS supercomputing operating system. The extensive disk, tape, and network I/O capabilities of the PVP host provide the hundreds of megabytes per second of I/O throughput required by the large MPPs. This heterogeneous operating system is called UNICOS MAX.

The support tools (editors, compilers, loaders, debuggers, performance analyzers) reside on the host and create code for execution on the MPP itself. The developers reused the existing Cray Fortran 77 (CF77) and Cray Standard C compilers, with modified front ends to support the MPP programming models and with new code generators to support the DECchip 21064 Alpha AXP microprocessors. They also reused and extended the heart of the compiling systems—the dependency-graph-analysis and optimization module.

The CRAFT Programming Model

The CRAFT programming model extends the Fortran 77 and Fortran 90 languages to support existing popular MPP programming methods (message passing and data parallelism) and to add a new method called work sharing. The programmer can combine explicit and implicit interprocessor communication methods in one program, using techniques appropriate to each algorithm. This support for existing MPP and PVP programming paradigms eases the task of porting existing MPP and PVP codes.

The CRAFT language designers chose directives such that codes written using the CRAFT model run

correctly on machines that do not support the directives. CRAFT-derived codes produce identical results on sequential machines, which ignore the CRAFT directives. Exceptions are hardware limitations (e.g., differing floating-point formats), nondeterministic behavior in the user's program (e.g., timing-dependent logic), and the use of MPP-specific intrinsic functions (i.e., intrinsics not available on the sequential machines).

A message-passing library and a shared memory access library (SMAL) provide interfaces for explicit interprocessor communication. The message-passing library is Parallel Virtual Machine (PVM), a public domain set of portable message-passing primitives developed at the Oak Ridge National Laboratory and the University of Tennessee.[5] The widely used PVM is currently available on all Cray systems. SMAL provides a function call interface to the distributed-shared memory hardware. This provides a simple interface to the programmer for shared memory access to any word of memory in the global address space. These two methods provide a high degree of control over the communication but require a significant programming effort; a programmer must code each communication explicitly.

The CRAFT model supports implicit data-parallel programming with Fortran 90 array constructs and intrinsics. Programmers often prefer this style when developing code on SIMD MPPs.

The CRAFT model provides an additional implicit programming method called work sharing. This method simplifies the task of distributing the data and work across the PEs. Programmers need not explicitly state which processors will have which specific parts of a distributed data array. Similarly, they need not specify which PEs will perform which parts of the work. Instead, they use high-level mechanisms to distribute the data and to assist the compiler in aligning the work with the data. This technique allows the programmers to maximize the locality of reference with minimum effort.

In work sharing, programmers use the SHARED directives to block the data across the distributed memories. They distribute work by placing DO SHARED directives in front of DO loops or by using Fortran 90 array statements. The compiler aligns the work with the data and doles out each iteration of a loop to the PE where most of the data associated with the work resides. Not all data needs to be local to the processor.

The hardware and the programming model can accommodate communication-intensive programs. The compiler attempts to prefetch data that resides in remote PEs, i.e., it tends to copy remote data to local temporaries before the data is needed. By prefetching multiple individual words over the fast interconnect, the compiler can mask the latency of remote memory references. Thus, locality of reference, although still important, is less imperative than on traditional MPP systems. The ability to fetch individual words provides a very fine-grained communication capability that supports random or strided access to remote memories.

The programming model is built on concepts that are also available in Fortran D, Vienna Fortran, and the proposed High-performance Fortran (HPF) language definition.[6,7,8] (Cray Research participates in the HPF Forums.) These models are based on Mehrotra's original Kali language definition and on some concepts introduced for the ILLIAC IV parallel computer by Millstein.[9,10]

Libraries

Libraries for MPP systems can be considered to consist of two parts: (1) the system support libraries for I/O, memory allocation, stack management, mathematical functions (e.g., SIN and COS), etc., and (2) the scientific libraries for Basic Linear Algebra Subroutines (BLAS), real and complex fast Fourier transforms, dense matrix routines, structured sparse matrix routines, and convolution routines. Cray Research used its current expertise in these areas, plus some third-party libraries, to develop high-performance MPP libraries with all these capabilities.

Tools

A wide variety of support tools is available to aid application developers working on the CRAY T3D system. Included in the Cray tool set are loaders, simulators, an advanced emulation environment, a full-featured MPP debugger, and tools that support high-level performance tuning.

Performance Analysis A key software tool is the MPP Apprentice, a performance analysis tool based in part on ideas developed by Cray Research for its ATExpert tool.[11] The MPP Apprentice tool has expert system capabilities to guide users in evaluating their data and work distributions and in suggesting ways to enhance the overall algorithm, application, and program performance.

The MPP Apprentice processes compiler and runtime data and provides graphical displays that relate performance characteristics to a particular subprogram, code block, and line in the user's original source code. The user can select a code block and obtain many different kinds of detailed information. Specific information on the amount of each type of overhead, such as synchronization constructs and communication time, let the user know precisely how and where time is being spent. The user can see exactly how many floating-point instructions, global memory references, or other types of instructions occur in a selected code block.

Debugging Cray Research supplies the Cray TotalView tool, a window-oriented multiprocessor symbolic debugger based on the TotalView product from Bolt Beranek and Newman Inc. The Cray TotalView tool is capable of debugging multiple-process, multiple-processor programs, as well as single-process programs, and provides a large repertoire of features for debugging programs written in Fortran, C, or assembly language.

An important feature of the debugger is its window-oriented presentation of information. Besides displaying information, the interface allows the user to edit information and take other actions, such as modifying the values of the variables.

The debugger offers the following full range of functions for controlling processes:

- Set and clear breakpoints (at the source or machine level)
- Set and clear conditional breakpoints and evaluation points
- Start, stop, resume, delete, and restart processes
- Attach to existing processes
- Examine core files
- Single-step source lines through a program, including stepping across function calls

Emulator Cray Research has implemented an emulator that allows the user to execute MPP programs before gaining access to a CRAY T3D system by emulating CRAY T3D codes on any CRAY Y-MP system. The emulator supports Fortran programs that use the CRAFT model, including message-passing and data-parallel constructs, and C programs that use message passing. Because it provides feedback on data locality, work distribution, program correctness, and performance comparisons, the emulator is useful for porting and developing new codes for the CRAY T3D system.

Hardware

A macro- and microarchitecture design was chosen to resolve the conflict of maximizing hardware performance improvements between generations of MPPs while preserving software investments. This architecture allows Cray Research to choose the fastest microprocessor for each generation of Cray MPPs. The macroarchitecture implements the memory system and the interconnection network with a set of Cray proprietary chips (shell circuitry) that supports switching, synchronization, latency-hiding, and communication capabilities. The macroarchitecture will undergo only modest changes over a three-generation life cycle of the design. Source code compatibility will be maintained. The microarchitecture will allow the instruction set to change while preserving the macroarchitecture.

Macroarchitecture

The CRAY T3D macroarchitecture has characteristics that are both visible and available to the programmer. These characteristics include

- Distributed memory
- Global address space
- Fast barrier synchronization, e.g., forcing all processors to wait at the end of a loop until all other processors have reached the end of the loop
- Support for dynamic loop distribution, e.g., distributing the work in a loop across the processors in a manner that minimizes the number of remote memory references
- Hardware messaging support
- Support for fast memory locks

Memory Organization

The CRAY T3D system has a distributed-shared memory built from DRAM parts. Any PE can directly address any other PE's memory, within the constraints imposed by security and partitioning. The physical address of a data element in the MPP has two parts: a PE number and an offset within the PE, as shown in Figure 2.

CRAY T3D memory is distributed among the PEs. Each processor has a favored low-latency,

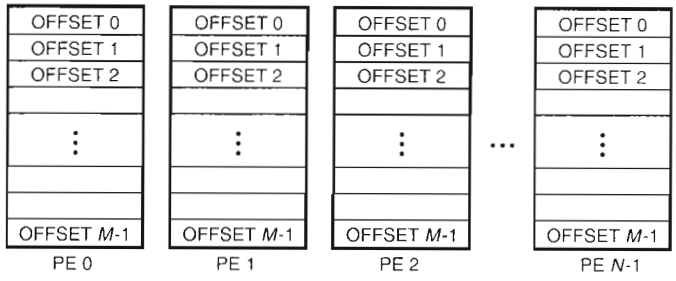


Figure 2 Memory Layout

high-bandwidth path to its local memory and a longer-latency, lower-bandwidth path to memory associated with other processors (referred to as remote or global memory).

Data Cache The data cache resident on Digital's DECchip 21064 Alpha AXP microprocessor is a write-through, direct-mapped, read-allocate cache. CRAY T3D hardware does not automatically maintain the coherence of the data cache relative to remote memory. The CRAFT programming model manages this coherence and guarantees the integrity of the data.

Local and Remote Memory Each PE contains 16 or 64 megabytes of local DRAM with a latency of 13 to 38 clock cycles (87 to 253 nanoseconds) and a bandwidth of up to 320 megabytes per second. Remote memory is directly addressable by the processor, with a latency of 1 to 2 microseconds and a bandwidth of over 100 megabytes per second (as measured in software). All memory is directly accessible; no action is required by remote processors to formulate responses to remote requests. The total size of memory in the CRAY T3D system is the number of PEs times the size of each PE's local memory. In a typical 1,024-processor system, the total memory size would be 64 gigabytes.

3-D Torus Interconnection Network

The CRAY T3D system uses a 3-D torus for the interconnection network. A 3-D torus is a cube with the opposing faces connected. Connecting the faces provides dual paths (one clockwise and one counterclockwise) in each of the three dimensions. These redundant paths increase the resiliency of the system, increase the bandwidth, and shorten the average distance through the torus. The three dimensions keep the distances short; the length of any one dimension grows as the cube root of the number of nodes. (See Figure 3.)

When evaluated within the constraints of real-world packaging limits and wiring capabilities, the 3-D torus provided the highest global bandwidth and lowest global latency of the many interconnection networks studied.[1,2,3] Using three dimensions was optimum for systems with hundreds or thousands of processors. Reducing the system to two dimensions would reduce hardware costs but would substantially decrease the global bandwidth, increase the network congestion, and increase the average latency. Adding a fourth dimension would add bandwidth and reduce the latency, but not enough to justify the increased cost and packaging complexity.

Network Design

The CRAY T3D network router is implemented using emitter-coupled logic (ECL) gate arrays with approximately 10,000 gates per chip. The router is dimension sliced, which results in a network node composed of three switch chips of identical design—one each for X-, Y-, and Z-dimension routing. The router implements a dimension-order, wormhole routing algorithm with four virtual channels that avoid potential deadlocks between the torus cycle and the request and response cycles.

Every network node has two PEs. The PEs are independent, having separate memories and data paths; they share only the bandwidth of the network and the block transfer engine (described in detail later in the paper). A 1,024-PE system would therefore have a 512-node network configured as a 3-D torus with XYZ dimensions of $8 \times 8 \times 8$.

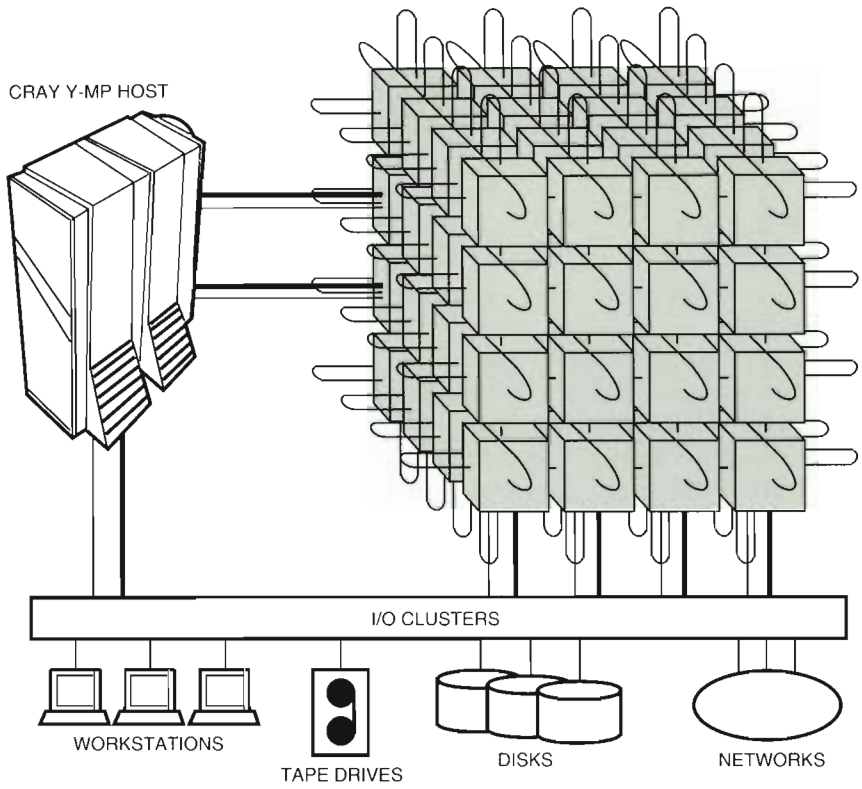


Figure 3 CRAY T3D System

The network moves data in packets with payload sizes of either one or four 64-bit words. Efficient transport of single-word payloads is essential for sparse or strided access to remote data, whereas the 4-word payload minimizes overhead for dense data access.

For increased fault tolerance, the CRAY T3D system also provides spare compute nodes that are used if nodes fail. There are two redundant PEs for every 128 PEs. A redundant node can be electronically switched to replace a failed compute node by rewriting the routing tag lookup table.

Latency of the switch is very low. A packet entering a switch chip requires only 1 clock cycle (6.67 nanoseconds at 150 megahertz [MHz]) to select its output path and to exit. The time spent on the physical wires is not negligible and must also be included in latency calculations. In a CRAY T3D system, all network interconnection wires are either 1 or 1.5 clock cycles long. Each hop through the network requires 1 clock cycle for the switch plus 1 to 1.5 clock cycles for the physical wire. Turning a corner is similar to routing within a dimension. The time required is 3 clock cycles: 1 clock cycle inside the first chip, 1 clock cycle for the connection between chips, and 1 clock cycle for the second chip, after which the packet is on the wires in the next dimension.

The result is an interconnection network with low latency. As stated previously in the Memory Organization subsection, the latency for a 1,024-PE system, including the hardware and software overhead, is between 1 and 2 microseconds.

Each channel into a switch chip is 16 bits wide and runs at 150 MHz, for a raw bandwidth of 300 megabytes per second. Seven channels enter and seven channels exit a network node: one channel to and one channel from the compute resource, i.e., the pair of local PEs, and six two-way connections to the nearest network neighbors in the north, south, east, west, up, and down directions. All fourteen channels are independent. For example, one packet may be traversing a node from east to west at the same time another packet is traversing the same node from west to east or north to south, etc.

The bandwidth can be measured in many ways. For example, the bandwidth through a node is 4.2 gigabytes per second (300 megabytes per second

times 14). A common way to measure system bandwidth is to bisect the system and measure the bandwidth between the two resulting partitions. This bisection bandwidth for a 1,024-PE CRAY T3D torus network is 76 gigabytes per second.

Microarchitecture—The Core Microprocessor

The CRAY T3D system employs Digital's DECchip 21064 Alpha AXP microprocessor as the core of the processing element. Among the criteria for choosing this reduced instruction set computer (RISC) microprocessor were computational performance, memory latency and bandwidth, power, schedule, vendor track record, cache size, and programmability. Table 1, the *Alpha Architecture Reference Manual,* and the *DECchip 21064-AA Microprocessor Hardware Reference Manual* provide details on the Alpha AXP microprocessor.[12,13]

For use in a shared address space MPP, all commercially available microprocessors contemporaneous with the DECchip 21064 device have three major weaknesses in common:[14]

1. Limited address space
2. Little or no latency-hiding capability
3. Few or no synchronization primitives

These limitations arise naturally from the desktop workstation and personal computer environments for which microprocessors have been optimized. A desktop system has a memory that is easily addressed by 32 or fewer bits. Such a system possesses a large board-level cache to reduce the number of memory references that result in the long latencies associated with DRAM. The system usually is a uniprocessor, which requires little support for multiple processor synchronization. Cray Research designed a shell of circuitry around the core DECchip 21064 Alpha AXP microprocessor in the CRAY T3D system to extend the microprocessor's capabilities in the three areas.

Address Extension

The Alpha AXP microprocessor has a 43-bit virtual address space that is translated in the on-chip data translation look-aside buffer (DTB) to a 34-bit address space that is used to address physical bytes of DRAM. Thirty-four bits can address up to 16 gigabytes (2^{34} bytes). Since the CRAY T3D system has up to 128 gigabytes (2^{37} bytes) of distributed-shared memory, at least 37 bits of physical address are required. In addition, several more address bits are needed to control caching and to facilitate control of the memory-mapped mechanisms that implement the external MPP shell. The CRAY T3D system uses a 32-entry register set called the DTB Annex to

Table 1 CRAY T3D Core Microprocessor Specifications

Characteristic	Specification
Microprocessor	Digital's DECchip 21064 Alpha AXP microprocessor
Clock cycle	6.67 nanoseconds
Bidirectional data bus	128 bits data, 28 check bits
Data error protection	SECDED
Address bus	34 bits
Issue rate	2 instructions/clock cycle
Internal data cache	8K bytes (256 32-byte lines)
Internal instruction cache	8K bytes (256 32-byte lines)
Latency: data cache hit	3 clock cycles
Bandwidth: data cache hit	64 bits/clock cycle
Floating-point unit	IEEE floating-point and floating-point–to–integer
Floating-point registers	32 (64 bits each)
Integer execution unit	Integer arithmetic, shift, logical, compare
Integer registers	32 (64 bits each)
Integrated circuit	CMOS, 14.1 mm × 16.8 mm
Pin count	431 (229 signal)
Typical power dissipation	−23 watts

extend the number of physical address bits beyond the 34 provided by the microprocessor.

Shell circuitry always checks the virtual PE number. If the number matches that of the local PE, the shell performs a local memory reference instead of a remote reference.

Latency-hiding Mechanisms

As with most other microprocessors, the external interface of the DECchip 21064 is not pipelined; only one memory reference may be pending at any one time. Although merely an annoyance for local accesses, this behavior becomes a severe performance restriction for remote accesses, with their longer latencies, unless external mechanisms are added to extend the processor's memory pipeline.

The CRAY T3D system provides three mechanisms for hiding the startup time (latency) of remote references: (1) the prefetch queue, (2) the remote processor store, and (3) the block transfer engine. As shown in Table 2, each mechanism has its own strengths. The compilers, communication libraries, and operating system choose among these mechanisms according to the specific remote reference requirements. Typically, the prefetch queue and the remote processor store are the most effective mechanisms for fine-grained communication, whereas the block transfer engine is strongest for moving large blocks of data.

The Prefetch Queue The DECchip 21064 instruction set includes an operation code FETCH that permits a compiler to provide a "hint" to the hardware of upcoming memory activity. Originally, the FETCH instruction was intended to trigger a prefetch to the external secondary cache. The CRAY T3D shell hardware uses FETCH to initiate a single-word remote memory read that will fill a slot reserved by the hardware in an external prefetch queue.

The prefetch queue is first in, first out (FIFO) memory that acts as an external memory pipeline. As the processor issues each FETCH instruction, the shell hardware reserves a location in the queue for the return data and sends a memory read request packet to the remote node. When the read data returns to the requesting processor, the shell hardware writes the data into the reserved slot in the queue.

The processor retrieves data from the FIFO queue by executing a load instruction from a memory-mapped register that represents the head of the queue. If the data has not yet returned from the remote node, the processor will stall while waiting for the queue slot to be filled.

The data prefetch queue is able to store up to 16 words, that is, the processor can issue up to 16 FETCH instructions before executing any load instructions to remove (pop) the data from the head of the queue. Repeated load instructions from the memory-mapped location that addresses the head of the queue will return successive elements in the order in which they were fetched.

The Remote Processor Store The DECchip 21064 stores to remote memory do not need to wait for a response, so a large number of store operations can be outstanding at any time. This is an effective communication mechanism when the producer of the data knows which PEs will immediately need to use the data.

The Alpha AXP microprocessor has four 4-word write buffers on chip that try to accumulate a cache line (4 words) of data before performing the actual external store. This feature increases the network packet payload size and the effective bandwidth.

The CRAY T3D system increments a counter in the PE shell circuitry each time the DECchip 21064 microprocessor issues a remote store and decrements the

Table 2 Latency-hiding Attributes

	Prefetch Queue	Remote Processor Store	Block Transfer Engine
Source	Memory	Register	Memory
Destination	Local queue	Memory	Memory
Data Size	1 word	1–4 words	Up to 256K words
Startup (6.67-nanosecond clock cycles)	18–47	6–53	>480
Latency (nanoseconds)	80	40	40–80

counter each time a write operation completes. For synchronization purposes, the processor can read this counter to determine when all of its writes have completed.

The Block Transfer Engine The block transfer engine (BLT) is an asynchronous direct memory access controller used to redistribute data between local and remote memory. To facilitate reorganization of sparse or randomly organized data, the BLT includes scatter-gather capabilities in addition to constant strides. The BLT operates independently of the processors at a node, in essence appearing as another processor in contention for memory, data path, and switch resources. Cray Research has a patent pending for a centrifuge unit in the BLT that accelerates the address calculations in the CRAFT programming model.

The processor initiates BLT activity by storing individual request information (for example, starting address, length, and stride) in the memory-mapped control registers. The overhead associated with this setup work is noticeable (tens of microseconds), which makes the BLT most effective for large data block moves.

Synchronization

The CRAY T3D system provides hardware primitives that facilitate synchronization at various levels of granularity and support both control parallelism and data parallelism. Table 3 presents the characteristics of these synchronization primitives.

Barrier The CRAY T3D has specialized barrier hardware in the form of 16 parallel logical AND trees that permit multiple barriers to be pipelined and the resource to be partitioned. When all PEs in the partition have reached the barrier and have set the same bit to a one, the AND function is satisfied and the barrier bit in each PE's barrier register is cleared by hardware, thus signaling the processors to continue.

Table 3 Synchronization Primitives

Primitive	Granularity	Parallelism
Barrier	Coarse	Control
Fetch-and-increment	Medium	Both
Lightweight messaging	Medium	Both
Atomic swap	Fine	Data

The barrier has a second mode, called eureka mode, that supports search operations. A eureka is simply a logical OR instead of a logical AND and can be satisfied by any one processor.

The barrier mechanism in the CRAY T3D system is quite fast. Even for the largest configuration (i.e., 2,048 PEs), a barrier propagates in less than 50 clock cycles (about 330 nanoseconds), which is roughly the latency of a local DRAM read.

Fetch and Increment The CRAY T3D system has specialized fetch-and-increment hardware as part of a shared register set that automatically increments the contents each time the register is read. Fetch-and-increment hardware is useful for distributing control with fine granularity. For example, it can be used as a global array index, shared by multiple processors, where each processor increments the index to determine which element in an array to process next. Each element can be guaranteed to be processed exactly once, with minimal control overhead.

Messaging A messaging facility in the CRAY T3D system enables the passing of packets of data from one processor to another without having an explicit destination address in the target PE's memory. A message is a special cache-line-size write that has as its destination a predefined queue area in the memory of the receiving PE. The shell circuitry manages the queue pointers, providing flow control mechanisms to guarantee the correct delivery of the messages. The shell circuitry interrupts the target processor after a message is stored.

Atomic Swap Atomic swap registers are provided for the exchange of data with a memory location that may be remote. The swap is an atomic operation, that is, reading the data from the memory location and overwriting the data with the swap data from the processor is an indivisible operation. As with ordinary memory reads, swap latency can be hidden using the prefetch queue.

I/O

System I/O is performed through multiple Cray high-speed channels that connect the CRAY T3D system to a host CRAY Y-MP system or to standard Cray I/O subsystems. These channels provide hundreds of megabytes per second of throughput to the wide array of peripheral devices and networks already supported on Cray Research mainframes.

Cray has demonstrated individual high-speed channels that can transfer over 100 megabytes per second in each direction, simultaneously. There are two high-speed channels for every 128 processors in a CRAY T3D system.

Benchmark Results

The following benchmarks show results as of May 1994, six months after the release of the CRAY T3D product. The results indicate that in this short span of time, the CRAY T3D system substantially outperformed other MPPs.

As shown in Figure 4, a CRAY T3D system with 256 processors delivered the fastest execution of all eight NAS Parallel Benchmarks on any MPP of any size.[15] (The NAS Parallel Benchmarks are eight codes specified by the Numerical Aerodynamic Simulation [NAS] program at NASA/Ames Research Center. NAS chose these codes to represent common types of fluid dynamics calculations.) The CRAY T3D system scaled these benchmarks more efficiently than all other MPPs, with near linear scaling from 32 to 64, 128, and 256 processors. Other MPPs scaled the benchmarks poorly. None of these other MPPs reported all eight benchmarks scaling to 256 processors, and the scaling reported showed more nonlinear scaling than on the CRAY T3D system. These benchmark results confirm that the superior speed of the CRAY T3D interconnection network is important when scaling a wide range of algorithms to run on hundreds of processors.

Note that a 256-processor CRAY T3D system was the fastest MPP running the NAS Parallel Benchmarks. Even so, the CRAY C916 parallel vector processor ran six of the eight benchmarks faster than the CRAY T3D system. The CRAY T3D system (selling for about $9 million) showed better price/performance than the CRAY C916 system (selling for about $27 million). On the other hand, the CRAY C916 system showed better absolute performance. When we run these codes on a 512-processor CRAY T3D system (later this year), we expect the CRAY T3D to outperform the CRAY C916 system on six of the eight codes.

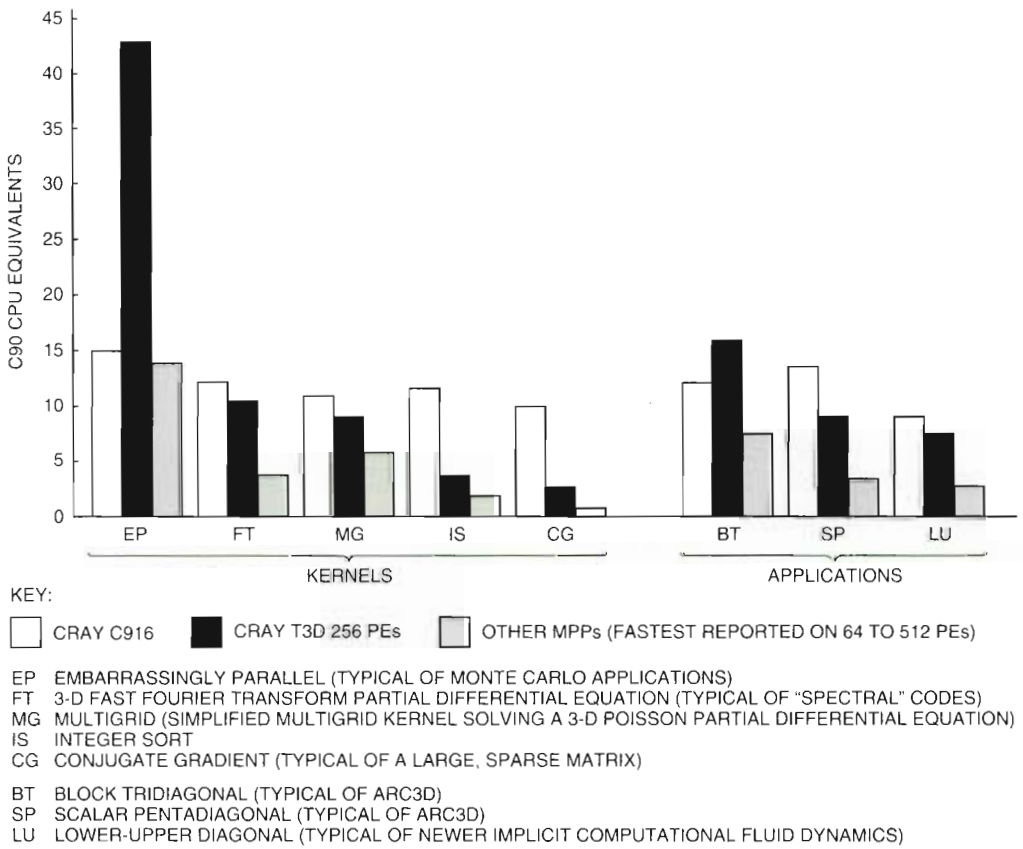


Figure 4 NAS Parallel Benchmarks

Heterogeneous benchmark results are also encouraging. We benchmarked a chemistry application, SUPERMOLECULE, that simulates an imidazole molecule on a CRAY T3D system with a CRAY Y-MP host. The application was 98 percent parallel, with 2 percent of the overall time spent in serial code (to diagonalize a matrix). We made a baseline measurement by running the program on 64 CRAY T3D processors. Quadrupling the number of processors (256 PEs) showed poor scaling—a speedup of 1.3 times over the baseline measurement. When we moved the serial code to a CRAY Y-MP processor on the host, leaving the parallel code on 256 CRAY T3D processors, the code ran 3.3 times faster than the baseline, showing substantially more efficient scaling. Figure 5 shows SUPERMOLECULE benchmark performance results on both homogeneous and heterogeneous systems. Ninety-eight percent may sound like a high level of parallelism, but after dividing 98 percent among 256 processors, each processor ran less than 0.4 percent of the overall parallel time. The remaining serial code running on a single PE ran five times longer than the distributed parallel work, thus dominating the time to solution. Speeding up the serial code by running it on a faster vector processor brought the serial time in line with the distributed-parallel time, improving the scaling considerably.

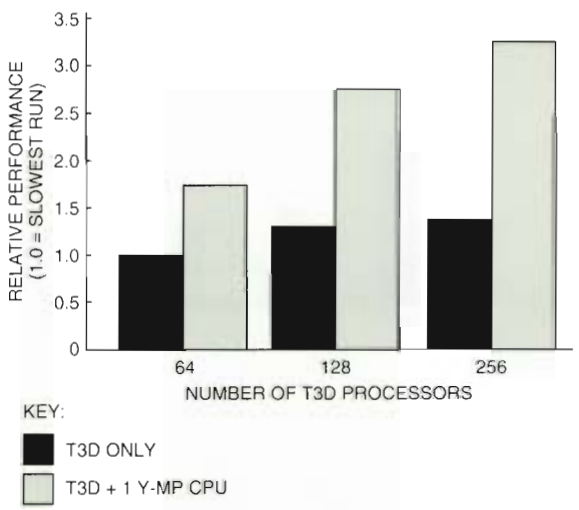


Figure 5 SUPERMOLECULE Benchmark Performance Results for Homogeneous and Heterogeneous Systems

The CRAY T3D system demonstrated faster I/O throughput than any other MPP. A 256-processor system sustained over 570 megabytes per second of I/O to a disk file system residing on a solid-state device on the host. The system sustained over 360 megabytes per second to physical disks.

Summary

This paper describes the design of the CRAY T3D system. Designers incorporated applications profiles and customer suggestions into the CRAFT programming model. The model permits high-performance exploitation of important computational algorithms on a massively parallel processing system. Cray Research designed the hardware based on the fundamentals of the programming model.

As of this writing, a dozen systems have shipped to customers, with results that show the system design is delivering excellent performance. The CRAY T3D system is scaling a wider range of codes to a larger number of processors and running benchmarks faster than other MPPs. The sustained I/O rates are also faster than on other MPPs. The system is performing as designed.

References

1. R. Numrich, P. Springer, and J. Peterson, "Measurement of Communication Rates on the CRAY T3D Interprocessor Network," *Proceedings HPCN Europe (Munich)* (April 1994).

2. R. Kessler and J. Schwarzmeier, "CRAY T3D: A New Dimension for Cray Research," *Proceedings of COMPCON*, 1993: 176–182.

3. S. Scott and G. Thorson, "Optimized Routing in the CRAY T3D," extended abstract for the *Parallel Computing Routing and Communication Workshop* (1994).

4. D. Pase, T. MacDonald, and A. Meltzer, "The CRAFT Fortran Programming Model," CRAY Internal Report (Eagan, MN: Cray Research, Inc., February 1993) and *Scientific Programming* (New York: John Wiley and Sons, forthcoming).

5. A. Geist et al., *PVM 3 User's Guide and Reference Manual* (Oak Ridge, TN: Oak Ridge National Laboratory, ORNL/TM-12187, May 1993).

6. G. Fox et al., *Fortran D Language Specification* (Houston, TX: Department of Computer Science, Rice University, Technical Report TR90-141, December 1990).

7. B. Chapman, P. Mehrotra, and H. Zima, *Vienna Fortran—A Fortran Language Extension for Distributed Memory Multiprocessors* (Hampton, VA: ICASE, NASA Langley Research Center, 1991).

8. *High Performance Fortran (High Performance Fortran Language Specification, Version 1.0)* (May 1993). Also available as technical report CRPC-TR 92225 (Houston, TX: Center for Research on Parallel Computation, Rice University) and in *Scientific Computing* (forthcoming).

9. P. Mehrotra, "Programming Parallel Architectures: The BLAZE Family of Languages," *Proceedings of the Third SIAM Conference on Parallel Processing for Scientific Computing* (December 1988): 289–299.

10. R. Millstein, "Control Structures in ILLIAC IV Fortran," *Communications of the ACM,* vol. 16, no. 10 (October 1973): 621–627.

11. J. Kohn and W. Williams, "ATExpert," *The Journal of Parallel and Distributed Computing,* 18 (June 1993): 205–222.

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

13. *DECchip 21064-AA Microprocessor Hardware Reference Manual,* 1st ed. (Maynard, MA: Digital Equipment Corporation, Order No. EC-N0079-72, October 1992).

14. D. Bailey and R. Schreiber, "Problems with RISC Microprocessors for Scientific Computing" (Moffet Field, CA: NASA/Ames, RNR Technical Report, September 1992).

15. D. Bailey, E. Barszcz, L. Dagum, and H. Simon, "NAS Parallel Benchmark Results" (Moffet Field, CA: NASA/Ames, RNR Technical Report, March 1994 [updated May 1994]).

Robert Couranz |

The E²COTS System and Alpha AXP Technology: The New Computer Standard for Military Use

The translation of Digital products applicable to military application has been affected by the DoD's need for lower cost products. Products developed for military application must retain robust mechanical characteristics; however, each product may be tailored to meet government specifications such as mean time between failure and temperature range. Design changes for military use have had a beneficial second effect. Militarized products may be readily modified to meet a severe industrial environment that previously could only be accomplished with commercial products in special enclosures. As a result of the close cooperation between Digital and Raytheon, cost-effective, severe environment products can be provided to the DoD and the industry.

In 1986, the Raytheon Company and Digital Equipment Corporation entered into a licensing agreement to equip Raytheon's militarized computer system with the best commercial computer technology of the time, Digital's VAX processor. The agreement had two major objectives. The first was to incorporate VAX computer technology into a configuration that complied with the government's existing military specifications. The second was to make the militarized VAX technology available as a strictly commercial effort. The concept was not unique. The Rolm Corporation had militarized a number of the commercial computers designed originally by Data General Corporation, and Norden Systems, Inc. had militarized and marketed Digital's PDP-11 system and earlier VAX processors. Under the Raytheon/Digital agreement, the first computer converted to a configuration usable by the military was the VAX 6200 system. The VAX 6200 incorporated very large-scale integration (VLSI) device technology.

Prior to the introduction of VLSI technology, the militarization of computers was difficult but manageable. The military was a major customer of semiconductor vendors, who would commonly manufacture parts to meet both commercial and military standards. The semiconductors, resistors, capacitors, switches, and other parts were tested and certified to be used in military computers, and the mechanical and electrical structure was also tested to meet extremes of temperature, shock, and vibration. It was, and still is, not unusual to encounter a requirement for computer operation over the temperature range of −54 degrees Celsius to 70 degrees Celsius with a 30-minute period of 85 degrees Celsius.[1] In contrast, the commercial units operate in a benign office environment of 0 degrees Celsius to 50 degrees Celsius.[2–4]

With the evolution of the proprietary VLSI computer in 1986, the cost of developing a new military computer would have strained the government's ability to fund the development of modern architectures to support the advances made in the field of software. The funding of new custom VLSI devices to become the core of military computers required that a large market was available, and the military sector offered only a small percentage of the total market.

Military specifications require the costly and time-consuming testing and documentation that have been in place since World War II. With the end of the Cold War and the serious decline of the Department of Defense (DoD) budget, the military began looking for new ways to procure the weapons

systems using VLSI computers. For many new procurements, the DoD approach has been to buy commercial computers for applications in which the environment is expected to be office-like. The forward edge of the battle area (FEBA), however, is anything but office-like and usually presents environmental challenges that are not normally those anticipated by designers of commercial systems. For example, when one thinks of the climate conditions encountered in the Gulf War, a vision of blowing sand and dry, hot weather comes to mind. In reality, the desert sand is a fine caustic dust, and the air over the Persian Gulf contains significant moisture. The combination is lethal to conventionally designed electronic equipment, etching away unprotected circuit board runs and contacts.

To address the combined budgetary and performance dilemma, Raytheon developed the extended environment, commercial off-the-shelf (E²COTS) computer. To provide the best microprocessor performance available in 1990 and for the forseeable future, the E²COTS computer is powered by Digital's commercial Alpha AXP technology and is constructed to meet the extended environmental needs of defense projects. In addition, that technology is made available to the government via weapon system integrators as a non-developmental item (NDI) and on a strictly commercial basis. As a result, the first of the E²COTS line, Digital's DEC 4000 AXP Model 500 workstation is already flying as the Raytheon Model 920 on the JSTARS aircraft.

This paper explores some of the changes made in the militarization process. It describes the characteristics of the E²COTS computer combined with Alpha AXP technology and the versatile microprocessor (VME) 64 bus. It then discusses the relevance of conduction cooling for the militarized module and design trade-offs based on space and thermal differences.

Characteristics of an E²COTS Computer

There are three major characteristics of an E²COTS computer with Alpha AXP technology:

1. It is software identical to the commercial equivalent.
2. The basic commercial design is modified only to the extent necessary to meet the extended environmental and reliability requirements of the system in which it is employed.
3. It is tested at the unit level to meet the military operational and logistical specifications required of the hardware.

The commercial software (operating system, high-level languages, and development environments) executed on the commercial product can be captured for the E²COTS counterpart. Software executed on the commercial computer can be executed on the E²COTS computer without change at the binary level. Further, the system developer can use benign environment commercial equipment to start developing and testing the military design. Finally, standardized code for high-level languages such as Ada can be readily transported to subsequent E²COTS computers as technology advances.

VLSI computers must be carefully designed to take into consideration even the length of the interconnect etch on the circuit board. A seemingly minor change in the characteristics of the etch may affect the signal timing, cross talk, or similar parameters, resulting in either unreliability or total failure to operate. Thus, any change in the component layout to meet the E²COTS configuration must be undertaken with extreme care and then only when required to meet environmental, reliability, or physical space requirements.

Finally, the historical test methodology of design validation tests every component used in the design. The completed computer is then tested for throughput, power consumption, electromagnetic compatibility, and durability. For the E²COTS system, this expensive and time-consuming test cycle is replaced with the review of the commercial components used in the original design. Based on this review, some components may be replaced with higher quality or specially screened components, and environmental and performance verification testing of the completed computer follows. It should be noted here that testing may be accomplished at the circuit card assembly (CCA) level where such CCAs may be separately developed. CCAs that are used in conjunction with a standardized backplane bus such as the VME bus are typically developed at this level.

Development of an E²COTS Single Module Computer for the VME Bus

The close cooperation between Raytheon and Digital led to an early identification of the DECchip 21066 and DECchip 21068 processors and VME 64 bus based on Alpha AXP technology as an excellent choice for translation into an E²COTS design. Table 1

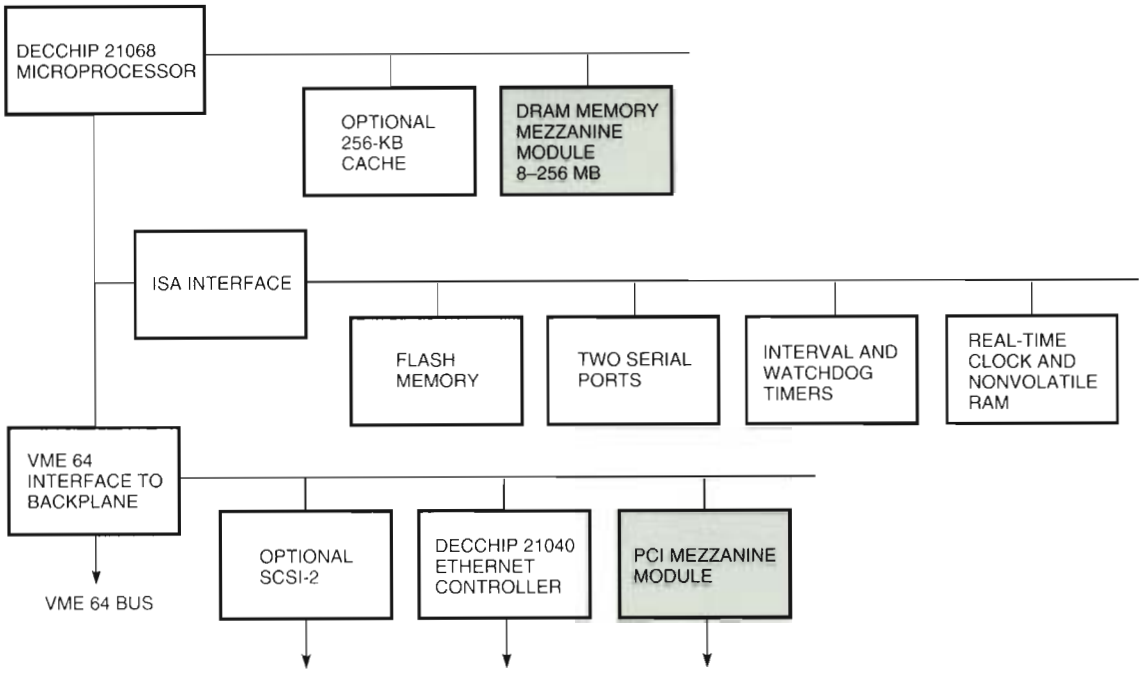
compares the technical specifications of Digital's and Raytheon's modules.[2,5] There are, at present, a number of manufacturers of NDI single module computers that build to a configuration much like the E²COTS specifications. Most, although not all, are based on the Motorola MC68000 series processors. Vendors include Motorola Inc., Radstone Technology, and DY-4 Corporation.

The major reasons for choosing Digital's AXPvme 64 system were the performance and extensive software support desired by embedded processor users. Further, the computer was being designed for the VME 64 backplane bus. The VME bus has been selected by numerous military design organizations to be the backplane bus of choice, providing for an open systems architecture. In addition, the AXPvme 64 system incorporated the peripheral component interconnect (PCI) bus, thereby offering flexibility of I/O design with a minimum of component overhead.[6]

Figure 1 shows the functional block diagram of Digital's AXPvme 64 single module computer. In most applications, computers of this class are used to handle the real-time control of a complex system. The computer uses the DECchip 21068, capable of 40 SPECfp92, as the base processor. It provides standard I/O: small computer systems interface (SCSI-2), Ethernet, two serial ports, and a VME 64 backplane bus interface as well as three configurable timers. Further configuration of the module has been made possible through provisions for two mezzanine modules. The first contains dynamic random-access memory (DRAM) for program and data storage. The second interfaces to the PCI bus and provides the user the option of adding a custom interface to the module.

Table 1 Technical Specifications Comparing the Digital Commercial and the Raytheon E²COTS Single Module Computers

Physical Characteristics	Digital Commercial Module	Raytheon E²COTS Module
Single board computer	Standard Eurocard format (6U) 233 mm × 160 mm (9 inch × 6.25 inch) (20.3 mm) wide	Standard Eurocard format (6U) 233 mm × 160 mm (9 inch × 6.25 inch) (20.33 mm) wide
PCI mezzanine card	5.9 inch × 2.95 inch	2.5 inch × 5.5 inch
Software Support		
Operating systems	DEC OSF/1 AXP, VxWorks for Alpha AXP	DEC OSF/1 AXP, VxWorks for Alpha AXP
Compilers	Ada, Fortran, C/C++	Ada, Fortran, C/C++
Power Requirements		
Power supply voltage	With 32 MB and no PCI options: 7.64 amperes @ 5 VDC and 0.6 ampere @ 12 VDC	With 32 MB and no PCI options: 4.6 amperes @ 5 VDC and 0.6 ampere @ 12 VDC
Environmental Specifications		
Operating temperature	0°C to +50°C with forced air cooling of 200 linear feet per minute at ambient	−54°C to +55°C system ambient (70°C sidewall), +85°C side rail for 30 minutes
Storage temperature	−40°C to +66°C	−62°C to +95°C
Temperature change	20°C per hour	1°C per minute
Relative humidity	10% to 95% (noncondensing)	0% to 100% (condensing)
Mechanical shock	7.5 G peak (±1 G) half sine pulse of 10 ms (±3 ms)	Per MIL-STD-810D Method 516.3
Acceleration	Not specified	9 G continuous operation
Vibration	5–10 Hz 0.02 in double amplitude, 10–500 Hz 0.1 G peak	Sinusoidal 5 Gs 50–2,000 Hz random 0.10 g^2/Hz



Note: Shaded functions are on mezzanine modules.

Figure 1 Block Diagram of Digital's AXPvme 64 Single Board Computer

Conduction Cooling of the Module

The design of a commercial VME module must be modified to meet the needs of the military. Commercial VME modules (as shown in Figure 2) use both the front panel and the connector edges of the module for interconnect. Military systems preclude front (top) of module interfacing because one or more cables may be required to be moved for servicing. This increases maintenance time and the risk of interconnect damage by battlefield personnel.

Standard commercial modules are normally cooled by blowing air over the module. In a commercial installation, the air is drawn from an air-conditioned office environment and is therefore devoid of excess humidity or damaging chemicals. In the military environment, cooling air is expected to contain impurities that will have an adverse effect on the long-term, worldwide reliability of the module. The AXPvme 64 module is convection cooled.[7] One technique used to extend the environmental range of the E²COTS unit is conduction cooling. Conduction cooling eliminates the need to bring air, and with it potentially damaging contaminants, into the computer enclosure. Conformal coating, covering the board and components with a moisture-resistant material similar to plastic, further ensures no contact between the circuit card assembly and contaminants. It also provides protection from condensing humidity. For these reasons, the E²COTS module (shown in Figure 3) is configured to be conduction cooled.

The decomposition of the module assembly in Figure 4 shows a number of techniques used to reduce the thermal resistance between the individual components and the module/side rail interface. The first is the design of the circuit card on which all components are mounted. Figure 5 shows the layer stackup on the circuit board. Power, both 5.0 volt (V) and 3.3 V, and the associated ground planes provide a low-impedance power distribution path for the various components and allow the transmission of heat from the component to the frame and sidewalls. Figure 6 shows the thermal path from a typical surface to the sidewall/heat exchanger. The heat from the component is passed into the copper power planes for transmission to the sidewall/heat exchanger. Due to the low thermal resistance of copper plus the increased thickness of these planes, the thermal resistance is significantly reduced. In addition, the combined copper and polyimide

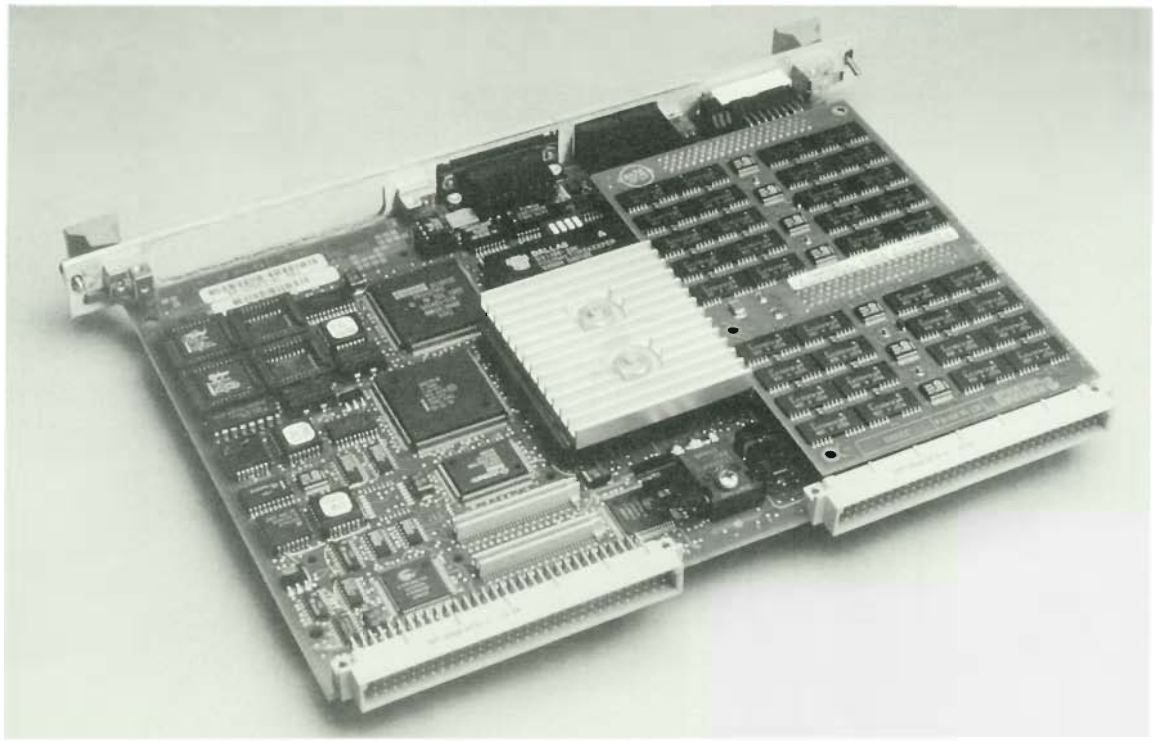


Figure 2 Digital's AXPvme 64 Single Module Computer

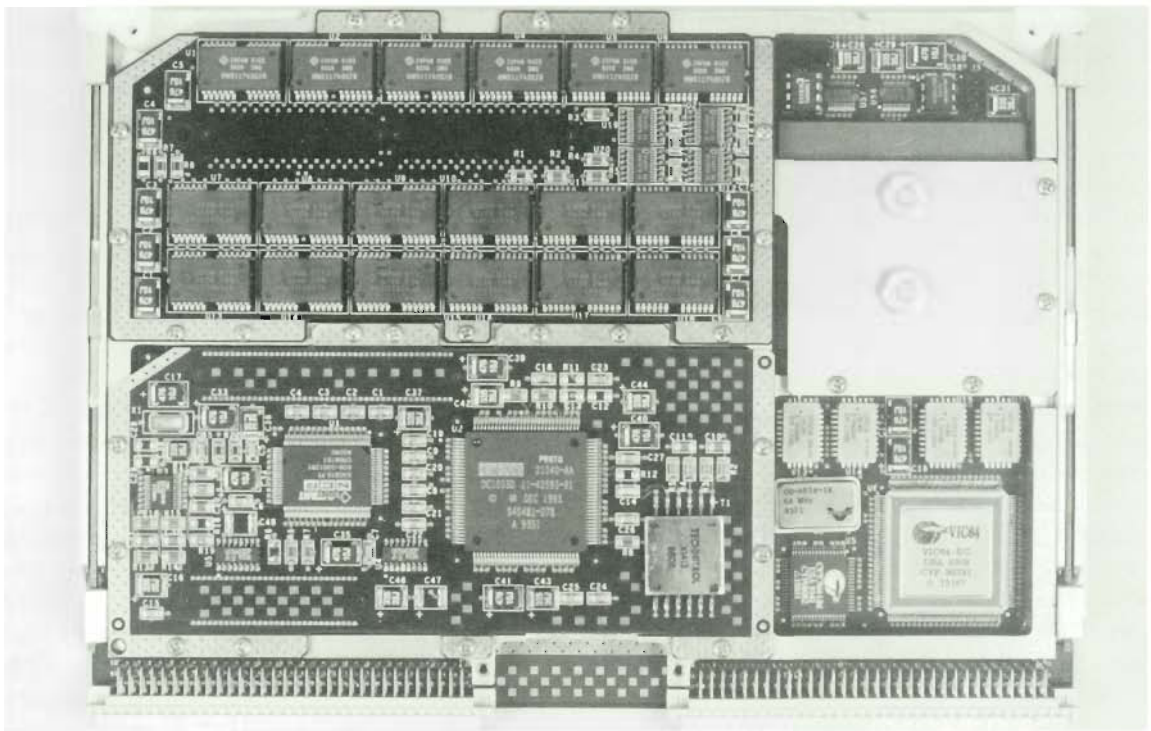


Figure 3 Raytheon Model 910 VME Single Module Computer with Alpha AXP Microprocessor

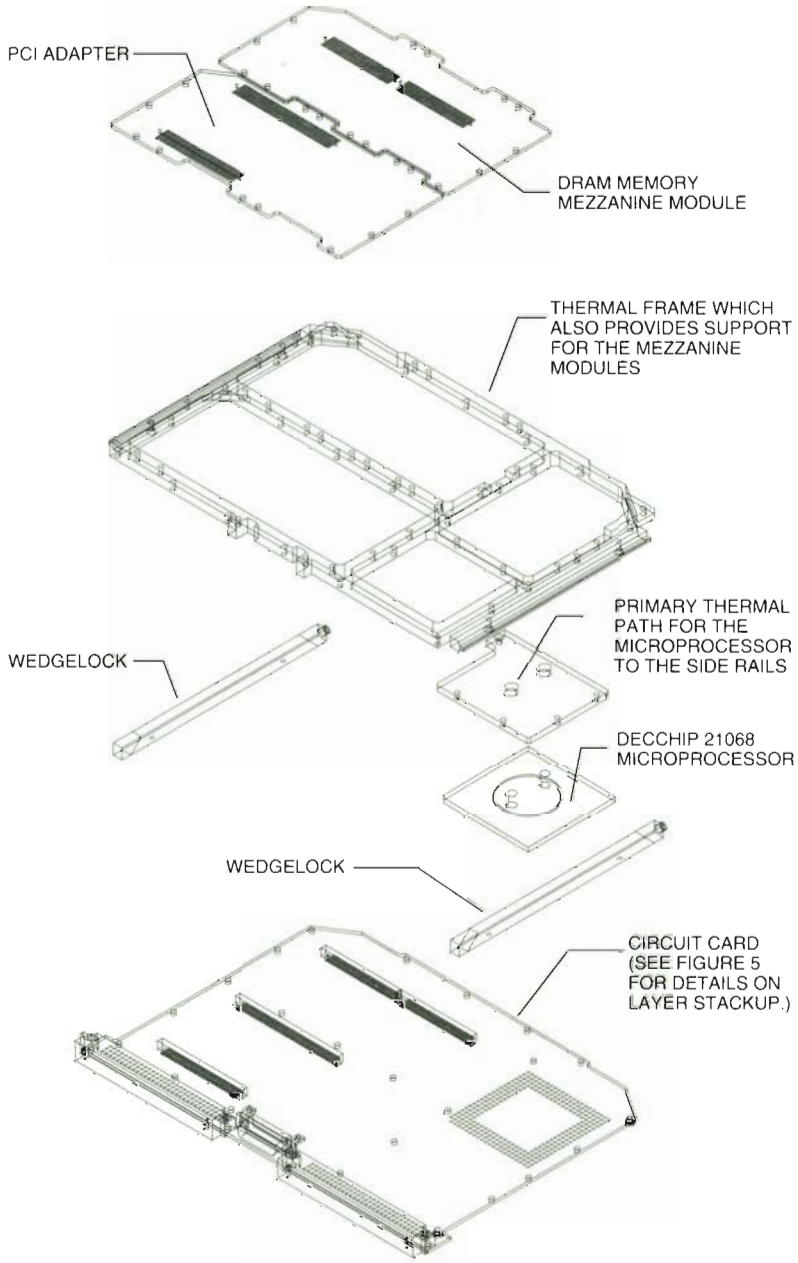


Figure 4 Exploded View of the AXPvme 64 Single Module Computer

layers provide a circuit board with the necessary strength to support the components without an additional backbone, although one is used for other purposes as noted in the next paragraph.

A second technique is the use of a combination thermal and support frame for the memory module and PCI adapter. The use of copper-loaded circuit cards extends to the PCI and memory modules. The thermal path for components mounted on these mezzanine modules is from the component through the circuit board embedded copper to the heat frame. From the heat frame, the thermal path is directly to the sidewall/heat exchangers. The mezzanine modules are designed to be screwed into the heat frame for both minimal thermal resistance and structural support against the shock, vibration, and "g" loading indicated in the technical specifications.

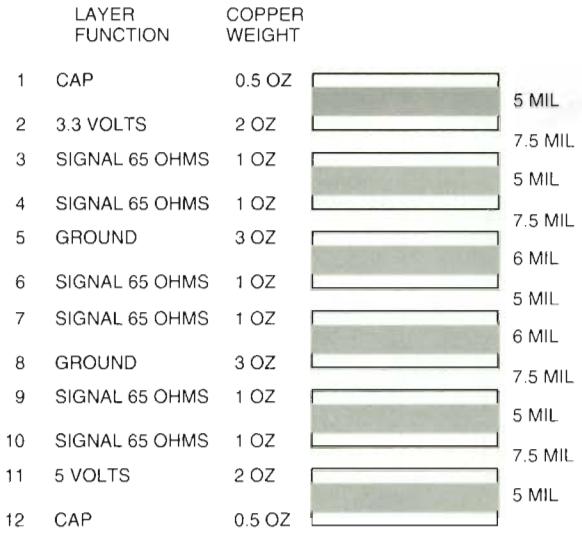


Figure 5 Printed Circuit Board Layer Stackup

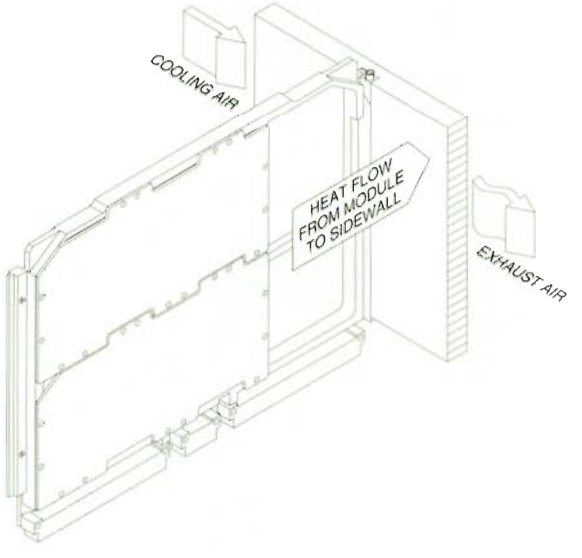


Figure 6 Thermal Flow to the Heat Exchanger

Finally, the two most active thermal radiators are the Alpha AXP processor and the 5.0-VDC to 3.3-VDC regulator. These components have been placed on opposite sides of the circuit board and directly adjacent to the wedgelocks to achieve a minimal thermal path. Because the DECchip 21068 processor is mounted cavity down in the ceramic pin grid array (PGA), its primary thermal path has been provided in the form of a cover plate.

Instead of cooling air passing over the surface of the module, the air is passed through a heat exchanger. Normally this is a brazed sidewall that provides both the outer structural shell of the computer and a duct, which has embedded heat fins for improved heat transfer. Individual modules are structurally connected to the sidewall/heat exchanger by wedgelocks that force a strong mechanical and a relatively low thermal interface.

The nominal temperature rise in the heat exchanger for an air transport rack (ATR) chassis and a total thermal load of approximately 300 watts (W) is 14 degrees Celsius.[5,8] Thus, with a nominal inlet air temperature of 25 degrees Celsius, the wedgelock interface of an E²COTS module is at 39 degrees Celsius. For modules with total thermal dissipation of 20 to 25 W, a nominal 7 degrees Celsius rise is anticipated between the sidewall and the module, yielding a module temperature of 46 degrees Celsius. The heavy aluminum cover essentially maintains the base module temperature to the microprocessor's case. Measurements of the DECchip 21068 processor on the computer have shown an average power of 5.3 W. With a Θ_{J-C} of 1.1 degrees Celsius per watt, the junction temperature is ~52 degrees Celsius. At the normal high end of the temperature range, 70 degrees Celsius inlet air, the chip temperature will increase to 97 degrees Celsius. It should be noted that the examples of temperature rise are nominal and must be computed accurately for each module type, total chassis dissipation, and the position of the module in the chassis.

As part of the thermal analysis of the design, a thermal map of the base module was developed as shown in Figure 7. The figure is an overlay of the thermal profile on the mechanical outline of the E²COTS single module computer. Although planning for the dissipation of power from the microprocessor and the voltage regulator proved successful, the computer-simulated thermal plot indicated a high-temperature region at the top center of the module. This area corresponds to the location of the 256-kilobyte (kB) cache. The junction temperature of the cache static RAMs (SRAMs) could approach 76 degrees Celsius given an inlet air temperature of 25 degrees Celsius.

Although it might be anticipated that the microprocessor would be the board hot spot, the higher thermal resistance of the printed circuit board results in a potentially higher junction temperature of the lower dissipating SRAM devices. Operating at 70 degrees Celsius inlet air temperature, the resultant

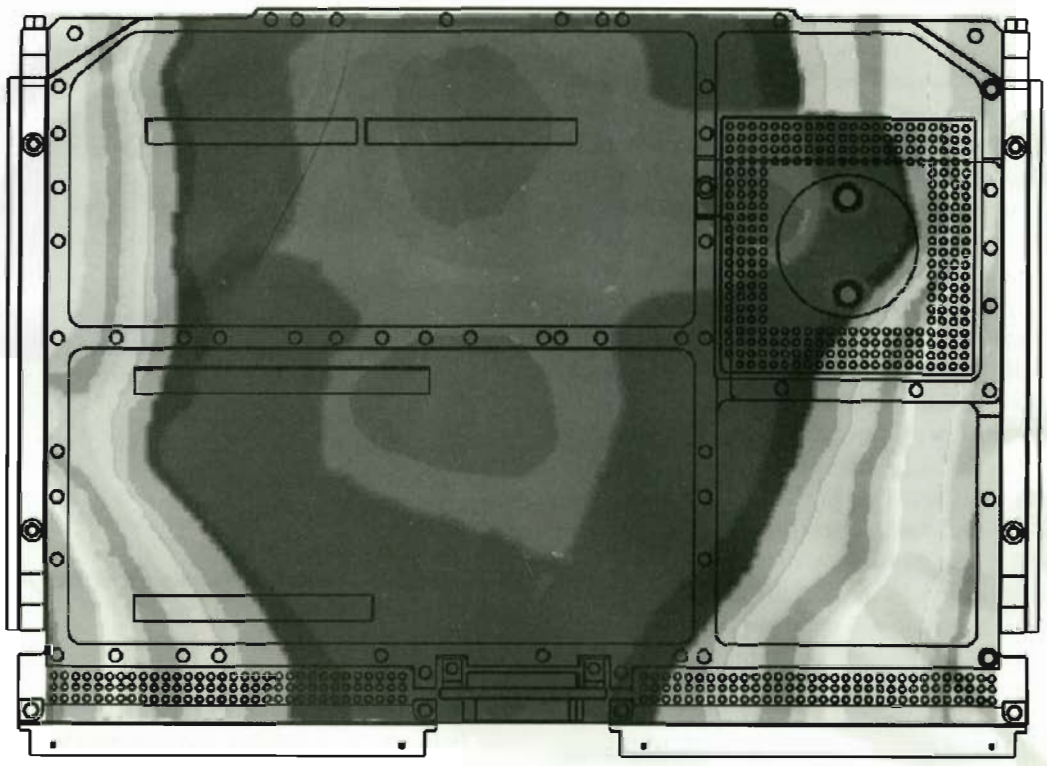


Figure 7 Thermal Map for the Circuit Card

SRAM junction temperature would be 104 degrees Celsius. Although this high junction temperature is still acceptable, it is not desirable because it decreases product reliability. Thus, an appropriate modification in the thermal design will be made to the circuit board stackup before release to production.

Design Trade-offs

This section discusses design trade-offs for the single module computer based on space and thermal differences.

Space Trade-offs

The conduction-cooled module has significantly less surface area for mounting components than its convection-cooled counterpart. This is due to the use of a thermal frame that serves the dual purposes of conducting heat to the heat exchanger and structurally supporting the mezzanine modules to meet shock and vibration specifications. In addition to the component mounting constraints already identified, Digital's mezzanine module provides approximately 17 square inches per side for mounting components whereas the Raytheon conduction-cooled PCI mezzanine module provides approximately 13.8 square inches per side. An additional constraint was that the module layout, including pad dimensions, had to support a range of components from commercial to Class B-1 components. As a result, it was necessary to reduce the area required for components to fit on the board.

The necessary reduction in component area was accomplished by the incorporation of a number of functions into a programmable gate array. The functions include

1. Fault logic
2. Interrupt multiplexer
3. All control/status registers (CSRs)
4. All address decoding
5. Interval timer glue logic

A second and more difficult selection was module I/O functionality. In Raytheon's planning stages, it was determined that each single module

computer needed a SCSI bus port for interfacing with a disk. Ethernet support was important, but this interface seemed to be needed on every computer module only in the development phase of a new project. Since the development of a PCI adapter to verify the performance of the adapter interface was an obvious requirement, an adapter was developed for the single module computer that contained two interfaces: SCSI and Ethernet. An alternate objective of this adapter development was to test the capability of the PCI drive circuitry to support two interfaces on a single PCI adapter. Although exhaustive signal integrity testing has not been accomplished over the temperature range, the Ethernet portion of the adapter was used in initial debug of the module, including download of the system console. It has consistently performed without problem.

A final decision was the establishment of package lead geometries that could be supported by both commercial and military components. In many cases, both commercial and military components are available that meet the design criteria. In some cases, commercial components are supplied from one vendor and military components are procured from a second vendor. Unique cases required special solutions. The cache SRAMs are available in commercial-quality, J-leaded packages, but no military counterpart could be found. To resolve this problem, leadless chip carriers were procured from the military vendor and J-leads were welded on the basic components by a specialty supplier.

Thermal Trade-offs

The extremes of temperature over which an E²COTS module must operate require careful consideration of the effects of thermal cycling on the component solder joint with the circuit board. Leadless devices such as chip carriers, capacitors, and resistors have advantages in the manufacture of circuit boards. However, leadless devices also require special care in the process whereby these components are attached to the circuit card to ensure high solder joint reliability during thermal cycling. For example, Figure 8 shows a crack in the solder joint of a chip capacitor that had undergone thermal cycling to determine equipment lifetime under the anticipated operating environment. Although these failures can be eliminated by special manufacturing processes for soldering leadless components, the use of leaded, active components has been made a requirement. This is consistent with the use of leadless SRAM with

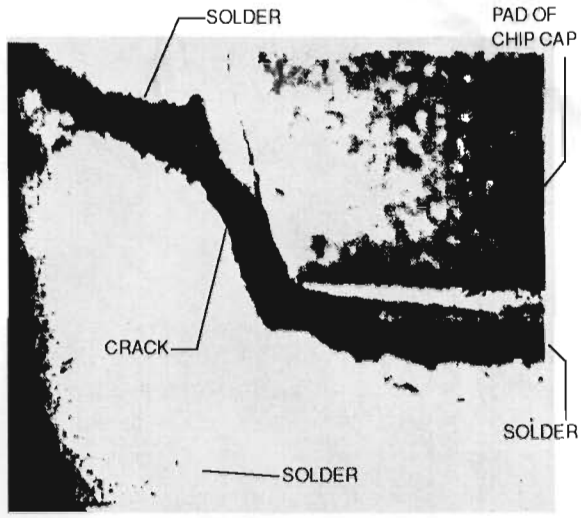


Figure 8 Failure Crack in the Solder Joint of a Chip Capacitor

welded-on J-leads described in the previous section to help ensure reliability and long module life.

A second aspect of the thermal environment range is the use of large PGA devices soldered into a circuit board of copper-polyimide. The Alpha AXP device has a diagonal dimension of ~2.96 inches. The expansion of the ceramic PGA between corner pins over a temperature range of −54 degrees Celsius to +70 degrees Celsius was studied using polyimide boards and ceramic PGAs fully inserted into the circuit board to the package standoffs. The PGAs did not contain semiconductor devices for reasons of cost.

Pin failures occurred at the corner positions of the PGA between 10 and 25 cycles. Additional tests were conducted with the PGA inserted so that the pin tips protruded slightly below the surface of the circuit board before soldering. Thus, the PGA was actually standing off the active component surface of the circuit board. In this configuration, the PGA withstood repeated thermal cycles because the pins had an opportunity to absorb the strain caused by the expansion mismatch. A negative element of this strategy is the inability to adequately inspect for solder bridging, which may occur in the area under the PGA and on the active component surface of the circuit board. It was concluded that repeated cycling of the module over even a moderate part of the temperature range would result in the deformation and eventual failure of the pins in the corners of the properly mounted PGA.

As an alternative to soldering the chip's PGA to the circuit board, a socket comprised of individual sleeves inserted into each hole was used successfully. This type of socketing provides sufficient contact flexibility to eliminate pin cracking of the PGA, yet provides a reliable contact during shock and vibration. With the use of a socket, the question of potential "walking out" of the socket by the PGA was raised. The primary thermal path for the Alpha AXP processor, as shown in Figure 9, provides the additional function of securing the device in the socket, thus eliminating the "walk out" problem.

PCI I/O

As previously noted, the standard PCI mezzanine module design for the single module computer has 19 percent less surface area than that of Digital's mezzanine module. In addition, all I/O from the PCI adapter must be routed through 50 pins on the P2 connector to the backplane to meet the criteria for the standard VME 64 bus. Figure 9 is a component side mechanical drawing of the single module computer.

In many single module computer applications, the interface to analog, video, and fiber optics is required to control or sense synchronous signals and status data such as temperature and air velocity, and to handle video signals (RS-170, RS-343). For this reason the PCI mezzanine module has been designed to include an impedance-controlled I/O interface by way of a third connector mounted between P1 and P2. Such an interface was found to be superior to routing analog and video signals out the P2 connector and made practical the inclusion of fiber-optic interfaces directly to the PCI adapter.

Parts Selection for the E²COTS Computer

The characteristics of Raytheon's E²COTS computer are detailed in the equipment performance

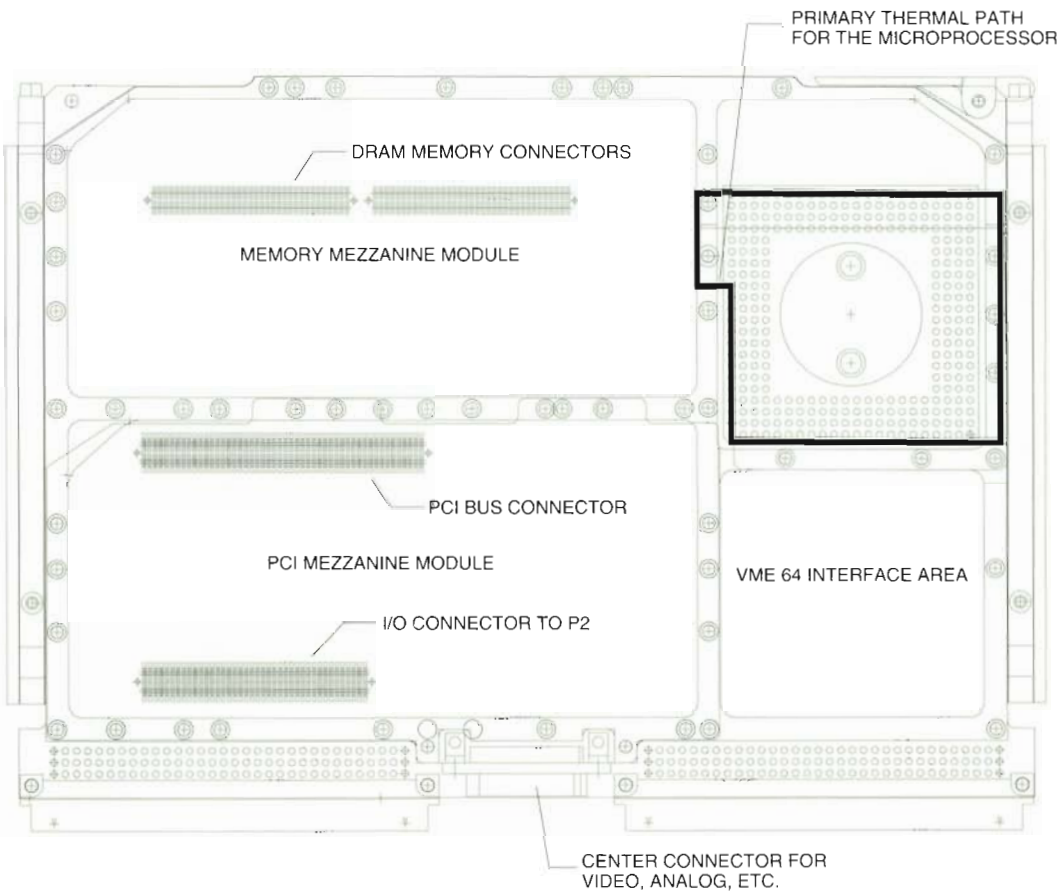


Figure 9 Mechanical Drawing of the Top Surface of the Raytheon Single Module Computer

specification. The mechanical features that make it compatible with military shock and vibration specifications are incorporated at the inception of the design. Once the mechanical features have been designed into the product, the additional cost at production is marginal. The primary factor affecting the cost is the quality of the semiconductor devices used for a given application. In previous DoD procurements, all parts were required to procure to MIL-STD-883 or MIL-STD-38510, the quality standards for all electronic components. Included in the requirements were hermetically sealed packages, semiconductor fabrication process validation, and in many cases extensive parts testing. All of these factors escalated cost substantially.

The E²COTS system allows the temperature and reliability requirements of a given application to determine the quality of semiconductor components utilized. In fact, reliability, much more than temperature range, forces the incorporation of military specification Class B-1 components. Clearly, there are some component types used by the commercial vendors that are inherently not suitable for military application. A prime example is that of oscillators in which the frequency drift over temperature range in commercial components is excessive. In the larger view, specified reliability is the determining factor because the DoD relies on MIL-HDBK-217F for the calculation of component, subsystem, and system reliability. MIL-STD-217F is the hardware benchmark against which all designs are evaluated.[9] Table 2 compares two part types that are typical of the single module computer design. In both cases the reliability improvement achieved in theory by using military-quality parts is a factor of five.

Since many of the passive components (e.g., resistors and capacitors) are normally procured to military specification, the ratio of calculated reliability for a full military-specification-compliant single module computer to a commercial single module computer is approximately 4.99. For a calculated increase in reliability of approximately 5.0, however, the full military-compliant module, subsystem, or system may cost 10 times that of the commercial system. This is an unacceptable cost-performance trade-off in today's defense environment.

Using an E²COTS computer, parts selection is conducted to meet the required mean time between failures (MTBFs) and temperature range. The "mil-spec" semiconductor parts cost is reduced to only those parts necessary for the application. The robust structure of the module is standard, thus providing protection against shock, vibration, and acceleration.

Built-in Test

Digital's built-in test (BIT), boot, and console code are used on an almost "as is" basis. The diagnostics provided on previous processors, such as Digital's VAX 6200 and VAX 6600 systems and the DEC 3000 Model 500 AXP workstation, have proven to be very robust. The exception is the incorporation of a system-level BIT strategy that is built upon the existing BIT design.

The BIT from each system component must be capable of being integrated into the overall system environment so that system-level test results may be easily obtained and the failed component rapidly replaced. To meet this requirement, Raytheon has extended the access to the BIT information at the system level by making test results available on the VME 64 bus. This is accomplished by using the VME interprocessor communication registers (ICRs) as mailboxes that may be accessed by any bus user. Upon initialization, the ICRs are set

Table 2 A Comparison of Reliability for Commercial-quality and Military-quality Parts

Device Type	Reliability Calculated for Class B-1 Parts at 25°C for a Transport Aircraft Environment	Reliability Calculated for Commercial Parts at 25°C for a Transport Aircraft Environment	Ratio of Calculated Military-quality-part Failure Rate to Commercial-quality-part Failure Rate
32K × 8 SRAMs for the cache	0.137 failures per million hours	0.686 failures per million hours	5.007
VIC-64 VME interface	0.613 failures per million hours	3.066 failures per million hours	5.001

to zero. At the end of the BIT, the results are written to the ICRs. Basically, there are three possible results available in the ICRs after BIT:

1. The ICRs contain zero, in which case the module has failed to execute the complete BIT and is therefore FAILED.
2. The ICRs contain the PASSED message.
3. The ICRs contain the FAILED message and identify the test(s) that were failed.

Supervisory processors may poll the single module computers and determine their status.

Planned Upgrades to the Model 910

The first deliveries of the Raytheon Model 910 utilize the 66-megahertz (66-MHz) DECchip 21068 processor. Since capabilities drive requirements, the availability of the DECchip 21066 necessitates the addition of a 160-MHz version of the Model 910. Key issues in the incorporation of the DECchip 21066 processor into the single module computer structure are the thermal dissipation of the design and the limited number of power and ground pins as provided under the VME bus specification.

Power dissipation of 23 W occurs on a system powered by the DECchip 21068 and having 32 megabytes (MB) of memory, a SCSI bus, and Ethernet running the DEC OSF/1 AXP operating system and a graphics demonstration on an X window terminal. When the same unit was exercised with the DECchip 21066, the power dissipation increased to 40 W, underscoring the need for more power/ground pins and additional thermal paths to the sidewall/heat exchanger. The memory capacity will also be expanded in 1994 to a maximum of 256 MB in increments of 128 MB.

Completion of these design upgrades is being conducted during 1994.

Acknowledgment

I would like to acknowledge the efforts of the Raytheon design team of Ted Rogers, project leader; Stewart Berke and Mark Lewin, electrical design; Jim Lanzafane, mechanical and thermal design; Ted King, firmware; and Dave Golden, operating system integration. The Raytheon team would also like to express their appreciation to the many fine Digital people who have supported the project and especially to Don DeRome, our Digital point of contact.

References

1. *MIL-E-5400T, General Specification for Electronic Equipment, Aerospace* (May 1990).
2. *Digital Standard 102-1 Environmental Standard for Computers and Peripherals—Temperature, Humidity, and Altitude Test Requirements* (Maynard, MA: Digital Equipment Corporation, Order No. EL-00102-01, Revision F, September 1992).
3. *Digital Standard 102-2 Environmental Standard for Computers and Peripherals—Mechanical Shock and Vibration Test Requirements* (Maynard, MA: Digital Equipment Corporation, Order No. EL-00102-02, Revision K, November 1992).
4. *Alpha AXPvme 64 Engineering Hardware Specification,* Version 0.9 (Maynard, MA: Digital Equipment Corporation, December 1992).
5. *AVSMC Development Specification,* Spec. No. CRG605203, Revision 1.2 (Lexington, MA: Raytheon Company, Computer Products Directorate, August 1993).
6. *PCI Local Bus Specification, Revision 2.0* (Hillsboro, OR: PCI Special Interest Group, April 1993).
7. *MEDULLA Engineering Hardware Specification* (Maynard, MA: Digital Equipment Corporation, March 1993).
8. *MIL-C-172C, Cases; Bases, Mounting; and Mounts, Vibration (for use with Electronic Equipment on Aircraft),* Amendment 5 (February 1977).
9. *MIL-HDBK-217F, Reliability Prediction of Electronic Equipment* (July 1992).

Ronald D. Levine |

Volume Rendering with the Kubota 3D Imaging and Graphics Accelerator

The Kubota 3D imaging and graphics accelerator, which provides advanced graphics for Digital's DEC 3000 AXP workstations, is the first desktop system to combine three-dimensional imaging and graphics technologies, and thus to fully support volume rendering. The power of the Kubota parallel architecture enables interactive volume rendering. The capability for combining volume rendering with geometry-based rendering distinguishes the Kubota system from more specialized volume rendering systems and enhances its utility in medical, seismic, and computational science applications. To meet the massive storage, processing, and bandwidth requirements associated with volume rendering, the Kubota graphics architecture features a large off-screen frame buffer memory, the parallel processing power of up to 20 pixel engines and 6 geometric transform engines, and wide, high-bandwidth data paths throughout.

The Kubota 3D imaging and graphics accelerator, which provides advanced graphics for Digital's DEC 3000 AXP workstations, enables interactive volume rendering—a capability that is unique among workstation-class systems. This paper begins with a discussion of the relations between imaging, graphics, and volume rendering techniques. Several sections then discuss the nature and sources of volume data sets and the techniques of volume rendering. The paper concludes with a description of how volume rendering is implemented on the Kubota accelerator.

This paper is also intended as a tutorial on volume rendering for readers who may not yet be familiar with its concepts and terminology. Following the body of the paper, the Appendix reviews the basic ideas and terminology of computer graphics and image processing, including digital images and geometry-based models, that lead to the ideas of volume rendering. Readers may wish to turn to the Appendix before proceeding to the next section.

This paper is a modified version of *Volume Rendering with Denali, Version 1.0*, which was written by Ronald D. Levine and published as a white paper by Kubota Pacific Computer Inc., June 1993.

Geometry, Pixels, and Voxels

Historically, computer graphics and image processing have been distinct technologies, practiced by different people for different purposes. Graphics has found application in computer-aided design, engineering analysis, scientific data visualization, commercial film, and video production. Image processing has found application in remote sensing for military, geophysical, and space science applications; medical image analysis; document storage and retrieval systems; and various aspects of digital video.

There is a recent trend for these two approaches to computer imaging to converge, and the Kubota 3D imaging and graphics accelerator is the forerunner of systems that enable the combination of imaging and graphics technologies. Users of either of the technologies increasingly find uses for the other as well. One result of this synergism in the combination of computer graphics and image processing is the advent of volume visualization and volume rendering methods, i.e., the production of images based on voxels.

The idea of the voxel-based approach is a natural generalization of the idea of pixels—digital picture elements; voxels simply add another dimension.

(See Figure 1 and Figure 2.) But voxel-based methods are slow to be adopted because their performance requirements are massive, in terms of processing power, storage, and bandwidth. The performance and capacity requirements of the voxel methods are indicative of the general fact that the size of the problem increases in proportion to the cube of the resolution. A two-dimensional (2-D) image with n pixels on a side has n^2 pixels, whereas a three-dimensional (3-D) volume data set with n voxels on a side has n^3 voxels. If the typical minimum useful linear resolution in an imaging application is 100, then a volume data set will have at least 100 times the data of a digital image.

The Kubota accelerator is an imaging *and* computer graphics system. That is, it contains hardware and firmware support both for producing images from geometry-based models and for accelerating certain fundamental image processing functions. As a graphics system, the Kubota accelerator produces raster images from 3-D geometric models. It offers hardware support for the graphics pipeline, including geometry processing, lighting computation, shading interpolations, depth buffering, rasterization, and high-quality rendering features such as antialiasing, texture mapping, and transparency. As an image processing system, the Kubota accelerator includes hardware support for basic image processing functions, such as pixel block transfers, image zooming and rotating, image compositing, and filtering. Some of these low-level image processing functions are used in the graphics pipeline and for advanced features such as antialiasing and texture mapping.

Volumetric rendering methods share certain features with both computer graphics and imaging. Common to volume rendering and graphics are the mathematical transformations of viewing and projection, as well as the surface shading methods when the volume rendering method is isosurface rendering. Common to volume rendering and image processing are the resampling and filtering operations, which are even more costly than in image processing because of the additional dimension. It is not surprising that a system that implements both imaging and graphics functionality is also amenable to volume rendering.

Like 3-D geometry-based graphics techniques, volume rendering techniques help the user gain understanding of a 3-D world by means of images, that is, projections to the 2-D viewing plane. As in geometry-based graphics, one of the most effective means of helping the viewer understand a 3-D arrangement through 2-D images is to provide interactive control over the viewing parameters, i.e., the 3-D position and orientation of the subject relative to the viewer.

In volume rendering, other visualization parameters beyond the viewing parameters need interactive control. For example, the exploration of volume data is greatly facilitated by interactive control of the position and orientation of section planes, of isosurface level parameters, and of sampling frequencies.

Therefore, volume rendering methods are most useful when they are feasible in interactive time. When the viewer turns a dial, the rendered image should be updated without noticeable delay. The upper limit on the response time implied by the interactive control requirement and the lower limit on problem size implied by the requirements of usable resolution combine to set extreme performance thresholds for practical volume rendering systems. The Kubota 3D imaging and graphics accelerator is able to meet these performance demands because it has a large image memory,

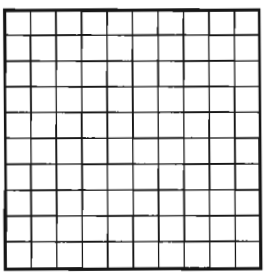


Figure 1 Two-dimensional Pixels

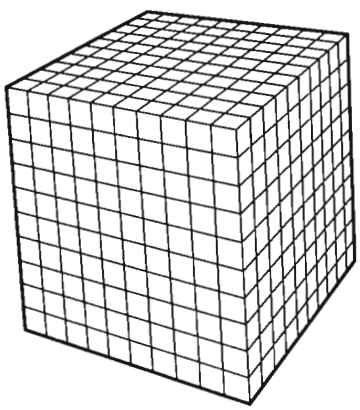


Figure 2 Three-dimensional Voxels

powerful parallel processing elements, and high-speed data paths that connect these components.

Volume Data Sets

After briefly defining a volume data set, this section describes three application areas that are sources of volume data sets.

Volume Data Set Definition

A volume data set, also called volumetric data, is a generalization to 3-D space of the concept of a digital image. A volume data set contains sample values associated with the points of a regular grid in a 3-D space. Each element of a volume data set is called a voxel, analogous to a pixel, which represents an element of a 2-D digital image. Just as we sometimes think of pixels as small rectangular areas of an image, it makes sense to think of voxels as small volume elements of a 3-D space.

Volume Data Sets in Medical Imaging

Because the real world has three spatial dimensions, virtually any application area that produces ordinary image data can also be a source of volume data sets. For example, in medical computed tomography (CT) imaging, when the 3-D structure of an organ is being studied, it is common to take a series of CT exposures through a set of parallel planar slices. These data slices represent a sampling of the underlying tissue (of some predetermined thickness) and can be stacked up to produce a volume data set. Magnetic resonance imaging (MRI), ultrasound imaging, and the positron emission tomography (PET) and single-photon emission computed tomography (SPECT) scan methods of nuclear medicine all can similarly produce volume data sets.

Volume Data Sets in Seismic Exploration

Another important application area that gives rise to massive volume data sets is seismic exploration for petroleum and mineral resources. In seismic exploration, an acoustic energy source (usually a dynamite charge or a mechanical vibrator) radiates elastic waves into the earth from the surface. Receivers on the surface detect acoustic energy reflected from geologic interfaces within the earth. The data gathered at each receiver is a time series, giving the acoustic wave amplitude as a function of time. A single pulse from the source yields a time series of pulses at each receiver. The amplitude of a reflected pulse carries information about the nature of the rock layers meeting at the interface, and its arrival time is directly related to the interface's depth in the earth. The receivers are generally placed on a regular 2-D grid on the surface. There may be more than a thousand receivers involved in a single seismic experiment. There may be hundreds of strata in the vertical direction that contribute pulses to the reflected signals, so the time-sampling resolution must also be measured in the hundreds or more. Thus, the seismic data comprises a large volume data set in which the value of each voxel is an acoustic amplitude, and the voxel array has two horizontal spatial dimensions and one vertical time dimension. The time dimension is directly related to depth in the earth. The relationship is not simple, however, because the acoustic wave velocity varies substantially from stratum to stratum. In fact, determination of the depth-time relationship is one initial objective of the interpretation activity.

The ultimate objective of the seismic interpreter is to locate the regions most likely to contain oil and gas deposits. These deposits occur only in porous permeable rock that is completely surrounded by impermeable rock. The strata are approximately horizontal, but slight tilting from the horizontal (called dip) and strata discontinuities caused by faults are extremely important clues for locating the regions where petroleum deposits may be trapped.

Proper interpretation of seismic data is not transparent; it requires trained specialists. Because the data for each seismic study can be so massive, interpreters have always relied heavily on graphical methods. For a long time, the conventional graphical methods have used long paper strip charts, each recording perhaps hundreds of parallel traces. Interactive workstations that use 2-D graphics and imaging methods have begun to be adopted in the seismic interpretation industry. The application of volume rendering methods is not yet widespread in seismic interpretation, but the advent of the Kubota accelerator, with its volume rendering capability, should stimulate the development of such applications.

Volume Data Sets in Computational Science

Volume data sets also arise naturally as computer-synthesized data in computational science. Three-dimensional fields are quantities that are attached to all the points in defined regions of 3-D space and generally vary from point to point and in time. The

laws of physics that govern the evolution of 3-D physical fields are most commonly expressed in terms of partial differential equations. The simulations of computational physics, such as computational fluid dynamics, stress and thermal studies in engineering, quantum physics, and cosmology, all study 3-D fields numerically by sampling them on finite sets of sample points. In many numerical methods, the sample points are arranged in regular grids or can be mapped to regular grids; thus, the field samples give rise to volume data sets.

Because the objects of study are continuous, there is no limit to the desired resolution or to the desired size of the volume data set. In practice, the grid resolutions are limited by the computing power of the machines used to perform the numerical solution, typically supercomputers for most 3-D problems. With today's supercomputers, a single 3-D field simulation may use millions of sample points.

Oil reservoir simulation—the modeling of the flow and evolution of the contents of subterranean oil deposits as the oil is extracted through wells—is another example of computational simulation. It also makes use of supercomputers, and it presents the same kind of 3-D data visualization problems as computational basic science.

The computational scientist is absolutely dependent on graphical visualization methods to explore, comprehend, and present the results of supercomputer simulations. For three- (and higher) dimensional work, most supercomputer visualization has depended on geometry-based tools, using modeled isosurfaces, stream lines, and vector advection techniques. Now, with hardware readily available that allows interactive volume visualization, the use of volume rendering methods as a means of exploring the large data sets of computational science will grow.

Volume Rendering

Although it is easy to understand the sources and significance of volume data sets, it is not as easy to determine the best way to use a raster imaging system to help the user visualize the data. After all, the real images displayed on the screen and projected onto the retinas of the eyes of the viewer are ordinary 2-D images, made of pixels. These images necessarily involve the loss of some 3-D information. So one objective of volume visualization techniques should be to give the user 2-D images that communicate as much of the 3-D information as possible. As mentioned earlier, interactive control over viewing parameters is an excellent means of conveying the 3-D information through 2-D images. Moreover, some volume rendering applications require interactive control of other parameters, such as section planes, isosurface levels, or sampling frequencies, in order to allow the user to explore the volume data set.

Volume rendering refers to any of several techniques for making 2-D images from the data in volume data sets, more or less directly from the voxel data, respecting the voxels' spatial relationships in all three dimensions. We include in the definition certain methods that make use of rendering surfaces determined directly by the voxel data, such as interpolated isosurfaces. Foremost, volume rendering methods make images from the volume data that depend on the fully 3-D distribution of data and that are not necessarily limited to a fixed set of planar sections. The idea of volume rendering is to make images in which each pixel reflects the values of one or more voxels combined in ways that respect their arrangement in 3-D space, with arbitrary choice of the viewing direction and with control over the sampling function.

We can think of an ordinary X-ray image as the result of an analog volume rendering technique. The X-ray image is a result of the X-ray opacity throughout the exposed volume of the subject. The image density at any point of the image is determined by the subject's X-ray opacity integrated along the X-ray that comes to that image point from the source. Thus, the information about how the opacity is distributed along the ray is lost. This loss of information is in part responsible for the difficulty of interpreting X-ray images. The radiologist can make use of several different 3-D exposures of the subject acquired in different directions to help understand the 3-D situation. A predetermined sampling of views, however, cannot compare with true interactive view adjustment as an aid to 3-D comprehension. Moreover, the radiologist has no ability to vary the sampling function; it will always be the integral of the X-ray opacity along the rays.

Of course, the ordinary X-ray image does not provide us with a volume data set, but we can obtain volume data sets from CT imaging, as previously described. The following volume rendering methods enable the user to explore the X-ray opacity using arbitrary viewing parameters or different sampling functions.

The methods described here, in analogy with the X-ray as an analog volume renderer, all use families of parallel rays cast into the volume, usually one ray for each pixel in the displayed image. In general, the volume data set is resampled along the rays. The methods differ in the choice of the function that determines pixel color according to the sample values along a ray.

Without volume rendering, the radiologist must treat this CT volume data set as a sequence of independent image slices. To get a 3-D picture of the X-ray opacity function, the radiologist must mentally integrate the separate images, which are presented either sequentially or in an array of images on a display surface. The particular viewing direction for an acquired image sequence may not be optimal with respect to the real-world 3-D anatomical structures under study. Volume rendering provides the remedy—an ability to vary the viewpoint arbitrarily.

The Magnitude of the Volume Rendering Problem

Volume data sets tend to be very large, and their sizes increase rapidly as the resolution increases. The number of voxels in a volume data set increases in proportion to the cube of the linear resolution; for example, a $100 \times 100 \times 100$ grid contains 1 million points. A typical medical CT volume data set has $512 \times 512 \times 64$ (i.e., 2^{24}) sample points, or more than 16 million voxels.

The large amount of volume data implies a need for massive processing requirements. For instance, all volume rendering techniques involve resampling the volume data at at least as many points as there are pixels in the rendered image. Some of the current volume rendering methods require multiple samplings of the volume data for each pixel. Minimizing the aliasing effects of resampling requires interpolation of values from voxels near the sample point. Trilinear interpolation, the simplest interpolation scheme that accounts for the variation of the data in all three dimensions, requires accessing eight voxels and performing about 14 additions and 7 multiplications for each point. For sampling on a single plane section of the volume, the number of sample points is proportional to n^2, where n is the typical resolution of the resulting image. But for the volume rendering methods that involve tracing through the volume, such as several of the methods described in this section, the number of sample points is proportional to n^3, where n is the average resolution of the volume data set.

The combined requirements of (1) high resolution to achieve useful results, (2) trilinear interpolation for antialiasing, and (3) interactive response time imply that adequate processing power for volume rendering must be measured in hundreds of millions of arithmetic operations per second. These operations are either floating-point operations or fixed-point operations with subvoxel precision.

Memory access bandwidth and the bandwidth of the other data communication paths in the system are further potential limits to volume rendering. The volume data size, the amount of processing needed for each volume-rendered frame, the interpolation requirement of multiple accesses to each voxel for each frame, and the interactive requirement of multiple frames per second all contribute to massive requirements for data path bandwidth.

Kubota's architectural features address all these requirements. The large off-screen frame buffer memory accommodates the large volume data sets. The highly parallel processing elements, up to 20 pixel engines and 6 geometric transform engines, meet the demands for processing power. The Kubota accelerator has wide data paths and high bandwidth throughout. The pixel engines have short access paths and high aggregate memory bandwidth to the voxel storage and image display memories. With these architectural features, Kubota offers a level of hardware acceleration for volume rendering that is unique in the workstation world.

Combining Volume Rendering and Geometry-based Rendering

Most applications that produce volume data sets from sampling measurements also involve objects that are defined geometrically. Such applications can frequently make good use of a facility for producing images using both kinds of initial data together. The most familiar examples come from the area of medical imaging, but other areas are also potential sources.

In some cases, the goal is to use the volume data to derive a geometric description of a scanned object (such as the surface of an anatomical organ or tumor) from the medical image data that provides a voxel representation. Such an activity benefits from checking the derivation of the geometry by rendering it (using ordinary geometry rendering methods) onto an image that also directly displays

the original volume data by means of volume visualization methods.

In other cases, the 3-D scene contains geometric objects defined independently of the scanned data. In the simplest case, the geometric objects may be reference frames or fiducials, in the form of planes or wire-frame boxes. A more complex example is the design of a prosthesis, such as an artificial hip joint. The prosthesis must be built and machined to an exact fit with the patient's skeletal structure. The bone geometry is determined by CT scans and presented as voxel data. The prosthesis is designed and manufactured using CAD/CAM methods, which are geometry based. The fit of the prosthetic device can be verified visually in a display system that combines the rendered geometry data from the CAD system with the voxel data from the medical scanning system.

Other medical imaging applications that benefit from mixing geometry-based imagery with voxel-based imagery include surgical planning and radiation treatment planning. Information on the distribution of bones, blood vessels, and organs comes to the surgeon in the form of volume data sets from one of the 3-D medical imaging modes, whereas, X-ray beam geometries, surgical instruments, fracture planes, and incision lines are all future objects or events that are usually described geometrically at the planning stage. (See Figure 5 in the following section for an example from radiation treatment planning.)

Among the other application areas that combine volume rendering and geometry-based rendering techniques are seismic data analysis and industrial inspection and testing. In seismic data analysis, the imaging and volume data from sonic experiments coexists with geographic/topographic data or well-log data that can be described geometrically. In industrial inspection and testing, displaying the test image data together with an image rendered from the geometry-based CAD model of a part may aid in improving the quality of the part.

As shown in the following section, the Kubota accelerator subsystem is capable of supporting simultaneous presentation and merging of volume-rendered and geometry-based imagery.

Volume Rendering Techniques

This section describes the four volume rendering methods that have been implemented on the Kubota accelerator hardware: (1) multiplanar reformatting, (2) isosurface rendering, (3) maximum intensity projection, and (4) ray sum. These methods, which constitute an important subset of the volume rendering techniques currently in use, all employ the technique of casting a bundle of parallel rays into the volume and then resampling the volume data along the rays. Figure 3 illustrates the ray casting for a single ray. The figure shows

- A virtual view plane (or virtual screen), which appears ruled into pixels as it will be mapped to the display surface
- The volume data set, which is oriented arbitrarily (with respect to the view plane) and ruled to indicate voxels

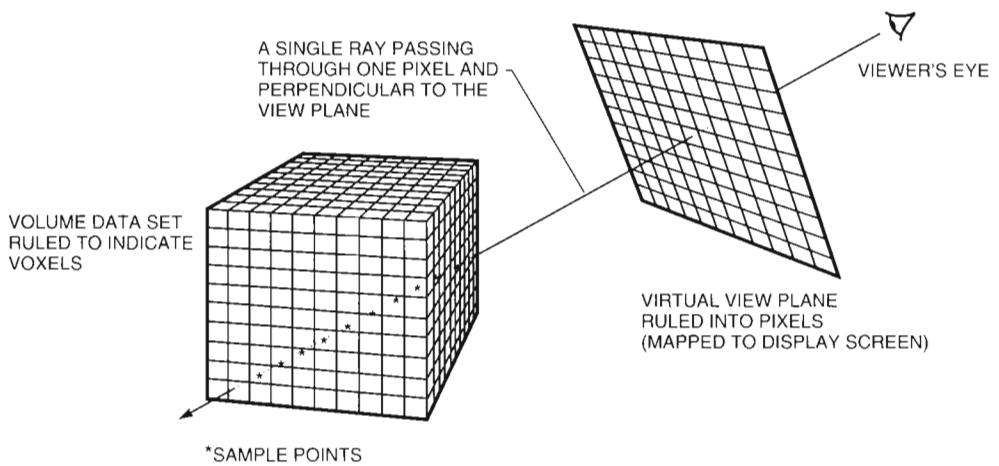


Figure 3 Ray Casting for Volume Rendering

- A single ray, which projects from one of the pixels and intersects the volume box
- Sample volume data points along the ray

For clarity, Figure 3 shows the pixel and voxel resolutions and the sample point density much lower than they are in actual practice, which is typically an order of magnitude higher. Also, although each pixel in the view plane has an associated ray, the figure illustrates only one ray of the parallel bundle.

The sample points generally do not coincide with voxel positions, so the sample values must be estimated from the values of the nearby voxels. The simplest sampling method uses the value of the nearest voxel for each sample point. To obtain better accuracy and to avoid unwanted artifacts, however, the sample value should be obtained by interpolating from nearby voxels. The interpolation must be at least linear (in terms of the algebraic degree) but should also respect all three spatial dimensions. The standard voxel sampling method uses trilinear interpolation, which computes each sample value as a blend of the eight voxels at the corners of a cube that contains the sample point.

The color (or gray level) used to display the pixel depends on the volume data values sampled along the ray. The set of sample points and how the sampled values determine the pixel color define the volume rendering method. Each pixel in the image has an associated ray for which the sampling operation must be performed.

Multiplanar Reformatting

One of the most direct means of giving the user views of volume data is through multiplanar reformatting. This method consists of displaying the volume data on one or more specified plane sections through the volume that do not need to be perpendicular to the viewing direction. As described previously, a bundle of rays, with one ray for each pixel in the image and all rays parallel to the viewing direction, is cast into the volume. Each ray has sample points where it intersects the specified section planes. A sample value determines the pixel color according to a defined color map. When there are multiple section planes, a depth buffer determines which section plane defines the color of each pixel.

Multiplanar reformatting is most effective when the user has interactive control over the section planes. Although some users may want to be able to change both the position and the orientation of the section plane, the interactive variation can be better understood if only one parameter is varied, usually the position of the section plane along a line perpendicular to it.

Multiplanar reformatting can be combined advantageously with isosurface rendering and with geometry-based surface rendering. Figure 4 displays an example of multiplanar reformatting together with isosurface rendering.

Isosurface Rendering

For a scalar field in 3-D space, the set of points on which the field value is a particular constant is called an isosurface or a level surface. A traditional method of visualizing a scalar field in 3-D space is to draw or display one or more isosurfaces using the rendering methods of geometry-based graphics. A volume data set represents a sampling of a scalar field. This set can be used to render the field's isosurfaces directly by a volume rendering method that resamples along rays projected from the virtual view plane.

The isosurface rendering method can be particularly effective when the object volume has sharp transitions where the field value changes rapidly in a small region, such as the interface between soft tissue and bone in a CT data set. Choosing the level value anywhere between the typical small opacity value of the soft tissue and the typical large opacity value of the bone produces an isosurface that is an accurate model of the actual surface of the bone. Choosing the level value between the very low opacity value of the air and the higher opacity value of the skin produces an isosurface that corresponds to the surface of the skin.

Kubota's isosurface rendering technique determines the visible isosurface by a depth buffer method. For each pixel in the image, the depth buffer records the estimated depth in the scene where the ray through the pixel first crosses the isosurface level value. Accurate determination of the threshold depth requires sampling each ray at many more points than with multiplanar reformatting. (The multiplanar reformatting method requires each ray to be sampled only at the relatively few points of intersection with the specified section planes.) Isosurface rendering is a truly 3-D sampling computation.

Once determined, the surface representation in the depth buffer must be shaded for display. That is, colors must be assigned to all the pixels in a way that makes the surface topography evident to the viewer. The Kubota implementation uses a simple lighting model for the surface reflectance, namely,

Figure 4 Isosurface Rendering with Multiplanar Reformatting

the standard model for diffuse reflection known as Lambert's law. The controlling parameter of the lighting computation is the surface normal vector, which specifies the orientation in space of the surface at a given point. The normal vector is simply related to the gradient of the depth function, which can be estimated numerically by differencing the depth values of neighboring pixels in both principal directions.

As a further aid to visualizing the shape of the isosurface in 3-D space, the method can apply a depth-cueing interpolation to the final pixel colors. For each pixel, the color determined by the shading is blended with a fixed depth-cue color (typically the background color), with the proportions of the blend dependent on the depth of the surface point in the scene. This interpolation simulates the general fact that more distant objects appear dimmer and thus can add 3-D intelligibility to the image.

Isosurface rendering can be combined with multiplanar reformatting by using a depth buffer to control the merging of the two images. An effective application of this capability to CT or MR data is to use isosurface rendering to display the outer surface (skin) and a moving section plane to display the interior data. The intersection of the section plane with the skin surface provides a good reference frame for the section data. Figure 4 shows an example of such an image. Note that in this image, the pixel value shown at each position comes from the surface further from the viewer, whereas the usual depth comparison used in geometry-based rendering shows the pixel value from the nearer surface.

Volumetric isosurface rendering and multiplanar reformatting can also be combined with geometry-based rendering using the depth buffer to merge the image data. Figure 5 shows an example relevant to radiation treatment planning. The surface of the head and the plane section are produced by volume rendering methods applied to a CT volume data set. The surface with several lobes is a radiation dosage isosurface described geometrically on the basis of the source geometry.

Maximum Intensity Projection

In maximum intensity projection, the color of each pixel is assigned according to the maximum of the voxel values found along its ray. This type of volume

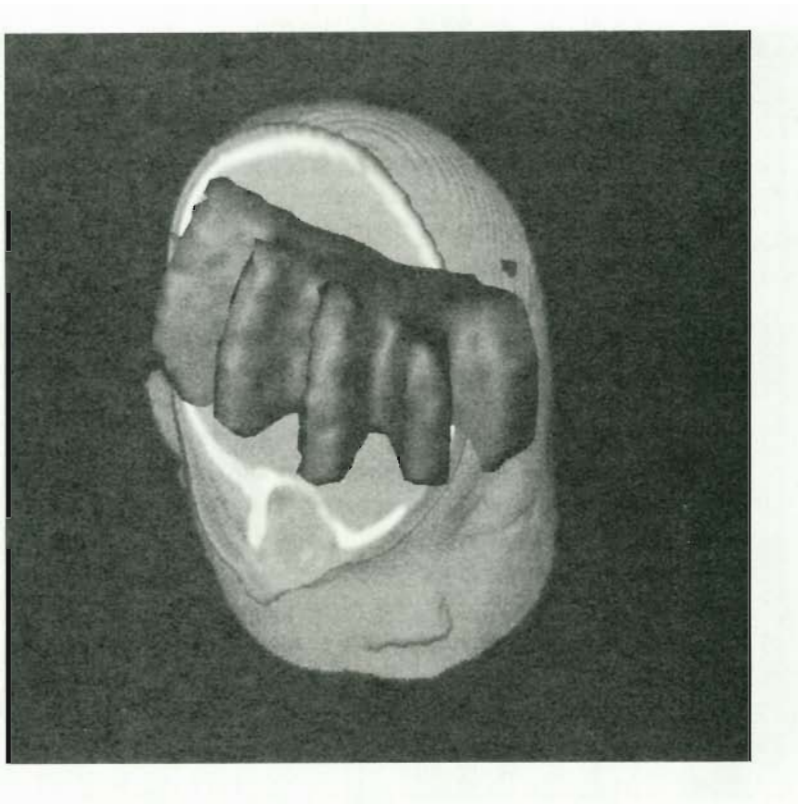


Figure 5 Volumetric Rendering Combined with Geometry-based Rendering

rendering is most useful in generating angiograms (or projection maps of the human vasculature) from MRI data sets. Special MR acquisitions are performed in which the signal from flowing blood (or from the hydrogen nuclei in the flowing blood) is more intense than that from the surrounding tissue.

To locate the maximum value, each ray must be sampled at many points along its entire intersection with the voxel volume. Thus, maximum intensity projection is also a truly 3-D resampling computation that requires Kubota's parallel processing capability. To be at least as accurate as the original volume data set, the sampling frequency should be comparable to the voxel resolution. If the sampling frequency is much smaller, there is a risk of large error in identifying and estimating the maximum value. On the other hand, using a sampling frequency that is much higher than the voxel resolution considerably increases the cost of the method and yields little benefit.

Ray Sum

In ray sum processing, rays are cast into a volume from a user-specified orientation, and intensities are accumulated from interpolated samples along the ray. The projected image produced by ray sum is the digital equivalent of an X-ray image generated from a volume of CT data. The ray sum technique permits an analyst to generate an X-ray–like image along an arbitrary direction for which directly obtained X-rays cannot produce a high-quality image. For instance, X-rays projected along a direction parallel to the spinal column of a human generally produce images of limited diagnostic value because too much matter is traversed between emission and absorption. Generating a ray sum operation on a reduced CT volume that contains only the tissue of interest results in a high-quality, clear image of the tissue structure.

Kubota Volume Rendering Implementation

The Kubota 3D imaging and graphics accelerator offers unique capabilities for hardware support of interactive volume visualization techniques in a general graphics workstation context. It is the first system on a desktop scale to provide useful volume rendering in interactive time. Moreover, the Kubota accelerator is unique among specialized volume rendering systems in its capability for combining

volume rendering and geometry-based rendering to produce images. (See *Denali Technical Overview* for more details on the rendering process and the architecture of the Kubota 3D imaging and graphics accelerator.)[1]

The power of the Kubota accelerator for volume rendering stems from

- A large off-screen frame buffer memory, which is used for volume data storage in volume rendering
- The parallel processing power of the pixel engines (PEs) and the transform engines (TEs)
- High-bandwidth data paths throughout

In particular, the short, wide data paths that connect the PEs to the large local memory on the frame buffer modules (FBMs) are important in enabling the resampling and interpolation of the voxel values, which is the most costly part of volume rendering.

The volume rendering implementation on the Kubota accelerator uses some of the same architectural elements that support the high-quality geometry rendering features. The resampling and interpolation functionality is similar to the support for 3-D texture mapping. Some volume rendering operations also use the geometry processing functionality of the TEs and the scanning and incremental interpolation functionality of the linear evaluator arrays on the TE modules. Several methods use depth merging to combine planar sections or to combine volume rendering with geometry rendering. The implementation of these methods exploits the depth-buffering and depth-compare features of the FBMs and the PEs.

Memory Usage and Volume Tiling

Volume rendering requires fast memory to handle the volume data set and the displayed image. All the methods discussed in this paper also require fast memory for intermediate results, principally the projected subimages computed in the first stage (described later in this section). The methods that use depth merging also require a fast depth buffer. The volume data set itself, the intermediate results, and the depth buffer all use the off-screen frame buffer memory (dynamic random-access memory [DRAM]) in the FBM draw buffers. The displayed image, which is the end result of the volume rendering operations, resides in the on-screen frame buffer memory (video random-access memory [VRAM]) in the FBM display buffer, which is used to refresh the display.

The Kubota accelerator offers two draw buffer memory configurations: 2M bytes (MB) per FBM and 4MB per FBM. There can be 5, 10, or 20 FBMs. Thus, the memory available for volume data, intermediate results, and depth buffer can range from 10MB to 80MB. As a rule of thumb, about half this memory can be used for the volume data set, and half is needed for intermediate results and depth buffer. Therefore, the largest configuration has 40MB of fast memory for volume data—enough to store the volume data sets of a wide range of potential applications.

Of course, the volume data must be distributed among the FBMs. To benefit from the Kubota architectural features, the volume must be partitioned so that most of the data flow in trilinear interpolation is within FBMs rather than between them. Trilinear interpolation is a local 3-D operation, that is, its computation involves combining data from each voxel with its neighbors in all three dimensions. Therefore, the volume data must be partitioned into approximately cubical, contiguous 3-D subvolumes.

The storage and accessing of the subvolumes use the same mechanisms as the 3-D texture-mapping capabilities. In the texture-mapping case, each FBM contains a copy of the same texture, which can have $64 \times 64 \times 64$ four-byte texture elements. For volume rendering, however, each FBM contains a different subvolume of the volume data set being rendered. Moreover, the Kubota volume rendering implementation treats only single-channel volume data, which can have either 1 or 2 bytes per channel. Therefore, each $64 \times 64 \times 64$ block of 4-byte texture elements can store four $64 \times 64 \times 64$ blocks of single-precision volume data or two $64 \times 64 \times 64$ blocks of double-precision volume data. Thus, an FBM with 4MB of DRAM can have eight $64 \times 64 \times 64$ single-precision blocks or four $64 \times 64 \times 64$ double-precision blocks.

When the volume data set is smaller than the maximum that the configuration supports, the subvolume blocks are smaller than $64 \times 64 \times 64$ and remain geometrically congruent and evenly distributed among the FBMs to maintain full parallelism.

To preserve locality at the edges of the subvolumes, the subvolume blocks are not completely disjoint; adjacent blocks overlap by one voxel slice. Because of the overlap and another constraint related to the way FBMs are grouped by scan line, the maximum size of the volume data that can be accommodated is slightly less than 4×64^3 bytes per FBM. A maximal Kubota accelerator configuration,

with 80MB of draw buffers, can accommodate single-precision volume data sets up to 256 × 256 × 505 or 512 × 512 × 127 and double-precision sets up to 256 × 256 × 253 or 512 × 512 × 64. Of course, configurations with fewer FBMs or smaller draw buffers accommodate proportionally smaller maximum volume data set sizes. In a single interactive study session, the volume data set needs to be downloaded to the FBMs and partitioned into subvolumes only once. The set may be rendered many times under the control of an interactive user who is varying viewing direction, sampling frequency, rendering method, and other parameters.

Kubota Volume Rendering Stages

The fundamental operation on which all the Kubota volume rendering operations are based is the resampling and interpolation of the volume data on parallel projected rays, as illustrated in Figure 3. In the Kubota implementation, the PEs work in parallel, each on the sample points within the subvolume stored on its local FBM. Thus, the unit for parallel processing is the subvolume. Several different sample points of a single ray, lying in different subvolumes, may be computed simultaneously.

Each PE produces a projected subimage according to the volume rendering method in use, based on the PE's local subvolume. This subimage is also stored on the local FBM. Data packets from one TE control the processing, but the great volume of data traffic is all within FBMs.

For each computed sample point on a projection ray, the PE updates the corresponding pixel of the subimage in a way that depends on the volume rendering method used. For isosurface rendering, the subimage is a depth buffer, which is updated subject to a depth comparison if the sample value exceeds the specified isosurface threshold value. For maximum intensity projection, the subimage is a voxel-value buffer, which is updated subject to a voxel-value comparison. For multiplanar reformatting, the update also consists of updating a voxel-value buffer, subject to a depth comparison. For ray sum, the subimage is an accumulation of voxel values multiplied by a constant.

The result of the parallel projection stage is a set of subimage tiles in the FBM draw buffers, with each tile representing a part of the projected image of the whole volume data set. Of course, the different image tiles represent overlapping portions of the image in screen space and are not yet stored with correctly interleaved addresses. The next volume rendering stage recombines the subimage tiles to form the whole image and redistributes the pixels correctly to the interleaved addresses. The recombination stage involves reading back the tiled subimage data to the TE modules, scan line by scan line, and then writing the data back to the FBMs. The write-back operation applies value comparison in each rendering mode.

Further processing stages are possible. The projected image resolution that determined two dimensions of the sampling frequency in the projection stage may not correspond to the desired screen image size. Thus, the next stage might be a 2-D zoom operation to produce an image of the desired size. This stage is implemented in TE module code with input coming from the stored image of the recombination stage. The 2-D zoom can use point sampling or bilinear interpolation, depending on the sampling chosen for the projection stage.

The isosurface rendering method requires a shading stage that involves another read-back cycle. This cycle computes the normal vectors by differencing the depth values and applies the depth-gradient shading and the depth cueing interpolations. This shading stage uses the ordinary geometry-based rendering support provided by the TE modules.

Finally, a further image merging stage may be used to combine the rendered isosurface with an image produced by multiplanar reformatting, using depth comparisons. To show a slice through a volume bounded by an isosurface, the depth comparison may show the pixel from the deeper surface rather than from the nearer surface, as is usually the case in geometry-based rendering.

All stages subsequent to the projection stage involve 2-D computations and so represent a small amount of computational work relative to the massive computation of the 3-D projection stage.

Performance and Speed/Resolution Trade-offs

A meaningful low-level volume rendering performance metric is trilinear interpolations per second (TRIPS). Most of the computational work in the expensive projection stage is for performing trilinear interpolations. The measured performance of the Kubota accelerator in this metric on 8-bit voxel data is 600,000 TRIPS per PE. As expected, this metric scales linearly with the number of PEs, so a 20-FBM configuration can achieve 12 million TRIPS. The corresponding measured performance on 16-bit

voxel data is 475,000 TRIPS per PE. A 20-FBM configuration can achieve 9.5 million TRIPS.

Currently, there are no recognized benchmarks to use as high-level volume rendering performance metrics. Practical tests can be expressed in terms of the size of the volume data sets that can be rendered with good interactive frame rates. Of course, the rendering speed depends strongly on the rendering parameters that affect quality, particularly the 3-D sampling frequency.

The ability to interactively change the rendering parameters abets the interactive use. For example, a considerable amount of the interaction typically consists of rotating the volume model about one or more axes (with respect to the view direction) and then stopping in a particular position to carefully examine the image. An application can set the sampling frequency to a coarser value (e.g., 10 frames per second) while rotating to get smooth motion with less accurate images, and then rerender the data with a finer sampling frequency to show a more accurate image when the user stops in the desired position.

Software Interface

The fundamental firmware routines that implement the Kubota volume rendering capability are accessible through an application programming interface. This interface permits users to perform volume rendering in a windows environment like the X Window System. The interface includes routines to manage image memory for volume rendering, to download and manipulate volume data sets, and to produce screen images by the volume rendering methods discussed in this paper—all while efficiently exploiting the parallel processing capabilities of the Kubota accelerator.

Appendix: Conceptual Review

The application of computing systems and computational methods to produce and manipulate images and pictures has historically involved two different kinds of data structures: geometry-based models and digital images. The body of this paper concerns a third kind of data structure, the volume data set, which has more recently become important in imaging applications. This Appendix seeks to clarify the natures of digital images and geometry-based models as a basis for the discussion of their roles in volume rendering. It reviews the principal concepts, data structures, and operations of computer graphics and image processing. The review is intended for the interested reader who may not be well versed in the subject. It is also intended to clarify for all readers the meanings of the terms used in the paper.

Pixels, Digital Images, and Image Processing

A digital image is simply a two-dimensional (2-D) array of data elements that represent color values or gray values taken at a set of sample points laid out on a regular grid over a plane area. The data elements of a digital image are commonly called pixels, a contraction of *picture elements.* A digital image can be obtained by scanning and sampling a real image, such as a photograph, or by capturing and digitizing a real-time 2-D signal, such as the output of a video camera. A digital image can be displayed on a raster output device. The raster device most commonly used in interactive work is a cathode-ray tube (CRT). The CRT is refreshed by repeated scanning in a uniform pattern of parallel scan lines (the raster), modulated by the information in a digital image contained in a frame buffer memory. Such a display will be a more-or-less faithful copy of the original image depending on the values of two parameters: the resolution or sampling frequency and the pixel depth, which is the precision with which the pixel values are quantized in the digital representation. In the context of a raster display, the pixels are regarded as representing small rectangular areas of the image, rather than as mathematical points without extent.

Image processing involves the manipulation of digital images produced from real images, e.g., photographs and other scanned image data. Image processing applications may have several different kinds of objectives. One set of objectives concerns image enhancement, i.e., producing images that are in some sense better or more useful than the images that come from the scanning hardware.

Some image processing applications, which can be characterized as image understanding, have the objective of extracting from the pixel data higher-level information about what makes up the image. The simplest of these applications classify the pixels in an image according to the pixel values. More sophisticated image understanding applications can include detection and classification of the objects, for example, in terms of their geometry. The term computer vision is also used for image understanding applications that strive for automatic extraction of high-level information from digital images.

Geometry-based Models and Computer Graphics

Geometry-based models are data structures that incorporate descriptions of objects and scenes in terms of geometric properties, e.g., shape, size, position, and orientation. The term computer graphics generally refers to the activity of synthesizing pictures from geometry-based models. The process of synthesizing pictures from models is called rendering.

The fundamental elements of the geometry-based models (frequently called primitives) are mathematical abstractions—typically points, lines, curves, polygons, and other surfaces. The graphics application usually defines certain objects made from primitives and assembles the objects into scenes to be rendered. Usually, the geometry-based models used in graphics contain additional data that describes graphical attributes and physical properties beyond the geometry of the displayed objects. Examples of these attributes and properties are surface color, the placement and colors of light sources, and the parameters that characterize how materials interact with light.

Applications use geometry-based models for purposes beyond producing graphics. Models are essential for analytical studies of objects, such as determination of structural or thermal properties, and for supporting automated manufacturing by computer-controlled machine tools.

In earlier eras, instead of raster devices, computer graphics systems used so-called stroke or vector graphics output devices. These devices were directly driven by geometric descriptions of pictures, rather than by digital images. The most familiar vector systems were the pen plotter and the CRT display operated in a calligraphic rather than a raster mode. Such stroke devices were driven by data structures called display lists, which were the forerunner of today's sophisticated three-dimensional (3-D) geometry-based models.

Digital Images and Geometry-based Models

We normally think of a digital image as a data structure that is of a lower level than a geometry-based model because the data contains no explicit information about the geometry, the physical nature, or the organization of the objects that may be pictured. The data tells how the colors or gray values are distributed over the plane of the image but not how the color distribution may have been produced by light reflected from objects. On the other hand, the digital image is a more generally applicable data structure than the geometry-based model and therefore may be used in applications that have no defined geometric objects.

Historically, image processing and geometry-based computer graphics have been distinct activities, performed by different people using different software and specialized hardware for different purposes. Recently, however, beginning with the advent of raster graphics systems, the distinction has become blurred as each discipline adopts techniques of the other.

For example, high-quality computer graphics uses image processing techniques in texture mapping, which combines digital images of textures with geometric surface descriptions to produce more realistic-looking or more interesting images of a surface. A good example of texture mapping is the application of a scanned image of a wood grain to a geometrically described surface to produce a picture of a wooden object. Because of its fine-scale detail, geometric modeling of the wood grain is impractical. More generally, one major problem of raster graphics is aliasing, which is the appearance of unwanted artifacts due to the finite sampling frequency in the raster. Some techniques now used in raster graphics to ameliorate the effects of aliasing artifacts are borrowed from image processing.

Conversely, the image understanding applications of image processing involve the derivation of geometric model information from given images. In other imaging application areas, one has information at the level of a geometric model for the same system that produced the image data, and there is naturally an interest in displaying the geometric modeling information and the imaging information in a single display. Thus, for example, a remote sensing application may want to combine earth images from satellite-borne scanning devices with geographic map drawings, which are based on geometrical descriptions of natural and political boundaries.

Dimensionality and Projection

The digital image is intrinsically 2-D because real images, even before the sampling that produces digital images, are all 2-D. That is, they have only two dimensions of extent, whether they exist on sheets of paper, on photographic film, on workstation screens, or on the retinas of our eyes. (True 3-D images exist in the form of holograms, but

these are not yet generally available as computer output devices, so we do not consider them further in this paper.)

In some application areas, such as integrated-circuit design, many geometry-based models may be strictly 2-D. But since the world is 3-D, many engineering and scientific application areas today use 3-D geometry-based models. In these models, the points, lines, curves, and surfaces are all defined in a 3-D model space. Although lines and curves have one dimension of extent and surfaces have two dimensions of extent, in a 3-D application, these figures all lie in an ambient 3-D space, not all contained in any single plane in the model space.

Hence, all 3-D visualization techniques, whether based on geometric models or based on the volume data sets discussed in this paper, use some kind of projection mapping from the 3-D model space to a 2-D view plane. The simplest kind of viewing projection, the one most frequently used in engineering graphics and in the volume rendering implementations described in this paper, is called orthographic projection. This projection is along a family of parallel lines to a plane that is perpendicular to all of them (see Figure 3). The common direction of the family of parallel projection lines is called the viewing direction.

The fact that the viewed images are 2-D poses a basic problem of 3-D graphics: How do you convey to the viewer a sense of the 3-D world by means of viewing 2-D images? An extremely important technique for solving this problem is to give the viewer interactive control over the viewing projection. The ability to change the viewpoint and viewing direction at will is a great aid to understanding the 3-D situation from the projected 2-D image, whether the image is produced by rendering from 3-D geometric models or by the volume rendering techniques discussed in this paper.

Visualization by Pixels and Voxels

The power of raster systems to display digital images vastly increases when we recognize certain aspects of data visualization. We can make digital images from data that are not intrinsically visual or optical and that do not originate from scanning real visible images or from rendering geometrical surfaces by using illumination and shading models. We can display virtually any 2-D spatial distribution of data by sampling it on a regular 2-D grid and mapping the sampled values to gray-scale values or colors. By viewing the displayed image, a viewer can gain insight into and understanding of the content of the 2-D data distribution. The term pseudocolor is used frequently to mean using colors to give visual representation to other kinds of data that have no intrinsic significance as color. This approach to data visualization provides a powerful tool for assimilating and interpreting 2-D spatially distributed data, in much the same way as geometry-based graphics have for centuries provided a powerful tool, graphing, for visualizing quantitative relationships in all realms of analytical science.

The most familiar examples of image renditions of data that are not intrinsically image data come from medical imaging. In ordinary X-ray imaging, real images are formed by exposing photographic film to X-radiation passing through the subject. However, the newer medical imaging modalities, such as CT scanning, MRI, and ultrasound, and the techniques of nuclear medicine (PET and SPECT) use various kinds of instrumentation to gather nonvisual data distributed over plane regions. These procedures then use computer processing to cast the data into the form of digital images that can be displayed in pseudocolor (or pseudo gray scale) for viewing by the medical practitioner or researcher.

Many other examples of data visualization by pixels abound. For example, in satellite-borne remote sensing of the Earth's surface, scanners gather data in several different spectral bands of electromagnetic radiation, both visible and nonvisible. The user can glean the geophysical information by viewing pseudocolor displays of the scanned information, usually after processing the information to classify surface regions according to criteria that involve combinations of several spectral values. Other examples come from the display of 2-D data distributions measured in the laboratory, as in fluid dynamics, or acquired in the field, as in geology.

Volume data sets and voxels are natural generalizations of digital images and pixels. They represent data sampled on regular grids but in three dimensions instead of two. The idea of volume visualization or volume rendering extends to volume data sets the idea of using images to represent arbitrary 2-D data distributions. Because the final viewed images are necessarily 2-D, however, volume rendering is substantially more complicated than simple pseudocolor representation of 2-D data. Although volume rendering uses ideas similar to those used in 2-D image processing, such as the methods

of resampling and interpolation, it also requires techniques similar to those used in rendering 3-D geometric models, such as geometric transformations and viewing projections. Thus, the Kubota 3D imaging and graphics accelerator, which is designed to provide both image processing and 3-D graphics, is especially well suited for volume rendering applications.

Reference

1. *Denali Technical Overview* (Santa Clara, CA: Kubota Pacific Computer Inc., 1993).

General Reference

A. Kaufman, ed., *Volume Visualization* (Los Alamitos, CA: IEEE Computer Society Press, 1991).

Samyojita A. Nadkarni
Walker Anderson
Lauren M. Carlson
David Kravitz
Mitchell O. Norcross
Thomas M. Wenners

Development of Digital's PCI Chip Sets and Evaluation Kit for the DECchip 21064 Microprocessor

The DECchip 21071 and the DECchip 21072 chip sets were designed to provide simple, competitive devices for building cost-focused or high-performance PCI-based systems using the DECchip 21064 family of Alpha AXP microprocessors. The chip sets include data slices, a bridge between the DECchip 21064 microprocessor and the PCI local bus, and a secondary cache and memory controller. The EB64+ evaluation kit, a companion product, contains an example PC mother board that was built using the DECchip 21064 microprocessor, the DECchip 21072 chip set, and other off-the-shelf PC components. The EB64+ kit provides hooks for system designers to evaluate cost/performance trade-offs. Either chip set, used with the EB64+ evaluation kit, enables system designers to develop Alpha AXP PCs with minimal design and engineering effort.

The DECchip 21071 and the DECchip 21072 chip sets are two configurations of a core logic chip set for the DECchip 21064 family of Alpha AXP microprocessors.[1] The core logic chip set provides a 32-bit PCI local bus interface, cache/memory control functions, and all related data path functionality to the system designer. It requires minimal external logic. The EB64+ kit is an evaluation and development platform for computing systems based on the DECchip 21064 microprocessor and the core logic chip set. The EB64+ kit also served as a debug platform for the chip sets. The DECchip 21071 and the DECchip 21072 chip sets and the EB64+ evaluation kit were developed to proliferate the Alpha AXP architecture in the industry by providing system designers with a means to build a wide range of uniprocessor systems using the DECchip 21064 processor family with minimal design and engineering effort.[2]

The core logic chip set and the EB64+ evaluation kit were developed by two teams that worked closely together. This paper describes the goals of both projects, the major features of the products, and the design decisions of the development teams.

The Core Logic Chip Set

This section discusses the design and development of the two configurations of the core logic chip set. After presenting the project goals and the overview, the section describes partitioning alternatives and the PCI local bus interface. It then details the memory controller and the cache controller and concludes with discussions of design considerations and functional verification.

Project Goals

The primary goal of the project was to develop a core logic chip set that would demonstrate the high performance of the DECchip 21064 microprocessor in desktop and desk-side systems with entry prices less than $4,000. The chip set had to be system independent and had to provide the system designer with the flexibility to build either a cost-focused system or a high-performance system.

Another key goal was ease of system design. The chip set had to include all complex control functions and require minimal discrete logic on the module so that a system could be built using a

personal computer (PC) mother board and off-the-shelf components.

Time-to-market was a major factor during the development of the chip set. The DECchip 21064 microprocessor had been announced nearly five months before we started to develop the core logic chip set. Digital wanted to proliferate the Alpha AXP architecture in the PC market segment; however, the majority of system vendors required some core logic functions in conjunction with the microprocessor to aid them in designing systems quickly and with low engineering effort. Providing these interested system vendors with core logic chip set samples as soon as possible was very important to enable the DECchip 21064 microprocessor to succeed in the industry.

To determine the feature set that would meet the project goals, we polled a number of potential chip set customers in the PC market segment to understand their needs and the relative importance of each feature. We kept this feedback in mind during the course of the design and made appropriate design decisions based on this data. The following subsections describe the final chip set partitioning, the trade-offs we had to make in the design, and the design process.

Chip Set Overview

The chip set consists of three unique designs:

- DECchip 21071-BA data slice
- DECchip 21071-CA cache/memory controller
- DECchip 21071-DA PCI bridge

It can be used in either a four-chip or a six-chip configuration.

The DECchip 21071 chip set consists of four chips: two data slices, one cache/memory controller, and one PCI bridge. This configuration was developed for a cost-focused system; it provides a 128-bit path to secondary cache and a 64-bit path to memory. Cache and memory data have 32-bit parity protection.

The DECchip 21072 chip set consists of six chips: four data slices, one cache/memory controller, and one PCI bridge. Intended for use in a performance-focused system, this configuration provides a 128-bit path to secondary cache and a 128-bit path to memory. The system designer can choose between 32-bit parity or 32-bit error correcting code (ECC) protection on cache and memory data.

Figure 1 is a block diagram of an example system using the core logic chip set. For a list of components used in a typical system built with this chip set, see the EB64+ Kit Overview section.

The processor controls the secondary cache by default. It transfers ownership of the secondary cache to the cache controller when it encounters a read or a write that misses in the secondary cache. The cache controller is responsible for allocating the cache on CPU memory reads and writes, and for extracting victims from the cache. The cache controller is also responsible for probing and invalidating the secondary cache on direct memory access (DMA) transactions initiated by devices on the PCI local bus.[3]

The ownership of the address bus, sysAdr, is shared by the processor and the PCI bridge. The processor is the default owner of sysAdr. When the PCI bridge needs to initiate a DMA transaction, the cache controller performs the arbitration.

Data is transferred between the processor, the secondary cache, the data slices, and the cache/memory controller over the sysData bus, which is 128 bits wide. In the 4-chip configuration, each of the two data slices connects to 64 bits of the sysData bus. In the 6-chip configuration, each of the four data slices connects to only 32 bits of the sysData bus, leaving 32 data bits available for use as ECC check bits for memory and cache data. The cache/memory controller connects to the lower 16 bits of the sysData bus to allow access to its control and status registers (CSRs).

Data transfers between the PCI and the processor, the secondary cache, and memory take place through the PCI bridge and the data slices. The PCI bridge and the data slices communicate through the epiBus. The epiBus contains 32 bits of data (epiData), 4 byte enables, and the data path control signals. We defined the epiBus control signals so that the PCI bridge chip operation is independent of the number of data slices in the system. Furthermore, the epiBus control signal definitions allow the epiData bus width to be expanded to 64 bits without changing the design of the data slice.

The system designer can link the system to an expansion bus, such as the Industry Standard Architecture (ISA) bus or the Extended Industry Standard Architecture (EISA) bus, by using a PCI-to-ISA bridge or a PCI-to-EISA bridge. The Intel 82378IB and 82375EB bridges, for example, are available in the market for the ISA and the EISA buses, respectively.[4]

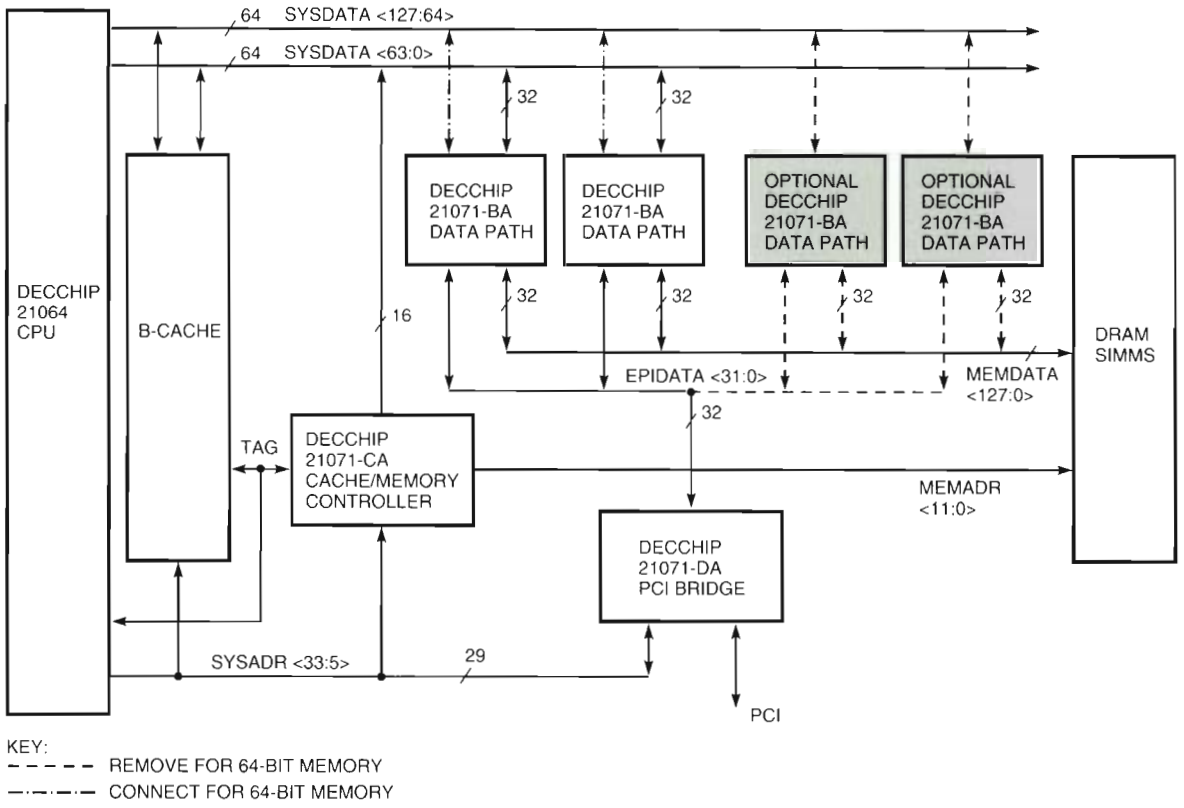


Figure 1 Core Logic Chip Set Configurations in a System Block Diagram

Partitioning Alternatives

As a result of our customer visits, we found that the following features were important for cost-focused systems. The features, which affect the partitioning, are listed in descending order of importance.

- Low cost for the chip set
- Low chip count
- Parity protection on memory
- Inexpensive memory subsystem

The following features were identified as important for performance-oriented, server-type systems (in descending order of importance).

- High memory bandwidth
- Chip set cost
- Low chip count
- ECC-protected memory (This is a requirement in a server system.)

During the feasibility stages, we decided to support a 128-bit secondary cache data path and not offer optional support for a 64-bit cache data path. We felt that a system based on the DECchip 21066 microprocessor, which supports a 64-bit cache interface, would meet the cost and performance needs in this segment of the market.[5] Keeping in mind the importance of time-to-market, we decided that the added flexibility in system design alternatives was not worth the additional design and verification time required to incorporate this feature.

We decided to provide an option between 64-bit-wide memory and 128-bit-wide memory. The wider memory data path provides higher memory bandwidth but at an additional cost. The minimum memory that the system can support with a 128-bit-wide memory data path is double that supported by a 64-bit memory data path. Memory upgrades are also more expensive. For example, with 4-megabyte (MB) single in-line memory modules (SIMMs), the minimum memory supported by a 64-bit memory data path is 8 MB (two SIMMs); with a 128-bit memory data path, it is 16 MB. Memory increments with

a 64-bit data path are 8 MB each, and with a 128-bit data path are 16 MB each. We decided that the performance of the 64-bit memory data path was sufficient for a cost-focused system; however, for memory-intensive applications in the server market, 128-bit-wide memory was necessary.

One alternative we explored could have provided all the features of a cost-focused system in a chip set of three chips, using two identical 208-pin data path slices and one 240-pin controller that provided the PCI bridge, cache controller, and memory controller functions. This configuration, however, would have been restricted to 64-bit memory width and parity protection on memory. Thus it would not have met two of the four desirable features of a high-performance system.

The partitioning that we chose permitted us to satisfy the requirements of both cost-focused and performance-oriented systems. By splitting the design into three unique chips: a data slice, a cache/memory controller, and a PCI bridge, we met the requirements of a cost-focused system with the 4-chip configuration. All 4 chips are 208-pin packages, costing roughly the same as the 3-chip alternative. This partitioning scheme allowed us to support a 128-bit-wide data path to memory and ECC protection with the addition of 2 data slices at relatively low incremental cost. Thus it met the requirements of a performance-focused system. We could not support ECC with the 64-bit-wide memory due to pin-count constraints, but we felt that this trade-off was reasonable given that cost was more important than ECC-protected memory in this market. This partitioning scheme had the added advantage of presenting a single load on the PCI local bus, as opposed to the two loads presented by the 3-chip configuration described above.

Another alternative was to provide a 4-chip configuration with 128-bit-wide, ECC-protected memory. This would have required the data slices to be of higher pin count and therefore higher cost, thus penalizing the cost-focused implementation.

PCI Local Bus Interface

The PCI local bus is a high-performance bus intended for use as an interconnect mechanism between highly integrated peripheral controller components, peripheral add-in boards, and processor/memory subsystems. Interfacing the DECchip 21064 family of CPUs to the PCI local bus opens up the Alpha AXP architecture to what promises to be an industry-standard, plug-and-play interconnect for PCs. The PCI bridge provides a fully compliant host interface to the PCI local bus. This section describes some features of the PCI bridge.

The PCI bridge includes a rich set of DMA transaction buffers that allows it to perform burst transfers of up to 64 bytes in length with no wait states between transfers. We optimized our design for naturally aligned bursts of 32 bytes and 64 bytes because this would eliminate the need for a large address counter and because we discovered through research that most PCI devices in development would not perform DMA bursts longer than 64 bytes.

DMA Write Buffering We chose a DMA write buffer size of four cache blocks. This size would allow for two PCI peripheral devices to alternate bursts of 64 bytes each, thus maximizing use of PCI bandwidth. We organized the DMA write buffer as four cache block entries (four addresses) to simplify the cache/memory interface. In addition, this would allow the data buffers to be used efficiently whenever 32-byte bursts were in use.

DMA Read Buffering We designed the DMA read buffer to be able to store a fetch cache block and a prefetch cache block. As with the DMA write buffer, the DMA read buffer is organized to allow for efficient operation during both 64-byte and 32-byte bursts. Prefetching is performed only if either the initiating PCI command type or a programmable enable bit indicates that the prefetch data will likely be used. This allows the system designer to combine 32-byte and 64-byte devices without sacrificing cache/memory bandwidth. To minimize typical DMA read latency while maintaining a coherent view of memory from the PCI, we designed the capability for DMA read transactions to bypass DMA write transactions, which are queued in the DMA write buffer, as long as the DMA read address does not conflict with any of the valid DMA write addresses. Because most DMA read addresses are not expected to conflict, typical DMA read latency does not suffer as a result of the relatively deep DMA write buffer.

Scatter Gather Address Mapping (S/G Mapping) The PCI bridge provides the ability to map virtual PCI addresses to physical locations in main memory. Because each 8-kilobyte (kB) page can be mapped to an arbitrary physical page in main memory, a virtual address range that spans one or more

contiguous pages can be mapped to pages that are physically scattered in main memory, thus the name S/G mapping. Using this mechanism, software designers can efficiently manage memory while performing multiple-page DMA transfers.

Although our inclusion of S/G mapping offers efficiency benefits to software designers, it also presented us with design challenges in the areas of performance and cost goals. The PCI bridge performs address translation by using incoming PCI physical addresses to index into a lookup table. Each incoming PCI transaction requires the PCI bridge to perform an address translation. A simple implementation might store the entire lookup table in local static random-access memory (RAM). To avoid use of this costly component and corresponding chip set pin allocations, our designers opted to store the lookup table in main memory. To minimize the performance impact of storing the table in main memory, the designers incorporated an on-chip translation lookaside buffer (TLB) for storing the eight most recently used translations. To keep things simple, we implemented a circular TLB replacement algorithm.

PCI Byte Access Support To successfully incorporate Alpha AXP CPUs into PC environments, we required collaboration across the corporation. Digital engineers defined a software/hardware mechanism that allows the 32-bit/64-bit Alpha AXP architecture to coexist with components on the PCI local bus that require arbitrary byte access granularity. This mechanism requires that low-order address bits be used to encode byte lane validity. Implementing this mechanism reduces the density of I/O registers in the address space and conveys byte lane validity information through the address itself.

I/O write performance in this address space suffers because each CPU-initiated I/O transaction can convey only up to 64 bits (a quadword) of data and byte lane validity information. To allow for full utilization of the DECchip 21064 microprocessor's 32-byte internal write buffer during I/O writes to devices that do not require byte granularity, the chip set designers implemented an address range that does not perform byte lane decoding. In this space, up to 32 bytes can be transferred from the CPU and burst onto the PCI in a single transaction. This allows for efficient bandwidth utilization during writes to I/O devices that exhibit memory-like interfaces, such as video adapters with directly accessible frame buffers.

Guaranteed Access Time Systems that support EISA or ISA expansion buses must be able to provide a guaranteed maximum read latency from EISA/ISA peripherals to main memory (2.5 microseconds for EISA, 2.1 microseconds for ISA). This requirement presented a challenge for us during our design. In the worst case, a simple memory read request from an EISA/ISA peripheral can result in significant latency due to our use of deep DMA write buffering and S/G mapping. Although our decision to allow DMA reads to bypass DMA writes provides systems with a typically low latency, this feature does not avoid worst-case high latency. To meet the EISA/ISA worst-case requirements, we included in our design PCI sideband signals and cache/memory arbitration sequences that allow for guaranteed main memory access time. When guaranteed access time is required, the EISA/ISA bridge must signal the PCI bridge by asserting a PCI sideband signal. In response, the PCI bridge will flush its DMA write buffers, hold ownership of the cache/memory, and signal readiness to the EISA/ISA bridge. When the EISA/ISA transaction starts, this sequence guarantees that the path to main memory is clear and will therefore have guaranteed access time.

Memory Controller

The memory controller supports up to eight banks of dynamic random-access memory (DRAM) and one bank of dual-port video random-access memory (VRAM). Each memory bank can be selectively programmed to enable two subbanks, which allows the memory controller to support double-sided SIMMs that have two row address strobe (RAS) lines per bank. The memory controller thus has the flexibility to support system memory sizes of 8 MB to 4 gigabytes (GB) of DRAM and 1 MB to 8 MB of VRAM. System designers can choose to implement memory by banks of individual DRAMs or SIMMs, either on board or across connectors. The memory controller is able to support a wide range of DRAM sizes and speeds across multiple banks in a system, by providing separate programmable bank base address, configuration, and timing registers on a per-bank basis.

We designed the memory controller for system flexibility by supporting fully programmable memory timing with 15-nanosecond (ns) granularity. This programmability supports SIMM speeds ranging from 100 ns down to 50 ns. Each memory bank's timing is programmed through registers that consist of DRAM timing parameters to control counters.

Some examples of programmable timing parameters used to control the memory interface are "row address setup," "read CAS width," and "CAS precharge." As the memory controller sequences through a memory transaction, these programmed counters control the exact timing of RAS, column address strobe (CAS), the DRAM address bits, and write enables. At the same time, the memory controller sends commands from the cache/memory controller chip to the data slice chips to control the clock edge for sending and receiving memory data on DRAM writes and reads, respectively.

One customer is currently using one of the banks in combination with medium-scale integration (MSI) components to interface to a very slow memory bus that supports flash read-only memories (ROMs), nonvolatile RAM, and light-emitting diodes (LEDs). Since the original design was not done with a very slow memory interface in mind, this demonstrates that the chip set provides flexible, programmable timing functionality independent of the system.

The memory controller allows the system designer to build an inexpensive graphics subsystem using a video frame buffer on the memory data bus, and a low-cost video controller on an expansion bus like the ISA bus. The system designer can achieve competitive graphics performance by using the processing power of the CPU for graphics computations and the existing high-bandwidth memory data path for transferring data between the graphics computation engine (the CPU) and the frame buffer. The interface between the memory controller and the video controller is very economical: only two control signals are required to time the transfer of screen data from the random-access memory of the VRAM to the serial-access memory of the VRAM. The video controller is responsible for transferring the data from the serial memory of the VRAM to the screen.

Although we designed the memory controller to be flexible, we also included features that improved performance. Two such features are optimizations to reduce memory read latency and selective support for use of page mode between memory transactions.

To minimize memory read latency, the memory controller prioritizes reads above writes pending in the memory write buffer. For a CPU memory read, the memory controller waits six system cycles after the last read data before servicing a pending write, unless the memory write buffer is full. At least six system cycles occur between the time the memory controller latches the last read data from the DRAMs and the time a subsequent read request could be issued by the DECchip 21064 processor. Because memory write transactions take longer than six cycles to complete, our choice to delay the execution of a pending write allows read latency to be reduced for the following read. Waiting six system cycles after a read is a significant performance improvement for successive reads with cache victims because every read is accompanied by a write.

We also chose to improve performance by selectively determining which memory transactions would benefit most by staying in page mode. The memory controller stays in page mode after a DMA read burst and between successive memory writes. Page mode is not supported between CPU memory read transactions since the RAS precharge time can typically be hidden between successive CPU read requests.

Cache Controller

The secondary cache interface logic is partitioned across the cache/memory controller chip and the data slice chips. The cache/memory controller chip contains the address path and control logic, and the data slice chips provide buffering for four cache lines of data to and from memory. We designed the cache controller to be system independent and flexible so that it could be designed into a wide range of systems.

The chip set supports a direct-mapped, write-back secondary cache with a data width of 128 bits and a cache line fixed at 32 bytes. The chip set allows the system designer to choose a secondary cache size ranging from 128 kB to 16 MB, as determined by software configuration. The speed of the cache RAMs must be fast enough to support the chip set's read access time of one system cycle. Writes to the cache can be programmed to take one or two system cycles. The write enables can be programmed to have a half-cycle or full-cycle pulse width when writing the cache during fill cycles. This feature was added to give the system designer flexibility in meeting SRAM write-enable specifications with various system cycle times.

Another feature added to the cache controller to provide flexibility is the support of an optional allocation policy on CPU writes. The write-back secondary cache is always allocated on CPU memory read misses. The option to allocate the cache on CPU memory write cache misses is programmable and can be disabled by software during system initialization. We chose to provide this option since

disabling cache write allocation can allow higher memory write bandwidth. This feature can be used by system designers to determine whether particular applications have better performance when secondary cache write allocation is disabled.

The cache controller provides arbitration between the CPU and the PCI bridge chip for secondary cache ownership. The arbitration policy is programmable and varies the level of control the PCI bridge has in keeping the ownership of the secondary cache during DMA transactions.

Although we designed the cache controller for system flexibility, we also included features that would give it performance advantages. One such feature is the memory write buffer. The cache controller uses the memory write buffer to store four cache lines of data for cache victims, DMA writes, CPU-noncacheable writes, and CPU-cacheable writes when write allocate mode is disabled. The buffer is organized as first in, first out (FIFO) on cache-line boundaries. Successive writes to the same cache line are not merged into the buffer because the CPU chip write buffer performs this function. The cache controller allows CPU and DMA reads to bypass the write buffer as long as the read address does not conflict with any of the write addresses. The memory write buffer improves performance by allowing timely acknowledgment of write transactions. Read bypassing of the write buffer improves performance by reducing memory read latency.

Global Design Considerations

This section briefly discusses some of the decisions concerning silicon technology, packaging technology, and internal clocking of the chip sets.

Silicon Technology The design team chose to use an externally supplied gate-array process that offered quick time-to-market and low cost. Most chips designed in the Semiconductor Engineering Group are manufactured using Digital's proprietary complementary metal-oxide semiconductor (CMOS) processes, which emphasize high speed and high integration. Our chips' performance and complexity—30-ns cycle time, approximately 35,000 gates per chip—did not require these capabilities. Gate-array technology offered shorter design times and quicker turnaround times than Digital's custom silicon technology.

Packaging Technology When choosing a package, the design team considered issues of package and system cost, design partitioning, and heat produced by power dissipation. Some of these issues are discussed in the Partitioning Alternatives section.

We chose to put all three chips in 208-pin plastic quad flat packages (PQFPs). The 208-pin PQFP is one of the most popular low-cost, medium pin-count, surface-mount packages. One drawback of PQFPs, however, is their low limit on power dissipation. To ensure a junction temperature of 85 degrees Celsius with 100 linear feet per minute of airflow, the power dissipation must be limited to 1.5 watts (W). The power dissipation of the data slice is about 1.7 W, resulting in a junction temperature approaching 100 degrees Celsius. We verified that reliability was not an issue at a junction of 100 degrees Celsius. However, we had to ensure that the chip timing worked at a junction temperature of 100 degrees Celsius, as opposed to the 85 degrees Celsius we would normally use. We could not use this approach on the PCI bridge chip because the additional timing optimization required would have adversely affected the schedule. We had to take special measures in the design to keep the power dissipation within the 1.5-W limit. We implemented conditional clock nets for large blocks of registers that are loaded infrequently, such as the CSRs and the TLB.

Internal Clocking To achieve the shortest possible cross chip set latencies, we implemented a four-phase clock system. A four-phase system allows data to be transferred from one section of the chip set to another in less than a full clock cycle if logic delays are sufficiently small.

In contrast to approaches based on latch designs, which can offer lower gate-count implementations, we chose to use mostly edge-triggered devices. We viewed this as an opportunity to simplify the design analysis and timing verification process by keeping the number of timing reference points to four clock edges.

To further simplify the clocking system, the designers chose to make the PCI clock and the cache/memory clock synchronous to each other. This approach avoids the need for synchronizers (and corresponding synchronizer delays) between clock domains; it also reduces the number of clock speed combinations to be verified. Although the synchronous approach does not allow the system designer to decouple the PCI clock speed from the cache/memory clock speed, we felt that the added complexity and verification effort required to support asynchronous clocks would not be worth the

small degree of flexibility that would be gained from such a design.

Functional Verification

Given the short design schedule and the requirement that first-pass prototypes be highly functional for customers, the team adopted a strategy of pseudorandom testing at the architectural level of the chip set as a whole. We felt that this strategy would test more of the design more quickly and would find more subtle and complex bugs than a testing methodology focused on the gate/register level of each separate chip.

The DECSIM simulation environment included models for the three chips, a DECchip 21064 bus functional model (BFM), a PCI BFM, a cache model, a memory model, and some "demon" models that could be programmed to pseudorandomly generate events such as the assertion of the video port inputs or the injection of errors. We developed SEGUE templates and used them in a variety of exercisers to generate DECSIM scripts pseudorandomly.[6]

To keep the testing environment from being overly complicated, we allowed users to pseudorandomly configure only those aspects of the design that significantly altered the operation of the control logic. Many configurable aspects of the chip set and simulation environment (for example, the PCI S/G map) were not varied in the exercisers and were tested with simple focused tests.

In addition to programming BFMs to read back and check data, we built a variety of checkers into the simulation environment to verify correct operation of RAM control timing, PCI protocol, tristate bus control, PCI transaction generation, data cache invalidate control on the DECchip 21064 CPU, and many other functions. At the end of every exerciser run, the secondary cache and memory were checked for coherence and correct error protection.

The verification efforts of the team resulted in the removal of over 200 functional bugs, ranging from simple bugs to quite complex and subtle bugs, prior to the fabrication of first-pass prototypes. We found no "show stopper" bugs in the core functions required for first-pass prototype chips, and we used simple work-arounds for the few bugs that we did find in the first-pass design.

The EB64+ Evaluation Kit

This section of the paper discusses the development of the EB64+ evaluation kit. After presenting the project's goals and the overview of the kit, it discusses some of the module design issues that were addressed during the design of the EB64+ module. This section concludes with performance results of benchmarks run on the EB64+ system.

Project Goals

The first and most important goal of the EB64+ evaluation kit project was to provide a sample design for customers using the DECchip 21064 microprocessor and the DECchip 21071 and the DECchip 21072 chip sets. Another major goal was to provide an evaluation and development platform that used standard PC components. These two goals would enable a customer to evaluate their design tradeoffs quickly and to complete their system design faster and with a better chance of success.

Secondary goals were to provide a development and debug environment for the core chip set and to provide a high-performance benchmarking system for the microprocessor and core chip set. The EB64+ kit also serves as a platform for hardware and software development for PCI I/O devices.

EB64+ Kit Overview

Figure 2 shows a block diagram of the EB64+ module, a full-size PC (12.0 inch by 13.0 inch) mother board. The major components on the module are given below:

- DECchip 21064 microprocessor (150 megahertz [MHz] to 275 MHz)
- Secondary cache (512 kB, 1 MB, or 2 MB)
- Secondary cache address buffer
- Interrupt/configuration programmable array logic (PAL) device
- Serial ROM interface for the microprocessor
- System clock generator: oscillator, phase-locked loop (PLL), clock buffers
- Core logic chip set
- Two secondary cache control PALs
- PCI bus peripherals: embedded small computer system interface (SCSI) and Ethernet
- PCI bus arbiter
- Intel 82378IB bridge between the PCI and ISA buses
- Three ISA expansion slots
- Eight slots of standard 36-bit memory SIMMs
- Memory control signal buffers

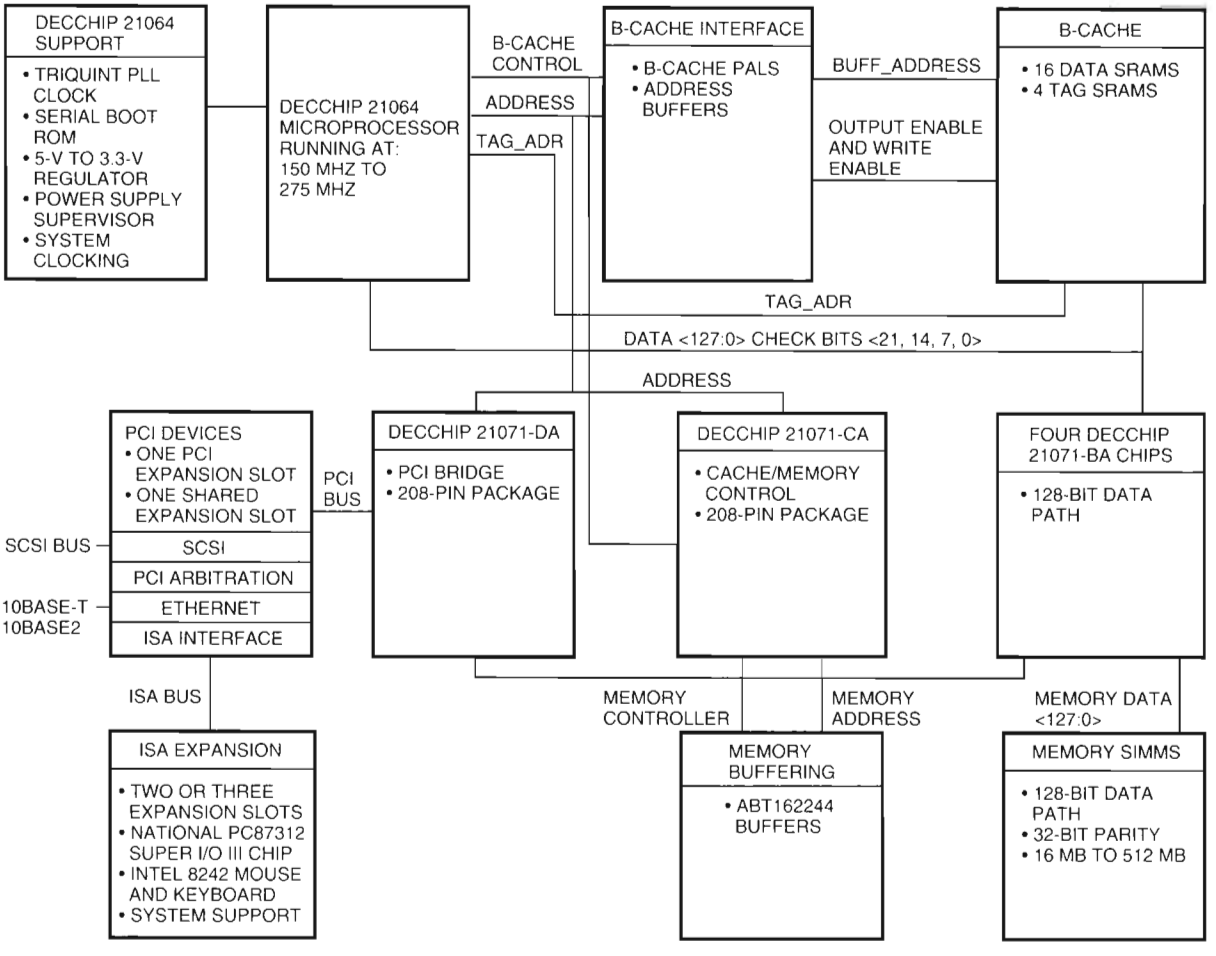


Figure 2 Block Diagram of the EB64+ Module

Secondary Cache Size and Speed

The DECchip 21064 processor has programmable secondary cache read and write access times with a granularity equal to the processor clock cycle time. For instance, if the read access time is 25 ns, the programmed value for a 150-MHz processor (6.6-ns cycle time) would be 4, and the programmed value for a 200-MHz processor (5-ns cycle time) would be 5.

One of the more difficult decisions for any system designer is to determine the optimal cache size and speed in terms of cost and performance. The EB64+ module supports various cache size and speed options in order to allow a system designer to evaluate the difference between a large, slow cache and a small, fast cache. The trade-off here is usually between lower cost for the 512-kB cache and higher performance for the 2-MB cache. The 2-MB cache uses four 128K by 9 SRAMs and twelve 128K by 8 SRAMs for the data store, and the 512-kB cache uses four 32K by 9 SRAMs and twelve 32K by 8 SRAMs.

We decided to share data RAM footprints between the 32K by 8 SRAMs and the 128K by 8 SRAMs, thus allowing the system designer to build two different modules: one with a 512-kB cache and the other with a 2-MB cache. The designer can evaluate the speed-to-size trade-off by using faster SRAMs for the smaller cache and slower SRAMs for the larger cache. The system designer can choose to evaluate the effect of varying the cache size from 512 kB, to 1 MB, to 2 MB, without varying the cache speed, by configuring jumpers to disable portions of the 2-MB cache on an EB64+ module.

System Clocking Design

System clocking for the EB64+ module presented a challenge in two different areas. The first area was the high-frequency input clocks needed by the

DECchip 21064 microprocessor. The input clocks operate at twice the frequency of the DECchip 21064 CPU, requiring a 300- to 550-MHz oscillator for the EB64+ module. Initially, an emitter-coupled logic (ECL) output oscillator was used for this purpose. The main drawback to this solution was the cost, which is in the $40 to $50 range. The other disadvantage was the long lead time and nonrecurring engineering charges associated with unique oscillator frequencies.

By working closely with a vendor of gallium arsenide (GaAs) devices, we were able to provide an alternative in the $8 to $18 range. The device consists of a low-frequency oscillator and a PLL that multiplies the low-frequency oscillator to provide the high-frequency input that the processor requires. For example, a 300-MHz frequency clock is generated using a 30-MHz oscillator connected to a PLL that multiplies this by 10 to provide the 300-MHz input. Since lower frequency oscillators are produced by more vendors, the lead times and nonrecurring engineering charges for unique frequencies are minimal when compared to the ECL output oscillators.

Generating the clocks for the other system components was quite challenging. The core logic chip set, PCI devices, and the cache control PALs together require three types of clock signals: the first clock is in phase with the processor's sysClkOut clock signal; another clock is 90 degrees phase shifted from the first; and a third clock has twice the frequency of and is in phase with the first. The frequency of sysClkOut is an integral divisor (between 2 and 17) of the processor's internal clock frequency. Some divisors may result in a sysClkOut duty cycle that is not 50 percent. A PLL is used to generate both the phase-shifted and the double-frequency clock. It also guarantees a 50 percent duty cycle, which is required for the PCI clock.

Figure 3 illustrates how the EB64+ module generates the three system clocks from the processor's sysClkOut signal. In addition to the PLL, the system clock generator uses low-skew clock buffers to drive the final device loads of the system. One output of the clock buffers is used to provide the feedback to the PLL. This causes the overall delay from sysClkOut to the system clock to be close to zero.

Design Evolution

As noted previously, the EB64+ kit was developed to provide an example design to external customers as well as provide a debug and development platform for the core logic chip set. The focus of the design evolved during the project.

Initially, we included several features on the EB64+ module to support the various modes of the chip set. As the design progressed, an updated version of the EB64+ module was developed. The final version focused more on being a sample design than a debug and development platform for the chip set. Some of the features that fell into this category are listed below.

- Initially, the EB64+ module supported both the 64-bit and 128-bit memory on the same module with configuration jumpers. This design affected

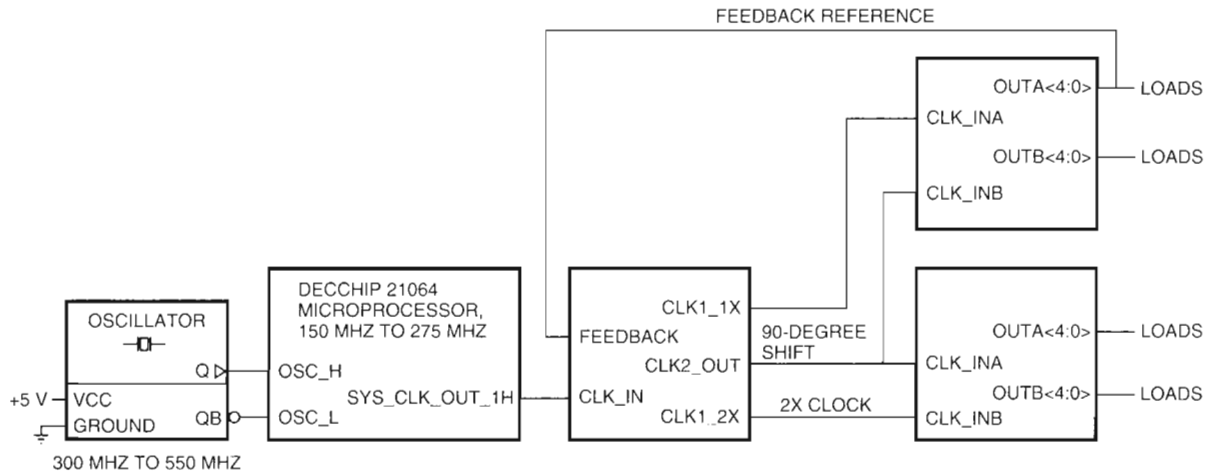


Figure 3 EB64+ System Clock Distribution

performance because 64 bits of the cache data bus were routed to two data slice chips. The final version of the EB64+ module supports only 128-bit memory. This change allowed us to reduce the cache read access time on the DECchip 21064 processor by 3 ns, thus reducing the programmed 2-MB cache read access time for a 200-MHz DECchip 21064 processor from 7 cycles to 6 cycles.

- Certain modes of the chip set were controlled by configuration jumpers initially. These have been redefined to support additional cache sizes and speeds to support a wider range of evaluation and benchmarking.

Performance

Figures 4 and 5 show the results of the *BYTE* magazine portable CPU/floating-point unit (FPU) benchmarks run on an EB64+ system running the Windows NT operating system. The EB64+ system has a 128-bit memory subsystem with 70-ns (RAS access time) DRAMs. The 150-MHz, 166-MHz, and 200-MHz benchmarks were run using a DECchip 21064 microprocessor with a 512-kB cache with a 28-ns read access time. The 275-MHz benchmark was run on a DECchip 21064A microprocessor with a 2-MB cache with a 35-ns read access time. The benchmarks for the DECchip 21066 processor were run on an EB66 system with a 256-kB cache. The figures show the performance relative to other Windows NT systems available in the market today. The benchmark data for the Intel486 DX2-66 and Pentium 60-MHz chips and for the MIPS Computer Systems' R4400SC processors was taken from the *Alpha AXP Personal Computer Performance Brief—Windows NT.*[7]

Table 1 compares the bandwidths on an EB64+ system using the two possible chip set configurations, a 200-MHz processor, and 70-ns DRAMs.

Summary

The DECchip 21071 and the DECchip 21072 chip sets and the EB64+ evaluation kit met their project goals by helping to proliferate the Alpha AXP architecture in the PC market. Several customers, as well as some groups within Digital, use the chip sets in their systems today. Many of these customers and internal groups have used the EB64+ platform as a basis for their designs and as a means of initiating

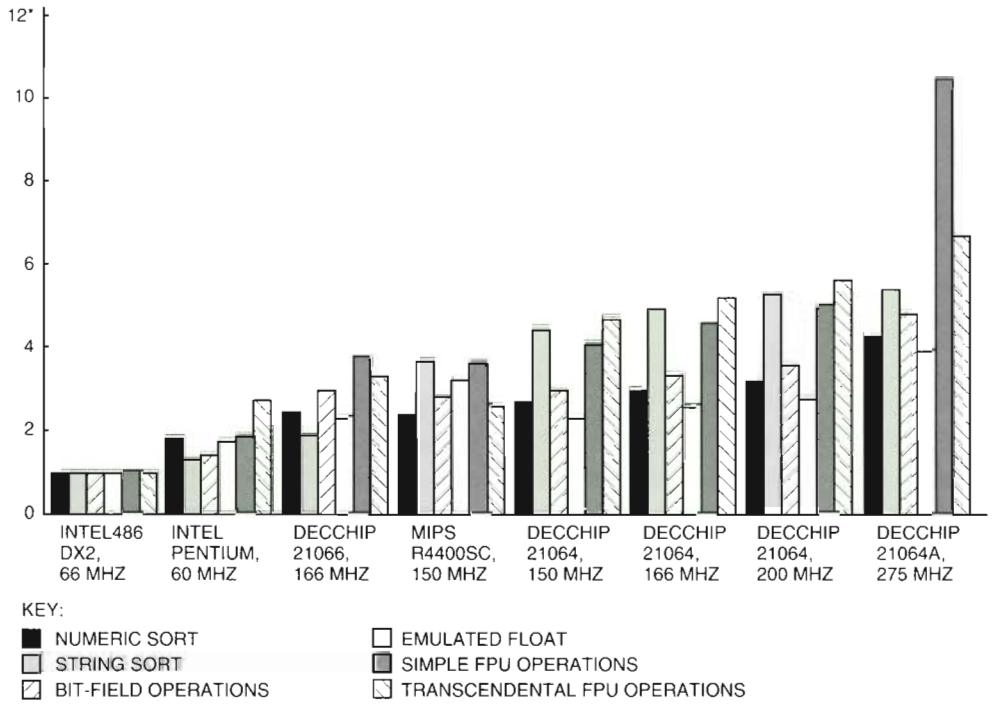
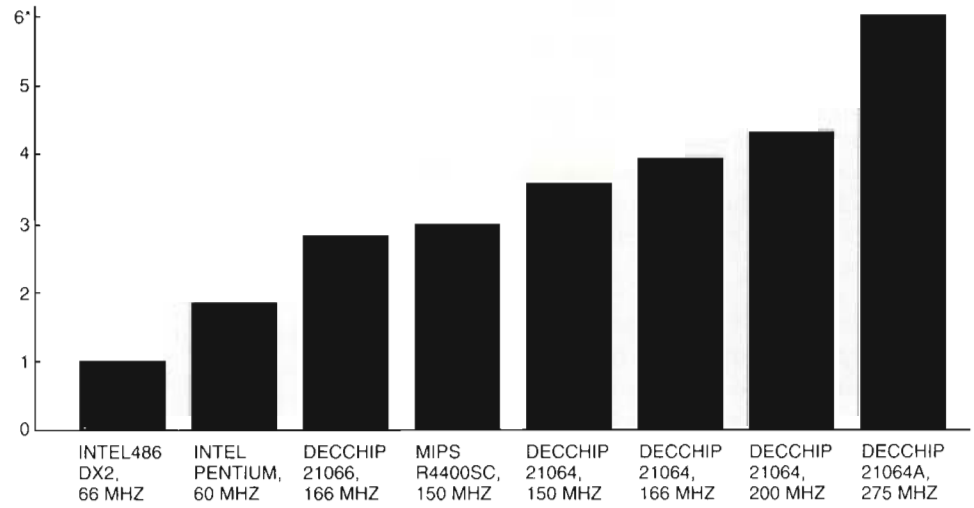


Figure 4 BYTE Portable CPU/FPU Benchmarks



* Measurements are relative to INTEL486 DX2, 66 MHZ.

Figure 5 EB64+ System Performance Benchmarks

Table 1 Comparison between a 64-bit Memory Data Path and a 128-bit Memory Data Path

Transaction Type	64-bit Memory 4-chip Configuration	128-bit Memory 6-chip Configuration
CPU Memory Writes:		
Write with secondary cache allocate	133 MB/s	133 MB/s
Write with no allocate	133 MB/s	267 MB/s
CPU Memory Read:		
Bandwidth	76 MB/s	107 MB/s
I/O Write:		
8 bytes	38 MB/s	38 MB/s
32 bytes (PCI dense memory space)	82 MB/s	82 MB/s
I/O Read:		
8 bytes	22 MB/s	22 MB/s
DMA Write:		
64-byte PCI burst	119 MB/s	119 MB/s
32-byte burst	107 MB/s	107 MB/s
DMA Read:		
Cache miss, 64-byte burst	55 MB/s	65 MB/s
Cache miss, 32-byte burst	41 MB/s	48 MB/s
Cache hit, 64-byte burst	74 MB/s	74 MB/s
Cache hit, 32-byte burst	51 MB/s	51 MB/s

their software development while they were developing their hardware. The EB64+ platform has also been used to develop device drivers for several PCI devices developed by Digital.

Acknowledgments

The authors would like to acknowledge the efforts of the following people. The projects would not have been successful without them. Aaron Bauch, Dick Bissen, Mike Blake, Gregg Bouchard, Mike Callander, Derrick Dacosta, Paul Dziekowicz, Greg Fitzgerald, Avi Godbole, Mike Goulet, Shaheed Haque, Franklin Hooker, Dave Ives, John Jakubowski, Mike Kagen, Elias Kazan, Don MacKinnon, Mike Martino, Mark Matulaitis, Kevin McCarthy, John Murphy, Mike Napier, Victor Peng,

Eric Rasmussen, Tracy Richardson, Mark Riggs, George Rzeznik, Debbie Salois, Raghu Shankar, Will Sherwood, Jai Singh, Wilson Snyder, Hemendra Talesara, Tom Walthall, Juanita Wickham, Mary Woodcome, Marco Zamora, and Beth Zeranski.

References

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

2. R. Sites, ed., *Alpha Architecture Reference Manual* (Burlington, MA: Digital Press, 1992).

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

4. *82420/82430 PCIset ISA and EISA Bridges* (Santa Clara, CA: Intel Corporation, 1993).

5. *DECchip 21066 and DECchip 21068 Hardware Reference Manual* (Maynard, MA: Digital Equipment Corporation, Order No. EC-N2681-72, 1994).

6. W. Anderson, "Logical Verification of the NVAX CPU Chip Design," *Digital Technical Journal*, vol. 4, no. 3 (Summer 1992): 38–46.

7. *Alpha AXP Personal Computer Performance Brief—Windows NT*, 2d ed. (Maynard, MA: Digital Equipment Corporation, Order No. EC-N2685-10, January 1994).

David C. Cressman |

Analysis of Data Compression in the DLT2000 Tape Drive

The DLT2000 magnetic tape drive is a state-of-the-art storage product with a 1.25M-byte-per-second data throughput rate and a 10G-byte capacity, without data compression. To increase data capacity and throughput rates, the DLT2000 implements a variant of the Lempel-Ziv (LZ) data compression algorithm. An LZ method was chosen over other methods, specifically over the Improved Data Recording Capability (IDRC) algorithm, after performance studies showed that the LZ implementation has superior data throughput rates for typical data, as well as superior capacity. This paper outlines the two designs, presents the methodology and the results of the performance testing, and analyzes why the LZ implementation is faster, when the IDRC hardware implementation had twice the bandwidth and was expected to have faster throughput rates.

Overview

Data compression, a method of reducing data size by coding to take advantage of data redundancy, is now featured in most tape drive products. Two compression techniques in widespread use are (1) an arithmetic coding algorithm called Improved Data Recording Capability (IDRC) and (2) variants of the general Lempel-Ziv (LZ) compression algorithm. Current tape products that implement these algorithms are IBM's fast (a maximum throughput rate of approximately 3M bytes per second [M bytes/s]) and relatively expensive (originally about $60K) family of half-inch, 36-track tape products, which have employed the IDRC algorithm for about five years. More recently, the 8-millimeter (mm) helical scan tape products began incorporating IDRC data compression. Also, some 4-mm helical scan digital audiotape (DAT) products now use a variant of the LZ algorithm, as do some quarter-inch cartridge (QIC) tape products.

In developing a complex product like an industry-leading tape drive, it is difficult to determine at the beginning of the project the design that will have the best performance characteristics and meet time/cost goals. When Digital included data compression in the plans for its DLT2000 tape product, the choice was not clear regarding which compression technology would best enhance the tape drive's data transfer rate and capacity. Keeping within cost constraints and incurring an acceptable level of risk to the development schedule were important factors as well. The options were greatly limited, however, because the schedule was too short for the engineering team to implement a compression method on a silicon chip designed specifically for the DLT2000 tape drive; therefore, the team needed to find a compression chip that was available already or would be soon.

Another important consideration was that the compression method used on the DLT2000 tape drive would likely be used on future digital linear tape (DLT) products. For media interchangeability, such products would have to be able to write and read media compatible with the DLT2000 tape drive. New products that used different compression methods would require extra hardware to handle both types of data compression. Since extra hardware adds significant cost and complexity to products, the use of different compression methods is undesirable. Also, to meet future data throughput needs, the compression method used on the DLT2000 tape drive had to support the significantly higher data transfer speeds planned. If the compression chip used initially was too slow for future products, it had to be at least possible to develop an implementation of the same compression algorithm that would be fast enough for future DLT products.

To gain more expertise in applying data compression technology to tape drives, the tape development group investigated several designs using

various data compression chips. Eventually, we created about 20 DLT2000 engineering prototype units, each of which used one of the two most common data compression methods: IDRC and an LZ variant. The specific Lempel-Ziv variant used was designated Digital Lempel-Ziv 1 (DLZ1).[1,2] We tested the performance of the prototype units and studied the results to check for consistency with our expectations. Such analysis was important since tape drive performance with data compression was a new area for the engineering team, and the interplay of higher tape transfer rates, new gate arrays, compression chip, memory buffers, new firmware, and host tape applications is complex.

Figure 1 shows the basic design of the data path on the DLT2000 tape drive's electronics module. (Microprocessors, most gate arrays, firmware read-only memories [ROMs], and other electronic components are not shown.) Note that the data cache size is effectively increased because it contains compressed data. The data processing throughput of the compression chip, however, can potentially be a bottleneck between the cache and the small computer systems interface (SCSI) bus. The IDRC compression chip can process data at throughput rates of up to 5M bytes/s, whereas the DLZ1 chip can process data at rates of up to about 2.5M bytes/s when compressing data and up to about 3M bytes/s when decompressing data. In each design, the memory and data paths outside the compression chip were designed to be adequate for the compression chip used.

One major goal of this study was to quantify the performance of each implementation to determine if the lower throughput of the DLZ1 chip was a practical disadvantage in the DLT2000 product. The IDRC version of the DLT2000 product, with its maximum throughput rate of 5M bytes/s, would seem to have a clear throughput advantage, but the typical compression ratio and the data rate to the tape media are significant factors in the overall throughput of the tape drive.

The development group expected the IDRC and DLZ1 chips to have approximately the same compression ratio (i.e., the result of dividing the number of units of data input by the number of units of data output). The DLZ1 ratio would possibly be slightly higher. The group based their expectation on comparisons of results from several studies.[2,3,4] These studies reported compression ratios for various types of data on implementations that used either the IDRC algorithm or an LZ algorithm but not both.

Compressing data within the tape drive has a multiplying effect on the drive's throughput rate, as seen by a host computer. If the uncompressed data throughput rate to the tape media is 1.25M bytes/s and the data compression ratio is 2.0:1 (or 2.0), the expected average data transfer rate is $1.25 \times 2.0 = 2.5$M bytes/s. Since the development group thought that the typical compression ratio of each implementation was 2.0:1, and because the DLZ1 chip would tend to become a bottleneck as data rates approached the chip's maximum throughput rate, the group expected the IDRC prototype to be at least as fast as the DLZ1 prototype for a given data set.

Testing showed, however, that the DLZ1 DLT2000 prototype consistently, and significantly, surpassed the IDRC prototype in both metrics! To ensure the correctness of the IDRC implementation used on the prototype DLT2000 and thus confirm the unexpected result, the group verified the IDRC compression efficiency results by testing two other tape products that use the IDRC algorithm. Given identical data sets, the benchmark test results were consistent with those of the IDRC DLT2000 prototype.

The marked difference between the DLZ1 and IDRC prototypes can be mainly attributed to the differences in the compression efficiencies of the two algorithms. Relatively low compression ratios on the IDRC unit limit its throughput capabilities. The

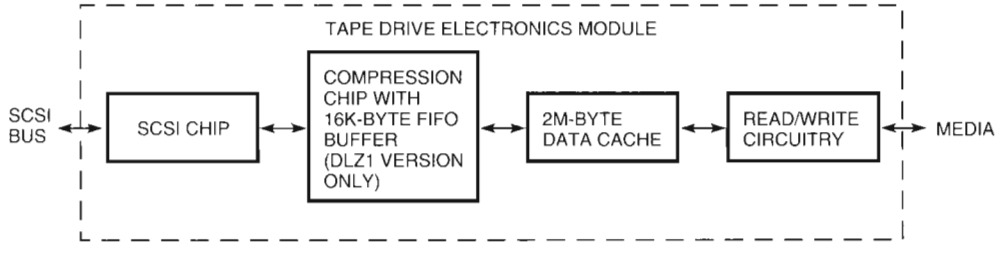


Figure 1 Tape Drive Data Path

author believes that the discrepancy between the results of the DLT2000 prototype testing and the results of the earlier studies can be explained by two factors: variations in the data sets used and differences in media format.

First, the compression efficiency for different samples of data, even if of the same type, e.g., PostScript data, can vary widely. The data sets tested on the DLT2000 prototypes were not identical to those tested in the earlier studies.

Second, some tape drive implementations combine IDRC data compression with a feature IBM calls autoblocking (also known as superblocking). This coupling occurs when the tape drive has a media format that contains interrecord gaps (IRGs) whose number is inversely proportional to the tape block (record) size used (sometimes linear). Autoblocking minimizes the number of IRGs by automatically using a large, fixed on-tape block size (e.g., 64K bytes). The autoblocking feature packs multiple compressed blocks from the host into the larger blocks on the media.[4] Reducing the number of IRGs on such tape formats is important because IRGs are wasted space. If block sizes are small, the number of IRGs will be large and the tape capacity significantly reduced. Tape products that combine autoblocking with IDRC compression derive an increased capacity from both techniques.

These two factors, however, were not relevant to the test results of our study, i.e., the favorable DLZ1 findings. We performed the DLT2000 prototype testing with tape drives that were virtually identical except for the compression technology used. Also, the data samples, tools, and test environments were the same.

From the test results and analysis we concluded that, when compared with the IDRC implementation, the DLZ1 implementation combines consistently superior cartridge capacity (25G bytes at a compression ratio of 2.5:1) and superior data throughput for most types of real data. The testing did not reveal any real data types that compressed better with the IDRC technique than with the DLZ1 technique. In addition, the DLZ1 technique is supported by the strong prospect of future DLZ1 compression chips that will greatly increase the maximum data throughput rates. This addresses the concern that the DLZ1 technique should support a growth path in data throughput rate for future members of the DLT product family.

The remainder of this paper outlines the operation of the IDRC and DLZ1 compression techniques, discusses what testing was done and how, presents the test data, and gives an analysis of the results.

Description of the IDRC and DLZ1 Compression Algorithms

This section provides some historical/industrial background on the IDRC and DLZ1 algorithms and some cursory information on how they work. An in-depth technical presentation of these (or other) compression techniques is beyond the scope of this paper. For more details on their operation and mathematics, please refer to the references.

The IDRC Compression Algorithm

IBM developed the IDRC algorithm and employs this technique on some members of the Model 3480 and Model 3490 tape subsystems. EXABYTE Corporation is currently licensing the IDRC algorithm from IBM.[4]

The IDRC algorithm is a lossless, adaptive arithmetic compression technique. Arithmetic compression encodes data by creating an output string that represents a sequence of fractional numbers between 0 and 1. Each fraction is the result of the product of the probabilities of the preceding input symbols.[4,5,6,7]

The IDRC technique has two modes: byte oriented and binary (bit) oriented. On input, bytes are compared with the last byte processed. If three or more consecutive bytes are found to be equal, processing occurs on a byte-by-byte basis. Otherwise, the data is compressed bit by bit.[6]

Parallel recording implementations for which the number of IRGs is a capacity issue (for example, the IBM Model 3490 product) usually combine IDRC compression with autoblocking. Since autoblocking reduces the number of IRGs (assuming that a smaller block size is commonly used), the effective increase in tape capacity due to autoblocking surpasses the increase that compression alone would yield.

In some tape implementations, though, data is packed into fixed-size blocks on the media whether or not compression is used. If done efficiently, this packing makes tape capacity on such products independent of block size.

The DLZ1 Compression Algorithm

A number of variations of the Lempel-Ziv algorithm (also referred to as the Ziv-Lempel algorithm) have been implemented and are in wide use in the industry today. Some examples are the common PC compression software tools PKARC, PKZIP, and ZOO; the compression method built into the

MS-DOS Version 6.0 system; and Hewlett-Packard's HP 7980XC tape drive. IBM recently announced that it has developed a high-speed (40M bytes/s) compression chip that uses the LZ algorithm. In addition, STAC Electronics' data compression products and the QIC-122 data compression standard use derivatives of the LZ algorithm.[4,5]

Lempel-Ziv methods generally replace redundant strings in the input data with shorter symbols. The methods are lossless and adapt to the input data. Implementations typically simplify the general algorithm in one or more ways for practical reasons, such as speed and memory requirements for string storage.[1,3,4,5,8]

The LZ variant used in the DLZ1 implementation maps variable-length strings in the input to variable-length output symbols. During compression, the algorithm builds a dictionary of strings, which is accessed by means of a hash table. Compression occurs when input data matches a string in the table and is replaced with the corresponding dictionary symbol. The dictionary itself is not output to the tape media but is rebuilt during decompression.[1]

When the dictionary fills up with strings, the algorithm cannot adapt to new patterns in the data. For this reason, the dictionary needs to be reset periodically. The DLT2000 DLZ1 algorithm resets the dictionary on each logical block boundary. Thus, the compression efficiency can vary according to the block size, as well as with the actual data. With small blocks, the dictionary is typically still adapting to the input data when the block ends and the dictionary is reset. This tends to keep the compression algorithm from reaching full efficiency. For example, with an LZ variant similar to the DLZ1, the LZW algorithm presented in Welch's "A Technique for High-Performance Data Compression," compression efficiency increases rapidly as the block size used goes from 1 byte to about 8K bytes.[3] The efficiency peaks at about 12K bytes, and larger block sizes show good but gradually decreasing compression efficiencies. The initial input block range that exhibits rapid improvement in compression efficiency (1 byte to 8K bytes, in this case) is referred to as the "adaptation zone."

Test Procedures

The development group carried out three main sets of tests.

1. Tests that measured the compression efficiency on an OpenVMS system and on an ULTRIX system, which is based on the UNIX system
2. Tests that measured the compression efficiency and the data throughput in a high-throughput test system environment
3. Benchmark tests that measured the IDRC compression ratios on two other tape products

The DLT2000 firmware measured the compression ratios precisely by comparing the block size (in bytes) before and after compression, during write command processing. In the benchmark tests, compression ratios were calculated from total tape capacities with and without compression enabled. We repeated the DLT2000 tests with minor variations in test parameters; the results suggested an uncertainty of approximately ± 1 percent in the measurements.

Test configurations were identical in system type, test software, and operating system versions. We often used the same test bed and varied only the tape unit under test, i.e., the DLZ1 or the IDRC. The hardware and firmware on the different DLT2000 prototypes were identical to ensure that factors such as diagnostic code overhead and clock speed did not skew test results between the DLZ1 and the IDRC units, or between test runs. We also varied some parameters and repeated tests to ensure that the measured performance characteristics were consistent with and reflective of the final product.

Operating System–based Tests

Since the system configurations used could not supply data fast enough for conclusions to be made regarding the DLT2000 tape drive's maximum throughput rates, compression efficiency was the focus of the operating system testing. Test parameters were still chosen to minimize throughput bottlenecks in the host system. For each test, the data was set up on a single disk on each of two systems—an OpenVMS system and a UNIX system.

OpenVMS Tests The OpenVMS system used in the tests was a clustered MicroVAX 3400 machine with a KZQSA adapter for the SCSI bus. The MicroVAX 3400 system was running the OpenVMS Version 5.5-2 operating system and used the standard backup utility (BACKUP) to write data to the DLT2000 tape drive. Although compression efficiency was the focus of the operating system testing, we selected the following BACKUP options to maximize system throughput as much as possible:

- /NOCRC. This option disables a cyclic redundancy check (CRC) calculated and stored in the

tape block by BACKUP for extra data integrity protection. Since the CRC calculations are CPU intensive, they were disabled to minimize system bottlenecks.

- /BLOCK_SIZE=65024. A block size of 65,024 minimizes host and SCSI bus overhead to a reasonable degree.
- /GROUP_SIZE=0. This option disables the creation of (and the writing to tape of) an exclusive OR (XOR) block calculated by BACKUP. By default, BACKUP would create one XOR block for every 10 data blocks. We disabled XOR blocks because their presence would probably decrease the compression ratio and system throughput.

We tested the following types of data on the OpenVMS system.

- Bin—the BACKUP of a set of binary files, mainly executable files
- Sys—the image BACKUP of the system disk
- C—the BACKUP of the DLT2000 product's firmware source library, primarily C code and include files

UNIX Tests The UNIX configuration used for testing was a DECsystem 5500 system running the ULTRIX Version 4.2c operating system. The SCSI common access model (CAM) software driver was used, running on this machine's native SCSI port. The standard ULTRIX tar and dd utilities were used to copy the following data to the tape:

- Text—ASCII text files of product documentation manuals
- PS—PostScript versions of the manuals
- tar—tar backup of the system disk
- HarGra—the chart and art files shipped with the standard Harvard Graphics software package
- ValLog—the files containing the gate array design database, which was built using Valid Logic tools

Throughput Tests

The throughput tests were performed on PC-based Adaptec SDS-3 SCSI development/test systems. The development team chose this test environment to do repeatable, high-performance testing because it is relatively unconstrained by disk, file system, CPU, or application software bottlenecks for the performance range of the DLT2000 tape drive.

We tested the following data types on the SDS-3 system:

- Binary—an OpenVMS VAX object file
- Source—C source code
- VAXcam—a VAXcamera image file in PostScript format
- HarGra—a collection of chart and art files shipped with the standard Harvard Graphics software package
- Paint—a complicated Paintbrush file, in bitmap format
- Ones—an all ones (hex FF) pattern
- Repeat—a string of 24 unique characters, repeated as needed

SCSI bus protocol overhead can be somewhat high on an SDS-3 system, and compression ratio and throughput rate can vary depending on the tape block size. Consequently, all measurements were taken using 64K-byte tape blocks. This block size minimizes per-command overhead on the SCSI bus, as well as in the host. With high enough compression ratios, however, this overhead was still a limiting factor for 64K-byte blocks on the IDRC testing, as will be shown later in the SDS-3 Test Results section.

Another factor in SCSI bus performance is whether synchronous or asynchronous data transfer mode is used. Asynchronous transfer mode requires a full handshake to transfer each data byte, which can seriously decrease the bandwidth of the SCSI bus in many configurations. Synchronous transfer mode (period/offset = 200/7) was enabled, which tends to minimize the effect of cable length on performance.

For a given data type, the same amount of data, i.e., from 50M bytes to 300M bytes, was transferred to both versions of the tape product. We often performed several test runs using different amounts of data to check the consistency of the test results.

To maximize the applicability of the test results, we wanted to use "real world" data. To do so in our test environment was not practical or would have introduced delays between blocks, thus ruining any throughput measurements. We obtained a compromise in the following manner. The SDS-3 tool we used is limited by a 64K-byte buffer for high-speed transfers. That buffer can be used repeatedly, and the direct memory access (DMA) pointers automatically "wrap around" back to the start when they reach the end of the buffer. We created a tool

that takes the first 64K bytes from a file with the desired test data, reformats the data, and writes the data to an output file compatible with the SDS-3 software. This "buffer file" can then be uploaded into the SDS-3 tool's memory buffer, thus duplicating the first 64K bytes of the data from the test file in SDS-3 memory. The tool has an obvious limitation; the first 64K bytes of data might not be representative of the rest of the data in the file. Using this tool was, however, a practical way to transfer at least subsets of real data into the throughput test environment.

Benchmark Tests

Since preliminary results of our study indicated that the IDRC chip has a lower compression ratio than that indicated by previous studies, the benchmark tests were performed primarily to confirm the compression efficiency of the IDRC DLT2000 implementation.[4] For the benchmark tests, we tested two tape products that use IDRC compression implementations.

The first product tested was Digital's TA91 tape drive (which is compatible with an IBM 3480E tape drive) configured on a Hierarchical Storage Controller (HSC) in a VAXcluster configuration. A collection of chart and art files included with the standard Harvard Graphics software package composed the data set. This identical data set was written to an IDRC DLT2000 tape drive for accurate comparison.

The second benchmark product tested was an EXB-8505 tape drive, which also uses IDRC compression.[9] We tested the EXB-8505 tape drive on an SDS-3 test system. The data set used was the first 64K bytes of the text of the U.S. Constitution. We compared the compression ratio obtained on the EXB-8505 with the compression ratio for the same data written to a DLZ1 DLT2000 unit and with text data compressed on an IDRC DLT2000 tape drive. (The text data on the IDRC implementation was different from the text data on the EXB-8505 and DLZ1 implementations because an IDRC prototype was no longer readily available when the U.S. Constitution data became part of the tests.) We also performed some throughput tests to compare the DLZ1 DLT2000 and the EXB-8505 drives.

We measured the native product capacity of the TA91 and EXB-8505 tape drives by writing to the end of tape (EOT) with compression disabled. We then repeated this test with compression enabled.

Test Results

The compression ratios shown in the test results are calculated by dividing the number of bytes of uncompressed data by the number of bytes of the same data when compressed. Therefore, a compression ratio of 2.0:1, or simply 2.0, means that the data compressed to one-half its original size, and if maintained for that whole tape, such compression would effectively double the data capacity of the tape drive.

Operating System Test Results

Figure 2 shows the measurements of compression ratio on the OpenVMS and UNIX systems. The difference between the compression ratios of the DLZ1 prototype and those of the IDRC prototype is striking on the graph. The DLZ1 prototype had significantly higher compression ratios for all the data types tested. Note that these results, as compared to the results of the SDS-3 testing, are more representative of the real world, since most of these data sets came from live multimegabyte databases.

We tested the ULTRIX dump utility on the same system and data on which we ran the tar utility. The dump utility compression ratios were almost identical to those obtained with the tar utility. This result was not surprising since the bulk of the data stored was identical—only the metadata created by the utility varied. For comparison purposes, the average compression ratio for these data types was 2.76 for the DLZ1 prototype and 1.54 for the IDRC prototype.

Although compression measurements were the focus of the operating system–based tests, for general information, we also took some throughput measurements. The DECsystem 5500 system running the dd utility achieved write rates of approximately 0.85M bytes/s for the data types. Running the tapex utility's performance test (which is not disk or file system limited) on a similar machine resulted in rates of more than 3M bytes/s. The 3M-byte/s rate implies that, when running dd or tar, the disk and/or file system is the likely bottleneck, since the ULTRIX drivers, SCSI channel, and tape driver were capable of three times the throughput. (Other possibilities are inefficiencies within dd and/or tar, inefficient handling of two devices on the SCSI bus, insufficient CPU horsepower, etc.)

OpenVMS tests showed similar results for the BACKUP utility, but the throughput is likely to have been limited by the KZQSA adapter. Other tests

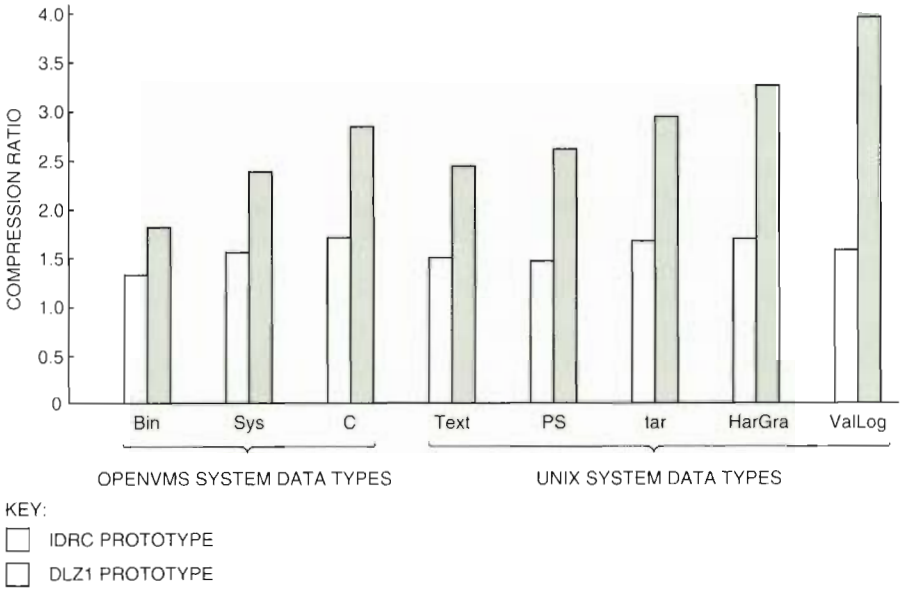


Figure 2 Operating System Data Compression Ratios

indicate that the KZQSA has a limit of 0.8M bytes/s to 0.9M bytes/s with the OpenVMS system.

The informal operating system throughput testing confirms that the particular configurations tested are not suitable for measuring the bandwidth limits of the DLT2000 tape drive, when using the standard backup utilities. Note that the newer VAX and the Alpha AXP platforms have much higher throughput capabilities and are able to more fully utilize the capabilities of the DLT2000 product. These platforms were not available when we performed this study.

SDS-3 Test Results

The SDS-3 tests measured compression ratios and data throughput rates.

Compression Figure 3 shows the SDS-3 data compression ratios. The ratios for the first four data types are in the normal range, i.e., the DLZ1 prototype averaged approximately 2.4 and the IDRC prototype averaged approximately 1.5. For the Paintbrush bitmap file, both prototype versions compressed at about the same efficiency.

Although the 30:1 compression ratio for the Ones pattern data is not representative of normal data, the ratio gives a sense of the maximum efficiency of the algorithms. The Repeat pattern test ratios highlight the ability of the DLZ1 algorithm to capitalize on redundant strings of moderate length (24 bytes, in this case). The IDRC algorithm lacks this ability. None of the many data sets tested compressed better with the IDRC algorithm than with the DLZ1 algorithm. (We tested six other data sets but did not include the test results in this paper because they showed little variation from those presented.)

Throughput Rates Figure 4 shows the data throughput rates for six of the data types; compression ratios are annotated at the bottom for convenience. The use of a line graph rather than a bar graph suggests some correlation between compression ratio and throughput. We tested variants of these data types to explore the strength of this correlation.

With the DLZ1 algorithm, we found data sets that had the same compression ratio but significantly different throughput rates. We saw variations of up to ±0.3M bytes/s from the "expected" rate, which is the native drive rate (1.25M bytes/s) multiplied by the compression ratio.

The throughput rate with the IDRC algorithm tends to correlate more strongly with the compression ratio, but we did see variations. For example, the VAXcamera data at a compression ratio of 1.4 transfers about 0.1M bytes/s faster than Harvard Graphics data, which compresses at 1.6.

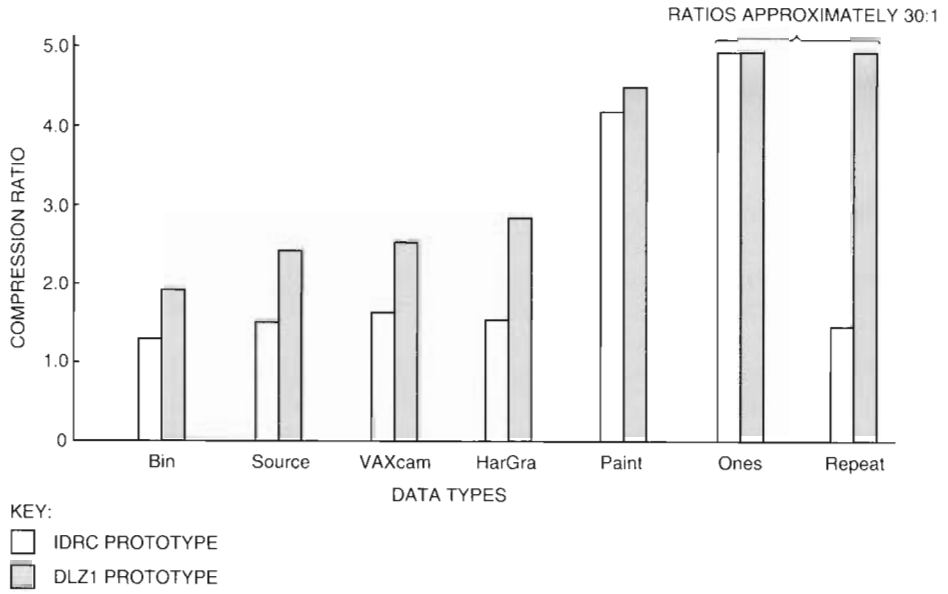


Figure 3 SDS-3 Data Compression Ratios

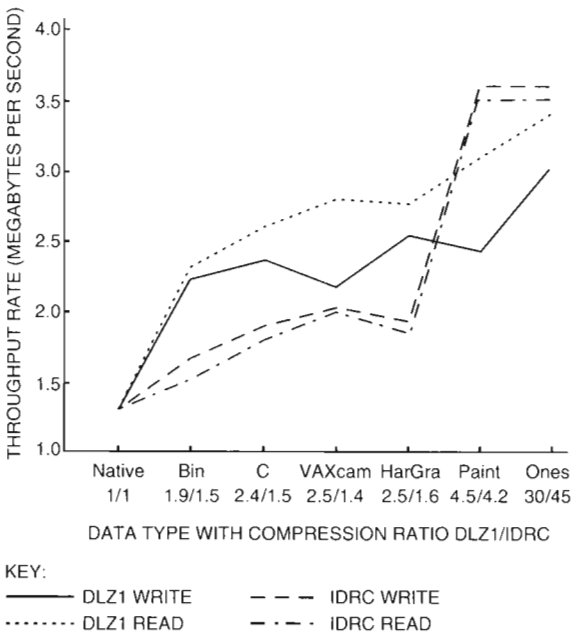


Figure 4 SDS-3 Data Throughput Rates

Even more striking is the difference on write and read transfer rates. The DLZ1 algorithm is almost always significantly faster on decompression. This feature is characteristic of this type of LZ algorithm. On the other hand, IDRC write and read rates match very closely, typically within 0.05M bytes/s.

The throughput limit of the SDS-3 system used was high enough to not usually be a factor. Knowing this fact was essential for the proper interpretation of test results. A bottleneck in the tape device must be distinguishable from an adapter or tester limitation. We measured the throughput limit of the SDS-3 system by writing and reading the Ones pattern and similar data patterns, which are highly compressible by the IDRC algorithm. With a 64K-byte block size, throughput on the SDS-3 system peaked at about 3.5M bytes/s. When we increased the block size 1M byte, the throughput jumped to nearly 4.5M bytes/s. This increase was due to reduction in the amount of command overhead for a given amount of data being transferred on the SCSI bus. None of the normal data types tested, except the Paintbrush bitmap files, could approach compression ratios high enough to begin to push the limits of the SDS-3 system.

These results indicate that at higher data rates, the SDS-3 system becomes a limiting factor. Analysis of SCSI protocol handling on the SDS-3 system shows that the nondata portions of a transaction (e.g., message, command, and status) are handled somewhat inefficiently. At high throughput rates, this overhead is significant enough to affect throughput to the device. Using a larger block size reduces this per-command overhead for a given amount of tape data and allows a higher throughput to be achieved on the SCSI bus.

Benchmark Test Results

We wrote the Harvard Graphics data set repeatedly to the TA91 tape drive. With compression disabled, about 132M bytes fit on the media. With compression enabled, 216M bytes were written, giving a compression ratio of 1.64. This ratio compares closely with the 1.66 obtained on the IDRC DLT2000 prototype.

We then used the SDS-3 tool to repeatedly write the first 64K bytes of the U.S. Constitution to the EXB-8505 tape drive. With compression disabled, about 5G bytes were written. With compression enabled, 7.6G bytes were written, giving a compression ratio of 1.52. Again, this corresponds closely with the compression ratio of 1.54 achieved when writing text data on the IDRC DLT2000 prototype.

We performed more testing for general comparison between the DLZ1 DLT2000 product and the EXB-8505 product. The U.S. Constitution data compressed at 2.23 on the DLT2000 drive and at 1.52 on the EXB-8505 drive. Figure 5 shows the results of throughput testing with this data on these two products, using two block sizes, 10K-byte blocks and 64K-byte blocks.

Conclusions

The compression efficiency testing outlined in this paper indicates that, for most data sets, the DLZ1 algorithm usually achieves a higher compression ratio than the IDRC algorithm and, therefore, yields a consistent capacity advantage over the IDRC algorithm. The reader should carefully note that regardless of the algorithm used, the actual capacity increase that a user might realize with data compression depends heavily on the specific mix of data. The following summarizes the compression results presented in this paper. Based on the compression testing in the operating system environment, a DLT2000 product using DLZ1 compression has a typical capacity of 25G bytes to 30G bytes. A DLT2000 product using IDRC compression would typically hold about 15G bytes of data.

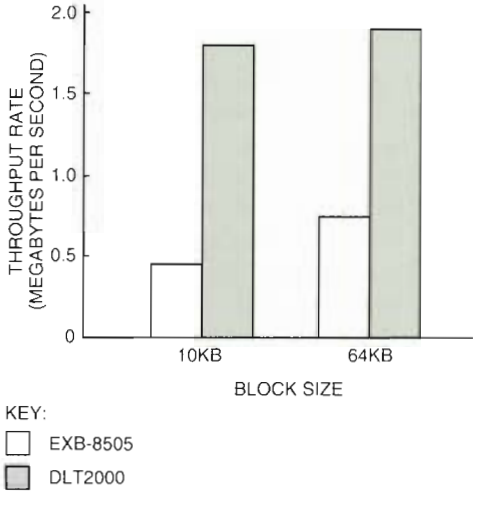


Figure 5 EXB-8505 and DLT2000 Data Throughput Rates

The data throughput testing showed that, in most cases, the DLZ1 DLT2000 prototype transferred data at a faster rate than the IDRC DLT2000 prototype—even though the IDRC prototype's hardware implementation was capable of almost twice the data rate (5M bytes/s for the IDRC drive and 2.5M/3.0M bytes/s for the DLZ1 drive). The IDRC implementation did not perform better for two reasons.

1. Given the same data set, the compression ratio of the IDRC implementation is almost always less than that of the DLZ1 implementation.
2. The typical compression ratio of the IDRC implementation is somewhat low, in an absolute sense (less than 1.8).

Since data compression in the tape device has a multiplying effect on data transfer rates seen by the host, a low compression ratio limits the practical rate at which compressed data can be made available to the tape media.

To transfer data faster than the DLZ1 prototype, the IDRC prototype must achieve a compression ratio that multiplies the drive's native data rate beyond the throughput limit of the DLZ1 prototype. This limit is about 2.5M bytes/s for write operations. Calculating the approximate minimum compression ratio (Cr) needed is straightforward, as the following steps show:

$$Cr \times (\text{native data transfer rate}) = \text{throughput limit}$$
$$Cr \times 1.25\text{M bytes/s} = 2.5\text{M bytes/s}$$
$$Cr = (2.5\text{M bytes/s})/(1.25\text{M bytes/s})$$
$$Cr = 2.0 \text{ (or } 2.0{:}1)$$

Thus, when the IDRC prototype compresses data at a rate greater than 2.0:1, its transfer rate should exceed that of the DLZ1 prototype. Indeed, with the Paintbrush and Ones data patterns, the

compression ratio was more than 4.0:1, and the transfer rate measurements show the throughput potential of the IDRC implementation over the DLZ1 implementation. These data patterns are not typical, however, and more realistic data sets (e.g., binary, source files, text, and databases) show the IDRC algorithm compression ratios to be only in the 1.5 to 1.7 range. The benchmark testing confirms these results and, therefore, the correctness of the IDRC DLT2000 implementation. These low IDRC compression ratios for typical data are what prevent the IDRC implementation from achieving its throughput potential on the DLT2000 tape product.

The DLZ1 DLT2000 implementation was adopted for the actual DLT2000 tape product. As the development team completed the design, they made hardware and firmware improvements to enhance the data throughput characteristics of the final product. For example, they increased the clock rate on the compression chip by 10 percent and optimized critical firmware code paths.

Acknowledgments

Other members of the firmware engineering team made contributions relevant to this paper. In particular, I would like to thank Brian LeBlanc for conducting performance SDS-3 test runs that confirmed my results and in some cases were incorporated into the data presented. I would also like to thank Haim Bitner for assisting me in digging into the theory behind the LZ and IDRC compression algorithms and for running the EXB-8505 benchmark tests.

References

1. D. Whiting et al., *Data Compression Apparatus and Method,* U.S. Patent 5,016,009 (May 14, 1991).

2. "9705 Data Compression Coprocessor Data Sheet," Revision 1.00, *STAC Electronics* (December 1991).

3. T. Welch, "A Technique for High-Performance Data Compression," *Computer 17* (June 1984): 8–19.

4. V. Chinnaswamy, "An Overview of Compression Techniques and TA90 Performance with Compression," internal report (Maynard, MA: Digital Equipment Corporation, July 1991). This internal document is unavailable to external readers.

5. D. Lelewer, *Current Techniques in Data Compression* (Irvine, CA: University of California, Instructional Television Network, 1993).

6. *Compaction Algorithm, Binary Arithmetic Coding,* 1st Draft, Proposed American National Standard X3B5 (November 17, 1989).

7. T. Bell, J. Cleary, and I. Witten, *Text Compression* (Englewood Cliffs, NJ: Prentice Hall, 1990).

8. J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory,* vol. IT-23, no. 3 (May 1977): 337–343.

9. *EXB-8505 8mm Cartridge Tape Subsystem User's Manual,* Revision 002 (Boulder, CO: EXABYTE Corporation, November 1992).

Further Readings

The Digital Technical Journal *publishes papers that explore the technological foundations of Digital's major products. Each* Journal *focuses on at least one product area and presents a compilation of refereed papers written by the engineers who developed the products. The content for the* Journal *is selected by the Journal Advisory Board. Digital engineers who would like to contribute a paper to the* Journal *should contact the editor at RDVAX::BLAKE.*

Topics covered in previous issues of the *Digital Technical Journal* are as follows:

High-performance Networking/OpenVMS AXP System Software/Alpha AXP PC Hardware
Vol. 6., No. 1, Winter 1994, EY-Q011E-TJ

Software Process and Quality
Vol. 5, No. 4, Fall 1993, EY-P920E-DP

Product Internationalization
Vol. 5, No. 3, Summer 1993, EY-P986E-DP

Multimedia/Application Control
Vol. 5, No. 2, Spring 1993, EY-P963E-DP

DECnet Open Networking
Vol. 5, No. 1, Winter 1993, EY-M770E-DP

Alpha AXP Architecture and Systems
Vol. 4, No. 4, Special Issue 1992, EY-J886E-DP

NVAX-microprocessor VAX Systems
Vol. 4, No. 3, Summer 1992, EY-J884E-DP

Semiconductor Technologies
Vol. 4, No. 2, Spring 1992, EY-L521E-DP

PATHWORKS: PC Integration Software
Vol. 4, No. 1, Winter 1992, EY-J825E-DP

Image Processing, Video Terminals, and Printer Technologies
Vol. 3, No. 4, Fall 1991, EY-H889E-DP

Availability in VAXcluster Systems/Network Performance and Adapters
Vol. 3, No. 3, Summer 1991, EY-H890E-DP

Fiber Distributed Data Interface
Vol. 3, No. 2, Spring 1991, EY-H876E-DP

Transaction Processing, Databases, and Fault-tolerant Systems
Vol. 3, No. 1, Winter 1991, EY-F588E-DP

VAX 9000 Series
Vol. 2, No. 4, Fall 1990, EY-E762E-DP

DECwindows Program
Vol. 2, No. 3, Summer 1990, EY-E756E-DP

VAX 6000 Model 400 System
Vol. 2, No. 2, Spring 1990, EY-C197E-DP

Compound Document Architecture
Vol. 2, No. 1, Winter 1990, EY-C196E-DP

Distributed Systems
Vol. 1, No. 9, June 1989, EY-C179E-DP

Storage Technology
Vol. 1, No. 8, February 1989, EY-C166E-DP

CVAX-based Systems
Vol. 1, No. 7, August 1988, EY-6742E-DP

Software Productivity Tools
Vol. 1, No. 6, February 1988, EY-8259E-DP

VAXcluster Systems
Vol. 1, No. 5, September 1987, EY-8258E-DP

VAX 8800 Family
Vol. 1, No. 4, February 1987, EY-6711E-DP

Networking Products
Vol. 1, No. 3, September 1986, EY-6715E-DP

MicroVAX II System
Vol. 1, No. 2, March 1986, EY-3474E-DP

VAX 8600 Processor
Vol. 1, No. 1, August 1985, EY-3435E-DP

Subscriptions and Back Issues

Subscriptions to the *Digital Technical Journal* are available on a prepaid basis. The subscription rate is $40.00 (non-U.S. $60.00) for four issues

and $75.00 (non-U.S. $115.00) for eight issues. Orders should be sent to Cathy Phillips, Digital Equipment Corporation, 30 Porter Road LJO2/D10, Littleton, Massachusetts 01460, U.S.A., Telephone: (508) 486-2538, FAX: (508) 486-2444. Inquiries can be sent electronically to DTJ@CRL.DEC.COM. Subscriptions must be paid in U.S. dollars, and checks should be made payable to Digital Equipment Corporation.

Single copies and past issues of the *Digital Technical Journal* are available for $16.00 each by calling DECdirect at 1-800-DIGITAL (1-800-344-4825). Recent back issues of the *Journal* are available on the Internet at gatekeeper.dec.com in the directory /pub/DEC/DECinfo/DTJ.

Digital Research Laboratory Reports

Reports published by Digital's research laboratories can be accessed on the Internet through the World Wide Web or FTP. For access information on the electronic or hard-copy versions of the reports, see http://gatekeeper.dec.com/hypertext/info/cra.reports.html.

Technical Papers by Digital Authors

P. Anick, "Integrating Natural Language Processing and Information Retrieval for a Computer Troubleshooting Help-Desk," *IEEE Expert* (December 1993).

B. Archambeault, "An Investigation into Alternative Construction Techniques to Reduce Shielded Room Resonance Effects," *Applied Computational Electromagnetics Society Symposium* (March 1994).

B. Archambeault, "Predicting EMI Emission Levels Using EMSCAN," *IEEE International Symposium on Electromagnetic Compatibility* (August 1993).

B. Archambeault and O. Ramahi, "Adaptive Absorbing Boundary Conditions in Finite Difference Time Domain Applications for EMI Simulation," *Applied Computational Electromagnetics Society Symposium* (March 1994).

N. Arora, "MOSFET Modeling for VLSI Circuit Simulation: Theory and Practice," *Computational Electronics* (New York: Springer-Verlag, 1993).

N. Arora, "MOSFET Modeling for VLSI Simulation," *Seventh International Conference on Physics of Semiconductor Devices* (December 1993).

E. Atakov, J. Clement, and B. Miner, "Two Electromigration Failure Modes in Polycrystalline Aluminum Interconnects," *IEEE International Reliability Physics Proceedings* (April 1994).

D. Bailey, "Diffuser Cooling Technology for Electronic High-Density Packaging," *Electro '94 International* (May 1994).

D. Bailey, "Improved Cooling of Electronic Equipment from the Use of a Diffuser," *ASME International Electronics Packaging Conference* (September 1993).

D. Bailey and S. Lindquist, "Improved Heat-Transfer Rates for Impingement Cooling Fins," *ASME International Electronics Packaging Conference* (September 1993).

D. Bhandarkar and Z. Cvetanovic, "Characterization of Alpha AXP Performance," *Twenty-first Annual International Symposium on Computer Architecture* (April 1994).

S. Bosworth, S. Hsu, and F. Polcari, "Exceptional Performance from the Development, Qualification and Implementation of a Silicone Adhesive for Bonding Heatsinks to Semiconductor Packages," *Forty-fourth Electronic Components and Technology Conference* (May 1994).

M. Bouziane, "Paradigm Translations in Integrated Manufacturing Information Using a Meta-Model: The TSER Approach," *Ingénierie des systèmes d'information* (November 1993).

M. Bouziane, "A Rulebase Model for Data and Knowledge Integration in Multiple Systems Environments," *International Journal on Artificial Intelligence Tools* (January 1994).

J. Bowman, "Heat Pipes—Operating Characteristics and Performance Stability," *National Electronic Packaging and Production Conference (NEPCON WEST '94)* (March 1994).

C. Brench and B. Archambeault, "Shielded Air Vent Design Guidelines from EMI Modeling," *IEEE Electromagnetic Compatibility Symposium* (August 1993).

K. Brown, "Manufacturing Challenges in Single-Chip Packaging," *ASME Winter Annual Meeting* (November 1993).

B. Cantell and J. Ramirez, "Optimization of a Wafer Stepper Alignment System Using Robust Design," *Sematech Technology Transfer* (December 1993).

B. Cantell and J. Ramirez, "Robust Design of a Polysilicon Deposition Process Using Split-Plot Analysis," *Quality and Reliability Engineering International* (April 1994).

R. Cappel and M. Nasr, "Framework for an Advanced Inspection Program," *IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop (ASMC '93 Proceedings)* (October 1993).

P. Catinella and B. Edwards, "The Use of Focused Ion Beam Milling Techniques as a Yield Enhancement Tool for Failure Analysis of VLSI Devices," *Proceedings of the Nineteenth International Symposium for Testing and Failure Analysis (ISTFA '93)* (November 1993).

R. Collica, "A Logistic Regression Yield Model for SRAM Bit Fail Patterns," *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems* (October 1993).

B. Doyle and K. Mistry, "Anomalous Hot-Carrier Behavior for LDD p-Channel Transistors," *IEEE Electron Device Letters* (November 1993).

M. Elbert, R. Howe, and T. Weyant, "Software Reliability Methodology," *IEEE Spring Reliability Symposium* (April 1994).

M. Elbert, C. Mpagazehe, and T. Weyant, "Stress Testing and Reliability," *Southcon* (March 1994).

M. Elbert, C. Mpagazehe, and T. Weyant, "Stress Testing—Key to Providing System Reliability," *IEEE Spring Reliability Symposium* (April 1994).

T. Ermolovich, "Mobile Data Networking: The Need for a Systems Perspective," *IEEE Fourth Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '93)* (September 1993).

R. Gandara, "Case Study: Workcell for Microprocessor Test," *SME AutoFact West '94 Conference* (March 1994).

D. Hassett, "High Pin Count VGA Sockets Raise Challenge," *Interconnection Technology* (August 1993).

F. Horwitz, E. Thomson, E. Aavov, and H. Levesque, "An Optical Waveguide Circuit Board with Surface Mounted Optical Receivers," *Optical Engineering* (March 1994).

M. Hsueh, "Online Workload, Performance, and Scalability of a Database Production System: A Case Study," *Proceedings of the Seventeenth Annual International Computer Software and Applications Conference (COMPSAC '93)* (November 1993).

C. Huang and N. Arora, "Characterization and Modeling of the n- and p-Channel MOSFETs Inversion-Layer Mobility in the Range 25 to 125C," *Solid State Electronics* (January 1994).

C. Huang and N. Arora, "Measurements and Modeling of MOSFET I-V Characteristics with Polysilicon Depletion Effect," *IEEE Transactions on Electron Devices* (December 1993).

R. Jain, "FDDI: Current Issues and Future Plans," *IEEE Communications Magazine* (September 1993).

C. Juszczak, "Improving the Write Performance of an NFS Server," *USENIX Winter 1994 Conference* (January 1994).

N. Khalil and J. Faricelli, "MOSFET Two-Dimensional Doping Determination," *Simulation of Semiconductor Devices and Processes (SISDEP)* (September 1993).

S. Kleinfeldt, "Design Methodology Management," *Proceedings of the IEEE* (February 1994).

K. Kodandapani and J. Grodstein, "A Simple Algorithm for Fanout Optimization Using High-Performance Buffer Libraries," *IEEE/ACM International Conference on Computer-aided Design (ICCAD-93)* (November 1993).

J. Lekas and J. Pew, "Photochemical Batches: E0 Modelling and Applications," *Proceedings of the International Society of Photo-Optical Instrumentation Engineers (SPIE): Integrated Circuit Metrology, Inspection, and Process Control VIII* (February 1994).

T. Michalka, "A High Performance HDI Based Pin Grid Array Package," *Forty-fourth Electronic Components and Technology Conference* (May 1994).

K. Mistry, B. Doyle, and B. Fishbein, "Effect of Plasma-Induced Charging Damage on n-Channel and p-Channel MOSFET Hot Carrier Reliability," *IEEE International Reliability Physics Proceedings* (April 1994).

K. Mistry, B. Doyle, R. Hokinson, B. Gieseke, F. Fox, and R. Preston, "Voltage Overshoots and N-MOSFET Hot Carrier Robustness in VLSI Circuits," *IEEE International Reliability Physics Proceedings* (April 1994).

E. Moy, "Implementing Network Protocols at User Level," *IEEE/ACM Transactions on Networking* (October 1993).

Q. Ng, "Composition and Performance of Hydrogenated Carbon Overcoats on Magnetic Storage Discs," *ASME/STLE Tribology Conference* (August 1993).

M. Nihart, O. Newkerk, and S. Wong, "The Common Agent: A Multiprotocol Management Agent," *IEEE Journal on Selected Areas in Communications* (December 1993).

D. Pan, "An Overview of the MPEG/audio Compression Algorithm," *Proceedings of the International Society of Photo-Optical Instrumentation Engineers (SPIE), Digital Video Compression on Personal Computers: Algorithms and Technologies* (February 1994).

R. Razdan, G. Bischoff, and E. Ulrich, "Clock Suppression Techniques for Synchronous Circuits," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems* (October 1993).

A. Rewari, "AI in Corporate Service and Support," *IEEE Expert* (December 1993).

R. Rios, N. Arora, and C. Huang, "An Analytic Polysilicon Depletion Effect Model for MOSFET's," *IEEE Electron Device Letters* (April 1994).

J. Rose, T. Sriram, R. Shuman, and T. Spooner, "Microscopy Methods for Integrated Interconnect Evaluation," *Proceedings of the Nineteenth International Symposium for Testing and Failure Analysis (ISTFA '93)* (November 1993).

S. Sathaye, "A Systematic Approach to Designing Distributed Real-Time Systems," *IEEE Computer* (September 1993).

J. Sauber, "The Manufacturing and Reliability of 'Footless' Gullwing SMT Solder Joints," *Materials Research Society* (Fall 1993).

J. Sauber, L. Lee, S. Hsu, and T. Hongsmatip, "Fracture Properties of Molding Compound Materials for IC Plastic Packaging," *Forty-fourth Electronic Components and Technology Conference* (May 1994).

J. Seyyedi, "Thermal Fatigue of Low-Temperature Solder Alloys in Insertion Mount Assembly," *ASME International Electronics Packaging Conference* (September 1993).

A. Shvartsman, "Controlling Memory Access Concurrency in Efficient Fault-Tolerant Parallel Algorithms," *Seventh International Workshop on Distributed Algorithms (WDAG '93)* (September 1993).

A. Shvartsman, "Dealing with History and Time in a Distributed Enterprise Management Director," *IEEE Network* (November 1993).

P. Sinha, "Redefining System Management," *CMG Transactions* (October 1993).

J. Steele, D. Siden, K. Gould, and H. Simmons, "Chlorinated Solvent Elimination in Chip Capacitor Attach Using Qualification by Comparison," *IEEE/CHMT International Electronics Manufacturing Technology (IEMT) Symposium* (October 1993).

K. Steeples, G. Tai, D. Fess, and D. Fletcher, "Cost of Ownership Benefits Using Multiply Charged Ion Implants on Conventional Medium and High Current Implanters," *IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop (ASMC '93 Proceedings)* (October 1993).

N. Sullivan, I. Fink, and J. Lekas, "Overlay Sample Plan Optimization for the Detection of Higher Order Contributions to Misalignment," *Proceedings of the International Society of Photo-Optical Instrumentation Engineers (SPIE): Integrated Circuit Metrology, Inspection, and Process Control VIII* (February 1994).

Digital Press

Digital Press, the authorized publisher for Digital Equipment Corporation, is an imprint of Butterworth-Heinemann, a major international publisher of professional books and a member of the Reed Elsevier group. The following are descriptions of computing titles available from Digital Press.

OPENVMS AXP INTERNALS AND DATA STRUCTURES
Ruth E. Goldenberg and Saro Saravanan, June 1994, hardbound, 1,800 pages, ISBN: 55558-120-X ($150.00).

This book describes in vivid detail the internals and data structures of the OpenVMS AXP operating system Version 1.5. Perhaps the most comprehensive and up-to-date description available for a commercial operating system, it is an irreplaceable reference for operating system development engineers, operating system troubleshooting experts, systems programmers, consultants, and customer support specialists. Some of the text and much of the book's structure are derived from its highly successful predecessor, *VAX/VMS Internals and Data Structures: Version 5.2*. The new work is divided into nine parts: Introduction; Control Mechanisms; Synchronization; Scheduling and Time Support; Memory Management; Input/Output; Life of a Process; Life of the System; and Miscellaneous Topics. Each of the 39 chapters is akin to a case study on the topic it covers, based on the depth and breadth of treatment.

THE UNIX PHILOSOPHY
Mike Gancarz, September 1994, softcover, 176 est. pages, ISBN: 55558-123-4 ($19.95 est.).

Unlike many books that focus on how to use the UNIX operating system, *The UNIX Philosophy* concentrates on answering the question, "Why use UNIX in the first place?" Readers will discover the rationale and reasons for such concepts as file system organization, user interface, and other system characteristics. In an informative, nontechnical fashion, *The UNIX Philosophy* explores the general principles for applying the UNIX philosophy to software development. This book describes complex software design principles and addresses the importance of small programs, code and data portability, early prototyping, and open user interfaces. Written for both the computer layperson and the experienced programmer, this book explores the tenets of the UNIX operating system in detail, dealing with powerful concepts in a comprehensive, straightforward manner.

VAXCLUSTER PRINCIPLES
Roy G. Davis, 1993, paperbound, 600 pages, ISBN: 55558-112-9 ($49.95).

This in-depth exploration of the VMS operating system is ideal for computer professionals who need a thorough understanding of VAXcluster components and functionality to support, manage, and develop applications in a VAXcluster environment.

DIGITAL'S CDD/REPOSITORY: A Comprehensive Guide
Lilian Hobbs and Ken England, 1993, paperbound, 259 pages, ISBN: 55558-113-7 ($34.95).

This comprehensive guide focuses on Version 5.0 of CDD/Repository—an extremely sophisticated and powerful repository based on an object-oriented approach. This active distributed repository system provides the functionality necessary for users to organize, manage, control, and integrate tools and applications across their companies. The repository simplifies application development by providing information management and environment management features.

WORKING WITH TEAMLINKS
Tony Redmond, 1993, paperbound, 446 pages (includes diskette), ISBN: 55558-116-1 ($44.95).

Working with TeamLinks is a practical guide to Digital's office system for the Microsoft Windows graphical user environment. Its thorough coverage will help experienced and inexperienced users, programmers, and system implementers realize the benefits while avoiding the pitfalls of using PCs in an integrated multivendor office system. The book shows how the TeamLinks File Cabinet works, how TeamLinks mail flows, how to streamline business processes with TeamRoute document-routing system, and how to integrate applications in a TeamLinks environment. It discusses the problems of implementing a PC-based office system and of managing the process of migration from ALL-IN-1 ISO, Digital's minicomputer-based office system. An appendix documents TeamLinks internal codes and presents other interesting information. A companion diskette contains many sample programs that can be used as a base for your own solutions.

NAS ARCHITECTURE REFERENCE MANUAL
Leo F. Laverdure, Patricia Srite, and John Colonna-Romano, 1993, paperbound, 564 pages, ISBN: 55558-115-3 ($34.95).

Designed for anyone interested in learning about the NAS architecture—including application developers, technical consultants, Independent Software Vendors (ISVs), Value-Added Resellers (VARs), and Digital's Integrated Business Units (IBUs)—the *NAS Architecture Reference Manual* provides information on the NAS services and the key public interfaces supported by each service.

NAS: Digital's Approach to Open Systems
James Martin and Joe Leben, 1993, paperbound, 412 pages, ISBN: 55558-117-X ($34.95).

Network Application Support (NAS) is both a comprehensive architecture and a set of software products. NAS provides a framework that makes it possible for applications developers to enhance those characteristics of computing applications that promote interoperability, application distributability, and application portability among applications that run on Digital's computing platforms as well as applications from other vendors, such as IBM, Hewlett-Packard, Sun Microsystems, and Apple Computer. For managers, executives, and information systems staff, the book describes the two types of NAS products: (1) the development toolkits that provide services directly to computing applications, both Digital applications and user-written applications—this important new class of software, called middleware, operates as an intermediary between application programs and the underlying hardware/software platform; and (2) the products that build on this NAS middleware to provide services directly to the end users of computing services.

USING MS-DOS KERMIT: Connecting Your PC to the Electronic World, Second Edition
Christine M. Gianone, 1992, paperbound, 345 pages (includes diskette), ISBN: 55558-082-3 ($34.95).

Using MS-DOS Kermit is a book/disk package designed to help technical and nontechnical PC users alike to link their IBM PCs, PS/2s, or compatibles to other computers and data services—e.g., Dow Jones News/Retrieval, MCI Mail, databases like BBS, DIALOG, and TYMNET, and any mainframe—throughout the world. Based on the author's close involvement with development and distribution of the Kermit transfer protocol, the guide supplies easy-to-follow, step-by-step instructions, meticulously compiled tables, and at-a-glance information on important areas.

USING C-KERMIT: Communication Software for OS/2 Atari ST, UNIX, OS-9, VMS, AOS/VS, Amiga
Frank da Cruz and Christine M. Gianone, 1993, paperbound, 514 pages, ISBN: 55558-108-0 ($34.95).

Using C-Kermit describes the new release, 5A, of Columbia University's popular C-Kermit communication software—the most portable of all communication software packages. Available at low cost on a variety of magnetic media from Columbia University, C-Kermit can be used on computers of all sizes, ranging from desktop workstations to minicomputers to mainframes and supercomputers. The numerous examples, illustrations, and tables in *Using C-Kermit* make the powerful and versatile C-Kermit functions accessible for new and experienced users.

USING DECWINDOWS MOTIF FOR OPENVMS
Margie Sherlock, 1993, paperbound, 363 pages, ISBN: 55558-114-5 ($34.95).

The book *Using DECwindows Motif for OpenVMS* is designed to help new OpenVMS DECwindows users explore and apply DECwindows techniques and features and to provide experienced DECwindows users with practical information about the Motif interface, ways to customize environments, and advanced user topics. OpenVMS DECwindows Motif is based on MIT's specification for the X Window System, Version 11, Release 4 and OSF/Motif 1.1.1.

X AND MOTIF QUICK REFERENCE GUIDE, Second Edition
Randi J. Rost, 1993, paperbound, 398 pages, ISBN: 55558-118-8 ($24.95).

Arranged in five sections—X Protocol Reference, Xlib Reference, X Toolkit Reference, Motif Reference, and General X Reference—and organized alphabetically with thumb tabs for quick and easy reference, the *X and Motif Quick Reference Guide* provides complete descriptions of routines and user-accessible data structures, including Xlib subroutines and macros, X Toolkit Intrinsics routines,

Motif routines, and all of the Motif Widgets. The Second Edition has been updated to reflect new functionality in both X Window System, Version 11, Release 5, and OSF/Motif Version 1.2, including routines and Xlib to support color management system and new routines on Xlib to better provide support for internationalization and localization.

ALL-IN-1: Managing and Programming in V3.0
Tony Redmond, 1993, paperbound, 552 pages, ISBN: 55558-101-3 ($52.95).

ALL-IN-1: Managing and Programming in V3.0 assists both new and experienced ALL-IN-1 system managers and programmers to make the best of ALL-IN-1 V3.0—the best single release of ALL-IN-1 since V2.0 (1985).

ALL-IN-1: Integrating Applications in V3.0
John Rhoton, 1993, paperbound, 265 pages, ISBN: 55558-102-1 ($52.95).

ALL-IN-1: Integrating Applications in V3.0 helps programmers experienced in third-generation languages to use code-level integration to (1) integrate non-Digital products and applications that may be either difficult to integrate using documented ALL-IN-1 features, integrated only by incurring significant performance overhead, or integratable without preservation of the ALL-IN-1 familiar look and feel; (2) build applications that surpass performance limitations; and (3) access external data stored in any format. The book gives system managers an overview of code-level integration and diagnostic help for product installations, including coverage on relinking the ALL-IN-1 image.

A BEGINNER'S GUIDE TO VAX/VMS UTILITIES AND APPLICATIONS, Second Edition
Ronald M. Sawey and Troy T. Stokes, 1992, paperbound, 399 pages, ISBN: 55558-066-1 ($27.95).

A Beginner's Guide to VAX/VMS Utilities and Applications offers a hands-on introduction to the EDT and EVE screen editor programs, the DECspell spelling checker, WPS-PLUS, phone and mail utilities, VAX notes, the DATATRIEVE database management program, the DECalc electronic spreadsheet, the BITNET network, and more. Included are a wealth of lively examples, exercises, and illustrations, plus "quick reference" charts summarizing commands and operations at the end of each chapter.

VAX/VMS OPERATING SYSTEM CONCEPTS
David Donald Miller, 1992, hardbound, 550 pages, 55558-065-3 ($44.95).

Combining discussions of operating system theory with examples of its application in key VAX/VMS operating system facilities, this book provides a thoughtful introduction for application programmers, system managers, and students. Each chapter begins with a discussion of the theoretical aspects of a key operating system concept—including generally recognized solutions and algorithms—followed by an explanation of how the concept is implemented, plus an example that shows the uses and implications of the approach.

DECNET PHASE V: An OSI Implementation
James Martin and Joe Leben, 1992, clothbound, 572 pages, ISBN: 55558-076-9 ($49.95).

Broaden your understanding of how large networks are designed with this introduction to the concepts surrounding Digital Network Architecture. This book blends an OSI tutorial with a complete look at how OSI technology is used in a Digital computer network. You will gain useful insights into OSI and the process of OSI standardization as well as implementation—all presented in a straightforward, easy-to-follow style.

INFORMATION IN THE ENTERPRISE: It's More than Technology
Geoffrey Darnton and Sergio Giacoletto, 1992, clothbound, 318 pages, ISBN: 55558-091-2 ($34.95).

This nontechnical book examines the role of information in the broader business enterprise—how to use it to gain competitive advantage and to redesign business processes for greater efficiency.

ENTERPRISE NETWORKING: Working Together Apart
Ray Grenier and George Metes, 1992, clothbound, 260 pages, ISBN: 55558-074-2 ($29.95).

Focusing on work environments in which knowledge workers use electronic networks and networking techniques to access, communicate, and share information, this book develops strategic and practical approaches that distributed organizations can use to succeed and compete.

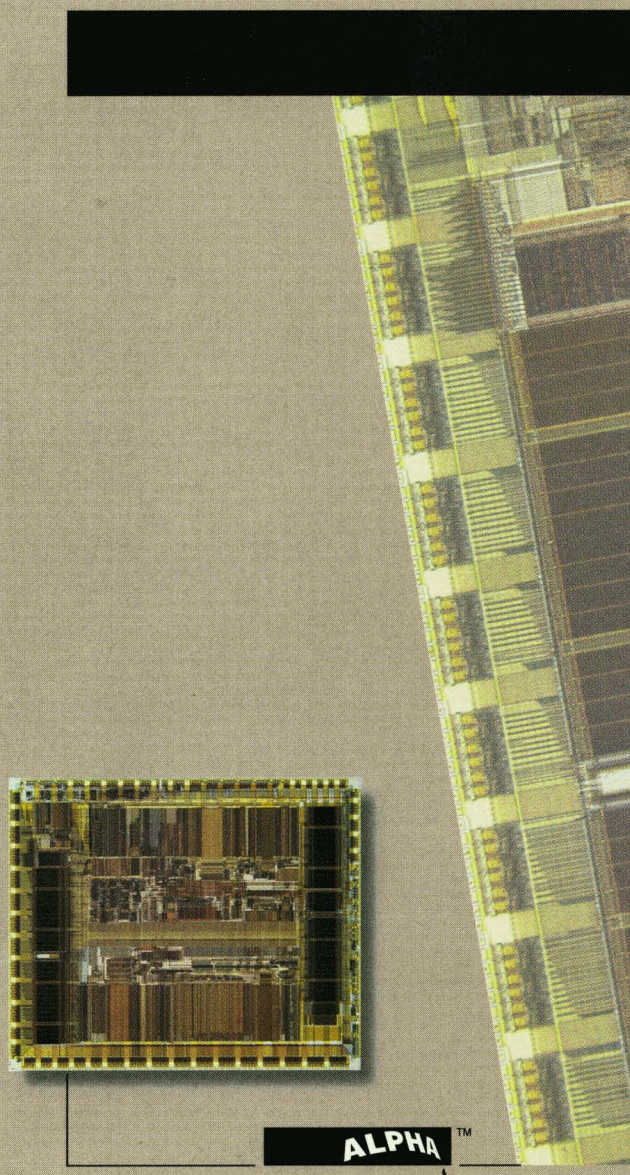
Call for Authors

Digital Press has become an imprint of Butterworth-Heinemann, a major international publisher of professional books and a member of the Reed Elsevier group. Digital Press remains the authorized publisher for Digital Equipment Corporation: the two companies are working in partnership to identify and publish new books under the Digital Press imprint and create opportunities for authors to publish their work.

Digital Press remains committed to publishing high-quality books on a wide variety of subjects. We would like to hear from you if you are writing or thinking about writing a book.

Contact: Frank Satlow, Publisher
Digital Press
313 Washington Street
Newton, MA 02158
Tel: (617) 928-2649

digital™



ALPHA™ GENERATION

ISSN 0898-901X