DEC 3000 Model 400/400S AXP

Technical Summary

Order Number: EK–SNDPR–TM. B01

Digital Equipment Corporation, Maynard, MA

First Printing, November 1992

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

FCC NOTICE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

This document is available on CD–ROM.

S1872

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

4 DEC 3000 Model 400S AXP Server

A Alpha AXP: A Computing Architecture for the 21st Century

B DECchip 21064: A Technical Summary

Glossary

Figures

Tables

Preface

About This Summary

This summary presents a technical description of the DEC 3000 Model 400/400S AXP base system, focusing on the architectural features and implementation details that most contribute to its performance.

For information on the installation and operation details of the system, refer to the *DEC 3000 Model 400/400S AXP Owner's Guide* and the *DEC 3000 Model 400/400S AXP Options Guide*.

Structure of This Summary

This summary contains four chapters, two appendices, and a glossary, as described in Table 1:

Table 1 Summary Structure Description

Chapter 1	This chapter introduces the DEC 3000 Model 400/400S AXP system, including an overview of the DECchip 21064 microprocessor and CPU performance parameters.
Chapter 2	This chapter describes the DEC 3000 Model 400/400S AXP system software and operating systems. The information includes the compilers, graphics interfaces, and network and application software.
Chapter 3	This chapter describes the various graphics options for the DEC 3000 Model 400 AXP system.

(continued on next page)

Table 1 (Cont.) Summary Structure Description

Chapter 4	This chapter gives an overview of the DEC 3000 Model 400S AXP server.
Appendix A	This appendix gives an overview of the Alpha AXP architecture and its features.
Appendix B	This appendix gives a technical summary of the DECchip 21064 microprocessor.

1

DEC 3000 Model 400/400S AXP System Overview

Introduction

The DEC 3000 Model 400/400S AXP system is available for the desktop or deskside. This chapter provides general information about the system.

In This Chapter

This chapter contains the following sections:

- System Features and Benefits
- System Configurations: Workstation and Server
- Physical Characteristics and Operating Environment
- System Architecture
- Operating Systems
- Graphics Options
- System Options

System Features and Benefits

System Features

The DEC 3000 Model 400/400S AXP system provides the following special features:

- **Alpha AXP 64-bit architecture.** Double the industry-standard 32-bit memory address.
- **Up to 128 megabytes of memory.** Expandable memory using 16, 32, and 64-megabyte memory boards.
- **Internal and external options.** Increased storage capacity, graphics, communications, and other capabilities for your system.
- **An AUI (thickwire) Ethernet port.** Connection to an AUI Ethernet network.
- **A 10BASE-T network port.** Connection to a twisted pair network.
- **TURBOchannel slots.** Access to high-performance module interconnect technology that allows a variety of options, including 2D and 3D graphics, multimedia, and other third-party applications.
- **A password security feature.** Additional system security for privileged commands in console mode.
- **Audio technology.** Built-in audio input and output capabilities.
- **Two operating systems.** Choice of OpenVMS AXP or DEC OSF/1 AXP operating systems.
- **Access to an integrated computing environment.** The best features of timesharing and the use of local or distributed applications.
- **DECwindows Motif software.** An industry-standard, windows-style user interface to multiple applications.

System Specifications

Table 1–1 lists the features of the DEC 3000 Model 400 AXP workstation and DEC 3000 Model 400S AXP server.

Table 1–1 System Features at a Glance

Processor	DECchip 21064, 64-bit CMOS-4
Processor clock rate	133 MHz
Number of processors	1
First-level cache (on chip)	8 KB I cache; 8 KB D cache
Second-level cache (on board)	512 KB
Min. memory	32 MB
Max. memory	128 MB[1]
Memory bus speed	106 MB/sec
Memory bus width	256 bits + ECC
Supported buses	SCSI-2 (7 connections: 3 internal, 7 external; 5 MB/sec total), TURBOchannel (3 connections, 90 MB/sec total)
Internal fixed storage	Two 3½-in., half-height drives
Internal removable storage	One 5¼-in. or 3½-in., half-height drive
Max. storage (integral SCSI)	9.45 GB (2 RZ26s internal, 7 RZ26s external expansion)
Graphics options[2]	HX (8-plane), PXG+ (8/24-plane), PXG Turbo+ (24-plane)
Supported input devices	mouse, tablet, keyboard, button box, dial box
Communications	sync/async, printer
Communications options	FDDI controller (DEFZA–AA), AUI Ethernet controller (PMAD–AA)
Ethernet	AUI, 10BASE-T
Audio	telephone quality input/output MJ

[1] 512 MB with 16 MB DRAM chips.

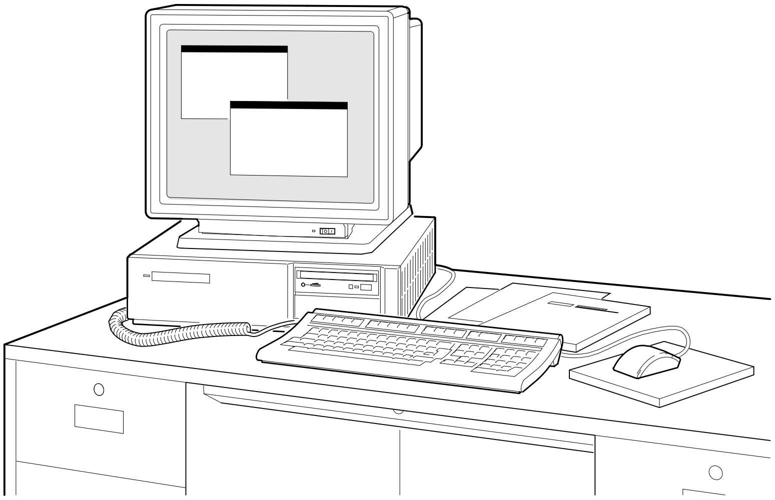
[2] Workstation only.

System Configurations: Workstation and Server

Introduction

The DEC 3000 Model 400/400S AXP system can be configured as a workstation or as a server. The workstation configuration includes a graphics board and monitor. The server version does not include these items. Figure 1–1 shows the DEC 3000 Model 400 AXP desktop version.

Figure 1–1 DEC 3000 Model 400 AXP Desktop System



MLO-009190

Workstation Configuration

As a workstation, the DEC 3000 Model 400 AXP system is a high-performance and expandable desktop system that provides a cost-effective solution for both two-dimensional and three-dimensional graphics environments. This workstation is an excellent choice for performance-oriented applications and for software development, engineering, and imaging applications.

Server Configuration

As a multiuser server, the DEC 3000 Model 400S AXP system offers good performance, storage expansion, and extensive communications—all in a compact desktop system or in a deskside unit utilizing the floor stand. The DEC 3000 Model 400S AXP server is well-suited to small businesses and distributed computing environments.

Physical Characteristics and Operating Environment

Table 1–2 details basic physical characteristics of the DEC 3000 Model 400/400S AXP.

Table 1–2 Physical Characteristics

Characteristics	
Dimensions (desktop unit)	12.7 cm x 50.8 cm x 44.5 cm (5 in. x 20 in. x 17.5 in.)
Dimensions (pedestal)	58.4 cm x 22.9 cm x 46.4 cm (23 in. x 9 in. x 18.25 in.)
Weight (desktop unit, diskless)	20.4 kg (45 lb)
Weight (pedestal, diskless)	21.8 kg (48 lb)
Power	120/240 V, single phase, 420 ac Watts
Max. Usable Power	1080–15 deskside 00 Watts
Operating Environment	Class B office
Operating Temperature	10° C to 35° C (50° F to 95° F)
Operating Humidity (noncondensing)	20% to 80%

System Architecture

DECchip 21064 CPU Chip

The DECchip 21064 CPU chip is a CMOS-4 superscalar (dual instruction issue), super-pipelined implementation of the Alpha AXP architecture.

DECchip 21064 Features

The DECchip 21064 has the following features:

- All instructions are 32 bits long, and have a regular instruction format
- Floating-point unit supports DEC and IEEE floating-point data types
- 32 integer registers, 64 bits wide
- 32 floating-point registers, 64 bits wide
- On-chip 8-KB, direct-mapped, write-through physical data cache
- On-chip 8-KB, direct-mapped, read-only virtual instruction cache
- On-chip 12-entry I-stream translation buffer
- On-chip 32-entry D-stream translation buffer
- Clock generator

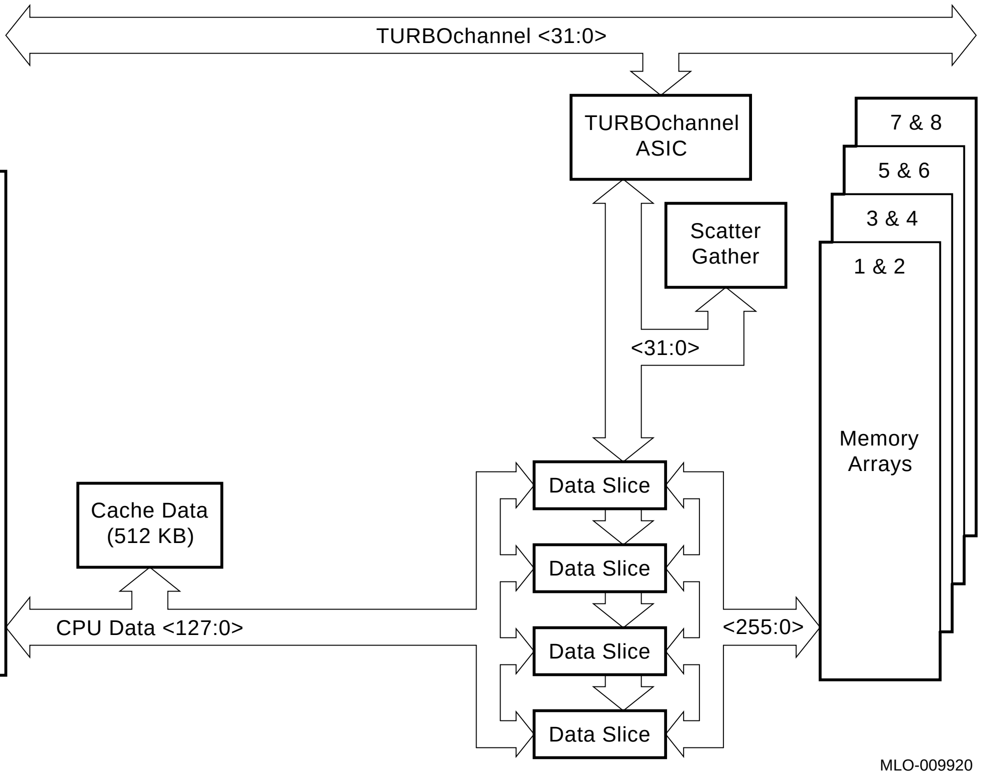
For more details on the DECchip 21064, see Appendix B.

System Logic Modules

The logic for the system consists of a CPU module, an I/O module, and four memory mother boards. To improve memory latency and bandwidth, the memory system is sliced among four memory mother boards. Thus, all four memory mother boards must be present to have an operational system.

Figure 1–2 illustrates the functional components of the system and how they are interconnected.

Figure 1–2 System Data and Address Paths



CPU Module

The CPU module contains the processor, cache, control logic, and TURBOchannel interface. It is a 12-layer board. It has connectors for the I/O module, four memory mother boards, and the power supply.

The system contains a single-chip processor running at 133 MHz. This processor is a super-scalar super-pipelined implementation of the Alpha AXP architecture. It contains two on-chip 8-KB direct-mapped caches, one for use as an I-cache, the other as a D-cache. It is packaged in a 431-pin PGA.

System Memory

The system implements a high-performance memory subsystem with ECC logic. It can be configured with up to 128 MB using 1Mx4 DRAMs, or up to 512 MB using 4Mx4 DRAMs.

The memory module (SIMM) is a four-layer board. The SIMM is populated with 1Mx4 DRAMs or 4Mx4 DRAMs on one or both sides. Therefore, each SIMM can have 2, 4, 8, or 16 MB.

The memory mother board (MMB) is an eight-layer board. Each MMB has four 100-pin SIMM connectors arranged two rows by two. The MMB connects to the CPU module through a 160-pin straddle mount connector. Memory SIMMs are added to the system in octets, two for each row in all four MMBs. Each octet must all be of the same type.

The latency for processor references which miss in both the primary and secondary caches is 180 ns for the first 16 bytes. The remaining 16 bytes are returned after an additional 45 ns. In the absence of victims, the sustained cache fill bandwidth is in excess of 114 MB/s. In the presence of victims, the sustained cache fill bandwidth drops to 80 MB/s, though the aggregate read+write bandwidth is 160 MB/s.

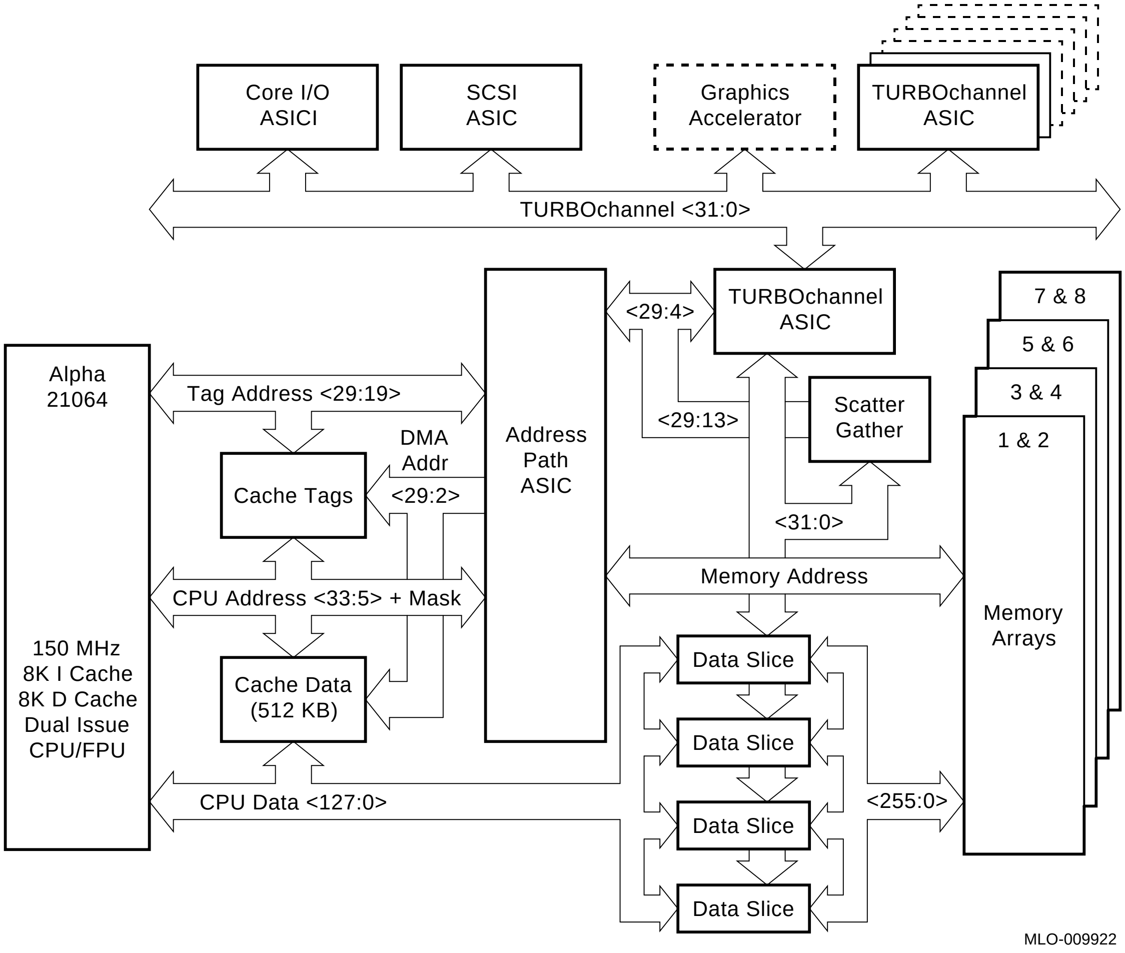
I/O Module

The I/O module contains all the I/O controllers and associated connectors. It is an eight-layer board. The I/O connectors include:

- AUI Ethernet
- 10BASE-T Ethernet
- ISDN
- Serial printer
- Keyboard/mouse
- Synchronous/asynchronous communications
- Internal and external SCSI-2
- Three TURBOchannel options
- Audio

The I/O module connects to the CPU module through a 210-pin connector. This integral I/O includes interfaces to serial lines, Ethernet, SCSI, audio input/output, and battery backed-up time-of-year (TOY). Figure 1–3 shows the I/O and graphics paths of the system.

Figure 1–3 I/O and Graphics Paths



Address Translation

Addresses generated by DMA devices in the I/O system can be translated via a scatter/gather map. This translation is optionally enabled on a per device basis. At any given time, the scatter/gather map is capable of mapping 32K pages of 8 KB each.

Serial Interface

The serial line interface supports the keyboard, mouse, printer, and communications port. The keyboard and mouse share a 15-pin D-sub connector. The printer connects to a 6-pin MMJ and is DEC-423-compliant. The communications port supports full modem control on a 25-pin D-sub connector.

Ethernet Interface

The Ethernet interface can connect to the LAN via either AUI (Attachment Unit Interface, also called thickwire) or 10BASE-T (twisted pair). The interface selection is controlled through firmware console command. The Ethernet interface is implemented using the LANCE chip.

SCSI Interface

The SCSI interface consists of two separate channels implemented using NCR 53C94 SCSI controller chips. These controller chips are interfaced to the TURBOchannel via the TCDS ASIC. The TCDS ASIC buffers data to and from the SCSI-2 controllers, providing 16-longword DMA transactions across the TURBOchannel for increased bus efficiency.

Audio Interface

An AMD 79C30 controller chip is used to provide telephone-quality audio input/output. Connectors in the rear of the box allow the user to connect the MJ handset.

Time-of-Year Chip

A battery backed-up time-of-year (TOY) chip provides an accurate time reference when the machine is powered off. The chip also contains 56 bytes of non-volatile RAM storage for firmware environment variables, such as the default boot device.

TURBOchannel Options

TURBOchannel is a high-performance, open I/O interconnect for desktop computers and servers. TURBOchannel performance is the consequence of its innovative architecture, which is specialized and optimized for the I/O function.

TURBOchannel Architecture

The TURBOchannel architecture departs from the pure bus topology that has previously been the norm for small computer interconnections. Instead, the TURBOchannel control signals have a radial point-to-point topology; a TURBOchannel-based system provides separate control lines for each of several peripheral slots. The benefits of this architectural innovation are:

- Simple, efficient protocol
- Low signal count per option module
- Low-latency transactions
- High-bandwidth DMA block transfer
- Simplified option module design

Kinds of Transactions

The TURBOchannel is a synchronous channel, transferring one 32-bit word of data or protocol overhead in each cycle. The TURBOchannel protocol implements two kinds of transactions, DMA transactions and I/O transactions, each transferring data in either direction between the option and the system.

In DMA transactions, the option transfers data directly between its buffers and system memory in blocks whose size is determined by the option, up to a maximum block size determined by the system (128 words in the DEC 3000 Model 400/400S AXP).

In I/O transactions, the option transfers single words of data between CPU registers and option memory locations. When a load/store instruction references TURBOchannel address space, the TURBOchannel control hardware generates the TURBOchannel I/O transaction signal sequences for the appropriate TURBOchannel slot. I/O transactions are typically used for controlling the peripheral device, but they can also be used for data transfer by low-cost options lacking DMA buffers and logic. (See the *TURBOchannel Technical Overview* for more detailed discussion of the TURBOchannel architecture.)

TURBOchannel Slots

There are a total of three TURBOchannel option slots on the system. Preconfigured workstation systems use one slot for a graphics option, leaving two remaining slots. These slots implement a full-speed, 25-MHz TURBOchannel. There is 128 MB of physical address space associated with each slot.

System Jumpers

There are two hardware jumpers on the DEC 3000 Model 400 /400S AXP system:

- One jumper disables the write feature of the two 256-KB console ROMs which connect to the Core I/O ASIC. This is accomplished by disabling the +12V to the FEPROMs. This jumper is labeled *ROM Update* on the DEC 3000 Model 400/400S AXP I/O module.
- One jumper selects a greater level of workstation security. This is accomplished by having the console software read the value of this jumper (set or clear) as a bit in a Core I/O register.

  If the value is clear, the workstation does not have the extra level of security and you can perform any privileged operation. If the value is set, the console asks you for a password before you can do any privileged operations. The jumper is labeled *Secure System* on the DEC 3000 Model 400/400S AXP I/O module. It is connected to an input pin of the Core I/O ASIC.

Operating Systems

Digital's Alpha AXP architecture supports multiple operating systems on DEC 3000 Model 400/400S AXP, including:

- **OpenVMS AXP.** The OpenVMS AXP operating system provides the power and functionality of the OpenVMS operating system on an Alpha AXP-based product.
- **DEC OSF/1 AXP.** DEC OSF/1 AXP is Digital's implementation of the Open Software Foundation's UNIX operating system.

This list represents the operating systems available in this first generation of the DEC 3000 AXP systems. Details on these operating systems may be found in Chapter 2. Other operating systems will be available at a later time.

Factory Installed Software

Preconfigured DEC 3000 Model 400/400S AXP systems that ship with an internal fixed disk have the operating system factory-installed on the disk.

Graphics Options

The DEC 3000 Model 400 AXP graphics options are TURBOchannel-based. TURBOchannel-based graphics provide a wide variety of Digital and third-party graphics options. Graphics support on the DEC 3000 Model 400 AXP is operating-system-dependent.

OpenVMS AXP Graphics

If your system is running the OpenVMS AXP operating system, the two-dimensional (2D) graphics option available is the HX, which allows 8-plane, accelerated 2D graphics and windowing operations.

Three-dimensional (3D) application support can be achieved by running a PHIGS (Programmer's Hierarchical Interactive Graphics System) or GKS (Graphics Kernel System).

DEC OSF/1 AXP Graphics

If your system is running the DEC OSF/1 AXP operating system, the graphics options listed in Table 1–3 are available.

Table 1–3 Graphics Capabilities for DEC OSF/1 AXP

Graphics Option	Description
HX	8-plane, accelerated 2D graphics and windowing operations.
PXG+	Configurable 3D graphics, either 8- or 24-plane, double-buffered and optional 24-bit Z-buffer.
PXGT+	24-plane, 3D graphics accelerator: 24-plane frame and double buffer, 24-bit Z-buffer, and additional 24-bit configurable buffer.

Chapter 3 provides more information about available graphics options.

System Options

Options Manual Reference

For detailed information on the DEC 3000 Model 400/400S AXP internal and external options, refer to the *DEC 3000 Model 400/400S AXP Options Guide*.

Internal Options

The DEC 3000 Model 400/400S AXP supports these internal options as follows:

- Two RZ25 and RZ26-series 3½-inch fixed disk drives, which provide up to 2 gigabytes of storage capacity.
- One 5¼-inch or one 3½-inch removable media devices (RMD). These devices include the RX26 diskette drive (3½-inch), the RRD42 compact disc drive, and the TLZ06, TZ30, and TZK10 tape drives. They provide access to software and media in a variety of industry-standard formats.
- Up to 128 megabytes of total memory.
- Up to three TURBOchannel modules, which provide a variety of TURBOchannel options, including 2D and 3D graphics options.

TURBOchannel Options

There are three TURBOchannel option slots on your system. Preconfigured OpenVMS AXP workstation systems use one slot for a graphics option, leaving two remaining slots available for additional TURBOchannel options. For information about adding TURBOchannel options to your system, see the *DEC 3000 Model 400/400S AXP Options Guide*.

External Options

You can add one or more of the following external options to your DEC 3000 Model 400/400S AXP system:

- A printer such as a color or PostScript laser printer
- A modem
- A puck or stylus tablet
- A 2- or 4-megabyte RX26 floppy disk drive
- A 4-gigabyte TLZ06, 4-mm DAT (digital audio tape) drive

- A PMTCE–AA TURBOchannel extender box
- A BA350–SA expansion box, which can hold the following devices:
  - RZ25 and RZ26 disk drives
  - TLZ06 tape drive
- A TZK10 one-quarter-inch cartridge tape drive

2

System Software

Introduction

Digital's Alpha AXP architecture allows use of multiple operating systems on your DEC 3000 Model 400/400S AXP system. The operating system is the core software installed on the system, which allows you to install and run applications. The two operating systems are OpenVMS AXP and DEC OSF/1 AXP.

In This Chapter

This chapter contains the following sections on the various software available for your DEC 3000 Model 400/400S AXP system:

- Operating Systems
- Software Development Languages and Compiler Technology
- Windowing Software
- Graphics Programming Interfaces
- Networking Software
- Application Software

Operating Systems

The OpenVMS AXP Operating System

The OpenVMS AXP operating system promotes ease-of-use and improved programming productivity, and facilitates system management. OpenVMS AXP offers a combination of commercial features and open system benefits:

- Integrated networking
- System security
- Distributed computing and multiprocessing
- Windowing capabilities

In addition, OpenVMS AXP supports a large number of industry standards to facilitate application portability and interoperability.

The DEC OSF/1 AXP Operating System

The DEC OSF/1 AXP operating system is Digital's implementation of Open Software Foundation (OSF) operating system components and DECwindows/Motif graphical user interface and programming environment.

Part of the Open Software Foundation's charter is to provide an interface for developing portable applications that run on a variety of hardware platforms. DEC OSF/1 AXP is compliant with the OSF Application Environment Specification (AES), which specifies the interface to support these portable applications.

Additionally, the DEC OSF/1 AXP operating system complies with standards and industry specifications, including:

- FIPS 151-1
- POSIX (IEEE STD. 1003.1-1988)
- XPG3 BASE branding
- XTI
- AT&T System V Interface Definition (SVID) Issue 2 (Base System and Kernel Extensions).

The DEC OSF/1 AXP operating system is an advanced kernel architecture based on Carnegie Mellon University's Mach V2.5 kernel design with components from Berkeley Software Distribution 4.3 (BSD) and other sources. DEC OSF/1 AXP provides numerous features to assist application programmers in developing applications that use shared libraries, multithread support, and memory mapped files.

To ensure a high level of binary compatibility with ULTRIX, the DEC OSF/1 AXP operating system is compatible with the Berkeley 4.3 programming interfaces.

Factory-Installed Software

If your DEC 3000 Model 400/400S AXP system contains an internal fixed disk drive, the operating system is factory installed on the disk. You can start your system using the OpenVMS AXP Factory Installed Software (FIS) procedure, or the DEC OSF/1 AXP Factory Installed Software Procedure.

Software Development Languages and Compiler Technology

Software Development Languages

Digital provides a full set of language compilers for software development. Compilers developed by Digital include:

- DEC Ada
- DEC C
- DEC COBOL
- DEC FORTRAN
- DEC Pascal

Compiler Technology

The Digital-developed compilers conform to applicable language standards where they exist.

Modern compilers are typically made up of two distinct parts:

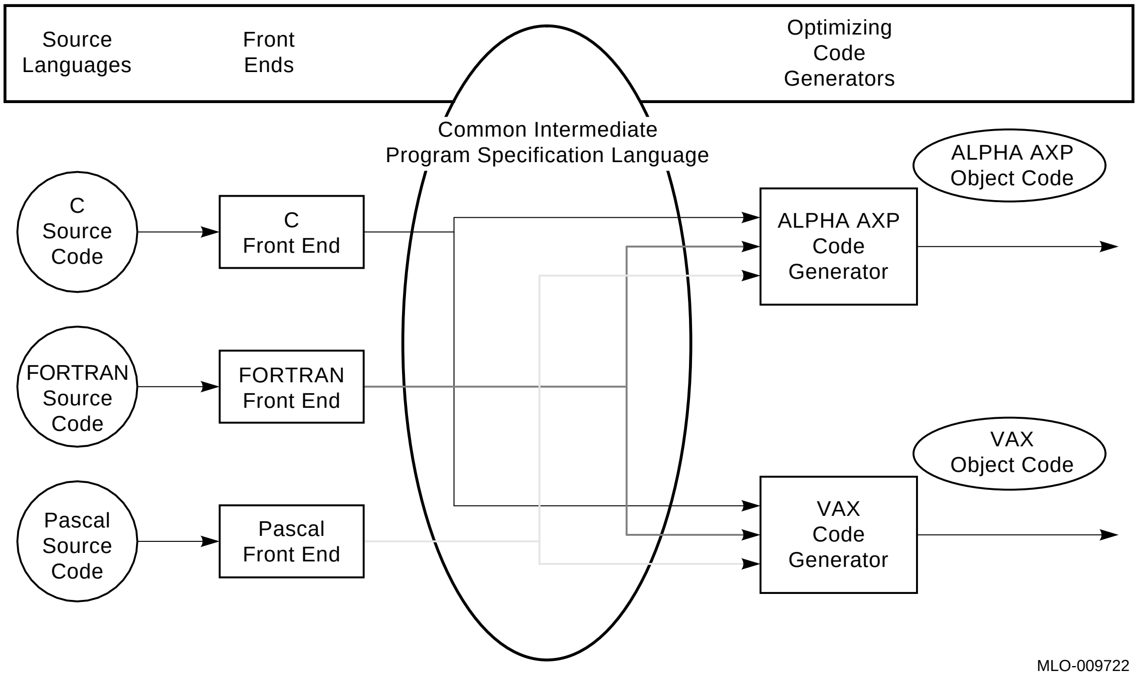
- A front end, which accepts the source code as input and parses the source language, but is independent of the target processor.
- A back end, which generates optimized machine code for a particular target processor, but is independent of the source language.

Compiler Architecture

The output of the front end, which is the input to the back end, is in an intermediate program specification language, which is independent of both the source language and the target machine architecture. A common intermediate language may be used for several different source languages and machine architectures. In this way, the benefits of optimizing the back end for a particular machine architecture can be enjoyed in several languages. Each language front end is used on several machine architectures, ensuring a common language definition and application portability.

Figure 2–1 illustrates compiler architecture.

Figure 2–1 Compiler Architecture



Software Engineering Tools—DEC FUSE

DEC FUSE is an integrated, graphically-oriented, multilanguage software engineering environment. The supported languages include C, FORTRAN, Pascal, and C++. FUSE is based on UNIX commands and utilities traditionally associated with programmer productivity and provides a set of OSF/Motif-based graphical tools. It makes extensive use of dynamic graphical capabilities to provide visual representations of a program's structure.

DEC FUSE includes its own editor, but also supports the GNU, Emacs, and vi editors. DEC FUSE includes:

- A debugger
- A program builder
- A call graph browser
- A profiler
- A cross-referencer

- A C++ class browser
- A code management system
- Extended annotation support

Another advanced feature is support for distributed build, that is, using several workstations on a network in parallel for compiling and linking large builds.

DEC FUSE EnCASE

DEC FUSE EnCASE is a facility that allows developers and end users to integrate additional development tools into DEC FUSE. The tools added to DEC FUSE by means of EnCASE can be third-party software development products or home-grown UNIX utilities.

You can link tools added to the DEC FUSE environment with existing DEC FUSE tools or with other newly added tools to enhance existing work procedures. DEC FUSE EnCASE uses object-oriented messaging principles to manage the integration of tools.

Windowing Software

Window Systems

Virtually all workstation and personal computer users are now familiar with window systems that allow you to work with several applications at the same time. Each window serves as a virtual graphics terminal to receive the output of an application or utility. Most window systems provide the convenience of transferring data between applications by interactive "cutting and pasting" in the corresponding windows.

The window system must be common to all the concurrent applications, and so forms part of the system software, placed between the operating system and the applications.

The Window Manager

A central software element is the window manager, which resolves contention for display area among the different applications and gives the interactive user a measure of control over the window layout. The window manager also associates operation of a single keyboard or pointing device with input for particular applications, identified with windows.

Graphical User Interfaces (GUI)

Window systems also serve as platforms for graphical user interfaces (GUI), in which interactive users enter data by directly manipulating screen objects—buttons, menus, scroll bars, dialog boxes—by pointing and clicking with mouse and cursor. If a number of applications are constructed using the same GUI, that is, a common set of interactive screen objects and a common style of manipulating them, they are easier to learn and use.

X Window System

The X Window System (also called X or X11) is an industry-standard networked window system developed at the Massachusetts Institute of Technology (MIT) with financial and technical support by an industry consortium of which Digital is a leading member. It is a networked window system constructed on a client/server model.

The server is the software that controls the interactive graphics workstation—the display and the associated interactive input devices.

The client is an application that carries on interactive I/O through requests to the server. The client may run on the same processor as the server, or it may reside on a remote host and communicate with the server over a network.

The X Protocol

The X client and server communicate through a precise protocol, the X Protocol, which is designed for communicating simple graphics and window control information. If the client and server are on different hosts, the X Protocol is implemented over the underlying network protocol. The protocol is independent of the system architecture and operating system on both ends. Therefore, the X Protocol supports interoperability for transparent network graphics among diverse systems.

X Software Architecture

The X software architecture has two parts, the server and the client-side support. The graphical device dependence is all in the server; the client side is device-independent. The basic application programming interface on the client side is Xlib, a library of almost 400 routines for 2D graphical primitives, window management functions, and input event management. Xlib is the lowest level access that an application programmer has to the X Window System.

Xtk Tool Kit

The standard X Window System also provides a higher-level programming interface, the tool kit Xtk, which provides the programmer with tools for constructing and managing interactive screen objects, such as buttons and sliders, called widgets in X.

Xtk comes with a set of widgets, called the intrinsics, but it defines no policy on look-and-feel standards. It can be used to implement graphical user interfaces with particular look and feel. Figure 2–2 illustrates the architecture of the X Window System.

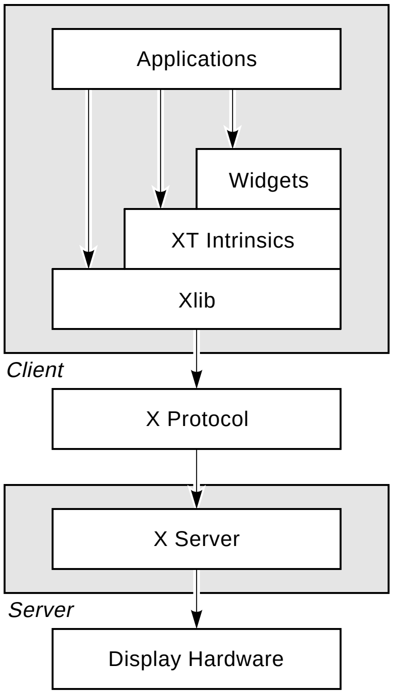
DECwindows

DECwindows is Digital's enhanced implementation of the X Window system. Some of Digital's enhancements to the standard system are:

- Faster algorithms
- International keyboard support
- Better security features
- Additional language bindings

On the DEC 3000 Model 400 AXP workstation, DECwindows is an exceptionally fast and robust implementation of X. The version of DECwindows shipped with the DEC OSF/1 AXP and OpenVMS AXP operating systems embody the X11R5 version of the X Window System.

Figure 2–2 X Window System Architecture



MLO-009723

OSF/Motif

OSF/Motif is an industry-standard graphical user interface adopted by the Open Software Foundation and based largely on XUI, the DECwindows GUI originally developed by Digital. OSF/Motif includes:

- A tool kit for constructing interactive widgets
- A user-interface language (UIL) for building applications with Motif user interfaces
- A style guide

Through adherence to the style guide, the application programmer can be sure that the application will have a standard look and feel, which will make it easy to learn and use by a user familiar with other Motif-based applications.

Extensions to the X Protocol

The standard X Protocol does not include semantics for addressing all conceivable functionality that a graphics server may provide. When new graphical functionality is added to the server side, the protocol and the server code must be extended to use the new functionality in the context of the X Window System. The design of the X Window System anticipated the need for such extensions and provides a standard way of making them.

PEX

The original X Protocol is limited to 2D graphics. However, some graphics subsystems, such as the PXG family of options, provide hardware and firmware for 3D graphics processing. The graphics subsystem is clearly on the server side of the X Protocol.

To use 3D hardware for making pictures in X windows, the X Consortium has defined PEX, a standard extension to the X Protocol. PEX stands for PHIGS Extension to X to reflect the fact that the semantics of PEX are based largely on PHIGS, the ANSI/ISO standard 3D graphics application programming interface (API). However, PEX is just a protocol for communicating 3D graphics information between client applications and 3D graphics servers; it can also be used with other graphics APIs.

Implementation of a PEX server on 3D graphics hardware makes the hardware available for network-transparent, intervendor distributed applications. Digital has been a leading member of the PEX development and standardization effort, and produced the industry's first PEX-based workstation in 1989. The Digital PEX server is bundled with the 3D acceleration options.

Display PostScript

Digital has developed a second extension of the X Protocol to support the Display PostScript graphics API. This extension has been adopted as a standard by the X Consortium.

Graphics Programming Interfaces

The Xlib API

The Xlib API provides drawing primitives only for simple 2D figures such as lines, arcs, polygons, and text. To support development of both 2D and 3D applications, programmers need much richer and higher-level APIs for defining, manipulating, editing, displaying, and storing complex graphical objects.

Additional APIs

Digital offers several additional higher-level graphics APIs both for 3D graphics and for 2D applications that need more graphics primitives and richer graphical data structuring capabilities. These APIs include:

- Display PostScript
- ANSI/ISO PHIGS
- DEC PHIGS
- PEXlib
- GKS
- DEC GKS–3D
- IrisGL
- OpenGL

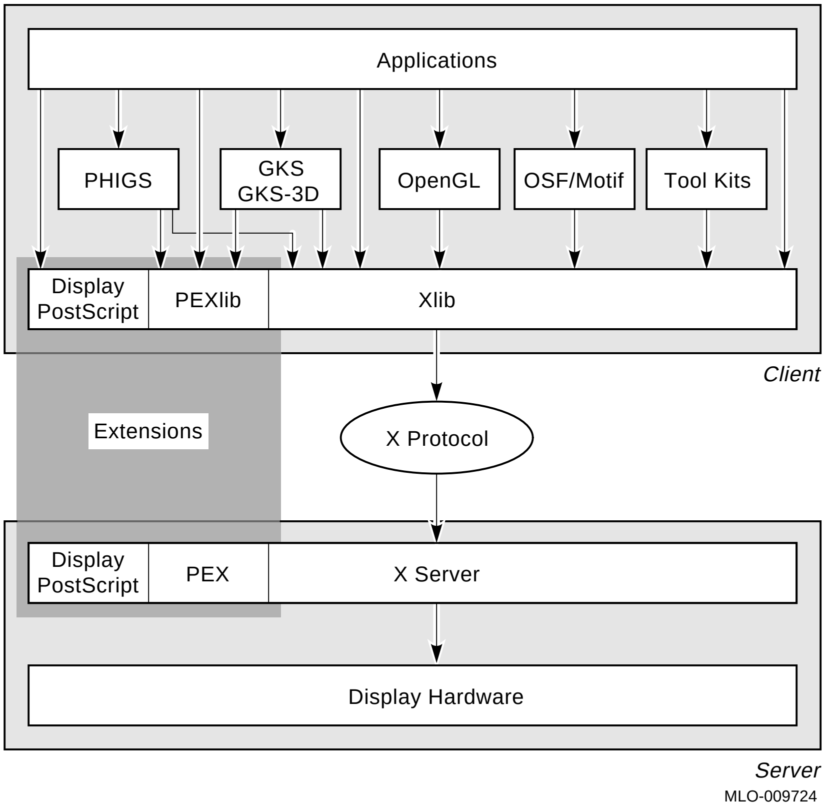
Graphics Programming Environmental Architecture

For display I/O, all of the higher level APIs work through the X Window System and its extensions. Figure 2–3 depicts the architecture of the graphics programming environment supported on the DEC 3000 Model 400 AXP.

Display PostScript

PostScript is a page-description language, developed by Adobe Systems Inc., which has become the industry-standard means of access to the advanced graphical and typographical capabilities of laser printers. Display PostScript is a language for programming interactive raster displays, adapted from PostScript. Digital's implementation of the Display PostScript interpreter is licensed directly from Adobe and is complete and fully compatible.

Figure 2–3 Graphics Programming Environment Architecture



Digital has implemented the Display PostScript interpreter as part of the X Server and extended the X Protocol to access it. Server-side implementation of the Display PostScript interpreter has certain advantages concerning performance and compatibility. But, just as PostScript is frequently implemented by a PostScript engine within a laser printer, it is natural to anticipate the advent of Display PostScript acceleration hardware in workstation displays, which will then require server-side implementation and protocol extension. By taking the lead in developing server-side implementation and standardization of the protocol extension, Digital ensures future compatibility of Display PostScript applications written today.

ANSI/ISO PHIGS

PHIGS (Programmer's Hierarchical Interactive Graphics System) is the preferred ANSI/ISO standard API for 3D device-independent graphics. PHIGS is a structure-oriented (as opposed to an immediate-mode) API. That is, through calls to the API library functions, the application builds data structures representing the picture to be drawn, structures which are retained and stored by the PHIGS runtime system and which can be redisplayed through subsequent calls. PHIGS structures admit hierarchical organization and structure editing, two important features that help distinguish PHIGS from the GKS and GKS–3D standards.

PHIGS PLUS

PHIGS PLUS is a proposed extension to the standard, which adds advanced surface primitives and lighting-and-shading rendering capability to the original PHIGS system. At this time, the PHIGS PLUS extension is close to final adoption; after adoption, the term PHIGS includes the PHIGS PLUS extension.

DEC PHIGS

DEC PHIGS is a very complete implementation of PHIGS, including the proposed PHIGS PLUS functionality and several further extensions. The advanced PHIGS PLUS features of DEC PHIGS include:

- Lighting
- Shading
- Depth cueing
- Non-uniform rational B-spline (NURBS) curves and surfaces.

The further extensions include circle and arc primitives, an immediate-mode feature, and certain other extensions that are not part of the standard but advocated and informally encouraged by a group of CAD/CAM software companies. DEC PHIGS offers C and FORTRAN language bindings and supports PHIGS programming in several other languages.

DEC PHIGS Integration

DEC PHIGS is very well integrated with DECwindows and provides the programmer with several choices for managing the interaction between PHIGS and X and for mixing PHIGS calls and Xlib calls. Further, DEC PHIGS can generate PEX protocol requests to make use of hardware graphics acceleration.

Alternatively, PHIGS can per form 3D graphics processing in software to generate drawing requests through the ordinary X Protocol. A few of the more compute-intensive capabilities of PHIGS PLUS are provided only when supported by 3D hardware acceleration. Still, the RISC processing power available in the system enables practical 3D applications through DEC PHIGS without 3D acceleration, so long as such features as smooth shading and hardware Z buffering are not needed.

DEC PHIGS Runtime Support

The PHIGS runtime license is bundled with the configurations of the DEC 3000 Model 400 AXP workstation, which include either the PXG+ or PXG Turbo+ graphics accelerators.

PEXlib

PEXlib is a 3D graphics API that allows programmers to access the PEX protocol at a slightly lower level than PHIGS, analogous to the Xlib API and the X Protocol. While PEXlib is not a formal standard, the latest version, V5.1, is widely accepted by members of the X Consortium as a defacto standard API for accessing the PEX protocol. PEXlib provides a lower level interface than PHIGS to allow programmers closer access to the features of PEX capable servers.

GKS

GKS (Graphics Kernel System) is an ISO standard for a 2D graphics API, which was originally developed in Europe and predates PHIGS.

DEC GKS–3D

GKS (Graphics Kernel System) is an ISO standard for a 2D graphics API, which was originally developed in Europe and predates PHIGS. GKS–3D is an extension of the standard to include 3D geometry. However, GKS–3D does not include the advanced primitives and lighting-based features of PHIGS PLUS.

The Digital implementation, DEC GKS, supports full-level 2D functionality of the standard, and is upward-compatible with the DEC GKS implementation. GKS and GKS–3D are optional products for the DEC 3000 AXP family.

While GKS and GKS–3D have a kind of retained structures (called segments), they lack the hierarchical structuring and structure-editing features of PHIGS, and the lighting and shading capabilities of PHIGS PLUS. Strictly 2D graphics applications can sometimes enjoy better performance when implemented in GKS than in PHIGS, because PHIGS is intrinsically 3D internally and therefore carries some 3D overhead even when used for 2D applications.

Iris GL

Iris GL has been popular with application developers in such markets as scientific visualization, molecular modelling, and arts/entertainment because of its power and advanced graphics feature set. It is anticipated that many existing Iris GL applications will be ported to OpenGL on Digital's Alpha AXP workstations.

OpenGL

OpenGL is an open, portable, windowing system independent 3D graphics programming environment which has its heritage in the Iris GL programming environment developed by Silicon Graphics. Because OpenGL is window system independent, it can be easily integrated with various window systems. Currently, OpenGL has been ported to the X Window System. Microsoft has announced plans to integrate OpenGL with future versions of Windows NT.

Digital was one of the first companies to license OpenGL from Silicon Graphics and was one of the companies that contributed to the OpenGL V1.0 specification, which was released to OpenGL licensees in July of 1992. Digital is also a founding member of the OpenGL Architecture Review Board, a group of OpenGL licensees who are responsible for determining the future direction of OpenGL based on user and vendor feedback.

Digital plans to implement OpenGL as a key piece of the Alpha AXP workstation graphics environment on both DEC OSF/1 AXP and OpenVMS AXP.

Networking Software

Introduction to Networking Software

Connectivity, interoperability, open networking, and distributed applications have become normal and expected features for workstation installations in industrial, commercial, and academic environments. DEC 3000 AXP systems provide the full range of options for connectivity, both in Digital proprietary networking environments and in multivendor open-networking situations.

Ethernet support through the I/O subsystem is standard on the DEC 3000 Model 400/400S AXP. In addition, connections to Fiber Distributed Data Interconnect (FDDI) networks are available through TURBOchannel options. These networking possibilities are all supported by available software.

TCP/IP

The TCP/IP network protocol is a set of software communications protocols, which were developed in government and defense contractor installations, and are now widely used in many networking environments. TCP/IP is the protocol for the Ethernet networking hardware. TCP/IP has become a *de facto* standard in UNIX operating environments. In addition, Alpha AXP systems can use TCP/IP to communicate with OpenVMS AXP systems using the FUSION TCP/IP or Digital UCX products.

Network File System

The DEC OSF/1 AXP operating system supports the Network File System (NFS), an industry-standard facility that governs file sharing among networked systems. NFS allows a workstation user to access disk storage on a remote file server as if it were local, and provides a network administrative service called Yellow Pages. NFS comes standard with the DEC OSF/1 AXP operating system on all DEC 3000 AXP systems.

DECnet/OSI

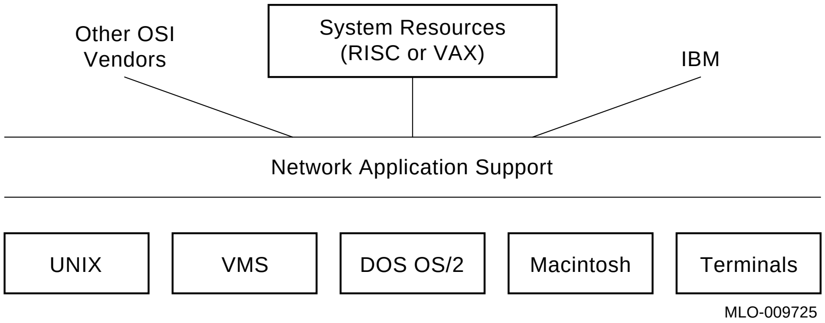
DECnet is Digital Equipment Corporation's optional networking product that implements the Open Systems Interconnect (OSI) model. DECnet offers task-to-task communications, establishing logical connection to other DECnet nodes in the network, remote file transfer, mail, coexistence with the Internet protocols on TCP/IP-based machines, and network-wide resource sharing and management.

Network Application Support

Network Application Support (NAS) is a backbone set of application services which allows integration of multiple applications in distributed, multivendor environments. NAS applications can, for example, combine graphics from an Apple Macintosh, a Lotus spreadsheet from a DOS PC, a drawing from an Alpha AXP workstation, data from an IBM mainframe, and a scanned image from a VAXstation all into a single report that can be sent electronically to others anywhere on the network.

Figure 2–4 illustrates NAS architecture.

Figure 2–4 NAS Architecture



NAS Components

NAS consists of well-defined industry-standard programming interfaces, tool kits, and products that help developers build applications that are well integrated and easily portable across diverse systems. Some of the components of NAS are:

- Remote Procedure Calls (RPC), which allow remote execution over loosely coupled CPUs in a heterogeneous network. DEC RPC is based on the Network Computing System (NCS) from HP/Apollo and is bundled with the DEC OSF/1 AXP operating system.
- Personal Computing Systems Architecture (PCSA), Digital Equipment Corporation's PC integration software that combines DOS or OS/2-based PCs into a PC Local Area Network (LAN) or corporate computing network.
- IBM 3270 terminal support and access to IBM environments through the DECnet/SNA family of IBM interconnect products.

- DECwindows, Digital's implementation of the X Window System.
- Compound Document Architecture (CDA), a method for creating, storing, and exchanging files that contain a number of integrated components including text, synthetic graphics, and scanned images.

Application Software

Introduction to Application Software

The DEC 3000 Model 400/400S AXP system is binary-compatible with the rest of the Alpha AXP family and DEC 3000 AXP workstation family and so enjoys the benefit of the large repertoire of application software written for or ported to the Alpha AXP architecture.

This repertoire includes:

- CAD (Computer-Aided Design) packages
- Desktop productivity packages
- SoftPC

CAD Packages

Many of the major third-party CAD packages and visualization tools have already been ported to the DEC 3000 Model 400/400S AXP system and the accelerated graphics options; these will all run on the DEC 3000 Model 400/400S AXP systems and the graphics options with substantially improved performance.

Desktop Productivity Packages

The DEC 3000 Model 400/400S AXP system user can run a wide range of the popular desktop productivity packages that have been the mainstay of the personal computer industry. Many of these packages have already been ported to the DEC 3000 AXP family.

DEC SoftPC

Digital provides DEC SoftPC, a software emulator of the IBM PC/AT class of personal computer, which allows use of the object code of most MS–DOS applications.

3

Graphics Options

Introduction

Digital offers a wide range of TURBOchannel-based graphics options for the DEC 3000 AXP models. All of the graphics options are TURBOchannel options, taking one to three TURBOchannel option slots.

In This Chapter

This chapter contains the following sections describing the various graphics options for your DEC 3000 Model 400 AXP workstation:

- Graphics Options Overview
- HX Graphics Option
- Accelerated 3D Graphics Options
- Graphics Performance Characteristics
- Multiscreen Support

Graphics Options Overview

Graphics Options Features

Table 3–1 and Table 3–2 list the relevant characteristics of the TURBOchannel-based graphics options.

Table 3–1 Graphics Options: Summary of Features, Part 1

Option	Resolution	Depth	Refresh Frequency
HX	1280x1024 1024x864 1024x768	8	72/66/60 Hz
PXG+	1280x1024	8/24+DB+24Z	72 Hz
PXG Turbo+	1280x1024	24+24+24+24	72 Hz

Table 3–2 Graphics Options: Summary of Features, Part 2

Option	TURBOchannel Slots	Acceleration Notes
HX	1	X primitives; Smart Frame Buffer
PXG+	2	Scan conversion; optional Z-buffer (for 8-plane versions) and 3D geometry
PXG Turbo+	3	Scan conversion; standard Z-buffer and 3D geometry

Raster Graphics Systems: Common Features

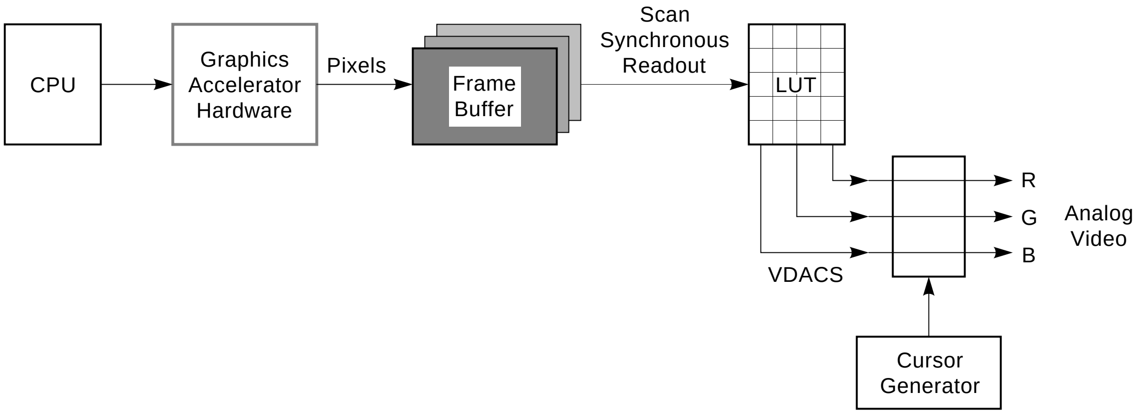
Raster graphics systems are built around a frame buffer. The frame buffer is a memory containing a pixel-level description of the displayed image that refreshes the raster display.

Refreshing the display involves clocking out the frame-buffer data synchronously with the sweep cycle of a cathode ray tube (CRT) monitor, using the pixel values to produce the analog signals that drive the color guns of the CRT. Frequently, the frame buffer is implemented in a special kind of dual-ported RAM, called video RAM (VRAM). A VRAM allows the CPU or graphics processing hardware to draw in the picture by writing to the frame buffer without interfering with the refresh cycle.

A Generic Raster Graphics System

Figure 3–1 depicts a generic raster graphics system. The box labeled "Graphics Accelerator Hardware" may be absent or may take any of several forms. The HX option and the PXG family options illustrate two different graphics acceleration architectures.

Figure 3–1 Generic Raster Graphics System



MLO-009726

Depth of Frame Buffer

The number of bits per pixel in the frame buffer is called its depth. If the depth is d, then the maximum number of different colors or intensities that the display can simultaneously contain is 2 d: specifically, 2 colors when d equals 1, and 256 colors when d equals 8.

About 16.7 million different colors are possible when d equals 24, but the maximum number of simultaneous colors is, of course, limited by the number of pixels, and this is somewhat less than 16.7 million for all of the graphics options.

Color Lookup Tables

On most 8-bit color systems and many 24-bit color systems, the pixel values are used as indices into a color lookup table whose entries determine the red, green, and blue (RGB) components of the color actually displayed. Typically, the lookup table entries each contain 24 bits, 8 bits for each of the red, green, and blue components.

The color lookup table entries are themselves under software control. Thus, when d equals 8, the lookup table has 256 24-bit entries, so the 256 different colors that can be simultaneously displayed at any moment are selected from a palette of 16.7 million colors. Such remapping of color components is frequently referred to as pseudo-color.

True-Color

A 24-bit system with no color lookup table is called a true-color system. The 24 bits at each pixel are partitioned into three 8-bit fields, which directly determine the RGB values for the pixel. In 24-bit systems with color lookup tables (such as the PXG family options), there are three 256-entry x 8-bit lookup tables, which allow remapping of each of the color components independently.

Bit Planes

Many frame buffers have additional bit planes (bits per pixel), for purposes other than encoding the displayed color information—for double buffering, for Z-buffering, and for pixmap storage.

Refresh Frequency

Video digital-to-analog converters (VDAC) convert the digital RGB values from the lookup table to the analog signals that drive CRT guns. One relevant parameter characterizing the raster system is the refresh frequency—the number of times per second that the entire CRT screen is scanned. A display refresh frequency that is too low will produce annoying flicker. The bandwidth of the VDACs is a major limiting factor for the maximum refresh frequency. All the TURBOchannel-based options discussed in this chapter provide 72-Hz refresh frequency.

Frequently, the color lookup table and VDAC circuitry are integrated on a single chip. Hardware cursor generation may also be included on the same circuit or on a different circuit. Mixing the hardware cursor signal with the image signal in the analog domain allows the cursor to move about the display without having to disturb the underlying image values in the frame buffer.

How Options Differ

The frame buffer and VDACs are elements common to all raster graphics systems. The options differ in the frame buffer parameters—its resolution (rows and columns of pixels), depth (bits per pixel), and off-screen planes (double buffer, Z-buffer). Options differ in details of the VDAC/lookup table/hardware cursor implementation and the refresh frequency.

Graphics Acceleration

The DEC 3000 Model 400 AXP graphics options also differ in graphics acceleration—the graphics processing implemented in hardware or firmware on the option cards.

The HX option provides acceleration for a set of 2D line drawing, polygon filling, and image copying operations, which are heavily used in X11-based applications.

The PXG+ and PXG Turbo+ options are based on a common architecture providing hardware acceleration both for the 2D scan conversion and for the 3D geometry processing—the most compute-intensive parts of the 3D rendering pipeline.

Graphics Acceleration Hardware

Graphics acceleration implemented in hardware or in dedicated high-performance processors running firmware can speed up applications in three ways:

- The graphics acceleration hardware is specialized to perform a particular graphics processing task and so can perform it faster than the general-purpose CPU working with system memory.
- Graphics acceleration hardware usually reduces the amount of data that has to be communicated over the I/O channel between the CPU and the graphics subsystem. For example, if the hardware does line drawing, only the endpoints of the line need be sent to the graphics acceleration hardware, whereas if the CPU is doing the line drawing, all the pixels in the line have to be sent.
- The graphics acceleration hardware frees the CPU from routine graphics processing to do other work on the application simultaneously with the offloaded graphics processing. The following sections present more details on the graphics acceleration features.

HX Graphics Option

Introduction to the HX Option

The HX graphics option takes one TURBOchannel slot and has 8-bit planes with 1280x1024 resolution. On the video output side, it also uses the Bt459 RAMDAC/Cursor chip, including a 256-entry, 24-bit color lookup table. The primary refresh frequency is 72 Hz. The HX option is available in several versions, each having a secondary mode with lower resolution and/or lower refresh frequency, so that a wide range of monitor types can be supported.

Smart Frame Buffer

The unique feature of the HX graphics option is the Smart Frame Buffer (SFB) custom ASIC, which provides acceleration for 2D line drawing, stippled polygon filling, and pixel copy and boolean operations (raster operations).

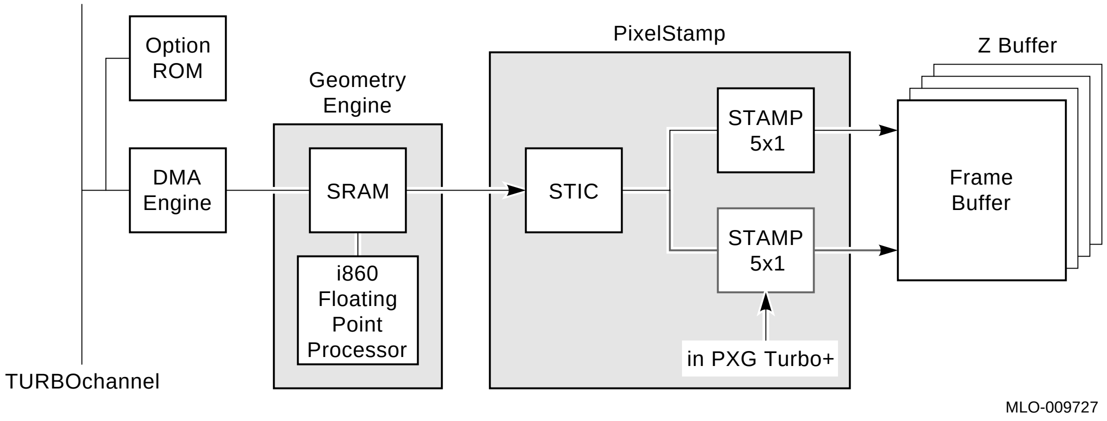
The SFB implements a fast-integer Bresenham algorithm for the line drawing. The polygon fill and pixel copy operations use considerable parallelism, processing 32 pixels at a time. All operations allow any of 16 boolean operations to be applied on the source and destination pixel. Such operations have been called raster-ops. It is worth noting that the particular graphical functions that the HX accelerates are all very heavily used by the DECwindows and Motif window managers, as well as many DECwindows applications. The fundamental raster-op is simply copying a pixmap from system memory to the frame buffer.

Accelerated 3D Graphics Options

Common Graphics Acceleration Architecture

The TURBOchannel graphics options offering 3D acceleration are the PXG+ and the PXG Turbo+, which are different implementations of a common graphics acceleration architecture, depicted in Figure 3–2. The graphics acceleration is applied to two compute-intensive stages of the 3D rendering pipeline. This section discusses the rendering pipeline in general and the common architecture of these three options, delineating the parameters that distinguish these options.

Figure 3–2 Graphics Acceleration Architecture



The Rendering Pipeline

The rendering pipeline is the sequence of processing steps for producing the pixel-level description of the image, which is to be written into the frame buffer, from the application-level representation of the 3D scene to be depicted. This sequence is called a pipeline because it consists of steps to be applied identically to a stream of graphical elements, the same processing to each element.

Application Data Structures

In 3D acceleration options, the rendering pipeline is implemented as a hardware pipeline, that is, as a sequence of processing stages, each performing one of the rendering functions, operating simultaneously in assembly-line fashion on the incoming stream of elements.

Applications represent the 3D scene by data structures whose elements are the primitives and attributes defined by the application programming interface (API). The primitives are 3D geometric objects, typically lines and polygonal areas in an abstract 3D model space, and more complex objects made up of such lines and polygons.

Typical attributes are line style, surface color, light-source color, and surface-reflectance properties. For smooth shading and other advanced features, the application must supply further data with each primitive, such as vertex colors or vertex normal vectors.

Steps in the Pipeline

The rendering pipeline begins with traversal of the application data structures defining the scene. The traversal is performed either by the API run-time software (in a structure-oriented API), or by the application itself (with an immediate-mode API). In either case, the API run-time software running in the client CPU generates PEX requests and sends them to the PEX server running in the CPU to which the graphics acceleration option is attached. The PEX requests themselves are in terms of geometric primitives, very similar to those of the API, so generation of PEX requests amounts essentially to reformatting the data structures.

The PEX server converts the high-level 3D PEX requests to 3D request packets acceptable to the 3D acceleration hardware on the graphics option. PEX converts this by breaking complex primitives into simpler components, such as line segments, convex polygons, and triangle strips, still in the 3D model space of the application.

Hardware Pipeline

The first stage of the hardware pipeline on the 3D acceleration graphics option is the direct memory access (DMA) engine, which fetches from a first in-first out (FIFO) in system memory the 3D request packets generated by the PEX server and stores them in a FIFO in the fast SRAM memory on the option.

Geometry Processing

The first computational stage on the graphics acceleration option consists of the geometry processor, implemented by an Intel i860 high-performance, 64-bit floating-point processor, running firmware stored in the fast SRAM. The function of geometry processing is to map the 3D lines and polygons to the plane of the display surface.

This processing involves a projection from 3D space to 2D space, as well as a transformation of coordinates from the device-independent 3D coordinates suitable for the application to the device-specific 2D coordinates that locate elements of the image on the display surface. Geometry processing consists mostly of matrix multiplication and requires floating-point arithmetic. Geometry processing may also include clipping, which is the elimination of primitives and parts of primitives that lie outside of a defined region of interest.

Processing Results

The results of geometry processing are stored in another FIFO in the option RAM. These results include the description of 2D lines and triangles in device coordinates. The results of geometry processing may also include some per-vertex or per-polygon data, such as colors or depth values, which carry information about the original 3D arrangement of the scene. These residual 3D data are needed for certain advanced rendering features such as smooth shading, depth cueing, or Z-buffering, which are computed using interpolation at the scan conversion-stage.

Scan Conversion

Scan conversion is the process of determining which pixels must be set and to what values they must be set in order to draw a given 2D primitive (e.g., lines and polygons). In the PXG family 3D options, scan conversion is performed by the PixelStamp, which computes the pixel addresses and pixel values for drawing lines and triangles to the frame buffer.

Although the PixelStamp computations are in a 2D image space, they include the interpolation in 2D space of certain quantities carrying information about the original 3D scene, such as colors or depth values. These interpolations are needed for some rendering features, such as smooth shading or Z-buffer hidden-surface removal. The PixelStamp design also supports antialiasing of lines.

The PixelStamp scan conversion works by computing the linear equations defining a given primitive for each pixel in a set of pixels determined to have a chance of intersecting the primitive. Therefore, the architecture is scalable; that is, it can be implemented to perform the computations for a number of pixels simultaneously in parallel.

PixelStamp Implementation

In the PXG family graphics options, the PixelStamp is implemented in two custom ASICs:

- The Stamp, which is essentially the linear solver and interpolator. The Stamp operates simultaneously on five consecutive pixels in a scan line. Stamp chips can be ganged to operate in parallel on several consecutive scan lines, as in the PXG Turbo+.
- The STIC (STamp Interface Chip), which provides its interface with the incoming 2D request packets left in the SRAM by the geometry engine. The STIC chip performs certain set-up operations for the Stamp and also is the source of video timing signals for the VDACs.

PXG+ Option

The PXG+ option has 1280x1024 frame-buffer resolution, and implements the 3D acceleration architecture described earlier. The PXG+ uses faster versions of both the i860 and STIC/Stamp chips than did the PXG. The i860 runs at 44 MHz on the PXG+. The STIC/Stamp in the PXG+ are 33 percent faster than in the PXG. The PXG+ option contains 128 KB of fast SRAM.

The PXG+ is available in 8-plane and 24-plane versions, each with a full-screen double buffer. The 8-plane version can be upgraded to 24 planes. A 24-plane Z-buffer is standard on the 24-plane version and optional on the 8-plane version. Both versions are double-width TURBOchannel options.

Both the 24-plane version and 8-plane versions can be color mapped through the table in the Bt459 RAMDAC/Cursor chip. The 24-plane version contains three Bt459s, to have three 256-entry x 24-bit lookup tables, one for each of the RGB components of the pixel value.

PXG Turbo+ Option

The PXG Turbo+ uses the same i860 and STIC/Stamp parts as the PXG+, but contains two Stamp chips and 256 KB of SRAM. The two Stamps work in parallel on two consecutive scan lines, each in parallel on five consecutive pixels.

The PXG Turbo+ option is a triple-width TURBOchannel option, and is available with 96 raster planes only:

- 24 image planes
- 24-plane double buffer
- 24-plane Z-buffer
- 24-plane buffer for off-screen storage of pixmaps

Monitors for PXG+ and PXG Turbo+

Both options can be used with either the 40.6-centimeter (16-inch) VRT16 or the 48.3-centimeter (19-inch) VRT19 66-Hz or 72-Hz Trinitron monitors.

Graphics Performance Characteristics

Benchmark Reference

For information on the latest DEC 3000 Model 400/400S AXP performance benchmarks, contact your Digital sales support representative. The information which follows will help you to interpret the benchmarks you receive from your Digital representative.

Varying Benchmarks

Characterizing graphics performance in ways which can help a user discriminate among competing systems is not a straightforward problem. The relative performance of two different systems will differ from benchmark to benchmark, depending not only on the sort of graphics processing inherent in the application, but also on the way in which it was coded.

Systems that favor line drawing, for example, may perform relatively poorly for polygon fill. Aggregating objects into large primitives may greatly enhance the performance on some systems, while on other systems aggregation may provide small benefit.

Speed Measures

Certain gross drawing speed measures have become current in the industry: Pixels per second for area filling, vectors per second for wireframe drawing, and polygons per second for shaded surface rendering. Such quoted numbers usually represent peak rates obtained with artificial timing programs that create most favorable conditions; they might not pertain to typical applications.

Standard Benchmarks

In an attempt to provide a more relevant means of performance characterization, an industry committee has defined a Picture Level Benchmark (PLB) suite. These benchmarks consist of drawing complete pictures deemed to be more typical of applications than the instruments which are used to determine the peak rates.

Peak Performance Measurements

In measurements of several of the peak graphics speed parameters, the appropriate application programming interface was used: Xlib for the 2D parameters, and DEC PHIGS for the 3D. Therefore, the measurements reflect the overhead costs of the supplied APIs as well as the CPU and graphics hardware. The 2D numbers were obtained using programs from x11perf, the performance measurement suite supplied with the X System.

Area-Fill Performance

Area-fill performance is especially relevant in routine window system operation. The prototypical area fill operation is simply copying a rectangular pixmap from system memory to the frame buffer. This simple copy reflects performance characteristics which will affect many windowing and image-manipulation functions. The x11perf pixmap copy test uses a 500x500 pixmap. The speed of the operation is expressed in megapixels per second (Mpixel/s), where 1 Mpixel equals 1,048,576 pixels.

HX Performance

HX performance is substantially affected by CPU performance and TURBOchannel clock speed, so the numbers are higher for the DEC 3000 Model 400 AXP system than for other models using the same graphics options.

Note that the HX performance benefits a great deal from the Smart Frame Buffer acceleration feature. The pixel copy test makes no use of the geometry processing and scan conversion support of the PXG family of options and so derives no benefit from these options. Therefore, the HX is clearly a preferable choice over the PXG family for 8-plane image manipulation applications that have no substantial drawing component.

2D Line Drawing

Beyond area-fill, the next popular gross 2D performance parameter is line-drawing performance, measured in vectors per second. This test also was provided by one of the standard x11perf benchmarks. Vectors are 10-pixel line segments, randomly oriented, grouped in 10-segment polylines.

3D Line Drawing

Line drawing also remains important in 3D graphics. The speed of rendering shaded polygons replaces simple area fill as a crucial performance question. The performance measurements are obtained with benchmarks written in DEC PHIGS. Vectors are now 3D vectors, still 10 pixel-long, randomly oriented, grouped into 10-segment polylines and clip-checked. 3D polygons are triangles, 100 pixels in area, combined into 10-triangle strips, shaded, illuminated with two light sources, Z-buffered, and clip-checked. Z-buffering and smooth shading in good interactive time requires hardware acceleration, so these results are given only for the PXG family of options.

Graphics Performance Characterization Committee

The Graphics Performance Characterization (GPC) committee is a volunteer group of vendors, users, and consultants who provide and support standardized benchmarks for measuring performance of computer graphics systems. GPC operates under the administration of the National Computer Graphics Association (NCGA).

Picture-Level Benchmark Results

The first performance measurement instrument developed by GPC is called the Picture-Level Benchmark (PLB). PLB characterizes performance in terms of the time to draw specified complete pictures. The initial PLB pictures adopted by GPC are:

- pc_board, a typical 2D electrical CAD application
- sys_chassis, a 3D wire-frame model of a computer chassis
- cyl_head, a 3D model of an automobile engine's cylinder head
- head, a 3D model of a human head based on laser scanner data
- shuttle, a low-end 3D simulation with the Space Shuttle

Multiscreen Support

HX Option Configurations

The DEC 3000 AXP workstation family supports multiscreen configurations for the HX option. A multiscreen configuration consists of up to three graphics options of the same type, installed in a single workstation. Presently, you cannot mix options in a multiscreen configuration. This is a limitation of the DECwindows Motif software, which could, in principle, change in the future.

Multiscreen Software Support

Software support for the multiscreen capability resides in the X Window server code, so is available to all X-based applications and APIs. Windows must lie all on one screen, but the cursor tracks across all the screens.

4

DEC 3000 Model 400S AXP Server

Introduction

The DEC 3000 Model 400S AXP server is based on the DEC 3000 Model 400 AXP workstation, utilizing the same CPU subsystem and system enclosure. The difference between the two systems is that the server does not include a monitor. In the absence of a graphics option and monitor, the DEC 3000 Model 400S AXP must have a terminal attached via the serial port to provide console functions.

In This Chapter

This chapter contains the following sections on the DEC 3000 Model 400S AXP server:

- Server Uses
- Server Performance Parameters

Server Uses

Introduction

The client/server model for distributed computing is based on the premise that there remains a place for shared centralized computing resources, even in an era when individual users enjoy the benefits of powerful deskside workstations. Shared server nodes may function in several ways:

- As a disk file server for client workstations that are diskless or have only limited local disk space
- As host for a shared database
- As host for specialized peripherals that are used only occasionally by each individual user: Printers, plotters, FAX, or other multimedia devices
- As a gateway to non-local networks and long-haul communication interfaces, serving mail, news, and other communication functions
- As host of specialized shared computing facilities such as attached array processors for scientific computation
- As a timeshared basic computing resource for distributed users with dumb character cell terminals, X Window terminals, or PCs.

Disk File Server

In the case of a file server application, a number of users with their own interactive workstations may each own directories on the server's file system. These directories can be mounted on their local systems via NFS for each interactive session and used transparently as if they were local.

Host for Shared Database

In the database server application, a large shared database and access software reside on the server and respond to database query commands from the remote users.

Timesharing System

The server may also be used as a timesharing system. The server provides a basic computing capability to remote users, using either character terminals or X Window terminals.

Deskside Server

Because it is powerful, expandable, and upgradeable, and because of its TURBOchannel-based open I/O capabilities, extensive peripheral options, and networking capabilities, the DEC 3000 AXP family is especially well-suited for the deskside server role, both for work groups and for large PC installations.

Server Performance Parameters

AIM Suite Benchmarks

AIM Technology developed a family of benchmarks for UNIX systems. Each benchmark evaluates systems differently and provides an independent measurement. AIM benchmarks discussed in this section are:

- AIM Milestone Benchmark
- AIM Suite III Multiuser Benchmark

Additionally, AIM benchmarks verify the results of the Suite III tests run by licensed vendors and uses the source data for the AIM Performance Report service.

AIM Milestone Benchmark

The AIM Milestone benchmark evaluates the performance of a system by simulating a pre-defined collection of user types:

- Administrative Assistant
- Spreadsheet User
- Database User
- Manager
- Scientist
- Software Engineer
- Technical Writer

Bourne Shell Command Language and Shell Scripts

The Bourne shell command language provides control, and shell scripts and standard UNIX commands perform the user tasks. Equal weight is assigned to each of the user types. Think times within and between commands are supported and tunable. Because each user performs different tasks repetitively, excessive CPU instruction and data caching caused by locality of reference is avoided. Milestone was designed to functionally represent real user loads.

System performance is reported for one to seven simultaneous user loads and a Milestone Index is generated. This rating is used to compare system performance across systems. Also reported are elapsed time and load average for each simulated userload level. The elapsed time tells you the speed of the system. The load average indicates how well the system handled multiprocessing and contention for I/O devices.

AIM Suite III Multiuser Benchmark

AIM Suite III is designed to measure, evaluate, and predict UNIX multiuser system performance of multiple systems. AIM Suite III uses 33 functional tests and these tests can be grouped to reflect the computing activities of various types of applications. AIM Suite III is designed to stress schedulers and I/O subsystems, and includes code that will exercise TTYs, tape subsystems, printers, and virtual memory management. The benchmark will run until it reaches either the user-specified maximum number of simulated users or the system capacity.

The 33 subsystem tests, each of which exercises one or more basic functions of the UNIX system under test, are divided into six categories based on the type of operation involved:

- RAM
- Floating Point
- Pipe
- Logic
- Disk
- Math

Within each of these six categories, the relative frequencies of the subsystem tests are evenly divided (with the exception of small biases for add-short, add-float, disk reads, and disk writes).

AIM Suite III contains no application level software. Each simulated user runs a combination of subsystem tests. The load that all simulated users put on the system is said to be characteristic of a UNIX time-sharing environment. The mix of subsystem tests can be varied to simulate environments with differing resource requirements. AIM provides a default model as a representative workload for UNIX multiuser systems and the competitive data that AIM Technology publishes is derived from this mix of subsystem tests.

TPC Benchmarks

The Transaction Processing Performance Council (TPC) defines standard benchmark specifications for commercial transaction processing systems. Their TPC Benchmark A (TPC-A) exercises the system components necessary to perform tasks associated with that class of on-line transaction processing (OLTP) environments emphasizing update-intensive database services. Such environments are characterized by:

- Multiple on-line terminal sessions
- Significant disk input/output
- Moderate system and application execution time
- Transaction integrity

The TPC Benchmark B exercises the database components necessary to perform tasks associated with that class of transaction processing environments emphasizing update-intensive database services. Such environments are characterized by:

- Significant disk input/output
- Moderate system and application execution time
- Transaction integrity

Two metrics are used in TPC-A and TPC-B:

- Throughput as measured in transactions per second (tps) subject to response time constraint
- Associated price-per-tps (K$/tps)

However, in spite of similarities, TPC-A and TPC-B contain substantial differences, which make their results incomparable. Both the TPC Benchmark A Standard and the TPC Benchmark B Standard require that test sponsors publish a full disclosure report of the implementation details along with the published test results.

A

Alpha AXP: A Computing Architecture for the 21st Century

What Is a Computing Architecture?

Beneath every computing solution is a set of rules about how data is addressed and instructions are handled. This set of rules is the foundation for how a computer—the hardware that performs operations, the compiler that turns software instructions into hardware operations, the operating environment for applications and applications development—actually works.

This set of rules is called a computing architecture. To the extent that a computing architecture is robust and long-lived—spanning a large range of systems, many kinds of applications, and multiple generations of technology—it delivers important benefits. Transitions to new systems and new technologies are smooth and nondisruptive. Investments in hardware, software, and data are protected. Operational and support requirements are simplified and the overall cost of ownership is lowered.

Throughout the history of computing there have been many different computing architectures, optimized for many different kinds of technology and applications. Some span only one system and last only for one generation. Others, like IBM's System 360 architecture, have delivered a consistent computing environment for more than two decades. Still others, like Digital's VAX architecture, are able to incorporate new performance technologies (clusters, integrated vector processing, and multiprocessing) that last more than a decade and span a broad range of systems.

Why a New Architecture?

Even the best computing architectures must eventually be replaced. In fact, the most popular computer systems in the world today, both complex instruction set computers (CISC) and reduced instruction set computers (RISC), share an underlying limitation in their architectures that made the development of a new computing architecture inevitable. They are all 32-bit systems.

The 32-bit computing architectures the world uses today are running out of address space. While systems based on both CISC and RISC 32-bit architectures will continue to deliver leading-edge performance for years to come, they will not be able to keep pace with improvements in memory technology to meet the demands of the “supercomputing-like” applications of the 21st century, including visualization, very large databases, imaging, multimedia, simulation, and modeling.

In This Appendix

This appendix contains the following sections:

- Planning for the Next Century
- A 25-Year Design Horizon
- Characteristics of 21st Century Architecture
- Introducing Alpha AXP
- Characteristics of Alpha RISC Architecture

Planning for the Next Century

Introduction

Since most computer users will have to choose a new computing architecture sometime within the next decade, it is important to ask what are the characteristics of a computing architecture that will last and will carry them well into the 21st century?

A careful analysis of past architectures, as well as a projection of current technology trends, yields a number of core characteristics that a computing architecture must have to be viable into the 21st century.

Architectures of the 1960s and 1970s

In the 1960s and 1970s, the most successful computing architectures were designed for a single, controlled, and proprietary operating environment. The goal was to provide binary compatibility across a range of systems—and multiple generations. This compatibility over time and across systems protected investments in applications and operational expertise, as well as provided a stable environment for application development. Despite their single, proprietary operating systems, these architectures supported many different languages, a broad range of applications, and a broad range of systems.

Price/Performance Architectures of the 1980s

In the 1980s, computing architectures focused on exploiting performance for single-user PCs and workstations. The development of RISC architectures achieved new levels of price/performance. Compared to the broad multilanguage and multisystem range of the computer-family architectures, architectures in the 1980s focused on a narrow range of systems (PCs, workstations, and, recently, small servers) and a narrow range of operating environments (single-user PC operating systems and the UNIX operating system). For the most part, these architectures did not provide binary compatibility for applications from one implementation of hardware to the next.

Learning from the Past

To be successful in the 1990s, a computing architecture must deliver the benefits of both the family architectures of the 1960s and 1970s and the highly competitive price/performance architectures of the 1980s. In other words, a 21st century architecture must provide both investment protection and competitive performance across a broad range of systems.

A 25-Year Design Horizon

Introduction

From the beginning, the Alpha AXP computing architecture was designed to last at least 25 years. Most major computer users are unwilling and unable to undergo a major technology transition any more frequently than once every 10 years—and only then, for significantly more performance or functionality.

A computer architecture that delivers competitive performance and binary compatibility for 25 years would be highly desirable. But is it possible?

Lessons from the Past

The IBM System 360 and 370 family of computers, designed in the 1960s, have provided a viable, binary-compatible computing family for almost 25 years.

Digital's VAX systems, designed in the 1970s, deliver among the best price/performance in the industry today. They have provided binary compatiblity across a broad range of systems for more than a decade. And VAX systems will continue to deliver both binary compatibility and highly competitive price/performance for many more years.

Given these past accomplishments, designing a 25-year architecture certainly seems feasible. However, the challenges of designing such a computing architecture with no compromise in performance should not be minimized. Today there are far more computer technology vendors and computer users than when the System 360 or VAX were designed.

Characteristics of 21st Century Architecture

Introduction

A 25-year architecture must start with a clean slate. Technology, markets, and applications are changing much more rapidly and unpredictably than they were even 5 years ago. There are so many fast changing, interdependent variables, that a 21st century computing architecture cannot be built with any assumptions about what will happen in the world of computing (or the world itself) in the next 25 years.

In other words, no assumptions about technology and no features that could get in the way of future innovation or implementation can be built into the architecture. Every classic computing architecture technique must be re-examined. Assumptions about instruction sets and physical memory must be discarded. For the architecture to endure, instruction decoding and data paths must be stripped down to their very simplest level.

Even from the vantage point of the early 1990s, it is obvious that the 25-year architecture must have the following characteristics:

- Include address space for growth
- Exhibit speed
- Include a broad family of systems
- Be targeted for multiple operating environments
- Be open to the products of multiple vendors

Address Space

Lack of address space is one thing that can curtail the life of a computing architecture. Many in the industry say that the world's next computing architecture will be a full-speed 64-bit computing architecture, including: 64-bit data, 64-bit registers, 64-bit addressing, 64-bit operation, and 64-bit compilers.

But is a 64-bit architecture big enough to last 25 years?

Given the historical consumption of 6/10ths of a bit per year, moving from a 32-bit to 64-bit architecture actually should provide a comfortable 50 years of growth. The move from a 32-bit to 64-bit architecture is not just a doubling, it provides 4 billion times the address space—and the architectural underpinnings and performance to last 25 years.

Performance Scaling of 1000X

A computing architecture for the 21st century must have a very fast implementation. Not just the first generation, but every implementation, must be among the most competitive in terms of price and performance for its time.

In addition, the architecture must be able to scale effectively over a range of 1 to 1000. Over the last 10 years, computing performance has improved by a factor of 100. Given the ever-accelerating rate of technical advancement and demand for performance, it is very likely that a 25-year architecture will have to be able to scale over a performance range of 10 times 100, or 1000.

One prediction that can confidently be made about performance gains in the next century is that they will not be achieved through CPU clock speed alone.

Amazingly, computer cycle times are already approaching the speed of light. Given a chip of some dimension and the limits of physics, cycle times in the 1/4- to 1/2-nanosecond range are the best that can be achieved. With clock speeds today already approaching 5 nanoseconds, the best performance gain that can be achieved through clock speed alone is by a factor of 10.

This means that a 21st-century architecture must be able to take advantage of performance improvements in all three dimensions of performance: CPU clock speed, multi-issue instruction (superscalar), and multiple processors, including massively parallel processing.

A Broad Family of Systems

Thirty years of 20th-century computing have shown that today’s mainframe is tomorrow’s notebook. Customers need a range of computing solutions to address different sized problems. A computing architecture that does not scale well across a broad range of systems is soon limited in its usefulness. At a minimum, an architecture for the 21st century must be able, from the begining, to be implemented on a single chip on the low end and to support very large, even massively parallel multiprocessing systems on the high end.

Multiple Operating Environments

Unlike all past architectures, a computing architecture for the 21st century will not be targeted for, nor show any bias toward, any particular operating system, language, or style of computing.

Very few organizations today rely on one operating environment. Production, client/server, PC, realtime, high-security — different applications require different operating environments for optimal performance. And new operating systems and environments are likely to be invented and become important over the next 25 years.

Today, computing architectures are optimized for a particular operating environment via different interrupt modes, types of memory management, and physical memory addressing—all of which are defined in the computing architecture.

In contrast, an architecture that can be optimized for multiple computing environments can solve a wider variety of customer problems. This enables it to become more pervasive, and so attract more solutions development, and so solve a wider variety of customer problems, and so on.

Such a truly unbiased architecture would have no “also ran” operating systems, languages, or styles of computing. An unbiased architecture could be optimized (in interrupts, operating modes, physical memory mapping, and so forth) to support multiple high-performing operating environments, regardless of whether the performance desired is SPECmarks, transactions per second, floating point, realtime, integer, COBOL processing speeds, or a performance dimension not even defined today.

Multiple Vendors

As the computing industry moves towards the 21st century, computer vendors and computer users alike are experiencing the benefits and the necessities of enabling solutions from many vendors to work together in a single solution.

Perhaps the most dramatic change in computing architectures is the concept of an open computing architecture—one that can be implemented by many different vendors at all levels of integration.

To be open, an architecture must be designed in such a way that a combination of support for standards and licensing will make it profitable for other vendors to implement portions of the architecture – and feasible for customers to mix and match chips, systems, compilers, and software from multiple sources to meet their needs.

In the past, a single vendor assumed both the authority and responsibility for being the "sole source implementer" of all aspects of the architecture: Chip design, fabrication, systems, compilers, operating environment—and, at one time, even applications.

A 21st century architecture, however, will be open to implementation by other vendors at all levels. For one thing, customers are demanding open solutions, available from multiple vendors. For another, practically speaking, the life of a computing architecture is increasingly dependent on its ability to achieve high volume in the marketplace. The large amounts of capital required for designing and fabricating new computer chips (approaching $1 billion in the next few years!) means that only those architectures that achieve high volume will generate the capital needed to produce the next generation of chips.

Introducing Alpha AXP

A New Architecture

Moving to a new computing architecture is a major transition. Even with services, tools, and a migration strategy that enables customers to move gradually and modularly at their own pace, moving to a new architecture represents both effort and risk for most computer users.

At Digital, we believe, that customers moving to a new computing architecture deserve a high-performance, flexible, and viable foundation for computing that will last for 25 years.

Because, while a computing solution reflects quality of execution at all levels of implementation—chip design, fabrication, compilers, operating system, development tools, and applications—a superior, long-lasting computing architecture is the the foundation that every subsequent layer of computing technology leverages.

The VAX Example

In 1975, the engineers working on the new VAX computer architecture made a prediction about when their new architecture would begin to face the limits of its endurance as a leading-edge technology. In October of 1992, they calculated, the world would begin to need a new computing architecture. Until then, theirs would be the best computing architecture.

The Alpha AXP Answer

Almost on schedule with that estimate comes the Alpha AXP architecture. Alpha AXP is a 64-bit load/store RISC architecture that is designed with particular emphasis on the three elements that most affect performance: clock speed, multiple instruction issue, and multiple processors. Alpha AXP is the first 21st century computer architecture.

The Alpha AXP architects examined and analyzed current and theoretical RISC architecture design elements and developed high-performance alternatives for the Alpha AXP architecture. The architects adopted only those design elements that appeared valuable for a projected 25-year design horizon.

Characteristics of Alpha RISC Architecture

A True 64-bit Architecture

Alpha AXP was designed as a 64-bit architecture. All registers are 64 bits in length and all operations are performed between 64-bit registers. It is not a 32-bit architecture that was later expanded to 64 bits.

A Design for Very High-Speed Implementations

The instructions are very simple. All instructions are 32 bits in length. Memory operations are either loads or stores. All data manipulation is done between registers.

The Alpha AXP architecture facilitates pipelining multiple instances of the same operations because there are no special registers and no condition codes.

The instructions interact with each other only by one instruction writing to a register or memory and another instruction reading from the same place. That makes it particularly easy to build implementations that issue multiple instructions every CPU cycle. (The first implementation issues two instructions per cycle.)

Alpha AXP makes it easy to maintain binary compatibility across multiple implementations and easy to maintain full speed on multiple-issue implementations. For example, there are no implementation-specific pipeline timing hazards, no load-delay slots, and no branch-delay slots.

The Approach to Byte Manipulation

The Alpha AXP architecture does byte shifting and masking with normal 64-bit register-to-register instructions, crafted to keep instruction sequences short.

Alpha AXP does *not* include single-byte store instructions. This has several advantages:

- Cache and memory implementations need not include byte shift-and-mask logic, and sequencer logic need not perform read-modify-write on memory locations. Such logic is awkward for high-speed implementation and tends to slow down cache access to normal 32-bit or 64-bit aligned quantities.

- The Alpha AXP approach to byte manipulation makes it easier to build a high-speed error-correcting write-back cache, which is often needed to keep a very fast RISC implementation busy.
- The Alpha AXP approach can make it easier to pipeline multiple byte operations.

The Approach to Arithmetic Traps

Alpha AXP lets the software implementor determine the precision of arithmetic traps. With the Alpha AXP architecture, arithmetic traps (such as overflow and underflow) are imprecise—they can be delivered an arbitrary number of instructions after the instruction that triggered the trap. Also, traps from many different instructions can be reported at once. That makes implementations that use pipelining and multiple issue substantially easier to build.

However, if precise arithmetic exceptions are desired, trap barrier instructions can be explicitly inserted into the program to force traps to be delivered at specific points.

The Approach to Multiprocessor Shared Memory

As viewed from a second processor (including an I/O device), a sequence of reads and writes issued by one processor may be arbitrarily reordered by an implementation. This allows implementations to use multibank caches, bypassed write buffers, write merging, pipelined writes with retry on error, and so forth. If strict ordering between two accesses must be maintained, explicit memory barrier instructions can be inserted in the program.

The basic multiprocessor interlocking primitive is a RISC-style load-locked, modify, store-conditional sequence. If the sequence runs without interrupt, exception, or an interfering write from another processor, then the conditional store succeeds. Otherwise, the store fails and the program eventually must branch back and retry the sequence. This style of interlocking scales well with very fast caches, and makes Alpha AXP an especially attractive architecture for building multiple-processor systems.

Instructions with Optimization Hints

Several Alpha AXP instructions include hints for implementations, all aimed at achieving higher speed.

- A target hint for calculated jump instructions can allow much faster subroutine calls and returns.
- Prefetching hints for the memory system can allow much higher cache hit rates.
- Granularity hints for the virtual-address mapping can allow much more effective use of translation lookaside buffers for large contiguous structures.

PALcode—A Very Flexible Privileged Software Library

A Privileged Architecture Library (PALcode) is a set of subroutines that are specific to a particular Alpha AXP operating system implementation. These subroutines provide operating-system primitives for context switching, interrupts, exceptions, and memory management. PALcode is similar to the BIOS libraries in personal computers.

PALcode subroutines are invoked by implementation hardware or by software CALL_PAL instructions.

PALcode is written in standard machine code with some implementation-specific extensions to provide access to low-level hardware.

PALcode for OpenVMS AXP

One version of PALcode lets Alpha AXP implementations run the full OpenVMS AXP operating system by mirroring many of the OpenVMS VAX features. The OpenVMS PALcode instructions let Alpha AXP run OpenVMS with little more hardware than that found on a conventional RISC machine: The PAL mode bit itself, plus 4 extra protection bits in each Translation Buffer entry.

PALcode for DEC OSF/1 AXP

Another version of PALcode lets Alpha AXP implementations run the DEC OSF/1 AXP operating system by mirroring many of the RISC ULTRIX features. Versions of PALcode can also be developed for real-time, teaching, and other applications.

PALcode makes Alpha AXP an especially attractive architecture for multiple operating systems.

The Alpha AXP Approach to Multiple Operating Environments

The Alpha AXP architecture is designed to avoid bias toward any particular operating system or programming language. Alpha AXP initially supports the OpenVMS AXP and DEC OSF/1 AXP operating systems, and supports simple software migration from applications that run on those operating systems.

Alpha AXP is an attractive architecture for compiling a large variety of programming languages. Alpha AXP has been carefully designed to avoid bias toward one or two programming languages. For example:

- Alpha AXP does not contain a subroutine call instruction that moves a register window by a fixed amount. Thus, Alpha AXP is a good match for programming languages with many parameters and programming languages with no parameters.
- Alpha AXP does not contain a global integer overflow enable bit. Such a bit would need to be changed at every subroutine boundary when a FORTRAN program calls a C program.

B

DECchip 21064: A Technical Summary

Introduction

The DECchip 21064 CPU chip is a RISC-style microprocessor which operates at up to 150 MHz. The chip implements a new 64-bit architecture, designed to provide a huge linear address space and to eliminate bottlenecks that would impede highly concurrent implementations.

Fully pipelined and capable of issuing two instructions per clock cycle, the DECship 21064 can execute up to 400 million operations per second. The chip includes:

- An 8KB instruction cache
- An 8KB data cache
- Two associated translation buffers
- Four-entry (32 byte/entry) write buffer
- A pipelined 64-bit integer execution unit with a 32-entry register file
- A pipelined floating-point unit with an additional 32 registers

In This Appendix

This appendix contains the following sections:

- DECchip 21064 Architecture
- DECchip 21064 Technical Specifications

DECchip 21064 Architecture

Pin Interface

The pin interface includes integral support for an external secondary cache.

- The package is a 431-pin PGA with 140 pins dedicated to VDD/VSS.
- The chip is fabricated in 0.75 μ N-well CMOS with three layers of metallization.
- The die measures 16.8 mm by 13.9 mm and contains 1.68 million transistors.
- Power dissipation is 30 watts from a 3.3V supply at 150 MHz.

Microprocessor Features

Table B–1 lists some of the features of the microprocessor.

Table B–1 DECchip 21064 Features

Feature size	0.75 μ
Channel length	0.5 μ
Gate oxide	10.5 nm
VTXn/VTXp	0.5V/-0.5V
Power supply	3.3V
Substrate	P epi with N well
Silicide	Cobalt-disilicide
Buried contact	Titanium nitride
Metal 1	7.5 kA AlCu, 2.25 μ pitch (contacted)
Metal 2	7.5 kA AlCu, 2.625 μ pitch (contacted)
Metal 3	20 kA AlCu, 7.5 μ pitch (contacted)

The architecture includes comprehensive support for both 32- and 64-bit operations on an instruction-specific basis. It is designed to provide continuity to an earlier CISC architecture without sacrificing RISC performance characteristics. All operate instructions are register-to-register while memory operations are strictly load/store.

Instruction Issue Scheme

The instruction issue scheme supports pair-wise execution among combinations of four basic units:

- Load/store
- Integer operate
- Floating-point operate
- Branch

Pipeline

The pipeline depth is seven cycles for everything except floating-point operate which is ten cycles. Only integer multiply and floating divide are not fully pipelined. Integer latency is generally one cycle, load latency is three cycles, and floating-point latency is six cycles. Instructions are issued in order and under the control of register scoreboarding logic. The scoreboarding logic also controls result bypassing among the various units. The issue point is at pipeline stage four, beyond which the pipeline does not stall.

Floating-Point Unit

The floating-point unit is a fully pipelined 64-bit floating-point processor that supports both VAX standard and IEEE standard data types and roundings. It is capable of generating a 64-bit result every cycle for all operations except divide. The pipeline stages include:

- A 64-bit adder
- A leading-1 detector using input operands
- Two parallel exponent adders in stage 1
- A radix-8 pipelined Booth multiplier organized as 8 odd and 8 even rows
- A 64-bit shifter capable of shifting both left and right by up to 63 bits
- A 64-bit double adder to enable parallel addition and rounding in the final stage

Clock Unit

Many novel circuit structures and detailed analysis techniques were developed to support the clock rate in conjunction with the complexity demanded by the concurrence and wide data paths. The clocking method is level-sensitive, single phase. Since there is no "dead time", clock integrity must be ensured to avoid race-through in latches. It is also important to avoid clock skew contribution to delay paths.

The DECchip 21064 uses a single-driver approach to clock distribution, using the third metal layer to do the majority of clock routing (as well as VDD/VSS). Total capacitive load on the clock driver is 3250 pF, requiring a final driver width of 250Ku (25 cm; 10 inches) and 100Ku (10 cm; 4 inches) for PMOS and NMOS, respectively. The clock driver resides in the horizontal center of the chip and extends vertically from top to bottom of the core. A method to extract and display clock skew was developed to analyze the grid.

A difficult circuit problem was the 64-bit adder portion of the integer and floating point ALUs. To achieve single-cycle latency in this unit, a combination of logic and circuit techniques was used. The logical scheme is a hybrid of two techniques—Manchester carries for the initial 8-bit groups, followed by a logarithmic carry select tree. The Manchester scheme is unique in that the NMOS chain is precharged low and is conditionally pulled high. This avoids threshold delays in the pass transistors and improves performance of the carry chain by 10 percent over the pull-down approach.

Operation Parameters

To provide maximum flexibility in applications, the external interface allows for several different operation parameters, including:

- A choice of logic family (CMOS/TTL or ECL)
- Bus width (64-bit or 128-bit)
- External cache size and access time
- BIU clock rate

These parameters are set into mode registers during chip power-up. The logic family choice provided an interesting circuit challenge. The input receivers are differential amplifiers that utilize an external reference level. To maintain signal integrity of this reference voltage, it is resistively isolated and RC-filtered at each receiver. The output driver presented a more difficult problem due to the 3.3-V V_{DD} chip power supply.

To provide a good interface to ECL, it is important that the output driver pull to the V_{DD} rail (for ECL operation V_{DD}=0 V, V_{SS}=-3.3 V). This precludes using NMOS pull-ups. PMOS pull-ups have the problem of well junction forward bias and PMOS turn-on when bidirectional outputs are connected to 5 V logic in CMOS/TTL mode. The solution is a unique floating well driver circuit which avoids the cost of series PMOS pull-ups in the final stage.

DECchip 21064 Technical Specifications

Technical specifications for the 150-megahertz DECchip 21064 are listed in Table B–2.

Table B–2 DECchip 21064 (150 MHz) Technical Specifications

Process technology	0.75 μ CMOS
Cycle time	150 MHz (6.6 ns)
Die size	13.9 mm x 16.8 mm
Transistor count	1.68 million
Package	431 pin PGA
Number of signal pins	291
Power dissipation	23 W at 6.6 ns cycle
Power supply	3.3 V
Clocking input	300 MHz differential
On-chip D-cache	8-Kbyte, physical, direct-mapped, write-through, 32-byte line, 32-byte fill
On-chip I-cache	8-Kbyte, physical, direct-mapped, 32-byte line, 32-byte fill, 64 ASNs
On-chip DTB	32-entry; fully-associative, 8-Kbyte, 64-Kbyte, 256-Kbyte, 4-Mbyte page sizes
On-chip ITB	8-entry, fully associative, 8-Kbyte page plus 4-entry, fully-associative, 4-Mbyte page
Floating-point unit	On-chip FPU supports both IEEE and VAX floating point
Bus	Separate data and address bus; 128-bit /64-bit data bus
Serial ROM interface	Allows the chip to directly access serial ROM
Virtual address size	64 bits checked; 43 bits implemented
Physical address size	34 bits implemented
Page size	8 Kbytes
Issue rate	2 instructions per cycle to A-box, E-box, or F-box
Integer pipeline	7-stage pipeline
Floating pipeline	10-stage pipeline

Glossary

ANSI

Acronym for American National Standards Institute, an organization that develops and publishes standards for the computer industry.

antialiasing

An advanced graphics term for a filtering operation applied to a rendered image to smooth the objectionable jagged appearance of straight lines that is caused by the finite pixel size.

arbiter

The entity responsible for controlling a bus. It controls bus mastership and may field bus interrupt requests.

assert

To cause a signal to change to its logical true state.

asymptotic

An equality statement about two functions indicating that the ratio of the two functions approaches 1 while the variable approaches some value, usually infinity. It is not a true equality. This term is sometimes used in reference to bus bandwidth (approaching, but never reaching, full theoretical bandwidth).

asynchronous system trap (AST)

A software-simulated interrupt to a user-defined routine. ASTs enable a user process to be notified asynchronously, with respect to that process, of the occurrence of a specific event. If a user process has defined an AST routine for an event, the system interrupts the process and executes the AST routine when that event occurs. When the AST routine exits, the system resumes execution of the process at the point where it was interrupted.

autoboot

The process by which the system boots automatically.

backplane

The main circuit module or panel that connects all of the modules in the system. In desktop systems, the backplane is analogous to the mother module.

backup cache

A second, very fast memory that is used in combination with slower large-capacity memories.

bandwidth

Often used to express a high rate of data transfer in an I/O channel. This usage assumes that a wide bandwidth may contain a high frequency, which can accommodate a high rate of data transfer.

baud rate

A data communication rate unit used similarly to bits per second (bps) for low-speed data. It is defined as the number of signal level changes per second regardless of the information content of those signals.

benchmark

A routine or program used in the evaluation of computer performance.

bit

A binary digit. The smallest unit of data in a binary notation system, designated as 0 or 1.

bit rate

The rate at which bits are transmitted over a communications path. Normally expressed in bits per second (bps). Not to be confused with baud rate.

BIU

See bus interface unit.

block exchange

A memory feature that improves bus bandwidth by paralleling a cache victim write-back with a cache miss fill.

boot

To load an operating system into memory or otherwise reach a needed state through internal action. A shorthand form of *bootstrap*.

bootblock

The first logical block on the boot device. It contains information about the location of the primary bootstrap on the device.

boot device

The device from which the system bootstrap software is acquired.

boot flags

Indicators that contain information that is read and used by the boot software during a system boot procedure.

bootp

An Internet protocol used for network booting. It is based on UDP (the Internet User Datagram Protocol) which is a simple, reliable datagram protocol. The bootp protocol allows a diskless machine to find its Internet address, the address of a bootserver, and the name of a file to boot.

boot primitives

Device handler routines that read the bootblock and, subsequently, the primary bootstrap program, into memory from the boot device. *See also* bootblock.

boot server

A system that provides boot services to remote devices such as network routers and VAXcluster satellite nodes.

bootstrap

See boot.

buffer

An internal memory area used for temporary storage of data records during input or output operations.

bug

An error or malfunction in a system, device, or program.

bugcheck

The operating system's internal diagnostic check. The system logs a failure and crashes the system.

bus

A group of signals that consists of many transmission lines or wires. It interconnects computer system components to provide communications paths for addresses, data, and control information.

bus interface unit (BIU)

Logic designed to interface internal logic, a module or a chip, to a bus.

byte

Eight contiguous bits starting on an addressable byte boundary. The bits are numbered right to left, 0 through 7.

cache

See cache memory.

cache block

The fundamental unit of manipulation in a cache. Also known as cache line.

cache line

See cache block.

cache memory

A small, high-speed memory placed between slower main memory and the processor. Cache memory increases effective memory transfer rates and processor speed. It contains copies of data recently used by the processor and fetches several bytes of data from memory, anticipating that the processor will access the next sequential series of bytes.

card cage

A mechanical assembly in the shape of a frame that holds modules against the system and storage backplanes.

CD–ROM

Acronym for compact disc read-only memory. The optical removable media used in a compact disc reader mass storage device.

channel

A path along which digital information can flow in a computer.

checksum

A sum of digits or bits that is used to verify the integrity of a piece of data.

CISC

Acronym for complex instruction set computer. An instruction set consisting of a large number of complex instructions that are managed by microcode. *Contrast with* RISC.

clean

In the cache of a system bus node, describes a cache line that is valid but has not been written.

client-server computing

An approach to computing that enables personal computer and workstation users—the client—to work cooperatively with software programs stored on a mainframe or minicomputer—the server.

clock

A signal used to synchronize the circuits in a computer system.

cluster

A group of systems and hardware that communicate over a common interface. *See also* VMScluster system.

CMOS

Acronym for complementary metal-oxide semiconductor, which is a silicon device formed by a process that combines PMOS and NMOS semiconductor material.

console subsystem

The subsystem that provides the user interface for a system when operating system software is not running. The console subsystem consists of the following components: console program, console terminal, console terminal port, remote access device, remote access port, and Ethernet ports.

console terminal

The terminal connected to the console subsystem. The console is used to start the system and direct activities between the computer operator and the computer system.

console terminal port

The connector to which the console terminal cable is attached.

CPU

The central processing unit. The unit of the computer that is responsible for interpreting and executing instructions.

CSR

Abbreviation for control and status register, which is a device or controller register that resides in the processor’s I/O space. The CSR initiates device activity and records its status.

cycle

One clock interval.

data bus

A bus used to carry signals between two or more components of the system.

D-cache

Data cache. A high-speed memory reserved for the storage of data. *Contrast with* I-cache.

deassert

To cause a signal to change to its logical false state.

DECnet network

Networking software designed and developed by Digital. DECnet is an implementation of the Digital Network Architecture.

DEC OSF/1 AXP operating system

A general-purpose operating system based on the Open Software Foundation OSF/1 1.0 technology. DEC OSF/1 AXP V1.2 runs on the range of Alpha AXP systems, from workstations to servers.

DEC VET

A verifier and exerciser tool developed by Digital. DEC VET is a multipurpose system-maintenance tool that performs exerciser-oriented maintenance testing.

depth cueing

An advanced graphics term for the gradual variation of color across a primitive element indicating the variation of depth in the 3D scene.

Dhrystone

A benchmark for computer system performance that measures integer performance. Dhrystone is part of the suite of SPEC benchmarks.

direct-mapping cache

A cache organization in which only one address comparison is needed to locate any data in the cache, because any block of main memory data can be placed in only one possible position in the cache.

direct memory access (DMA)

Access to memory by an I/O device that does not require processor intervention.

dirty

Describes a cache block in the cache of a system bus node that is valid and has been written so that it differs from the copy in system memory.

dirty victim

Describes a cache block in the cache of a system bus node that is valid but is about to be replaced due to a cache block resource conflict. The data must therefore be written to memory.

disk fragmentation

The writing of files in noncontiguous areas on a disk. Fragmentation can cause slower system performance because of repeated read or write operations on fragmented data.

disk mirroring

See volume shadowing.

distributed processing

A processing configuration in which each processor has its own autonomous operating environment. In a distributed processing environment, the processors are not tightly coupled and globally controlled as they are with multiprocessing. Instead, an application is distributed over more than one system. The application must have the ability to coordinate its activity over a dispersed operating environment.

double buffering

An advanced graphics term for rendering a picture into an off-screen frame buffer, then presenting the viewer with the completed picture all at once, either by very rapidly copying the completed raster image to the on-screen buffer or by making the drawing buffer the on-screen buffer by switching.

EEPROM

Acronym for electrically erasable programmable read-only memory. A memory device that can be byte-erased, written to, and read from.

emitter-coupled logic

A form of current-mode logic in which only one transistor conducts at a time. The emitters of the two transistors are tied together to a single current-carrying resistor.

environment variables

Global data structures that can be accessed from console mode. The setting of these data structures determines how a system powers up, boots operating system software, and operates.

error correction code (ECC)

Code that carries out automatic error correction by performing an exclusive OR operation on the transferred data and applying a correction mask.

Ethernet

A local area network that was originally developed by Xerox Corporation and has become the IEEE 802.3 standard LAN. Ethernet LANs use bus topology.

Ethernet ports

The connectors through which the Ethernet is connected to the system.

extents

The physical locations in a storage device allocated for use by a particular data set.

Factory Installed Software (FIS)

Operating system software that is loaded into a system disk during manufacture. On site, the FIS is bootstrapped in the system, prompting a predefined menu of questions on the final configuration.

fast SCSI

An optional mode of SCSI-2 that allows transmission rates of up to 10 MB/sec.

FDDI

Fiber Distributed Data Interface. A high-speed networking technology that uses fiber optics as the transmission medium.

FIFO

Acronym for first in/first out; the order in which data is accessed.

firmware

Software code stored in hardware.

FIS

See Factory Installed Software.

floating point

A number that may be positive or negative but that has a whole (integer) portion and a fractional (decimal) portion; an arithmetic operation in which the decimal point is not fixed, but placed automatically in a correct position in a computer word.

FRU

Acronym for field-replaceable unit. Any system component that the service engineer is able to replace on site.

full-height device

Standard form factor for 13-centimeter (5.25-inch) storage devices.

half-height device

Standard form for storage devices that are half the height of full-height devices.

halt

The action of transferring control to the console program.

hard error

An error that has induced a nonrecoverable failure in a system.

hit

Indicates that a valid copy of a desired memory location is currently in cache.

I-cache

Instruction cache. A high-speed memory reserved for the storage of instructions. *Contrast with* D-cache.

IEEE

Abbreviation for Institute of Electrical and Electronics Engineers.

initialization

The sequence of steps that prepare the system to start. Initialization occurs after a system has been powered up.

interleaving

See memory interleaving.

internal processor register (IPR)

A register internal to the CPU chip.

I/O

Abbreviation for system input and output functions.

LAN

Abbreviation for local area network, a network that supports servers, PCs, printers, minicomputers, and mainframe computers that are connected over limited distances.

latency

The amount of time it takes the system to respond to an event.

LED

Abbreviation for light-emitting diode. A semiconductor device that glows when supplied with voltage.

Linpack

A benchmark for computer system performance that measures floating-point performance. Part of the suite of SPEC benchmarks.

local area VMScluster system

Digital's VMScluster configuration in which cluster communication is carried out over the Ethernet by software that emulates certain computer interconnect (CI) port functions.

longword

Four contiguous bytes starting on an arbitrary byte boundary. The bits are numbered from right to left, 0 through 31.

loopback tests

Diagnostic tests used to isolate a failure by testing segments of a particular control or data path.

machine check

An operating system action triggered by certain system hardware-detected errors that can be fatal to system operation. Once triggered, machine check handler software analyzes the error.

masked write

A write cycle that only updates a subset of a nominal data block.

mass storage device

An input/output device on which data is stored. Typical mass storage devices include disks, magnetic tapes, and floppy disks.

memory interleaving

The process of assigning consecutive physical memory addresses across multiple memory controllers. This process improves total memory bandwidth by overlapping system bus command execution across two or four memory modules.

memory-like

Describes regions that have predictable behavior. For example, all locations are read/write; a write to a location followed by a read from that location returns precisely the bits written.

MFLOPS

Acronym for millions of floating-point operations per second. An estimation of the theoretical peak performance of vector operations. A more useful means of measurement is to express MFLOPS in an industry-standard benchmark, such as Linpack MFLOPS.

MIPS

Acronym for millions of instructions per second.

miss

Indicates that a copy of a desired memory location is not in a cache.

mixed-interconnect VMScluster system

Digital's VMScluster system that uses multiple interconnect types between systems; for example, CI, Ethernet, DSSI, FDDI, or OpenVMS AXP.

module

A board on which logic devices (such as transistors, resistors, and memory chips) are mounted and connected to perform a specific system function.

MOP

Acronym for maintenance operations protocol. The transport protocol for network bootstraps and other network operations.

multiplex

Describes a system that can transmit several messages or signals simultaneously on the same circuit or channel.

multiprocessing system

A system that executes multiple tasks simultaneously.

node

A device that has an address on, is connected to, and is able to communicate with other devices on the bus. In a computer network, a node is an individual computer system connected to the network that can communicate with other systems on the network.

non-memory-like

Describes regions that may have arbitrary behavior. For example, there may be unimplemented locations or bits anywhere; some locations or bits may be read-only and others write-only.

NVRAM

Acronym for nonvolatile random-access memory. Memory that retains its information in the absence of power such as magnetic tape, drum, or core memory.

octaword

Sixteen contiguous bytes starting on an arbitrary byte boundary. The bits are numbered from right to left, 0 through 127.

open system

A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable applications software to:

- Be ported across a wide range of systems with minimal changes
- Interoperate with other applications on local and remote systems
- Interact with users in a style that facilitates user portability

OpenVMS AXP operating system

Digital's open version of the OpenVMS AXP operating system, which runs on Alpha AXP machines. *See also* open system.

operating system mode

The state in which the system console terminal is under the control of the operating system software. Also called program mode.

PAL

Acronym for programmable array logic. A hardware device that can be programmed by a process that blows individual fuses to create a circuit.

PALcode

Alpha AXP Privileged Architecture Library code, written to support Alpha AXP processors. PALcode implements architecturally defined behavior.

parity

A method for checking the accuracy of data by calculating the sum of the number of ones in a piece of binary data. Even parity requires the correct sum to be an even number; odd parity requires the correct sum to be an odd number.

pipeline

A CPU design technique whereby multiple instructions are simultaneously overlapped in execution.

power down

To initiate the sequence of steps that stops the flow of electricity to a system or its components.

power up

To initiate the sequence of events that starts the flow of electrical current to a system or its components.

primary cache

The cache that is the fastest and closest to the processor.

processor module

The module that contains the CPU chip.

program mode

See operating system mode.

quadword

Eight contiguous bytes starting on an arbitrary byte boundary. The bits are numbered from right to left, 0 through 63.

redundant

Describes duplicate or extra computing components that protect a computing system from failure.

register

A temporary storage or control location in hardware logic.

reliability

The probability a device or system will not fail to perform its intended functions during a specified time interval when operated under stated conditions.

ring topology

A circular LAN configuration that connects a group of nodes.

RISC

Acronym for reduced instruction set computer. Describes a computer with an instruction set that is reduced in complexity.

script

A data structure that points to various tests and exercisers and defines the order in which they are run.

SCSI

Acronym for Small Computer System Interface. An ANSI-standard interface for connecting disks and other peripheral devices to computer systems. *See also* fast SCSI.

SDD

Abbreviation for symptom-directed diagnostics.

self-test

A test that is invoked automatically when the system powers up.

shadowing

See volume shadowing.

shadow set

In volume shadowing, the set of disks on which the data is duplicated. Access to a shadow set is achieved by means of a virtual disk unit. After a shadow set is created, applications and users access the virtual disk unit as if it were a physical disk. *See also* volume shadowing.

smooth shading

An advanced graphics term for the gradual variation of color across a primitive element used to make more realistic pictures by hiding the facets when polygonal meshes are used to approximate smooth surfaces. Such color variation is also used for data mapping in data visualization applications.

SPEC

Acronym for System Performance Evaluation Cooperative. A nonprofit organization that creates and maintains a suite of benchmarks to compare the performance of computer systems across vendors.

SPECmark

The geometric mean of the normalized results from the benchmarks defined by SPEC. SPECmarks are used to measure the performance of a single CPU. *See also* SPEC.

star topology

A LAN configuration in which nodes are connected through a central point.

storage array

A group of mass storage devices, frequently configured as one logical disk.

superpipelined

Describes a pipelined machine that has a larger number of pipe stages and more complex scheduling and control. *See also* pipeline.

superscalar

Describes a machine that issues multiple independent instructions per clock cycle.

system disk

The device on which operating system software resides.

system fatal error

An error that is fatal to the system operation, because the error occurred in the context of a system process or the context of an error cannot be determined.

TCP/IP

Abbreviation for Transmission Control Protocol/Internet Protocol. A set of software communications protocols widely used in UNIX operating environments. TCP delivers data over a connection between applications on different computers on a network; IP controls how packets (units of data) are transferred between computers on a network.

thickwire

An IEEE standard 802.3-compliant Ethernet network made of standard Ethernet cable. Also called standard Ethernet. *Contrast with* ThinWire.

ThinWire

A Digital trademark used to describe its Ethernet products used for local distribution of data communications. ThinWire Ethernet is an IEEE standard 802.3-compliant (10base2) network. *Contrast with* thickwire.

TPS

Abbreviation for transactions per second. Measurement of CPU and I/O performance of a system.

UETP

Abbreviation for User Environment Test Package. An OpenVMS AXP software package designed to test whether the OpenVMS AXP operating system is installed correctly. UETP puts the system through a series of tests that simulate a typical user environment, by making demands on the system that are similar to those that might occur in everyday use.

VMScluster system

A highly integrated organization of Digital's OpenVMS AXP systems that communicate over a high-speed communications path. VMScluster configurations have all the functions of single-node systems, plus the ability to share CPU resources, queues, and disk storage.

volume shadowing

The process of maintaining multiple copies of the same data on two or more disk volumes. When data is recorded on more than one disk volume, a user has access to critical data even when one volume is unavailable. Also called disk mirroring.

Whetstone

A benchmark for computer system performance that measures floating-point performance. Part of the suite of SPEC benchmarks.

word

Two contiguous bytes (16 bits) starting on an arbitrary byte boundary. The bits are numbered from right to left, 0 through 15.

write back

A cache management technique in which data from a write operation to cache is written into main memory only when the data in cache must be overwritten. This results in temporary inconsistencies between cache and main memory. *Contrast with* write through.

write-enabled

The state of a device when data can be written to it. *Contrast with* write-protected.

write-protected

The state of a device when transfers are prevented from writing information to it. *Contrast with* write-enabled.

write through

A cache management technique in which data from a write operation is copied to both cache and main memory. Cache and main memory data is always consistent. *Contrast with* write back.

Z-buffering

An advanced graphics term for attaching a depth value to every pixel as a means of determining which parts of the objects in a scene must be discarded from the processing because they would be hidden from view by other opaque objects (hidden surface removal).