# AlphaServer 4100

## System Drawer User's Guide

Order Number: EK–4100A–UG. D01

This manual is for anyone who manages, operates, or services the BA30A system drawer in an AlphaServer 4100 system. It covers operation, firmware, initial troubleshooting, and component installation.

**Digital Equipment Corporation**
**Maynard, Massachusetts**

**First Printing, September 1996**

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

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 software, if any, described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license. No responsibility is assumed for the use or reliability of software or equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright © 1996 by Digital Equipment Corporation. All rights reserved.

The following are trademarks of Digital Equipment Corporation: AlphaGeneration, AlphaServer, OpenVMS, StorageWorks, VAX, the AlphaGeneration logo, and the DIGITAL logo.

The following are third-party trademarks: Lifestyle 28.8 DATA/FAX Modem is a trademark of Motorola, Inc. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Ltd. U.S. Robotics and Sportster are registered trademarks of U.S. Robotics. Windows NT is a trademark of Microsoft, Inc. All other trademarks and registered trademarks are the property of their respective holders.

**FCC Notice:** The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense will be required to take whatever measures are required to correct the interference.

**Shielded Cables:** If shielded cables have been supplied or specified, they must be used on the system in order to maintain international regulatory compliance.

**Warning!** This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

**Achtung!** Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

**Avertisement!** Cet appareil est un appareil de Classe A. Dans un environnement résidentiel, cet appareil peut provoquer des brouillages radioélectriques. Dans ce cas, il peut être demandé à l'utilisateur de prendre les mesures appropriées.

# Contents

**Preface** ..... xi

## Part 1: System Drawer

### Chapter 1 System Drawer Overview

1.1 Characteristics ..... 1-2
1.2 Architecture ..... 1-4
1.3 Parts of the System Drawer ..... 1-6
1.3.1 System Motherboard ..... 1-8
1.3.2 CPU Module ..... 1-10
1.3.3 Memory Pair ..... 1-12
1.3.4 Power Control Module ..... 1-14
1.3.5 System Bus to PCI Bus Bridge Module ..... 1-16
1.3.6 PCI Motherboard ..... 1-18
1.3.7 Server Control Module ..... 1-20
1.3.8 Control Panel ..... 1-22
1.3.9 Power Supply ..... 1-24
1.4 Console Terminal ..... 1-26

### Chapter 2 Operation

2.1 Powering Up the System Drawer ..... 2-2
2.2 Power-Up Display ..... 2-4
2.3 Booting Digital UNIX ..... 2-9
2.4 Installing Digital UNIX ..... 2-11
2.5 Booting OpenVMS ..... 2-13
2.6 Installing OpenVMS ..... 2-16
2.7 Booting Windows NT ..... 2-18
2.8 Installing Windows NT ..... 2-20
2.9 Updating Firmware ..... 2-22
2.9.1 Updating Firmware from the Internal CD-ROM ..... 2-24
2.9.2 Updating Firmware from the Internal Floppy Disk — Creating the Diskettes ..... 2-28

2.9.3 Updating Firmware from the Internal Floppy Disk — Performing the Update....2-30
2.9.4 Updating Firmware from a Network Device....2-34
2.9.5 LFU Commands....2-38
2.10 Operating the System Remotely....2-41
2.10.1 Modem Usage....2-42
2.10.2 Entering and Leaving Command Mode....2-46
2.10.3 RCM Commands....2-47
2.10.4 Dial-Out Alerts....2-57
2.10.5 Resetting the RCM to Factory Defaults....2-60
2.10.6 Troubleshooting Guide....2-61
2.10.7 Modem Dialog Details....2-64

## Chapter 3 Troubleshooting

3.1 System Drawer Does Not Power Up....3-2
3.2 Control Panel Display Is Blank....3-4
3.3 Control Panel Display Indicates an Error....3-6

## Chapter 4 Installing Components

4.1 Exposing the System Drawer in a Pedestal....4-2
4.2 Exposing the System Drawer in a Cabinet....4-4
4.3 Installing a CPU Module....4-6
4.4 Installing a Memory Pair....4-8
4.5 Installing a PCI or EISA Card....4-10
4.6 Installing a Power Supply....4-12

# Part 2: User Interface

## Chapter 5 SRM Console

5.1 Switching from SRM to the AlphaBIOS Console....5-2
5.2 Displaying the System Configuration....5-4
5.3 Booting the Operating System....5-12
5.4 Configuring the System....5-14
5.4.1 Configuring DSSI....5-14
5.4.2 Configuring a PCI NVRAM Module....5-16
5.4.3 Configuring EISA and RAID....5-17
5.5 Testing the System....5-18
5.6 Stopping and Starting CPUs....5-20
5.7 Updating Firmware....5-22
5.8 Forcing a System Crash Dump....5-24
5.9 Using Environment Variables....5-25
5.10 Depositing and Examining Data....5-28

5.11 Reading a File ....5-31
5.12 Initializing the System ....5-32
5.13 Finding Help ....5-34
5.14 Command Summary ....5-35
5.15 Environment Variable Summary ....5-40

## Chapter 6 AlphaBIOS Console

6.1 Switching from AlphaBIOS to the SRM Console ....6-2
6.2 Starting AlphaBIOS ....6-4
6.3 Keyboard Conventions and Help ....6-6
6.4 Displaying the System Configuration ....6-8
6.4.1 System Board Configuration ....6-10
6.4.2 Hard Disk Configuration ....6-12
6.4.3 PCI Configuration ....6-14
6.4.4 EISA Configuration ....6-18
6.4.5 SCSI Configuration ....6-19
6.4.6 System Bus Configuration ....6-21
6.4.7 Memory Configuration ....6-22
6.4.8 Integrated Peripherals ....6-23
6.5 Updating Firmware ....6-25
6.6 Setting up the Hard Disk ....6-27
6.6.1 Manually Creating and Deleting Partitions ....6-29
6.6.2 Formatting a FAT Partition ....6-31
6.7 Performing Setup Tasks ....6-33
6.8 Installing Windows NT ....6-37
6.9 Running Utility Programs ....6-39
6.10 Selecting the Version of Windows NT ....6-41
6.10.1 Designating a Primary Operating System ....6-43
6.10.2 Primary Operating System and the Auto Start Option ....6-45

## Chapter 7 Utilities

7.1 Running Utilities from a Graphics Monitor ....7-2
7.2 Running Utilities from a Serial Terminal ....7-4

## Appendix A How to Find Firmware Updates

## Appendix B Hard Disk Partitioning for AlphaBIOS and Windows NT

B.1 Hard Disk Error Conditions ....B-1
B.1.1 Disk Initialization Failed ....B-1
B.1.2 No Hard Disks Found ....B-2
B.1.3 No Partitions on Disk ....B-2

B.2 System Partitions ..... B-2
B.3 How AlphaBIOS Works with System Partitions ..... B-3

## Appendix C Differences Between AlphaBIOS and ARC Firmware

C.1 Revamped User Interface ..... C-1
C.2 Hard Disk Setup ..... C-1
C.3 Changing CMOS and NVRAM Values ..... C-2
C.4 Working with Operating Systems ..... C-3
C.5 Running a Program from the BIOS ..... C-3
C.6 Differences in System Configuration Display ..... C-3
C.7 Resetting to Factory Defaults ..... C-4
C.8 Multilingual Support ..... C-4

## Appendix D Acoustical Data

## Index

## Examples

2–1 Power-Up Display ..... 2-4
2–2 Booting Digital UNIX from a Local Device ..... 2-9
2–3 Installing Digital UNIX ..... 2-11
2–4 Booting OpenVMS ..... 2-13
2–5 Installing OpenVMS ..... 2-16
2–6 Starting LFU from the SRM Console ..... 2-22
2–7 Updating Firmware from the Internal CD-ROM ..... 2-24
2–8 Creating Update Diskettes on an OpenVMS System ..... 2-29
2–9 Updating Firmware from the Internal Floppy Disk ..... 2-30
2–10 Selecting AS4X00FW to Update Firmware from the Internal Floppy Disk ..... 2-33
2–11 Updating Firmware from a Network Device ..... 2-34
2–12 Sample Remote Dial-In Dialog ..... 2-44
2–13 Entering and Leaving RCM Command Mode ..... 2-46
2–14 Configuring the Modem for Dial-Out Alerts ..... 2-57
2–15 Typical RCM Dial-Out Command ..... 2-57
5–1 Switching to the AlphaBIOS Console ..... 5-2
5–2 Show Config Command ..... 5-4
5–3 Show CPU Command ..... 5-5
5–4 Show Device Command ..... 5-6
5–5 Show FRU Command ..... 5-8
5–6 Show Memory Command ..... 5-9
5–7 Show Network Command ..... 5-9

5–8 Show PAL Command....5-10
5–9 Show Power Command....5-10
5–10 Show Version Command....5-11
5–11 Boot Command....5-12
5–12 Set Host Command....5-14
5–13 Prcache Command....5-16
5–14 Test Command....5-18
5–15 Start, Halt, and Continue Commands....5-20
5–16 Lfu Command....5-22
5–17 Crash Command....5-24
5–18 Clear, Set, and Show *envar* commands....5-25
5–19 Edit Command....5-27
5–20 Deposit Command....5-28
5–21 Examine Command....5-28
5–22 More Command....5-31
5–23 Initialize Command....5-32
5–24 Help (or Man) Command....5-34

## Figures

1–1 BA30A System Drawer....1-2
1–2 System Drawer Block Diagram....1-4
1–3 System Drawer Sections....1-6
1–4 System Motherboard....1-8
1–5 CPU Modules....1-10
1–6 Memory Module....1-12
1–7 Power Control Module....1-14
1–8 Bridge Module....1-16
1–9 PCI Motherboard....1-18
1–10 Server Control Module....1-20
1–11 Control Panel....1-22
1–12 Power Supply....1-24
1–13 Comparison of Console Terminals....1-26
2–1 Power Button and Control Panel Display....2-2
2–2 AlphaBIOS Boot Screen....2-18
2–3 Installing Windows NT....2-20
2–4 Starting LFU from the AlphaBIOS Console....2-22
2–5 RCM Connections....2-42
3–1 Location of Cover Interlocks....3-2
3–2 Power Control Module LEDs....3-3
3–3 CPU and Bridge Module LEDs....3-4
3–4 Control Panel Display....3-6
4–1 Pedestal Enclosure....4-2
4–2 Cabinet Enclosure....4-4

4–3 CPU Module Installation ....4-6
4–4 CPU Module Slots....4-7
4–5 Memory Module Installation ....4-8
4–6 Memory Pair Slots....4-9
4–7 PCI/EISA Card Installation ....4-10
4–8 PCI and EISA Slots....4-11
4–9 Power Supply Cover....4-12
4–10 Power Control Module LEDs ....4-14
5–1 Configuring EISA and RAID Devices ....5-17
6–1 Switching to the SRM Console ....6-2
6–2 Boot Screen....6-4
6–3 AlphaBIOS Setup Screen ....6-5
6–4 Typical First-Level Help Screen ....6-6
6–5 Second-Level Help Screen ....6-7
6–6 Display System Configuration Screen ....6-8
6–7 System Board Configuration ....6-10
6–8 Hard Disk Configuration ....6-12
6–9 PCI Configuration ....6-14
6–10 Advanced PCI Information....6-17
6–11 EISA Configuration....6-18
6–12 SCSI Configuration ....6-19
6–13 System Bus Configuration....6-21
6–14 Memory Configuration....6-22
6–15 Integrated Peripherals....6-23
6–16 Updating Firmware ....6-25
6–17 Hard Disk Setup Screen....6-27
6–18 Create New Partition Dialog Box ....6-29
6–19 Delete Partition Dialog Box ....6-30
6–20 Formatting a FAT Partition ....6-31
6–21 Standard Formatting....6-32
6–22 Standard CMOS Setup Screen ....6-33
6–23 Advanced CMOS Setup Screen ....6-35
6–24 Installing Windows NT ....6-37
6–25 Run Maintenance Program Dialog Box ....6-39
6–26 Operating System Selections ....6-41
6–27 Primary Operating System....6-43
6–28 Operating System Selection Setup ....6-45
7–1 Running a Utility from a Graphics Monitor ....7-2
7–2 Run Maintenance Program Dialog Box ....7-3
B–1 System Partition Not Defined ....B-3

## Tables


1 AlphaServer 4100 Documentation ..... xii
1–1 BA30A System Drawer Characteristics ..... 1-3
1–2 PCI Motherboard Slot Numbering ..... 1-19
2–1 File Locations for Creating Update Diskettes on a PC ..... 2-28
2–2 LFU Command Summary ..... 2-38
2–3 RCM Command Summary ..... 2-47
2–4 RCM Status Command Fields ..... 2-56
2–5 RCM Troubleshooting ..... 2-61
2–6 RCM/Modem Interchange Summary ..... 2-66
3–1 Control Panel Display ..... 3-7
5–3 Device Naming Convention ..... 5-7
5–4 Summary of SRM Console Commands ..... 5-35
5–5 Syntax for SRM Console Commands ..... 5-37
5–6 Special Characters for SRM Console ..... 5-38
5–7 Environment Variable Summary ..... 5-40
5–8 Settings for boot_osflags Bootflags (OpenVMS) ..... 5-44
7–1 AlphaBIOS Option Key Mapping ..... 7-4
C–1 Differences Between AlphaBIOS and ARC Menu Selections ..... C-2

# Preface

## Intended Audience

This manual is for anyone who manages, operates, or services the BA30A system drawer in an AlphaServer 4100 system.

## Document Structure

This manual uses a structured documentation design. Topics are organized into small sections for efficient online and printed reference. Each topic begins with an abstract. You can quickly gain a comprehensive overview by reading only the abstracts. Next is an illustration or example, which also provides quick reference. Last in the structure are descriptive text and syntax definitions.

This manual has seven chapters and three appendixes, as follows:

**Part 1: System Drawer**

- **Chapter 1, System Drawer Overview,** describes the components in the BA30A system drawer.
- **Chapter 2, Operation,** gives instructions for powering the system drawer up and down and for booting the operating system.
- **Chapter 3, Troubleshooting,** provides basic troubleshooting information for the system drawer.
- **Chapter 4, Installing Components,** gives instructions for adding CPU, memory, PCI, and EISA cards to the system drawer.

**Part 2: User Interface**

- **Chapter 5, SRM Console,** describes the SRM console commands.
- **Chapter 6, AlphaBIOS Console,** describes the AlphaBIOS menu selections.
- **Chapter 7, Utilities,** describes utilities that configure options, update firmware, and reach a console when a power-up problem prevents the usual startup routine.
- **Appendix A, How to Find Firmware Updates,** explains the process for downloading AlphaServer firmware from the Internet.

- **Appendix B, Hard Disk Partitioning for AlphaBIOS and Windows NT,** describes the concept of partitioning for AlphaBIOS and Windows NT.
- **Appendix C, Differences Between AlphaBIOS and ARC Firmware,** describes the differences.
- **Appendix D, Acoustical Data,** lists values for the system drawer in various configurations.

## Documentation Titles

Table 1 lists the books in the AlphaServer 4100 documentation set.

### Table 1 AlphaServer 4100 Documentation

| Title | Order Number |
|---|---|
| **User and Configuration Documentation Kit** | QZ–00VAA–GZ |
| *System Drawer User's Guide* | EK–4100A–UG |
| *Configuration and Installation Guide* | EK–4100A–CG |
| **Service Information** | |
| *Service Manual* (hard copy) | EK–4100A–SV |
| *Service Manual* (diskette) | AK–QXBJA–CA |
| **Reference Information** | |
| *KN30n CPU Installation Card* | EK–KN300–IN |
| *MS3n0 Memory Installation Card* | EK–MS300–IN |
| *H7291 Power Supply Installation Card* | EK–H7291–IN |
| *ServerWORKS Manager Administrator User's Guide* | ER–4QXAA–UA |

# Part 1

# System Drawer

# Chapter 1

# System Drawer Overview

The BA30A system drawer is the enclosure for AlphaServer 4100 processor and memory modules, system bus, and PCI bus. The same drawer is used in various AlphaServer 4100 models, such as the pedestal system and the cabinet system.

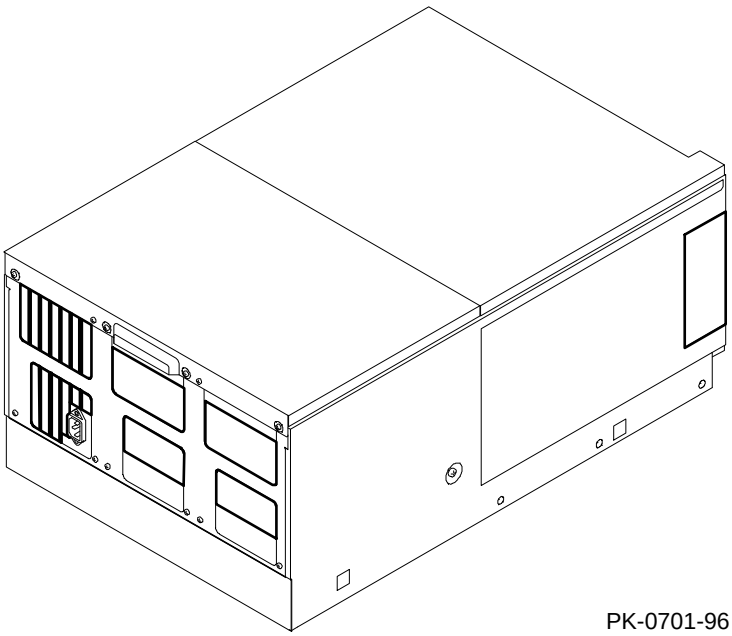
This chapter provides an overview of the BA30A system drawer in these sections:

- Characteristics
- Architecture
- Parts of the System Drawer
- Console Terminal

## 1.1 Characteristics

**The illustration shows the BA30A system drawer with covers in place. The table lists characteristics of the drawer.**

**Figure 1–1 BA30A System Drawer**



PK-0701-96

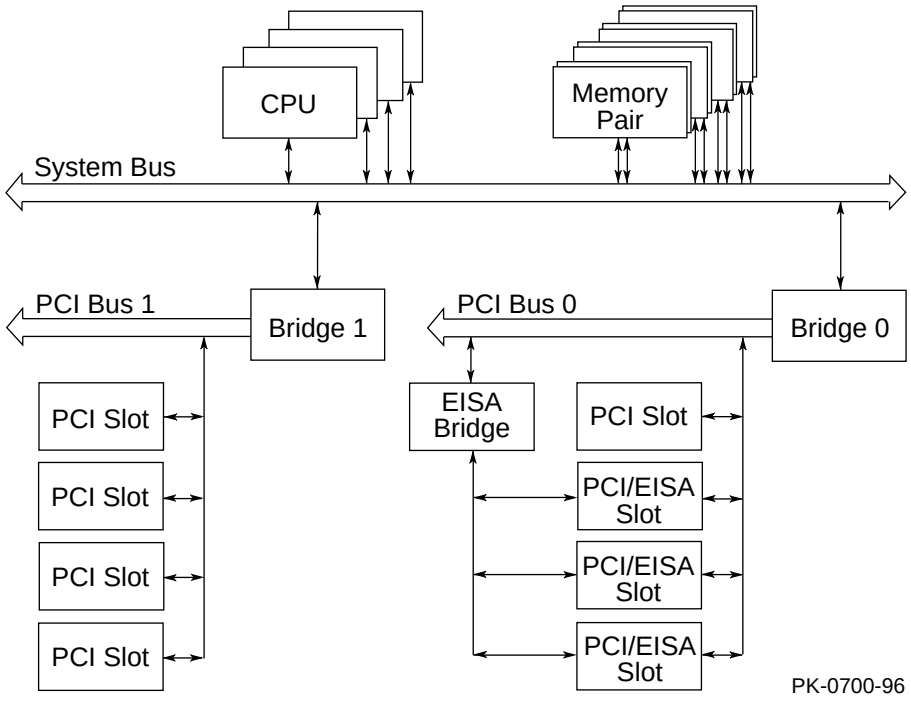
**Table 1–1 BA30A System Drawer Characteristics**

| Characteristic | Specification |
|---|---|
| Size | 30 cm H x 45 cm W x 69 cm D (11.8 in. x 17.7 in. x 27.2 in.) |
| Weight | 45.5 kg (100 lb) fully configured |
| Clearance | 20 cm (7.9 in.) front and rear |
| Maximum heat dissipation | 1100 w (3754 Btu/hr) |
| **Electrical** | |
| Voltage | 100–120/200–240 Vac |
| Phase | Single |
| Frequency | 50–60 Hz |
| Maximum rated current (single cord set) | 8.4 A at 100–120 V<br>4.2 A at 200–240 V |
| Maximum rated current (multiple cord sets) | 8.0 A at 100–120 V<br>4.0 A at 200–240 V |
| Maximum power consumption | 700 w |
| **Environmental** | |
| Temperature | Operating: 10° to 35°C (50 to 95°F)<br>Not operating: -40° to 66°C (-40° to 150.8°F)<br>Storage (60 days): -40° to 66°C (-40° to 150.8°F)<br>Rate of change: 11° to 19°C per hour (20° to 35°F per hour) |
| Humidity | Operating: 20% to 90% (noncondensing)<br>Not operating: 10% to 95%<br>Storage (60 days): 10% to 95% |
| Altitude | Operating: 2000 m (6560 ft)<br>Not operating: 3600 m (11808 ft) |

## 1.2 Architecture

**The system bus, with its logic on the system motherboard, connects the CPU, memory, and system bus to PCI bus bridge modules. The two 64-bit PCI buses each have four slots for PCI options. (Three of the eight slots can be used for either PCI or EISA options.)**

**Figure 1–2 System Drawer Block Diagram**



The **system bus** is a 144-bit data bus (128 bits for data and 16 for ECC) and a separate 40-bit command/address bus. The logic is on the system motherboard. This bus connects CPU modules, memory modules, and the system bus to PCI bus bridge module. The clock for the system bus is on the module in CPU slot 0.

**CPU modules** are available with and without an external cache. The Alpha chip has an 8-KB instruction cache, an 8-KB write-through data cache, and a 96-KB-write-back secondary data cache. The on-board cache version of the module has a 2- or 4-MB data cache. The default cache system is write-back. The system drawer supports up to four CPUs.

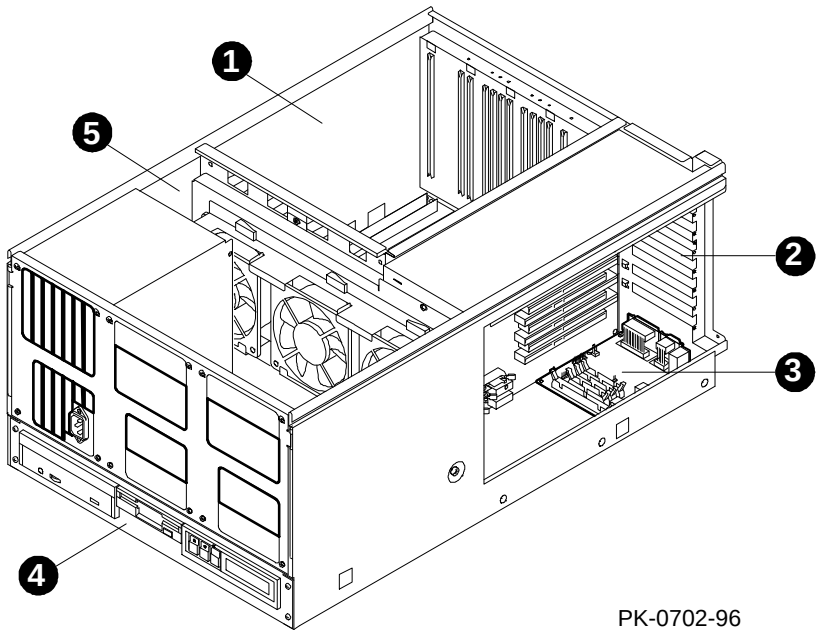
A **memory pair** is two modules of the same size and type, which together comprise one memory option. Memory pairs can be 128 or 512 MB or 1 GB. Each module in the pair drives 72 bits of the 144-bit system bus. The system drawer can have up to four memory pairs.

The **system bus to PCI bus bridge module** supports two 64-bit **PCI buses**. One PCI bus has four slots for PCI options; the other has one dedicated PCI slot and three slots that can be used for either PCI or EISA options.

## 1.3 Parts of the System Drawer

**The system drawer consists of four sections: system card cage, PCI card cage, control panel, and power and cooling section.**

**Figure 1–3 System Drawer Sections**



The callouts in Figure 1–3 point to these components of the BA30A system drawer:

❶ System card cage, which holds the system motherboard and the CPU, memory, bridge, and power control modules. Refer to Sections 1.3.1 through 1.3.5 for information about these modules.

❷ PCI card cage, which holds the PCI motherboard and option cards and the server control module. Refer to Section 1.3.6 for information about the motherboard.

❸ Server control module, which holds the I/O connectors and remote console monitor. Refer to Section 1.3.7.

❹ Control panel assembly, which includes the control panel, a floppy drive, and a CD-ROM drive. Refer to Section 1.3.8.

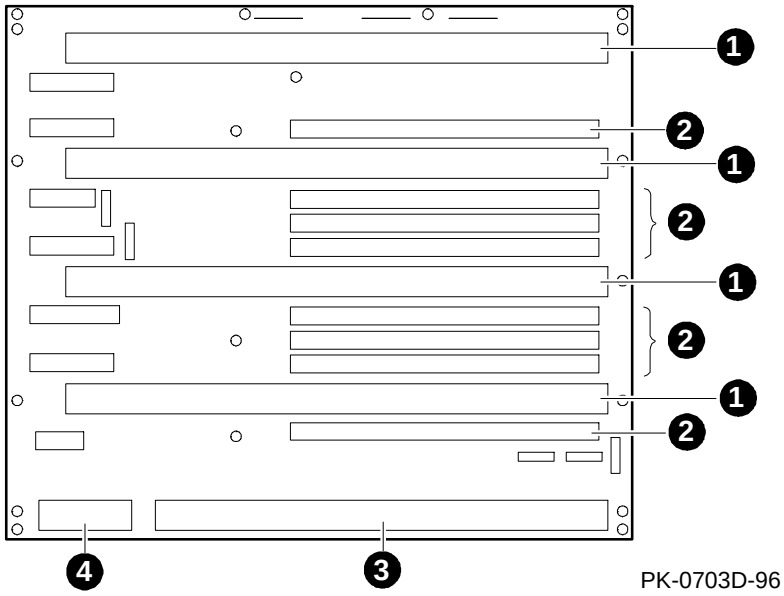
*NOTE: When the system drawer is mounted in a pedestal cabinet, the control panel assembly is mounted in a tray at the top of the drawer.*

❺ Power and cooling section, which contains one to three power supplies (refer to Section 1.3.9) and three fans.

### 1.3.1 System Motherboard

**The system motherboard is on the floor of the system card cage. It has slots for the CPU, memory, power control, and bridge modules.**

**Figure 1–4 System Motherboard**



The system motherboard has the logic for the system bus. It is the backplane that holds the CPU, memory, bridge, and power control modules. Figure 1–4 shows the locations of these modules and of connectors on the motherboard.

❶ CPU module

❷ Memory module

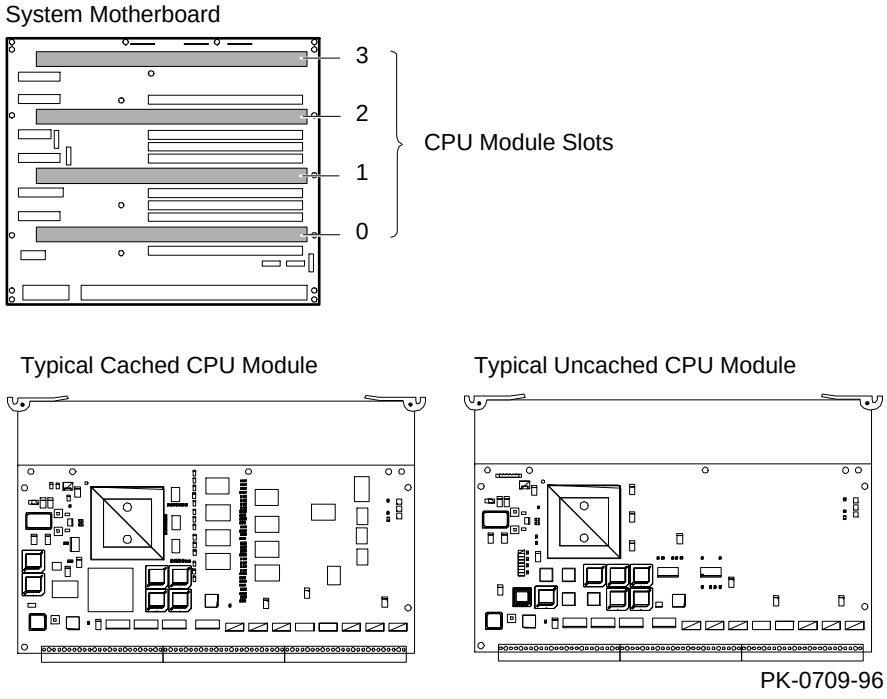
❸ Bridge module

❹ Power control module

## 1.3.2 CPU Module

**The CPU module comes with or without an on-board cache, and each module has a fan for cooling. The BA30A system drawer can hold up to four CPU modules.**

### Figure 1–5 CPU Modules



### Description

CPU modules are available in two types: the cached CPU module has a 2- or 4-MB synchronous data cache external to the Alpha microprocessor chip, and the uncached does not. The system drawer can have up to four CPU modules. Each CPU module has a fan for cooling.

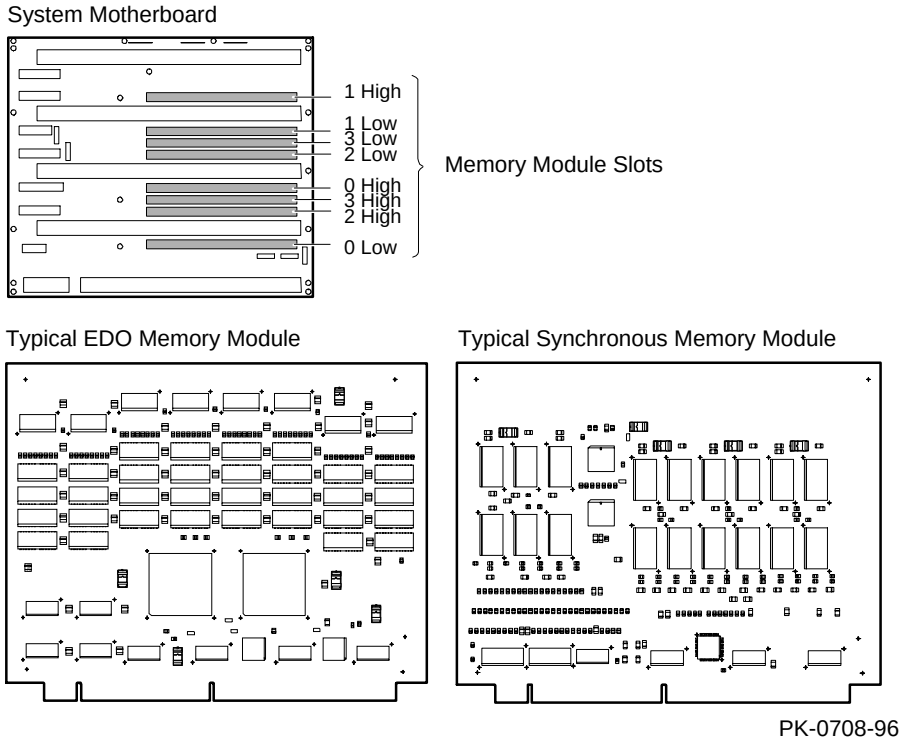
### Configuration

- The first CPU module must be in CPU slot 0 to provide the system clock.
- Additional CPU modules should be installed in ascending order by slot number.
- All CPU modules must have the same clock frequency.
- Combining cached and uncached CPU modules is not supported.

## 1.3.3 Memory Pair

**Memory modules are used only in pairs — two modules of the same size and type. Each module provides either the low half or the high half of the memory space. The system drawer can hold up to four memory module pairs.**

### Figure 1–6 Memory Module



### Description

Memory modules are used only in pairs. A memory module pair is two modules of the same size and type; one module in the pair provides the low half and the other the high half of that pair's memory region. Two types of memory modules are available: synchronous and extended data out (EDO).

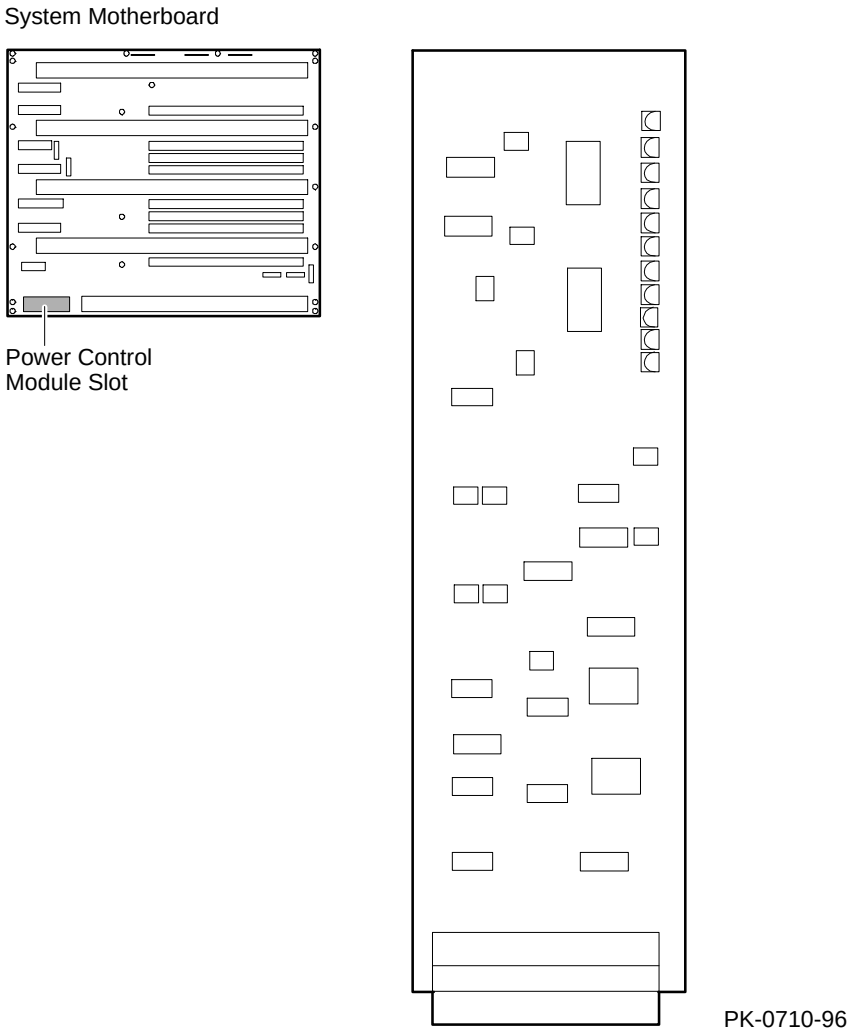
### Configuration

- Memory modules are installed and used in pairs. Both modules in a memory pair must be of the same size and type.
- Memory must be configured in descending order. That is, the largest memory pair must be in slots 0 Low and 0 High, the second largest pair in 1 Low and 1 High, and so on. (The first memory pair defines the size of the memory region for each of the other pairs.)

## 1.3.4 Power Control Module

**The power control module controls power sequencing and monitors power supply voltage, temperature, and fans.**

### Figure 1–7 Power Control Module



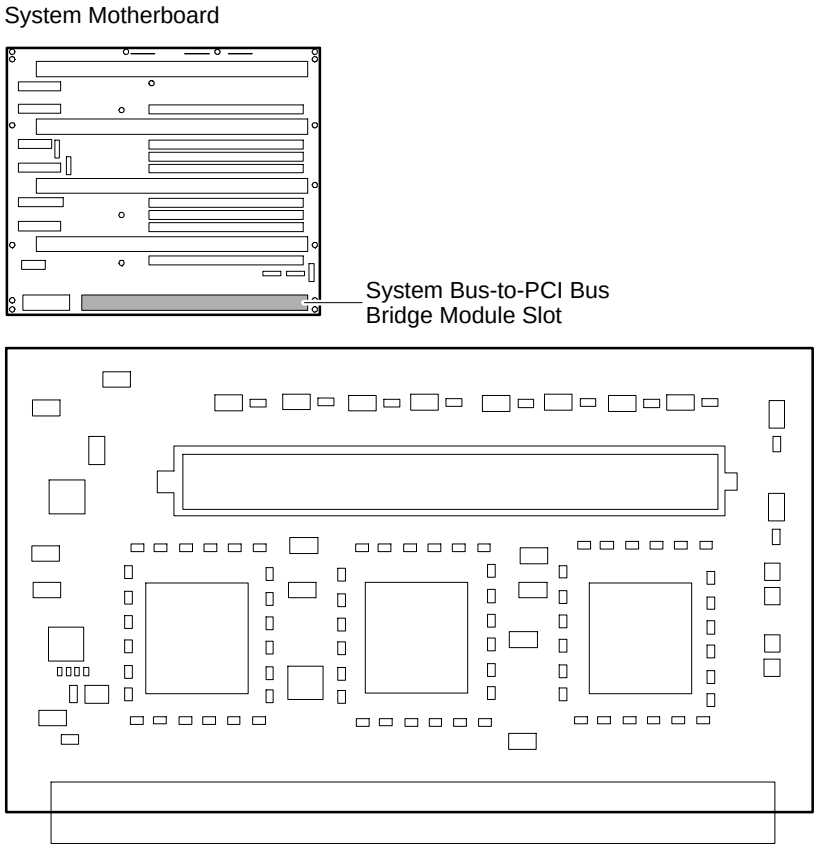
The power control module performs these functions:

- Controls power sequencing.
- Monitors the combined output of power supplies and shuts down power if it is not in range.
- Monitors system temperature and shuts off power if it is out of range.
- Monitors the fans in the system drawer and on the CPU modules and shuts down power if a fan fails.
- Provides visual indication of faults through LEDs.

## 1.3.5 System Bus to PCI Bus Bridge Module

**The bridge module is the physical interconnect between the system motherboard and the PCI motherboard.**

### Figure 1–8 Bridge Module



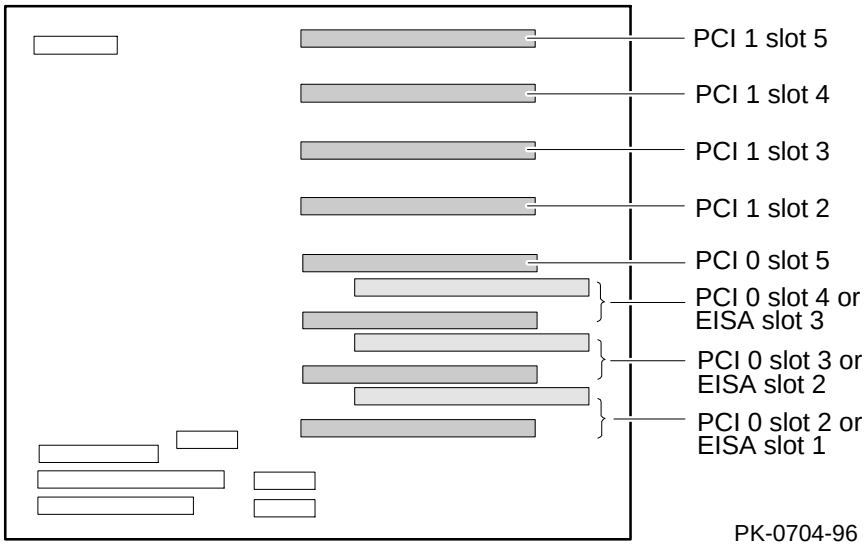
The bridge module is the physical interconnect between the system motherboard and the PCI motherboard. Each side of the module is a bridge to one PCI bus.

This module converts system bus commands and data addressed to I/O space to PCI commands and data, and it converts PCI bus commands and data addressed to system memory or CPUs to system bus commands and data.

### 1.3.6 PCI Motherboard

**The PCI motherboard has the logic for two PCI buses. PCI 0 consists of one dedicated PCI slot and three shared PCI/EISA slots; PCI 1 has four PCI slots.**

**Figure 1–9 PCI Motherboard**



The logic for two 64-bit PCI buses is on the PCI motherboard. PCI 0 has one dedicated PCI slot and three shared PCI/EISA slots, and PCI 1 has four PCI slots. This module has cable connections to remote I/O (mouse, keyboard, serial port, and parallel port) and to an internal floppy drive, an internal CD-ROM drive, the control panel, and 5V power. Also on this module are the chips for the PCI to EISA bridge and the internal CD-ROM controller.

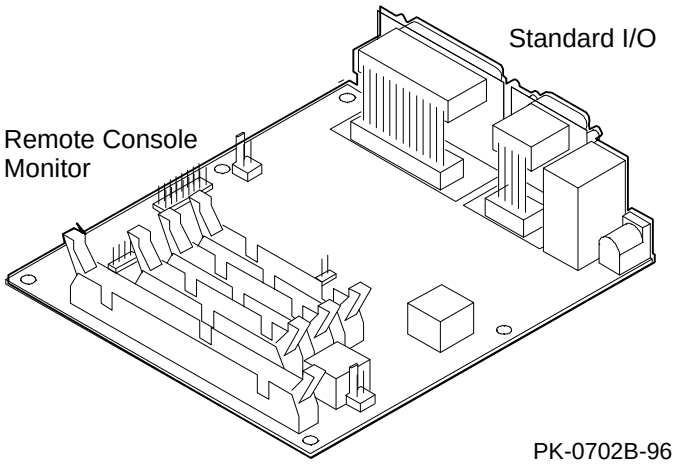
**Table 1–2 PCI Motherboard Slot Numbering**

| Slot | PCI 0 | PCI 1 |
|---|---|---|
| 0 | Reserved | Reserved |
| 1 | PCI to EISA bridge | Internal CD-ROM controller |
| 2 | PCI or EISA slot | PCI slot |
| 3 | PCI or EISA slot | PCI slot |
| 4 | PCI or EISA slot | PCI slot |
| 5 | PCI slot | PCI slot |

### 1.3.7 Server Control Module

**The server control module enables remote connections to the system drawer's console. It also holds the standard I/O connectors.**

**Figure 1–10 Server Control Module**



The server control module has two sections: the remote console monitor and the standard I/O connectors.

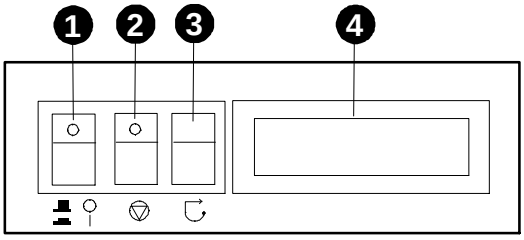
The remote console monitor connects to a modem through the modem port on the bulkhead, and it receives power through the 12V power connector.

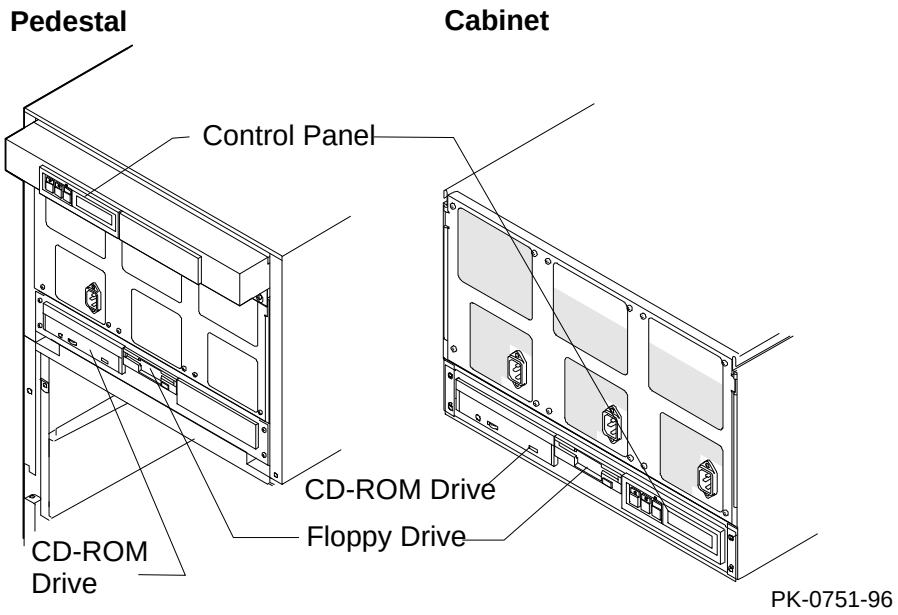
The standard remote I/O ports (keyboard, mouse, COM1 and COM2 serial, and parallel ports) are on the same bulkhead.

## 1.3.8 Control Panel

**The control panel includes the Power, Halt, and Reset buttons and a display. In a pedestal system the control panel is located in a tray at the top of the system drawer. In a cabinet system it is at the bottom of the system drawer with the CD-ROM drive and the floppy drive.**

**Figure 1–11 Control Panel**



The callouts in Figure 1–11 point to these components of the control panel:

❶ Power button — Powers on or off the system drawer. When the LED at the top of the button is lit, the power is on.

*NOTE: The LEDs on some modules are on when the line cord is plugged in, regardless of the position of the Power button.*

❷ Halt button — Pressing this button in (so the LED at the top of the button is lit) does the following:

- If Digital UNIX or OpenVMS is running, halts the operating system and returns to the SRM console.
- If the Halt button is in when the system is reset or powered up, the system halts in the SRM console, regardless of the operating system. Digital UNIX and OpenVMS systems that are configured for autoboot will not boot if the Halt button is in. Windows NT systems halt in the SRM console; AlphaBIOS is not loaded and started.

❸ Reset button — Initializes the system drawer.

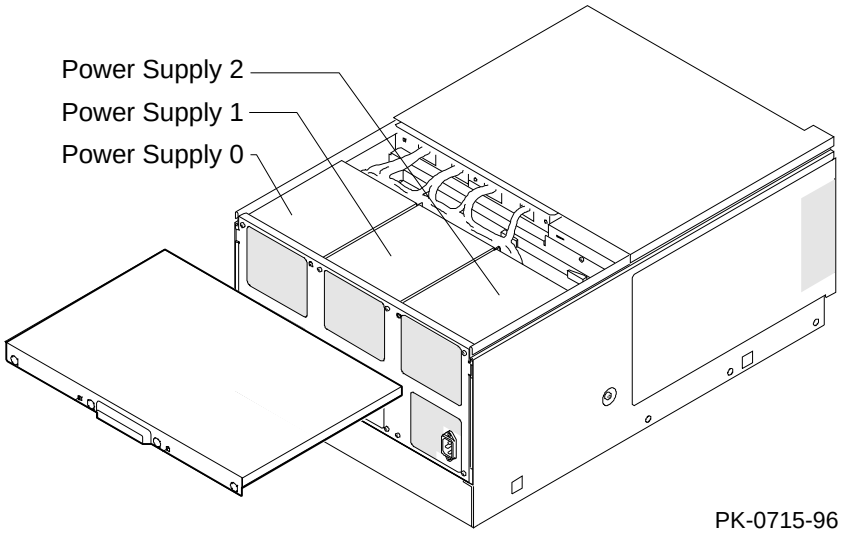
❹ Control panel display — Indicates status during power-up and testing. See Table 3–1 for the messages that are displayed and their meanings.

While the operating system is running, displays the system type as a default. This message can be changed by the user.

### 1.3.9 Power Supply

**The system drawer power supplies provide power only to components in the drawer. One or two power supplies are required, depending on the number of CPU modules, and a second or third can be added for redundancy.**

**Figure 1–12 Power Supply**



### Description

One to three power supplies provide power to components in the system drawer. (They supply power only for the drawer in which they are located.)

These power supplies share the load, and redundant configurations are supported. They autoselect line voltage (120V to 240V). Each has 450 W output and supplies up to 75A of 3.43V, 50A of 5.0V, 11A of 12V, and small amounts of -5V, -12V, and auxiliary voltage (Vaux).

*NOTE: The LEDs on some modules are on when the line cord is plugged in, regardless of the position of the Power button.*

### Configuration

- Systems with one or two CPUs require one power supply (two for redundancy).
- Systems with three or four CPUs require two power supplies (three for redundancy).
- Power supply 0 is installed first, power supply 2 second, and power supply 1 third. See Figure 1–12. (The power supply numbering shown here corresponds to the numbering displayed by the SRM **show power** command.)

## 1.4 Console Terminal

**The console terminal can be either a serial terminal or a graphics monitor. The power-up display prints to this terminal.**

### Figure 1–13 Comparison of Console Terminals

| Condition | Serial Terminal | Graphics Monitor |
|---|---|---|
| Does the SROM power-up display print? | Yes | No |
| Does the XSROM power-up display print? | Yes | No |
| Does the SRM console power-up display print? | Yes | Yes |
| Is it supported by the SRM console? | Yes | Yes |
| Is it supported by the AlphaBIOS console? | Yes | Yes |
| Is it supported by the Digital UNIX operating system? | Yes | Yes |
| Is it supported by the OpenVMS operating system? | Yes | Yes |
| Is it supported by the Windows NT operating system? | No | Yes |

The console terminal can be a serial (character cell) terminal connected to the COM1 port or a graphics monitor connected to either a TGA adapter on PCI0 or PCI1 or a VGA adapter on PCI0. If the console terminal is connected to COM1, the entire power-up display prints. (See Section 2.2 for information about the power-up display.) If the console terminal is either a TGA or VGA monitor, console power-up messages are sent to either of those devices but SROM and XSROM power-up messages are lost.

When the console terminal is a serial terminal, the **console** environment variable is set to **serial**. When it is a graphics monitor, **console** is set to **graphics**. (See Section 5.15 for information about environment variables.)

*NOTE: Regardless of the setting of the **console** environment variable, all three power-up programs send messages to the control panel display.*

If the **console** environment variable is set to **serial**, **os_type** is set to **unix** or **openvms**, and no terminal is attached to COM1, pressing a carriage return on a graphics terminal attached to the keyboard port makes it the console device and the console prompt is sent to it.

If the **console** environment variable is set to **graphics** and no graphics monitor is attached to the keyboard port, pressing a carriage return on a serial terminal attached to COM1 makes it the console device and the console prompt is sent to it.

*NOTE: The console prompt displays only after the entire power-up sequence is complete. This can take up to several minutes if the memory is very large.*

# Chapter 2

# Operation

This chapter provides basic operating instructions, including powering up the system drawer and booting the operating system. It also provides information about updating firmware and operating the system from a remote location.

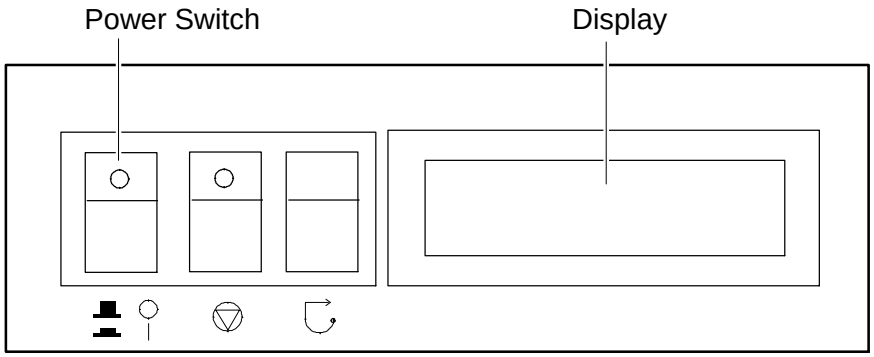
Sections in this chapter are:

- Powering Up the System Drawer
- Power-Up Display
- Booting Digital UNIX
- Installing Digital UNIX
- Booting OpenVMS
- Installing OpenVMS
- Booting Windows NT
- Installing Windows NT
- Updating Firmware
- Operating the System Remotely

## 2.1 Powering Up the System Drawer

**To power up the system drawer, press the Power button (the LED at the top of the button lights). Testing begins, and status shows on the screen and in the control panel display.**

**Figure 2–1 Power Button and Control Panel Display**




PK-0706C-96

Power up the system drawer by pressing the Power button (see Figure 2–1). Testing begins, and screen text similar to that in Example 2–1 displays (if the console terminal is a serial terminal connected to the COM1 port), along with status messages in the control panel display. (See Figure 2–1 for display location. Refer to Table 3–1 for information about the messages in the control panel display.)

*NOTE: If the console terminal is a graphics monitor, only the last few lines of the power-up display print. See Section 2.2 for more information.*

## 2.2 Power-Up Display

**The entire power-up display prints to a serial terminal (if the console environment variable is set to serial); the last several lines print to either a serial terminal or a graphics monitor. Power-up status also displays on the control panel display.**

### Example 2–1 Power-Up Display

```
 SROM V2.0 on cpu0                                          ❶
 SROM V2.0 on cpu1
 SROM V2.0 on cpu2
 SROM V2.0 on cpu3
XSROM V2.0 on cpu2                                          ❷
XSROM V2.0 on cpu1
XSROM V2.0 on cpu3
XSROM V2.0 on cpu0
BCache testing complete on cpu2                             ❸
BCache testing complete on cpu0
BCache testing complete on cpu3
BCache testing complete on cpu1
mem_pair0 - 128 MB                                          ❹
mem_pair1 - 128 MB
20..20..21..20..21..20..21..21..23..24..24..24..24..        ❺
Memory testing complete on cpu0
Memory testing complete on cpu1
Memory testing complete on cpu3
Memory testing complete on cpu2
```

*Continued on next page*

❶ On power-up, the SROM code on each CPU module loads into that module's I-cache and tests the processor chip. If all tests pass, the module's LED lights. If any test fails, the LED remains clear and power-up terminates.

The primary CPU is determined for the first of three times. The primary CPU then executes a loopback test on each PCI bridge. If this test passes, the bridge LED lights. If it fails, the LED remains clear and power-up continues. The EISA system controller, PCI-to-EISA bridge, COM1 port, and control panel port are all initialized.

Each CPU prints an SROM message to the device attached to the COM1 port and to the control panel display. (The banner prints to the COM1 port if the **console** environment variable is set to **serial**. If it is set to **graphics**, nothing prints to the console terminal, only to the control panel display, until ❻. See Section 5.15 for information about environment variables.)

❷ The S-cache on each CPU module is initialized, and the XSROM code in the FEPROM on the PCI motherboard is unloaded into them. (If the unload is not successful, the SROM unloads XSROM code from a different FEPROM sector. If the second try fails, the CPU hangs.)

Each CPU starts the XSROM code, which prints an XSROM message to the COM1 port and to the control panel display.

❸ The three S-cache banks on each CPU are enabled, and the B-cache is tested. If a failure occurs, a message is printed to the COM1 port and to the control panel display immediately.

Each CPU prints a B-cache completion message to COM1.

❹ The primary CPU is determined for the second time. It then sizes memory.

Information on memory pairs is printed to the COM1 port. If an illegal memory configuration is detected, a warning message is printed to the COM1 port and the control panel display.

❺ Memory is initialized and tested, and test traces are printed to the COM1 port and the control panel display. (The numbers for tests 20 and 21 might appear interspersed; this is normal behavior. Test 24 can take several minutes if the memory is very large. The message "P0 TEST 24 MEM**" is displayed on the control panel display; the second asterisk rotates to indicate that testing is continuing. All CPUs in a multiprocessor system share the memory testing to minimize test time.) If a failure occurs, a message is printed to the COM1 port and to the control panel display immediately.

Each CPU prints a test completion message to the COM1 port.

*Continued on next page*

**Example 2–1 Power-Up Display (Continued)**

```
starting console on CPU 0                                    ❻
sizing memory                                                ❼
  0    128 MB SYNC
  1    128 MB SYNC
starting console on CPU 1
starting console on CPU 2
starting console on CPU 3
probing IOD1 hose 1                                          ❽
  bus 0 slot 1 - NCR 53C810
  bus 0 slot 2 - DEC PCI MC
  bus 0 slot 3 - NCR 53C810
  bus 0 slot 4 - DECchip 21040-AA
  bus 0 slot 5 - DEC KZPSA
probing IOD0 hose 0
  bus 0 slot 1 - PCEB
  bus 0 slot 4 - DEC PCI FDDI
  bus 0 slot 5 - PBXGA-AA
Configuring I/O adapters...
  ncr0, hose 1, slot 1, bus 0
  kzpsa0, hose 1, slot 2, bus 0
  ncr1, hose 1, slot 3, bus 0
  tulip0, host 1, slot 4, bus 0
  floppy0, host 0, slot 0, bus 1
  pfi0, hose 0, slot 4, bus 0

System temperature is 24 degrees C
AlphaServer 4100 Console V2.0-1, 24-JUL-1996 18:18:26 ❾
P00>>>
```

❻ The primary CPU is determined for the last time. The primary CPU unloads PALcode and decompression code from the FEPROM on the PCI motherboard to its B-cache. The primary CPU then jumps to the PALcode to start the SRM console.

The primary CPU prints a message indicating that it is running the console. Starting with this message, the power-up display is printed to any console terminal, regardless of the state of the **console** environment variable. (If **console** is set to **graphics**, the display from here to the end is saved in a memory buffer and printed to the graphics monitor after the PCI buses are sized and the VGA device is initialized.)

❼ The size and type of each memory pair is determined.

The console is started on each of the secondary CPUs. A status message prints for each CPU.

❽ The PCI bridges (indicated as IOD*n*) are probed and the devices are reported. I/O adapters are configured.

❾ The SRM console banner and prompt are printed. (The SRM prompt is shown in this manual as P00>>>. It can, however, be P01>>>, P02>>>, or P03>>>. The number indicates the primary processor.) If the **auto_action** environment variable is set to **boot** or **restart** and the **os_type** environment variable is set to **unix** or **openvms**, the Digital UNIX or OpenVMS operating system boots.

If the system is running the Windows NT operating system (the **os_type** environment variable is set to **nt**), the SRM console loads and starts the AlphaBIOS console.

Refer to Chapter 5 for information about the SRM console and to Chapter 6 for AlphaBIOS.

## 2.3 Booting Digital UNIX

**Digital UNIX® can be booted from a local disk, a disk connected through a cluster, or a remote disk through an Ethernet connection. Refer to the documentation shipped with the operating system for booting instructions.**

### Example 2–2 Booting Digital UNIX from a Local Disk

```
P00>>> show device                                              ❶
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1      DKa500                        RRD45   1645
polling ncr1 (NCR 53C810) slot 3, bus 0 PCI, hose 1   SCSI Bus ID 7
dkb100.1.0.3.1      DKb100                        RZ29B   0007
dkb200.2.0.3.1      DKb200                        RZ29B   0007
dkb400.4.0.3.1      DKb400                        RZ29B   0007 ❷
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0     DVA0                           RX23
polling tulip0 (DECchip 21040-AA) slot 2, bus 0 PCI, hose 1
ewa0.0.0.2.1: 08-00-2B-E5-B4-1A

P00>>> boot -file vmunix -flags a dkb400                        ❸
(boot dkb4.4.0.3.1 -file vmunix -flags a)
Building FRU table
block 0 of dkb4.4.0.3.1 is a valid boot block
reading 16 blocks from dkb4.4.0.3.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

OSF boot - Mon Aug 5 21:56:39 EDT 1996

Loading vmunix ...
.
.
.
```

*Continued on next page*

### Example 2–2 Booting Digital UNIX from a Local Disk (Continued)

```
The system is ready.


Digital UNIX Version 3.2D                                    ❹
```

❶ The **show device** command displays device information, including name and type of connection to the system. See Section 5.2 for a description of the **show device** command and the device naming convention.

❷ The operating system is on the third disk connected to the system through the controller in slot 3 of PCI1. The name of this device, dkb400, is used as an argument to the **boot** command.

❸ This command loads Digital UNIX from the disk dkb400, using the boot file vmunix and autobooting to multiuser mode. See Section 5.3 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 5.15 for information about environment variables.

❹ The operating system banner displays.

### Example 2–3 Booting Digital UNIX from a Disk on a Cluster

```
P00>>> show boot*                                                    ❶
boot_arg0               0
boot_arg1
boot_dev                dkb0.0.0.5.1
boot_file               vmunix
boot_osflags            a
boot_reset              OFF
bootdef_dev             dkb0.0.0.5.1
booted_dev
booted_file
booted_osflags

P00>>> boot                                                          ❷
(boot dkb0.0.0.5.1 -flags a)
halted CPU 1
halt code = 1
operator initiated halt
PC = fffffc0000264600
Building FRU table
block 0 of dkb0.0.0.5.1 is a valid boot block
reading 16 blocks from dkb0.0.0.5.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
Configuring I/O adapters...
jumping to bootstrap code

OSF boot - Mon Jul 29 21:56:39 EDT 1996

Loading vmunix ...

.
.
.

The system is ready.


Digital UNIX Version 3.2D                                            ❸
```

❶ The **show boot*** command lists all environment variables that begin with the string "boot." In this example, the **bootdef_dev** environment variable sets the default boot device to dkb0.0.0.5.1, and **boot_osflags** indicates that the system will autoboot to multiuser mode. See Section 5.15 for descriptions of these environment variables.

❷ The **boot** command needs no options, since the environment variables pass all necessary information.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 5.15 for information about environment variables.

❸ The operating system banner displays.

### Example 2–4 Booting Digital UNIX from a Remote Disk

```
P00>>> show device                                                    ❶
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1  SCSI Bus ID 7
dka500.5.0.1.1     DKa500                    RRD45  1645
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0    DVA0                       RX23
polling kzpsa0 (DEC KZPSA) slot 3, bus 0 PCI, hose 0  TPwr 1 Fast 1 Bus ID 7
kzpsa0.7.0.3.0     dkb     TPwr 1 Fast 1 Bus ID 7   N01  A09
dkb100.1.0.3.0     DKb100                    RZ29B  0007
dkb200.2.0.3.0     DKb200                    RZ29B  0007
dkb300.3.0.3.0     DKb300                    RZ29B  0007
dkb500.5.0.3.0     DKb500                    RZ29B  0007
polling ncr1 (NCR 53C810) slot 5, bus 0 PCI, hose 0  SCSI Bus ID 7
dkc400.4.0.5.0     DKc400                    RZ29B  0007
polling tulip0 (DECchip 21040-AA) slot 4, bus 0 PCI, hose 1
ewa0.0.0.4.1: 08-00-2B-E5-B4-1A                                       ❷
polling tulip1 (DECchip 21041-AA) slot 5, bus 0 PCI, hose 1
ewb0.0.0.5.1: 00-00-F8-00-AC-B4

P00>>> boot -flags an -protocols bootp ewa0                           ❸
(boot ewa0.0.0.4.1 -flags an)
Building FRU table

Trying BOOTP boot.

Broadcasting BOOTP Request...
Received BOOTP Packet File Name: /var/adm/ris/ris0.alpha/hvmunix
local inet address: 16.122.128.26
remote inet address: 16.122.128.59
TFTP Read File Name: /var/adm/ris/ris0.alpha/hvmunix
.....................................................................
.....................................................................
.................
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 9a0fa0
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

Secondary boot program - Thu Aug 1 22:33:13 EST 1996

Loading vmunix ...

.
.
.
```

### Example 2–4 Booting Digital UNIX from a Remote Disk (Continued)

```
The system is ready.

Digital UNIX Version 3.2D                                    ❹
```

❶ The **show device** command displays device information, including name and type of connection to the system. See Section 5.2 for a description of the **show device** command and the device naming convention.

❷ The operating system is on a remote disk accessed through the Ethernet controller in slot 4 of PCI1. The name of this device, ewa0, is used as an argument to the **boot** command.

❸ This command loads Digital UNIX from ewa0, autobooting to multiuser mode. See Section 5.3 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 5.15 for information about environment variables.

❹ The operating system banner displays.

## 2.4 Installing Digital UNIX

**Digital UNIX is installed from the internal CD-ROM. Refer to the documentation shipped with the CD-ROM for installation instructions.**

### Example 2–5 Installing Digital UNIX

```
P00>>> boot dka500                                                    ❶
(boot dka500.5.0.1.1 -flags a)
Building FRU table                                                    ❷
block 0 of dka500.5.0.1.1 is a valid boot block
reading 16 blocks from dka500.5.0.1.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

OSF boot - Mon Jul 24 21:56:39 EDT 1995

Loading hupdate ...
Starting hupdate at 0x20000000
Using hvmunix

Loading hvmunix ...
Current PAL Revision <0x20000000010111>
Switching to OSF PALcode Succeeded
New PAL Revision <0x20000000020115>
Loading into KSEG Address Space

Sizes:
text = 4460352
data = 5555024
bss  = 1498368
Starting at 0xfffffc0000239290

Alpha boot: available memory from 0x2dfe000 to 0x7fff6000
physical memory = 2048.00 megabytes.
available memory = 2001.96 megabytes.
using 7856 buffers containing 61.37 megabytes of memory
```

*Continued on next page*

### Example 2–5 Installing Digital UNIX (Continued)

```
Firmware revision: 2.0
PALcode: Digital-UNIX/OSF version 1.21
AlphaServer 4100
pci1 at mcbus0 slot 5
psiop0 at pci1 slot 1
.
.
.
Printer service started
The system is ready.


Digital UNIX Version 3.2D

login: root
Password:
```

❶ Use the **boot** command to install the operating system from the internal CD-ROM, which is always dka500.

❷ See your operating system documentation for installation instructions.

## 2.5 Booting OpenVMS

**OpenVMS can be booted from a local disk, a disk connected through a cluster, or a remote disk through an Ethernet connection. Refer to the documentation shipped with the operating system for booting instructions.**

### Example 2–6 Booting OpenVMS from a Local Disk

```
P00>>> show device                                                    ❶
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1     DKa500                      RRD45  1645
polling ncr1 (NCR 53C810) slot 2, bus 0 PCI, hose 1   SCSI Bus ID 7
dkb0.0.0.2.1       DKb0                        RZ29B  0016
dkb100.1.0.2.1     DKb100                      RZ29B  0016
.
.
.
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0    DVA0                         RX23
polling tulip0 (DECchip 21040-AA) slot 3, bus 0 PCI, hose 0
ewa0.0.0.3.0: 00-00-F8-20-14-61
P00>>> show boot_reset                                                ❷
boot_reset              OFF
P00>>> show bootdef_dev                                               ❸
bootdef_dev             dkb0.0.0.2.1
P00>>> boot                                                           ❹
 SROM V2.0 on cpu0                                                    ❷
 SROM V2.0 on cpu1
XSROM V2.0 on cpu1
XSROM V2.0 on cpu0
.
.
.
System temperature is 24 degrees C
AlphaServer 4100 Console V2.0-1, 2-AUG-1996 15:17:48
```

*Continued on next page*

### Example 2–6 Booting OpenVMS from a Local Disk (Continued)

```
CPU 0 booting

(boot dkb0.0.0.2.1 -flags 0)
Building FRU table
block 0 of dkb0.0.0.2.1 is a valid boot block
reading 1002 blocks from dkb0.0.0.2.1
bootstrap code read in
.
.
.
Welcome to OpenVMS Alpha (TM) Operating System, Version V6.2-1H3❺
```

❶ The **show device** command displays device information, including name and type of connection to the system. See Section 5.2 for a description of the **show device** command and the device naming convention.

❷ The **boot_reset** environment variable was previously set to "on," causing the power-up trace to display when the system initializes (see Section 2.2). See Section 5.9 for commands used with environment variables.

❸ The **bootdef_dev** environment variable specifies the default boot device. In this example, the default boot device was previously set to dkb0.0.0.2.1.

❹ No boot device is specified in the **boot** command; the default boot device was set with the environment variable. See Section 5.3 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 5.15 for information about environment variables.

❺ The operating system banner displays.

### Example 2–7 Booting OpenVMS from a Disk on a Cluster

```
P00>>> show bootdef_dev                                          ❶
bootdef_dev             dua1.8.0.5.1
P00>>> show device                                               ❷
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1     DKa500                    RRD45  1645
polling dac0 (Mylex DAC960) slot 4, bus 0 PCI, hose 1
dra0.0.0.4.1       DRA0                      1 Member JBOD
dra1.0.0.4.1       DRA1                      1 Member JBOD
dra2.0.0.4.1       DRA2                      4 Member RAID 5
polling cipca0 (CIPCA) slot 5, bus 0 PCI, hose 1   cixcd_poll_comp
cipca_a.4.0.5.1    dua     CI Bus ID 4                           ❸
dua24.2.0.5.1      $22$DUA24 (HSC702)        RA92
dua30.2.0.5.1      $22$DUA30 (HSC702)        RA92
.
.
.
dua0.8.0.5.1       $22$DUA0 (RAWHSJ)         HSX1
dua1.8.0.5.1       $22$DUA1 (RAWHSJ)         HSX1              ❸
.
.
.
P00>>>boot                                                       ❹
(boot dua1.8.0.5.1 -flags 0)
Building FRU table
.
.
.
Welcome to OpenVMS Alpha (TM) Operating System, Version V6.2-1H3❺
```

❶ The **bootdef_dev** environment variable specifies the default boot device. In this example, the default boot device was previously set to dua1.8.0.5.1. See Section 5.9 for commands used with environment variables.

❷ The **show device** command displays device information, including name and type of connection to the system. See Section 5.2 for a description of the **show device** command and the device naming convention.

❸ The disk dua1.8.0.5.1 is on the CI cluster that includes this system.

❹ No boot device is specified in the **boot** command; the default boot device was set with the environment variable. See Section 5.3 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 5.15 for information about environment variables.

❺ The operating system banner prints.

### Example 2–8 Booting OpenVMS from a Remote Disk

```
P00>>> show device                                                    ❶
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1      DKa500                        RRD45  1645
polling kzpsa0 (DEC KZPSA) slot 2, bus 0 PCI, hose 1  TPwr 1 Fast 1 Bus ID 5
kzpsa0.5.0.2.1      dkb      TPwr 1 Fast 1 Bus ID 5   N01  A10
polling kzpsa1 (DEC KZPSA) slot 3, bus 0 PCI, hose 1  TPwr 1 Fast 1 Bus ID 5
kzpsa1.5.0.3.1      dkc      TPwr 1 Fast 1 Bus ID 5   N01  A10
polling kzpsa2 (DEC KZPSA) slot 4, bus 0 PCI, hose 1  TPwr 1 Fast 1 Bus ID 5
kzpsa2.5.0.4.1      dkd      TPwr 1 Fast 1 Bus ID 5   P01  A10
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0     DVA0                           RX23
polling kzpsa3 (DEC KZPSA) slot 3, bus 0 PCI, hose 0  TPwr 1 Fast 1 Bus ID 5
kzpsa3.5.0.3.0      dke      TPwr 1 Fast 1 Bus ID 5   P01  A10
polling kzpsa4 (DEC KZPSA) slot 4, bus 0 PCI, hose 0  TPwr 1 Fast 1 Bus ID 5
kzpsa4.5.0.4.0      dkf      TPwr 1 Fast 1 Bus ID 5   P01  A10
polling kzpsa5 (DEC KZPSA) slot 5, bus 0 PCI, hose 0  TPwr 1 Fast 1 Bus ID 5
kzpsa5.5.0.5.0      dkg      TPwr 1 Fast 1 Bus ID 5   F01  A10
dkg0.0.0.5.0        DKg0                        RZ26J  589H
mkg300.3.0.5.0      MKg300                      TLZ09  0165
polling tulip0 (DECchip 21140-AA) slot 5, bus 0 PCI, hose 1
ewa0.0.0.5.1: 00-00-F8-00-42-04                                       ❶
P00>>> boot ewa0 -flags 0                                             ❷
(boot ewa0.0.0.5.1 -flags 0)
Building FRU table

Trying MOP boot.
.............

Network load complete.
.
.
.
Welcome to OpenVMS Alpha (TM) Operating System, Version V6.2-1H3      ❸
```

❶ The **show device** command displays device information, including name and type of connection to the system. In this example the Ethernet connection is ewa0. See Section 5.2 for a description of the **show device** command and the device naming convention.

❷ The **boot** command specifies ewa0 as the boot device. See Section 5.3 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 5.15 for information about environment variables.

❸ The operating system banner prints.

## 2.6 Installing OpenVMS

**OpenVMS is installed from the internal CD-ROM. Refer to the documentation shipped with the CD-ROM for complete installation instructions.**

### Example 2–9 Installing OpenVMS

```
P00>>> boot -flags 0,0 dka500                                    ❶
Initializing...
SROM V1.0 on cpu0
.
.  [The initialization display prints. See Section 2.2.]
.

AlphaServer 4100 Console V1.0-1, 2-APR-1996 15:17:48

CPU 0 booting

(boot dka500.5.0.1.1 -flags 0,0)
Building FRU table
block 0 of dka500.5.0.1.1 is a valid boot block
reading 1002 blocks from dka500.5.0.1.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7d400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

    OpenVMS (TM) Alpha Operating System, Version 6.2-1H3        ❷

%SMP-I-SECMSG, CPU #01 message:   P01>>>START
%SMP-I-CPUBOOTED, CPU #01 has joined the PRIMARY CPU in
multiprocessor operation
```

*Continued on next page*

### Example 2–9 Installing OpenVMS (Continued)

```
Installing required known files...

    Configuring devices...


******************************************************************

 You can install or upgrade the OpenVMS Alpha operating system
 or you can install or upgrade layered products that are included
 on the OpenVMS Alpha operating system CD-ROM.

 You can also execute DCL commands and procedures to perform
 "standalone" tasks, such as backing up the system disk.

 Please choose one of the following:

    1)  Install or upgrade OpenVMS Alpha Version 6.2-1H3
    2)  List layered product kits that this procedure can install
    3)  Install or upgrade layered product(s)
    4)  Execute DCL commands and procedures
    5)  Shut down this system

Enter CHOICE or ? to repeat menu: (1/2/3/4/5/?)
```

❶ Use the **boot** command to install the operating system from the internal CD-ROM, which is always dka500.

❷ See your operating system documentation for installation instructions.

## 2.7 Booting Windows NT

**Microsoft Windows NT is started from the AlphaBIOS Boot screen.**

### Figure 2–2 AlphaBIOS Boot Screen

PK-0728-96

The setting of the SRM **os_type** environment variable determines if AlphaBIOS is loaded and started on reset and power-up. If **os_type** is set to **nt**, after the power-up display (see Section 2.2) the SRM console is loaded and started, and it then loads and starts the AlphaBIOS console. AlphaBIOS must be running before Windows NT can be booted.

The method used for booting Windows NT is determined by the setting of **Auto Start** in the AlphaBIOS **Standard CMOS Setup** screen (see Section 6.7).

- If **Auto Start** is enabled, the primary version of Windows NT starts automatically.
- If **Auto Start** is disabled, use the arrow keys to select the Windows NT version to start. Press Enter to boot Windows NT.

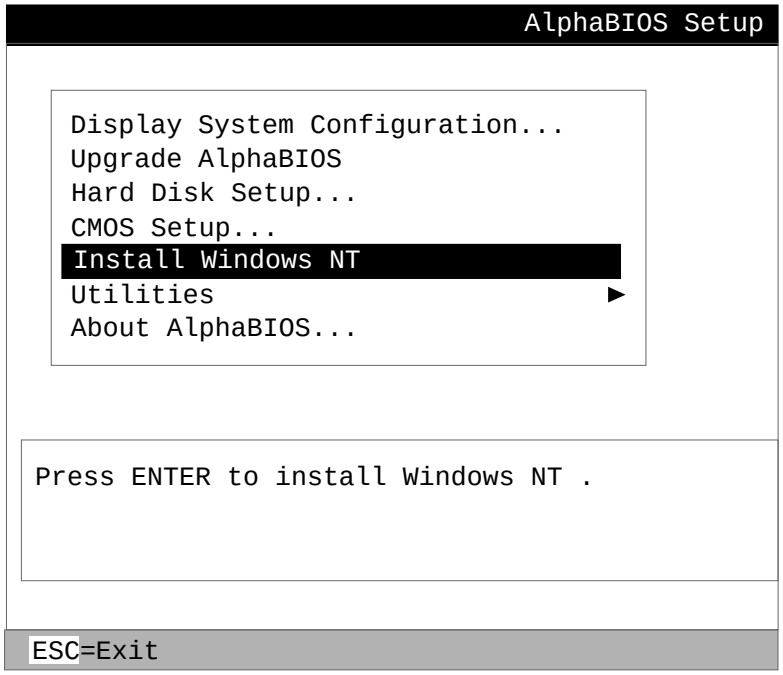
*NOTE: The SRM **console** environment variable must be set to **graphics** before booting Windows NT. (This setting is not necessary to run AlphaBIOS.)*

*If the Halt button is in (the LED on the button is lit), the SRM console will not load AlphaBIOS. This is true even when the **os_type** environment variable is set to **nt**.*

## 2.8 Installing Windows NT

**Windows NT is installed from CD-ROM. Insert the CD-ROM in the drive, start AlphaBIOS Setup, select the menu item Install Windows NT, and follow the prompts.**

### Figure 2–3 Installing Windows NT




PK-0726B-96

Up to four versions of Windows NT can be resident in a system at one time.

If this is a new Windows NT installation, start with this procedure:

1. Start AlphaBIOS. If the system is in the SRM console, issue the command **alphabios**.
2. From the **AlphaBIOS Boot** screen, enter **AlphaBIOS Setup** by pressing the F2 key.
3. From **AlphaBIOS Setup** select **CMOS Setup**, and press Enter.
4. From **CMOS Setup** set the system date and time.
5. Return to the **AlphaBIOS Setup** screen by pressing the F10 key.
6. Select **Hard Disk Setup**, and press Enter.
7. Perform an express hard disk setup by pressing the F7 key to enter **Express Setup**.
8. Continue the setup by pressing the F10 key.
9. Go to the procedure below.

This procedure is for all Windows NT installations:

1. Put the Windows NT CD into the CD-ROM drive.
2. Start **AlphaBIOS Setup**.
3. Select **Install Windows NT** and press Enter.
4. Follow the prompts to complete the installation. For more information on installing Windows NT, refer to the *Installation Guide* in your Windows NT software package.

## 2.9 Updating Firmware

**Start the Loadable Firmware Update (LFU) utility by issuing the lfu command at the SRM console prompt, booting it from the CD-ROM while in the SRM console, or selecting Update AlphaBIOS in the AlphaBIOS Setup screen.**

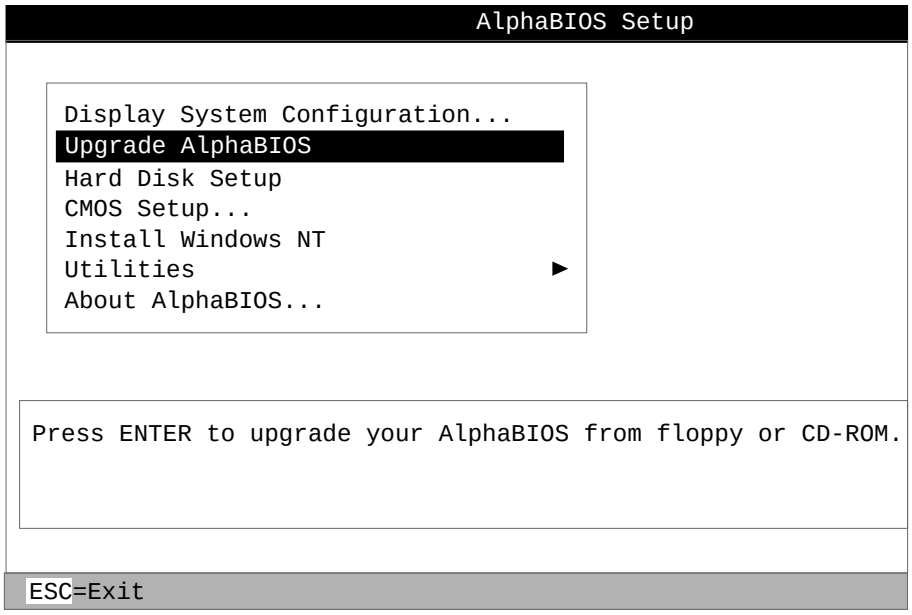
### Example 2–10 Starting LFU from the SRM Console

```
P00>>> lfu

          ***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: cda0
.
.
.
UPD>
```

### Figure 2–4 Starting LFU from the AlphaBIOS Console




PK-0726A-96

Use the Loadable Firmware Update (LFU) utility to update system firmware.

You can start LFU from either the SRM console or the AlphaBIOS console.

- From the SRM console, start LFU by issuing the **lfu** command (see Example 2–10). Also from the SRM console, LFU can be booted from the Alpha CD-ROM (V3.7 or later), as shown in Example 2–11.
- From the AlphaBIOS console, select **Update AlphaBIOS** from the **AlphaBIOS Setup** screen (see Figure 2–4).

A typical update procedure is:

1. Start LFU.
2. Use the LFU **list** command to show the revisions of modules that LFU can update and the revisions of update firmware.
3. Use the LFU **update** command to write the new firmware.
4. Use the LFU **exit** command to exit back to the console.

The sections that follow show examples of updating firmware from the local CD-ROM, the local floppy, and a network device. Following the examples is an LFU command reference.

## Example 2–11 Booting LFU from the CD-ROM

```
P00>>> show dev ncr0
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1     DKa500                    RRD45  1645
P00>>> boot dka500
(boot dka500.5.0.1.1 -flags 0,0)
block 0 of dka500.5.0.1.1 is a valid boot block
.
.
.
jumping to bootstrap code

The default bootfile for this platform is
        [AS4X00]AS4X00_LFU.EXE
Hit <RETURN> at the prompt to use the default bootfile.
Bootfile: <CR>

Starting Firmware Update Utility

          ***** Loadable Firmware Update Utility *****
.
.
.
UPD>
```

## 2.9.1 Updating Firmware from the Internal CD-ROM

**Insert the update CD-ROM, start LFU, and select cda0 as the load device.**

### Example 2–12 Updating Firmware from the Internal CD-ROM

```
          ***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: cda0        ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS4X00FW]: AS4X00CP  ❷

Copying AS4X00CP from DKA500.5.0.1.1 .
Copying [as4x00]RHREADME from DKA500.5.0.1.1 .
Copying [as4x00]RHSRMROM from DKA500.5.0.1.1 ...................
Copying [as4x00]RHARCROM from DKA500.5.0.1.1 .............

------------------------------------------------------------------
 Function    Description                                          ❸
------------------------------------------------------------------
 Display     Displays the system's configuration table.
 Exit        Done exit LFU (reset).
 List        Lists the device, revision, firmware name, and
             update revision.
 Lfu         Restarts LFU.
 Readme      Lists important release information.
 Update      Replaces current firmware with loadable data image.
 Verify      Compares loadable and hardware images.
 ? or Help   Scrolls this function table.
------------------------------------------------------------------

UPD> list                                                        ❹

Device          Current Revision    Filename       Update Revision
AlphaBIOS          V5.12-2          arcrom            V6.40-1
srmflash           V1.0-9           srmrom            V2.0-3
```

*Continued on next page*

❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In this example, the internal CD-ROM is selected.

❷ Select the file that has the firmware update, or press Enter to select the default file. The file options are:

| | |
|---|---|
| AS4X00FW (default) | SRM console, AlphaBIOS console, and I/O adapter firmware. |
| AS4X00CP | SRM console and AlphaBIOS console firmware only. |
| AS4X00IO | I/O adapter firmware only. |

In this example the file for console firmware (AlphaBIOS and SRM) is selected.

❸ The LFU function table and prompt (UPD>) display.

❹ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the resident firmware for each console (SRM and AlphaBIOS) is at an earlier revision than the firmware in the update file.

*Continued on next page*

### Example 2–12 Updating Firmware from the Internal CD-ROM (Continued)

```
UPD> update *                                                    ❺
WARNING: updates may take several minutes to complete for each
device.

Confirm update on: AlphaBIOS       [Y/(N)] y                     ❻

                          DO NOT ABORT!
AlphaBIOS     Updating to V6.40-1... Verifying V6.40-1... PASSED.

Confirm update on: srmflash        [Y/(N)] y

                          DO NOT ABORT!
srmflash        Updating to V2.0-3...  Verifying V2.0-3...
PASSED.

UPD> exit                                                        ❼
```

❺ The **update** command updates the device specified or all devices. In this example, the wildcard indicates that all devices supported by the selected update file will be updated.

❻ For each device, you are asked to confirm that you want to update the firmware. The default is no. Once the update begins, do not abort the operation. Doing so will corrupt the firmware on the module.

❼ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

### 2.9.2 Updating Firmware from the Internal Floppy Disk — Creating the Diskettes

**Create the update diskettes before starting LFU. See Section 2.9.3 for an example of the update procedure.**

**Table 2–1 File Locations for Creating Update Diskettes on a PC**

| Console Update Diskette | I/O Update Diskette |
|---|---|
| AS4X00FW.TXT | AS4X00IO.TXT |
| AS4X00CP.TXT | RHREADME.SYS |
| RHREADME.SYS | CIPCA214.SYS |
| RHSRMROM.SYS | DFPAA246.SYS |
| RHARCROM.SYS | KZPAAA10.SYS |

To update system firmware from floppy disk, you first must create the firmware update diskettes. You will need to create two diskettes: one for console updates, and one for I/O.

1. Download the update files from the Internet (see Appendix A).
2. On a PC, copy files onto two FAT-formatted diskettes as shown in Table 2–1.

   From an OpenVMS system, copy files onto two ODS2-formatted diskettes as shown in Example 2–13.

### Example 2–13 Creating Update Diskettes on an OpenVMS System

Console update diskette

```
$ inquire ignore "Insert blank HD floppy in DVA0, then continue"
$ set verify
$ set proc/priv=all
$ init /density=hd/index=begin dva0: rhods2cp
$ mount dva0: rhods2cp
$ create /directory dva0:[as4x00]
$ copy as4x00fw.sys dva0:[as4x00]as4x00fw.sys
$ copy as4x00cp.sys dva0:[as4x00]as4x00cp.sys
$ copy rhreadme.sys dva0:[as4x00]rhreadme.sys
$ copy as4x00fw.txt dva0:[as4x00]as4x00fw.txt
$ copy as4x00cp.txt dva0:[as4x00]as4x00cp.txt
$ copy rhsrmrom.sys dva0:[as4x00]rhsrmrom.sys
$ copy rharcrom.sys dva0:[as4x00]rharcrom.sys
$ dismount dva0:
$ set noverify
$ exit
```

I/O update diskette

```
$ inquire ignore "Insert blank HD floppy in DVA0, then continue"
$ set verify
$ set proc/priv=all
$ init /density=hd/index=begin dva0: rhods2io
$ mount dva0: rhods2io
$ create /directory dva0:[as4x00]
$ create /directory dva0:[options]
$ copy as4x00fw.sys dva0:[as4x00]as4x00fw.sys
$ copy as4x00io.sys dva0:[as4x00]as4x00io.sys
$ copy rhreadme.sys dva0:[as4x00]rhreadme.sys
$ copy as4x00fw.txt dva0:[as4x00]as4x00fw.txt
$ copy as4x00io.txt dva0:[as4x00]as4x00io.txt
$ copy cipca214.sys dva0:[options]cipca214.sys
$ copy dfpaa246.sys dva0:[options]dfpaa246.sys
$ copy kzpsaA10.sys dva0:[options]kzpsaa10.sys
$ dismount dva0:
$ set noverify
$ exit
```

### 2.9.3 Updating Firmware from the Internal Floppy Disk — Performing the Update

**Insert an update diskette (see Section 2.9.2) into the internal floppy drive. Start LFU and select dva0 as the load device.**

#### Example 2–14 Updating Firmware from the Internal Floppy Disk

```
          ***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0          ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS4X00IO,(AS4X00CP)]:
AS4X00IO                                                           ❷

Copying AS4X00IO from DVA0 .
Copying RHREADME from DVA0 .
Copying CIPCA214 from DVA0 .
Copying DFPAA252 from DVA0 ...
Copying KZPSAA11 from DVA0 ...
.
.  [The function table displays, followed by the UPD> prompt, as
.  shown in Example 2–12.]

UPD> list                                                          ❸

Device          Current Revision    Filename       Update Revision
AlphaBIOS          V5.12-3          arcrom         Missing file
pfi0                2.46            dfpaa_fw           2.52
srmflash           T3.2-21          srmrom         Missing file
                                    cipca_fw        A214
                                    kzpsa_fw         A11
```

*Continued on next page*

❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In this example, the internal floppy disk is selected.

❷ Select the file that has the firmware update, or press Enter to select the default file. When the internal floppy disk is the load device, the file options are:

| | |
|---|---|
| AS4X00CP (default) | SRM console and AlphaBIOS console firmware only. |
| AS4X00IO | I/O adapter firmware only. |

The default option in Example 2–12 (AS4X00FW) is not available, since the file is too large to fit on a 1.44 MB diskette. This means that when a floppy disk is the load device, you can update either console firmware or I/O adapter firmware, but not both in the same LFU session. If you need to update both, after finishing the first update, restart LFU with the **lfu** command and insert the diskette with the other file.

In this example the file for I/O adapter firmware is selected.

❸ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the update revision for console firmware displays as "Missing file" because only the I/O firmware files are available on the floppy disk.

*Continued on next page*

### Example 2–14 Updating Firmware from the Internal Floppy Disk (Continued)

```
UPD> update pfi0                                                    ❹
WARNING: updates may take several minutes to complete for each
device.

Confirm update on: pfi0            [Y/(N)] y                        ❺

                          DO NOT ABORT!
pfi0            Updating to 2.52... Verifying to 2.52... PASSED.

UPD> lfu                                                            ❻

          ***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS4X00IO,(AS4X00CP)]:❼

.
. (The function table displays, followed by the UPD> prompt.
.  Console firmware can now be updated.)

UPD> exit                                                           ❽
```

❹ The **update** command updates the device specified or all devices.

❺ For each device, you are asked to confirm that you want to update the firmware. The default is no. Once the update begins, do not abort the operation. Doing so will corrupt the firmware on the module.

❻ The **lfu** command restarts the utility so that console firmware can be updated. (Another method is shown in Example 2–15, where the user specifies the file AS4X00FW and is prompted to insert the second diskette.)

❼ The default update file, AS4X00CP, is selected. The console firmware can now be updated, using the same procedure as for the I/O firmware.

❽ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

### Example 2–15 Selecting AS4X00FW to Update Firmware from the Internal Floppy Disk

```
P00>>> lfu

          ***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0

Please enter the name of the firmware files list, or
Press <return> to use the default filename [AS4X00IO,(AS4X00CP)]: as4x00fw

Copying AS4X00FW from DVA0 .
Copying RHREADME from DVA0 .
Copying RHSRMROM from DVA0 ..........................
Copying RHARCROM from DVA0 ..............
Copying CIPCA214 from DVA0
Please insert next floppy containing the firmware,
Press <return> when ready. Or type DONE to abort.
Copying CIPCA214 from DVA0 .
Copying DFPAA246 from DVA0 ...
Copying KZPSAA10 from DVA0 ...
.
.
.
```

## 2.9.4 Updating Firmware from a Network Device

**Copy files to the local MOP server's MOP load area, start LFU, and select ewa0 as the load device.**

### Example 2–16 Updating Firmware from a Network Device

```
          ***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: ewa0          ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS4X00FW]:             ❷

Copying AS4X00FW from EWA0 .
Copying RHREADME from EWA0 .
Copying RHSRMROM from EWA0 ............................
Copying RHARCROM from EWA0 ............
Copying CIPCA214 from EWA0 .
Copying DFPAA246 from EWA0 ...
Copying KZPSAA11 from EWA0 ...

.
.  [The function table displays, followed by the UPD> prompt, as
.  shown in Example 2–12.]

UPD> list                                                          ❸

Device          Current Revision    Filename       Update Revision
AlphaBIOS          V5.12-2          arcrom            V6.40-1
kzpsa0            A10               kzpsa_fw         A11
kzpsa1            A10               kzpsa_fw         A11
srmflash           V1.0-9           srmrom            V2.0-3
                                    cipca_fw        A214
                                    dfpaa_fw           2.46
```

*Continued on next page*

Before starting LFU, download the update files from the Internet (see Appendix A). You will need the files with the extension .SYS. Copy these files to your local MOP server's MOP load area.

❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In this example, a network device is selected.

❷ Select the file that has the firmware update, or press Enter to select the default file. The file options are:

| | |
|---|---|
| AS4X00FW (default) | SRM console, AlphaBIOS console, and I/O adapter firmware. |
| AS4X00CP | SRM console and AlphaBIOS console firmware only. |
| AS4X00IO | I/O adapter firmware only. |

In this example the default file, which has both console firmware (AlphaBIOS and SRM) and I/O adapter firmware, is selected.

❸ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the resident firmware for each console (SRM and AlphaBIOS) and I/O adapter is at an earlier revision than the firmware in the update file.

*Continued on next page*

### Example 2–16 Updating Firmware from a Network Device (Continued)

```
UPD> update * -all                                                    ❹
WARNING: updates may take several minutes to complete for each
device.

                          DO NOT ABORT!
AlphaBIOS    Updating to V6.40-1... Verifying V6.40-1...  PASSED.

                          DO NOT ABORT!
kzpsa0        Updating to  A11    ...  Verifying A11...  PASSED.

                          DO NOT ABORT!
kzpsa1        Updating to  A11    ...  Verifying A11...  PASSED.

                          DO NOT ABORT!
srmflash      Updating to V2.0-3...  Verifying V2.0-3...  PASSED.

UPD> exit                                                             ❺
```

❹ The **update** command updates the device specified or all devices. In this example, the wildcard indicates that all devices supported by the selected update file will be updated. Typically LFU requests confirmation before updating each console's or device's firmware. The **-all** option eliminates the update confirmation requests.

❺ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

### 2.9.5 LFU Commands

**The commands summarized in Table 2–2 are used to update system firmware.**

**Table 2–2 LFU Command Summary**

| Command | Function |
|---|---|
| **display** | Shows the physical configuration of the system. |
| **exit** | Terminates the LFU program. |
| **help** | Displays the LFU command list. |
| **lfu** | Restarts the LFU program. |
| **list** | Displays the inventory of update firmware on the selected device. |
| **readme** | Lists release notes for the LFU program. |
| **update** | Writes new firmware to the module. |
| **verify** | Reads the firmware from the module into memory and compares it with the update firmware. |

These commands are described in the following pages.

### display

The **display** command shows the physical configuration of the system. **Display** is equivalent to issuing the SRM console command **show configuration**. Because it shows the slot for each module, **display** can help you identify the location of a device.

### exit

The **exit** command terminates the LFU program, causes system initialization and testing, and returns the system to the console from which LFU was called.

### help

The **help** (or **?**) command displays the LFU command list, shown below.

```
---------------------------------------------------------------------
 Function    Description
---------------------------------------------------------------------
 Display     Displays the system's configuration table.
 Exit        Done exit LFU (reset).
 List        Lists the device, revision, firmware name, and update
             revision.
 Lfu         Restarts LFU.
 Readme      Lists important release information.
 Update      Replaces current firmware with loadable data image.
 Verify      Compares loadable and hardware images.
 ? or Help   Scrolls this function table.
---------------------------------------------------------------------
```

### lfu

The **lfu** command restarts the LFU program. This command is used when the update files are on a floppy disk. The files for updating both console firmware and I/O firmware are too large to fit on a 1.44 MB disk, so only one type of firmware can be updated at a time. Restarting LFU enables you to specify another update file.

### list

The **list** command displays the inventory of update firmware on the CD-ROM, network, or floppy. Only the devices listed at your terminal are supported for firmware updates.

The **list** command shows three pieces of information for each device:

- Current Revision — The revision of the device's current firmware
- Filename — The name of the file used to update that firmware
- Update Revision — The revision of the firmware update image

### readme

The **readme** command lists release notes for the LFU program.

### update

The **update** command writes new firmware to the module. Then LFU automatically verifies the update by reading the new firmware image from the module into memory and comparing it with the source image.

To update more than one device, you may use a wildcard but not a list. For example, **update k*** updates all devices with names beginning with k, and **update *** updates all devices. When you do not specify a device name, LFU tries to update all devices; it lists the selected devices to update and prompts before devices are updated. (The default is no.) The **-all** option eliminates the update confirmation requests, enabling the update to proceed without operator intervention.

*CAUTION: Never abort an **update** operation. Aborting corrupts the firmware on the module.*

### verify

The **verify** command reads the firmware from the module into memory and compares it with the update firmware. If a module already verified successfully when you updated it, but later failed tests, you can use **verify** to tell whether the firmware has become corrupted.

## 2.10 Operating the System Remotely

**The remote console monitor (RCM) is used to monitor and control the system remotely. The RCM resides on the server control module (see Section 1.3.7) and allows the system administrator to connect remotely to a managed system through a modem, using a serial terminal or terminal emulator.**

### RCM Console Overview

The RCM has special console firmware that is used to remotely control an AlphaServer system. The RCM firmware resides on an independent microprocessor. It is not part of the SRM console that resides in the flash ROM. The RCM firmware has its own command interface that allows the user to perform the tasks that can usually be done from the system's serial console terminal. RCM console commands are used to reset, halt, and power the system on or off, regardless of the operating system or hardware state. The RCM console commands are also used to monitor the power supplies, temperature, and fans.

The user can enter the RCM console either remotely or through the local serial console terminal. Once in command mode, the user can enter commands to control and monitor the system.

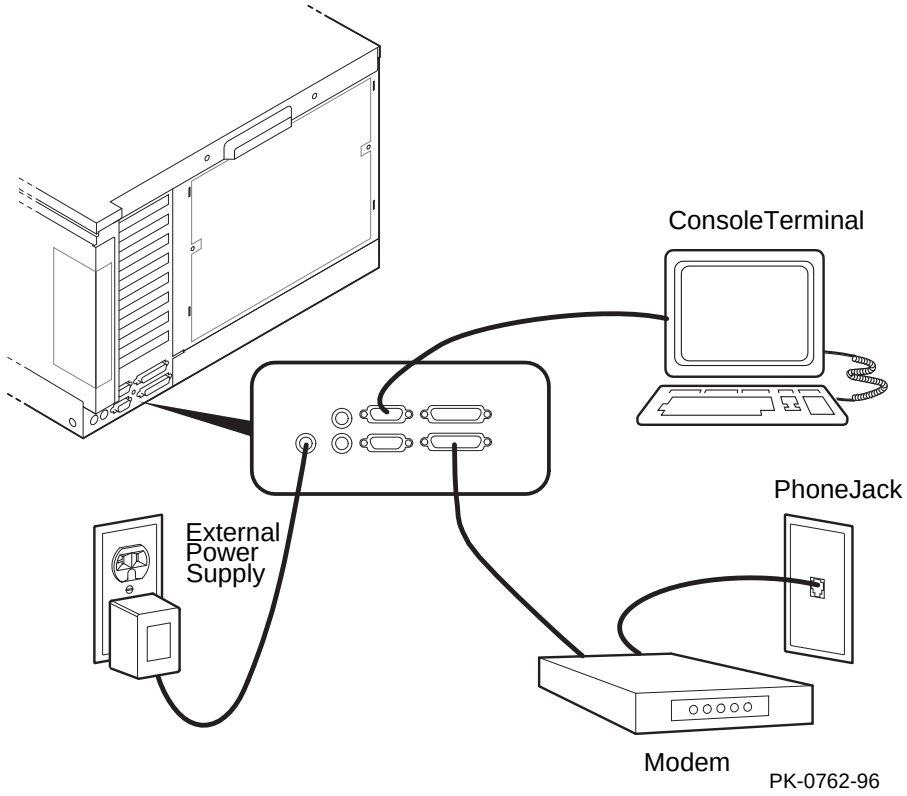
- To enter the RCM console remotely, the user dials in through a modem, enters a password, and then types a special escape sequence that invokes RCM command mode. You must set up the modem before you can dial in remotely. See Section 2.10.1.
- To enter the RCM console locally, the user types the escape sequence at the SRM console prompt on the local serial console terminal.

The RCM also provides an autonomous dial-out capability when it detects a power failure within the system. When triggered, the RCM dials a paging service at 30-minute intervals until the administrator clears the alert within the RCM.

## 2.10.1 Modem Usage

**To use the RCM to monitor a system remotely, first make the connections to the server control module, as shown below. Then configure the modem port for dial-in.**

**Figure 2–5 RCM Connections**



### Modem Selection

The RCM requires a Hayes-compatible modem. The controls that the RCM sends to the modem have been selected to be acceptable to a wide selection of modems. The modems that have been tested and qualified include:

Motorola LifeStyle Series 28.8
AT&T DATAPORT 14.4/FAX
Zoom Model 360

The U.S. Robotics Sportster DATA/FAX MODEM is also supported, but requires some modification of the modem initialization and answer strings. See Section 2.10.7.

### Modem Configuration Procedure

1. Connect a Hayes-compatible modem to the RCM as shown in Figure 2–5, and power up the modem.
2. From the local serial console terminal, enter the RCM firmware console by typing the following escape sequence:

   ```
   ^]^]rcm
   ```

   The character "^" is created by simultaneously holding down the Ctrl key and pressing the ] key (right square bracket). The firmware prompt, RCM>, should now be displayed.
3. Enter a modem password with the **setpass** command. See Section 2.10.3.
4. Enable the modem port with the **enable** command. See Section 2.10.3.
5. Enter the **quit** command to leave the RCM console.
6. You are now ready to dial in remotely.

### Dialing In to the RCM Modem Port

1. Dial the modem connected to the server control module. The RCM answers the call and after a few seconds prompts for a password with a "#" character.
2. Enter the password that was loaded using the **setpass** command. The user has three tries to correctly enter the password. On the third unsuccessful attempt, the connection is terminated, and as a security precaution, the modem is not answered again for 5 minutes.

   On successful entry of the password, the RCM banner message "RCM V1.0" is displayed, and the user is connected to the system COM1 port. At this point the local terminal keyboard is disabled except for entering the RCM console firmware. The local terminal displays all the terminal traffic going out to the modem.
3. To connect to the RCM firmware console, type the RCM escape sequence.

Refer to Example 2–17 for an example of the modem dial-in procedure.

A sample dial-in dialog would look similar to the following:

### Example 2–17 Sample Remote Dial-In Dialog

```
ATQ0V1E1S0=0                !When modem dial-in connection is made, a screen display
OK                          similar to this appears.
ATDT30167
CONNECT 9600
#                           !Enter password at this prompt.
RCM V1.0                    !RCM banner is displayed.
^]^]rcm                     !Enter the escape sequence after the banner is displayed.
                            The escape sequence is not echoed on the terminal.
RCM>                        !RCM prompt is displayed. Commands to control and
                            monitor the system can be entered.
```

### Terminating a Remote Console (Modem) Session

Terminate the modem session by executing a **hangup** command from the RCM console firmware. This will cleanly terminate the modem connection.

If the modem connection is terminated without using the **hangup** command, or if the line is dropped due to phone line problems, the RCM will detect carrier loss and initiate an internal **hangup** command. This process can take a minute or more, and the local terminal will be locked out until the auto hangup process completes.

If the modem link is idle for more than 20 minutes, the RCM initiates an auto hangup.

## 2.10.2 Entering and Leaving Command Mode

**Use the default escape sequence to enter RCM command mode for the first time. You can enter RCM command mode from the SRM console level, the operating system level, or an application. The RCM quit command reconnects the terminal to the system console port.**

### Example 2–18 Entering and Leaving RCM Command Mode

```
^]^]rcm                ❶
RCM>

RCM> quit              ❷
Focus returned to COM port
```

#### Entering the RCM Firmware Console

To enter the RCM firmware console, enter the RCM escape sequence. See ❶ in Example 2–18 for the default sequence.

The escape sequence is not echoed on the terminal or sent to the system. Once in the RCM firmware console, the user is in RCM command mode and can enter RCM console commands.

#### Leaving Command Mode

To leave RCM command mode and reconnect to the system console port, enter the **quit** command, then press Return to get a prompt from the operating system or system console. (See ❷).

### 2.10.3 RCM Commands

**The RCM commands summarized below are used to control and monitor a system remotely.**

**Table 2–3 RCM Command Summary**

| Command | Function |
|---|---|
| **alert_clr** | Clears alert flag, stopping dial-out alert cycle |
| **alert_dis** | Disables the dial-out alert function |
| **alert_ena** | Enables the dial-out alert function |
| **disable** | Disables remote access to the modem port |
| **enable** | Enables remote access to the modem port |
| **hangup** | Terminates the modem connection |
| **halt** | Halts server |
| **help** or **?** | Displays the list of commands |
| **poweroff** | Turns off power to server |
| **poweron** | Turns on power to server |
| **quit** | Exits console mode and returns to system console port |
| **reset** | Resets the server |
| **setesc** | Changes the escape sequence for entering command mode |
| **setpass** | Changes the modem access password |
| **status** | Displays server's status and sensors |

### Command Conventions

- The commands are not case sensitive.
- A command must be entered in full.
- If a command is entered that is not valid, the command fails with the message:

```
*** ERROR - unknown command ***
```

Enter a valid command.

The RCM commands are described on the following pages.

### alert_clr

The **alert_clr** command clears an alert condition within the RCM. The alert enable condition remains active, and the RCM will again enter the alert condition when it detects a system power failure. See Section 2.10.4.

```
RCM>alert_clr
```

### alert_dis

The **alert_dis** command disables RCM dial-out capability. It also clears any out - standing alerts. See Section 2.10.4. The alert disable state is nonvolatile. Dial-out capability remains disabled until the **alert_enable** command is issued.

```
RCM>alert_dis
```

### alert_ena

The **alert_ena** command enables the RCM to automatically dial out when it detects a power failure within the system. The RCM repeats the dial-out alert at 30-minute intervals until the alert is cleared. The alert enable state is nonvolatile. Dial-out capability remains enabled until the **alert_disable** command is issued.

```
RCM>alert_ena
```

In order for the **alert_enable** command to work, two conditions must be met:

- A modem dial-out string must be entered with the system console.
- Remote access to the RCM modem port must be enabled with the **enable** command.

See Section 2.10.4.

If the **alert_enable** command is entered when remote access is disabled, the following message is returned:

```
*** error ***
```

### disable

The **disable** command disables remote access to the RCM modem port.

```
RCM>disable
```

The module's remote access default state is DISABLED. The modem enable state is nonvolatile. When the modem is disabled, it remains disabled until the **enable** command is issued. If a modem connection is in progress, entering the **disable** command terminates it.

### enable

The **enable** command enables remote access to the RCM modem port. It can take up to 10 seconds for the **enable** command to be executed.

```
RCM>enable
```

The module's remote access default state is DISABLED.

The modem enable state is nonvolatile. When the modem is enabled, it remains enabled until the **disable** command is issued.

The **enable** command can fail for two reasons:

- There is no modem access password configured.
- The modem is not connected or is not working properly.

If the **enable** command fails, the following message is displayed:

```
*** ERROR - enable failed ***
```

### hangup

The **hangup** command terminates the modem session. When this command is issued, the remote user is disconnected from the server. This command can be issued from either the local or remote console.

```
RCM>hangup
```

### halt

The **halt** command attempts to halt the managed system. It is functionally equivalent to pressing the Halt button on the system operator control panel to the "in" position and then releasing it to the "out" position. The RCM console firmware exits command mode and reconnects the user's terminal to the server's COM1 serial port.

```
RCM>halt
Focus returned to COM port
```

*NOTE: Pressing the Halt button has no effect on systems running Windows NT.*

### help or ?

The **help** or **?** command displays the RCM firmware command set.

### poweroff

The **poweroff** command requests the RCM module to power off the system It is functionally equivalent to turning off the system power from the operator control panel.

```
RCM>poweroff
```

If the system is already powered off, this command has no effect.

The external power to the RCM must be connected in order to power off the system from the RCM firmware console. If the external power supply is not connected, the command will not power the system down, and displays the message:

```
*** ERROR ***
```

### poweron

The **poweron** command requests the RCM module to power on the system. For the system power to come on, the following conditions must be met:

- AC power must be present at the power supply inputs.
- The DC On/Off button must be in the "on" position.
- All system interlocks must be set correctly.

The RCM firmware console exits command mode and reconnects the user's terminal to the system console port.

```
RCM>poweron
Focus returned to COM port
```

*NOTE: If the system is powered off with the DC On/Off button, the system will not power up. The RCM will not override the "off" state of the DC On/Off button. If the system is already powered on, the* ***poweron*** *command has no effect.*

### quit

The **quit** command exits the user from command mode and reconnects the user's terminal to the system console port. The following message is displayed:

```
Focus returned to COM port
```

The next display depends on what the system was doing when the RCM was invoked. For example, if the RCM was invoked from the SRM console prompt, the console prompt will be displayed when you enter a carriage return. Or, if the RCM was invoked from the operating system prompt, the operating system prompt will be displayed when you enter a carriage return.

## reset

The **reset** command requests the RCM module to perform a hardware reset. It is functionally equivalent to pressing the Reset button on the system operator control panel.

```
RCM>reset
Focus returned to COM port
```

The following events occur when the **reset** command is executed:

- The system restarts and the system console firmware reinitializes.
- The console exits RCM command mode and reconnects the user's terminal to the server's COM1 serial port.
- The power-up messages are displayed, and then the console prompt is displayed or the operating system boot messages are displayed, depending on the state of the Halt button.

## setesc

The **setesc** command allows the user to reset the default escape sequence for entering console mode. The escape sequence can be any character string. A typical sequence consists of 2 or more characters, to a maximum of 15 characters. The escape sequence is stored in the module's on-board NVRAM.

*NOTE: If you change the escape sequence, be sure to record the new sequence. Although the module factory defaults can be restored if the user has forgotten the escape sequence, this involves accessing the server control module and moving a jumper.*

The following sample escape sequence consists of five iterations of the Ctrl key and the letter "o".

```
RCM>setesc
^o^o^o^o^o
RCM>
```

If the escape sequence entered exceeds 15 characters, the command fails with the message:

```
*** ERROR ***
```

When changing the default escape sequence, avoid using special characters that are used by the system's terminal emulator or applications.

Control characters are not echoed when entering the escape sequence. To verify the complete escape sequence, use the **status** command.

### setpass

The **setpass** command allows the user to change the modem access password that is prompted for at the beginning of a modem session. The password is stored in the module's on-board NVRAM.

```
RCM>setpass
new pass>*********
RCM>
```

The maximum password length is 15 characters. If the password entered exceeds 15 characters, the command fails with the message:

```
***  ERROR ***
```

The minimum password length is one character, followed by a carriage return. If only a carriage return is entered, the command fails with the message:

```
*** ERROR - illegal password ***
```

If the user has forgotten the password, a new password can be entered.

### status

The **status** command displays the current state of the server's sensors, as well as the current escape sequence and alarm information.

```
RCM>status

Firmware Rev: V1.0
Escape Sequence: ^]^]RCM
Remote Access: ENABLE/DISABLE
Alerts: ENABLE/DISABLE
Alert Pending: YES/NO (C)
Temp (C): 26.0
RCM Power Control: ON/OFF
External Power: ON
Server Power: OFF

RCM>
```

The status fields are explained in Table 2–4.

**Table 2–4 RCM Status Command Fields**

| Item | Description |
|---|---|
| Firmware Rev: | Revision of RCM firmware. |
| Escape Sequence: | Current escape sequence to enter RCM firmware console. |
| Remote Access: | Modem remote access state. (ENABLE/DISABLE) |
| Alerts: | Alert dial-out state. (ENABLE/DISABLE) |
| Alert Pending: | Alert condition triggered. (YES/NO) |
| Temp (C): | Current system temperature in degrees Celsius. |
| RCM Power Control: | Current state of RCM system power control. (ON/OFF) |
| External Power: | Current state of power from external power supply to server control module. (ON/OFF) |

## 2.10.4 Dial-Out Alerts

**The RCM can be configured to automatically dial out through the modem (usually to a paging service) when it detects a power failure within the system. When a dial-out alert is triggered, the RCM initializes the modem for dial-out, sends the dial-out string, hangs up the modem, and reconfigures the modem for dial-in. The RCM and modem must continue to be powered, and the phone line must remain active, for the dial-out alert function to operate.**

### Example 2–19 Configuring the Modem for Dial-Out Alerts

```
P00>>> set rcm_dialout "ATDTstring#;"  ❶

RCM>enable
RCM>status
.
.
Remote Access:  Enable                ❷
.
RCM>alert_ena                         ❸
```

### Example 2–20 Typical RCM Dial-Out Command

```
P00>>> set rcm_dialout "ATXDT9,15085553333,,,,,,5085553332#;"
```

Use the **show** command to verify the RCM dial-out string:

```
P00>>> show rcm_dialout
rcm_dialout       ATXDT9,15085553333,,,,,,5085553332#;
```

### Enabling the Dial-Out Alert Function:

1. Enter the **set rcm_dialout** command, followed by a dial-out alert string, from the SRM console (see ❶ in Example 2–19).

   The *string* is a modem dial-out character string, not to exceed 47 characters, that is used by the RCM when dialing out through the modem. See the next topic for details on composing the modem dial-out string.

2. Enter the RCM firmware console and enter the **enable** command to enable remote access dial-in. The RCM firmware **status** command should display "Remote Access: ENABLE." (See ❷.)

3. Enter the RCM firmware **alert_ena** command to enable outgoing alerts. (See ❸.)

### Composing a Modem Dial-Out String

The modem dial-out string emulates a user dialing an automatic paging service. Typically, the user dials the pager phone number, waits for a tone, and then enters a series of numbers.

The RCM dial-out string (Example 2–20) has the following requirements:

- The entire string following the **set rcm_dialout** command must be enclosed by quotation marks.
- The characters ATDT must be entered after the opening quotation marks. Do not mix case. Enter the characters either in all uppercase or all lowercase.
- Enter the character X if the line to be used also carries voice mail. Refer to the example that follows.
- The valid characters for the dial-out string are the characters on a phone keypad: 0–9, *, and #. In addition, a comma (,) requests that the modem pause for 2 seconds, and a semicolon (;) is required to terminate the string.

### Elements of the Dial-Out String

| | |
|---|---|
| ATXDT | AT = Attention<br>X = Forces the modem to dial "blindly" (not look for a dial tone). Enter this character if the dial-out line modifies its dial tone when used for services such as voice mail.<br>D = Dial<br>T = Tone (for touch-tone)<br>, = Pause for 2 seconds. |
| 9, | In the example, "9" gets an outside line. Enter the number for an outside line if your system requires it. |
| `15085553333` | Dial the paging service. |
| `,,,,,,` | Pause for 12 seconds for paging service to answer |
| `5085553332#` | "Message," usually a call-back number for the paging service. |
| ; | Return to console command mode. Must be entered at end of string. |

## 2.10.5 Resetting the RCM to Factory Defaults

**If the escape sequence has been forgotten, you can reset the controller to factory settings.**

### Reset Procedure

1. Power down the server and access the server control module, as follows:

   Expose the PCI bus card cage. Remove three Phillips head screws holding the cover in place and slide it off the drawer. If necessary, remove several PCI and EISA options from the bottom of the PCI card cage until you have enough space to access the server control module.

2. Unplug the external power supply to the server control module.

   Locate the password and option reset jumper. The jumper number, which is etched on the board, depends on the revision of the server control module.

   *NOTE: If the RCM section of the server control module does not have an orange relay, the jumper number is J6. If the RCM section of the server control module has an orange relay, the jumper number is J7.*

3. Move the jumper so that it is sitting on both pins.

4. Replace any panels or covers as necessary so you can power up the system. Press the Halt button and then power up the system to the SRM console prompt. Powering up with the password and option reset jumper in place resets the escape sequence, password, and modem enable states to the factory default.

5. When the console prompt is displayed, power down the system and move the password and option reset jumper back onto the single pin.

6. Replace any PCI or EISA modules you removed and replace the PCI bus card cage cover.

7. Power up the system to the SRM console prompt and type the default escape sequence to enter RCM command mode:

   ```
   ^]^]RCM
   ```

8. Configure the module as desired. You must reset the password and modem enable states in order to enable remote access.

## 2.10.6 Troubleshooting Guide

**Table 2–5 lists a number of possible causes and suggested solutions for symptoms you might see.**

### Table 2–5 RCM Troubleshooting

| Symptom | Possible Cause | Suggested Solution |
|---|---|---|
| The local terminal will not communicate with the system or the RCM. | System and terminal baud rate set incorrectly. | Set the system and terminal baud rates to 9600 baud. |
| | Cables not correctly installed. | Review external cable installation. |
| RCM will not answer when the modem is called. | Modem cables may be incorrectly installed. | Check modem phone lines and connections. |
| | RCM remote access is disabled. | Enable remote access. |
| | RCM does not have a valid password set. | Set password and enable remote access. |
| | The local terminal is currently in the RCM console firmware. | Issue a **quit** command on the local terminal. |
| | On power-up, the RCM defers initializing the modem for 30 seconds to allow the modem to complete its internal diagnostics and initialization. | Wait 30 seconds after powering up the system and RCM before attempting to dial in. |
| | Modem may have had power cycled since last being initialized or modem is not set up correctly. | Enter **enable** command from RCM console. |

**Table 2–5 RCM Troubleshooting (Continued)**

| Symptom | Possible Cause | Suggested Solution |
|---|---|---|
| After the system and RCM are powered up, the COM port seems to hang and then starts working after a few seconds. | This delay is normal behavior. | Wait a few seconds for the COM port to start working. |
| RCM installation is complete, but system will not power up. | RCM Power Control: is set to DISABLE. | Enter RCM console and issue the **poweron** command. |
| New password, escape sequence, and modem enable state are forgotten when system and RCM module are powered down. | The password and option reset jumper is still installed. If the RCM section of the server control module does not have an orange relay, the jumper number is J6. If it does have an orange relay, the number is J7. | After resetting RCM to factory defaults, move the jumper so that it is sitting on only one pin. |
| The remote user sees a "+++" string on the screen. | The modem is confirming whether the modem has really lost carrier. This occurs when the modem sees an idle time, followed by a "3," followed by a carriage return, with no subsequent traffic. If the modem is still connected, it will remain so. | This is normal behavior. |
| The message "unknown command" is displayed when the user enters a carriage return by itself. | The terminal or terminal emulator is including a linefeed character with the carriage return. | Change the terminal or terminal emulator setting so that "new line" is not selected. |

*Continued on next page*

**Table 2–5 RCM Troubleshooting (Continued)**

| Symptom | Possible Cause | Suggested Solution |
|---|---|---|
| Cannot enable modem or modem will not answer. | The modem is not configured correctly to work with the RCM. | Modify the modem initialization and/or answer string as described in Section 2.10.7. |

## 2.10.7 Modem Dialog Details

**This section provides further details on the dialog between the RCM and the modem and is intended to help you reprogram your modem if necessary.**

### Phases of Modem Operation

The RCM is programmed to expect specific responses from the modem during four phases of operation:

- Initialization
- Ring detection
- Answer
- Hang-up

The initialization and answer command strings are stored in the RCM NVRAM. The factory default strings are:

Initialization string: `AT&F0EVS0=0S12=50<cr>`

Answer string `ATXA<cr>`

*NOTE: All modem commands must be terminated with a <cr> character (0x0d hex).*

### Initialization

The RCM initializes the modem to the following configuration:

Factory defaults (`&F0`)

No Echo (`E`)

Numeric response codes (`V`)

No Auto Answer (`S0=0`)

Guard-band = 1 second (`S12=50`)

Fixed modem-to-RCM baud rate

Connect at highest possible reliability and speed

The RCM expects to receive a "**`0<cr>`**" (OK) in response to the initialization string. If it does not, the **enable** command will fail.

This default initialization string works on a wide variety of modems. If your modem does not configure itself to these parameters, the initialization string will need to be modified. See the topic in this section entitled Modifying Initialization and Answer Strings.

### Ring Detection

The RCM expects to be informed of an in-bound call by the modem signaling the RCM with the string, "2<cr>" (RING).

### Answer

When the RCM receives the ring message from the modem, it responds with the answer string. The "X" command modifier used in the default answer string forces the modem to report simple connect, rather than connect at *xxxx*. The RCM expects a simple connect message, "1<cr>" (CONNECTED). If the modem responds with anything else, the RCM forces a hang-up and initializes the modem.

The default answer string is formatted to request the modem to provide only basic status. If your modem does not provide the basic response, the answer string, and/or initialization string will need to be modified. See the topic in this section entitled Modifying Initialization and Answer Strings.

After receiving the "connect" status, the modem waits for 6 seconds, and then prompts the user for a password.

### Hangup

When the RCM is requested to hang up the modem, it forces the modem into command mode and issues the **hangup** command to the modem. This is done by pausing for a minimum of the guard time, sending the modem "+++". When the modem responds with "0<cr>" (OK), the hang-up command string is sent. The modem should respond with "3<cr>" (NO CARRIER). After this interchange, the modem is reinitialized in preparation for the next dial-in session.

### RCM/Modem Interchange Overview

Table 2–6 summarizes the actions between the RCM and the modem from initialization to hangup.

## Table 2–6 RCM/Modem Interchange Summary

| Action | Data to modem | Data from Modem |
|---|---|---|
| Initialization command | AT&F0EVS0=0S12=50<cr> | |
| Initialization successful | | 0<cr> |
| Phone line ringing | | 2<cr> |
| RCM answering | ATXA<cr> | |
| Modem successfully connected | | 1<cr> |
| Force modem into command mode | <guard_band>+++ | |
| Modem in command mode | | 0<cr> |
| Hangup | ATH<cr> | |
| Successful hangup | | 3<cr> |

### Modifying Initialization and Answer Strings

The initialization and answer strings are stored in the RCM's NVRAM. They come pre-programmed to support a wide selection of modems. In the circumstance where the default initialization and answer strings do not set the modem into the desired mode, the following SRM **set** and **show** commands are provided to enable the user to define and examine the initialization and answer strings.

**To replace the initialization string**:

```
P00>>>set rcm_init " new_init_string"
```

**To replace the answer string:**

```
P00>>>set rcm_answer "new_answer_string"
```

**To display all the RCM user settable strings:**

```
P00>>>show rcm*
rcm_answer      ATXA
rcm_dialout
rcm_init        AT&F0EVS0=0S12=50
P00>>>
```

### Initialization and Answer String Substitutions

The following modems have been identified as requiring modified initialization and/or answer strings.

| Modem Model | Initialization String | Answer String |
|---|---|---|
| RCM Defaults | "AT&F0EVS0=0S12=50" | "ATXA" |
| USRobotics Sportster 28,800 Data/Fax Modem | RCM default | "ATX0&B1&A0A" |

# Chapter 3

# Troubleshooting

This chapter contains basic troubleshooting information. For more about troubleshooting, see the *AlphaServer 4100 Service Manual*.

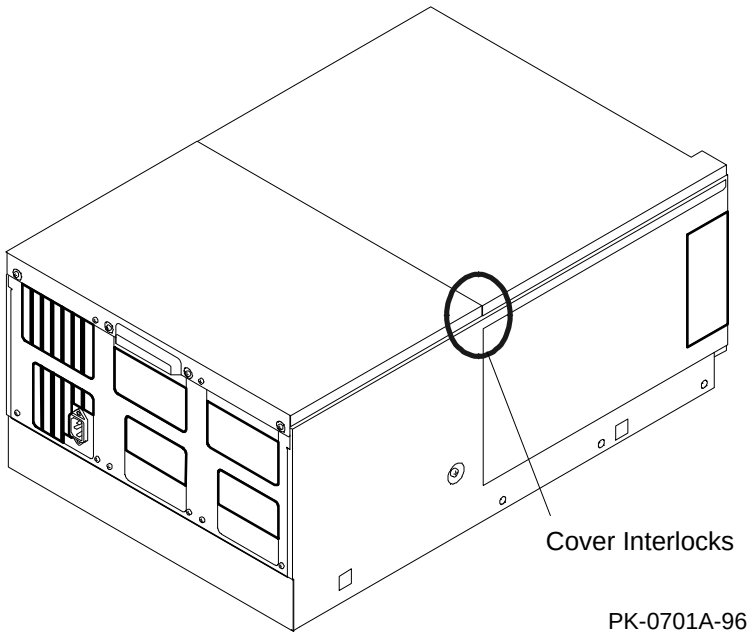
Sections in this chapter are:

- System Drawer Does Not Power Up
- Control Panel Display Is Blank
- Control Panel Display Indicates an Error

## 3.1 System Drawer Does Not Power Up

**When the system drawer does not power up, the problem may be a disconnected power cord, a faulty fan, or a disengaged cover interlock.**

**Figure 3–1 Location of Cover Interlocks**



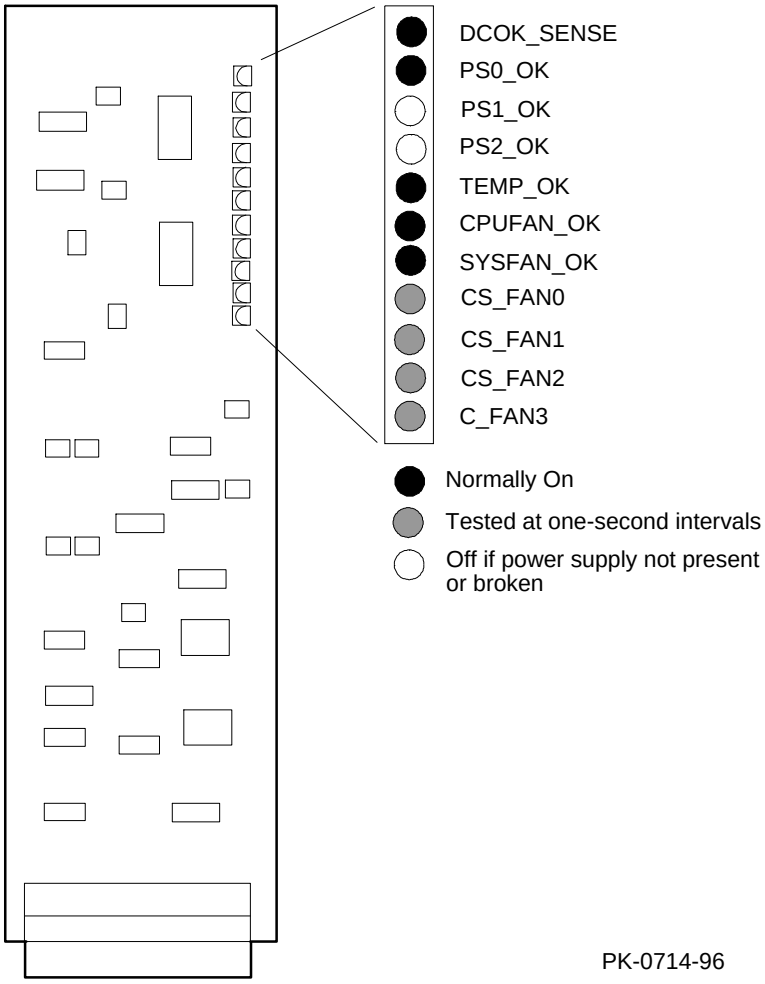

### If the system drawer does not power up

- Is the system drawer power cord plugged into the power strip? Is it secure in the system drawer connector?
- Is the pedestal or cabinet AC power cord plugged in?
- Are the covers on the system drawer closed and the interlocks engaged?

  The cover interlocks must be engaged to enable power-up. See Figure 3–1 for interlock locations. (The three interlocks are all in the area highlighted in the illustration.)

- Is the power supply operating?
- Are the system drawer fans operating?
- Are the CPU fans operating?
- Is the power control module securely seated in its connector?
- What is the status of the power control module LEDs?

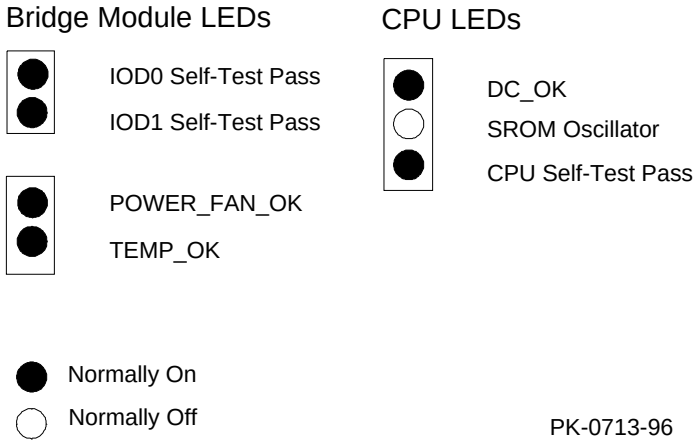
**Figure 3–2 Power Control Module LEDs**



## 3.2 Control Panel Display Is Blank

**If the system drawer powers up but the control panel display is blank, the problem may be a CPU module, the bridge module, the PCI motherboard, the control panel display, or the cable connecting the PCI motherboard and the display.**

**Figure 3–3 CPU and Bridge Module LEDs**



### If the system drawer powers up but the control panel display is blank

- Did all Self-Test Pass (STP) LEDs on the CPU modules light?

  *NOTE: Running AlphaBIOS causes the CPU STP LEDs to turn off. This is normal behavior.*

  If all STP LEDs are on, check the connection between the PCI motherboard and the control panel. If the cable is connected and not damaged, the fault may be in either the PCI motherboard or the control panel.

  If all STP LEDs are off and the DC_OK LED is on, the fault may be the CPU module or, less likely, the motherboard.

- Did all "OK" LEDs (DC_OK, POWER_FAN_OK, and TEMP_OK) on the CPU modules light?

  If any OK LED is off, the fault may be in the power system. Check the power supplies, power control module, and all connections to both.

- Did all STP LEDs on the bridge module light?

  If all bridge STP LEDs are off, the fault is likely the motherboard or the bridge module. Another possible, but less likely, cause may be the oscillator on CPU 0. If this is a multiprocessor system, insert a different module in the CPU 0 slot.

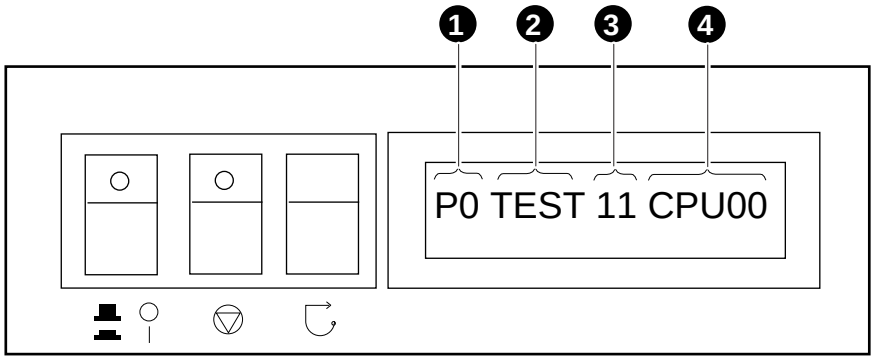
  If one bridge STP LED is off, the fault may be the bridge module or the motherboard.

*NOTE: In all cases listed above, the memory modules can be ruled out as a possible cause.*

## 3.3 Control Panel Display Indicates an Error

**The control panel display indicates the likely device when testing fails.**

### Figure 3–4 Control Panel Display




PK-0706E-96

#### If the control panel display indicates an error

Check the failing device called out in the control panel display. See Figure 3–4 and Table 3–1 for the location and meaning of the fields in the display.

**Table 3–1 Control Panel Display**

| Field | Content | Display | Meaning |
|---|---|---|---|
| ❶ | CPU number | P0–P3 | CPU reporting status |
| ❷ | Status | TEST | Tests are executing |
| | | FAIL | Failure has been detected |
| | | MCHK | Machine check has occurred |
| | | INTR | Error interrupt has occurred |
| ❸ | Test number (for Digital use only) | | |
| ❹ | Suspected device | CPU0–3 | CPU module number[1] |
| | | MEM0–3 and L, H, or * | Memory pair number and low module, high module, or either[2] |
| | | IOD0 | Bridge to PCI bus 0[3] |
| | | IOD1 | Bridge to PCI bus 1[3] |
| | | FROM0 | Flash ROM[4] |
| | | COMBO | COM controller[4] |
| | | PCEB | PCI-to-EISA bridge[4] |
| | | ESC | EISA system controller[4] |
| | | NVRAM | Nonvolatile RAM[4] |
| | | TOY | Real time clock[4] |
| | | I8242 | Keyboard and mouse controller[4] |

[1] CPU module

[2] Memory module

[3] Bridge module

[4] PCI motherboard

# Chapter 4

# Installing Components

The system drawer can be upgraded to four CPU modules, four memory pairs, and a total of eight PCI cards and EISA cards (three EISA cards maximum). When upgrading to three CPU modules, a second power supply is required: a third can be installed for redundancy.

The procedure for opening the system drawer is different for pedestal systems and cabinet systems. Instructions for both systems are included.

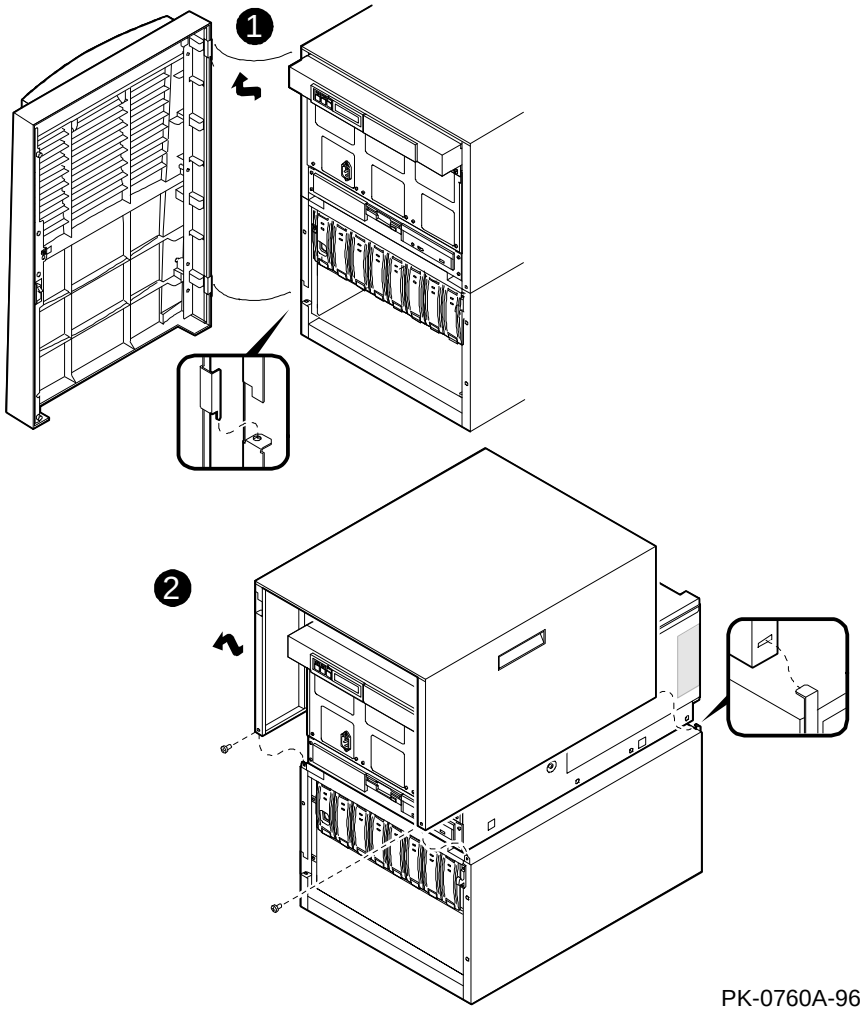
Sections in this chapter are:

- Exposing the System Drawer in a Pedestal
- Exposing the System Drawer in a Cabinet
- Removing Covers from the System Drawer
- Installing a CPU Module
- Installing a Memory Pair
- Installing a PCI or EISA Card
- Installing a Power Supply

## 4.1 Exposing the System Drawer in a Pedestal

**Open and remove the front door, and remove the top tray. For a power supply installation, also remove the pedestal tray.**

**Figure 4–1 Pedestal Door and Cover Removal**



*CAUTION: Wear an antistatic wrist strap when working on this system.*

### For CPU and memory modules, PCI/EISA cards, and power supplies

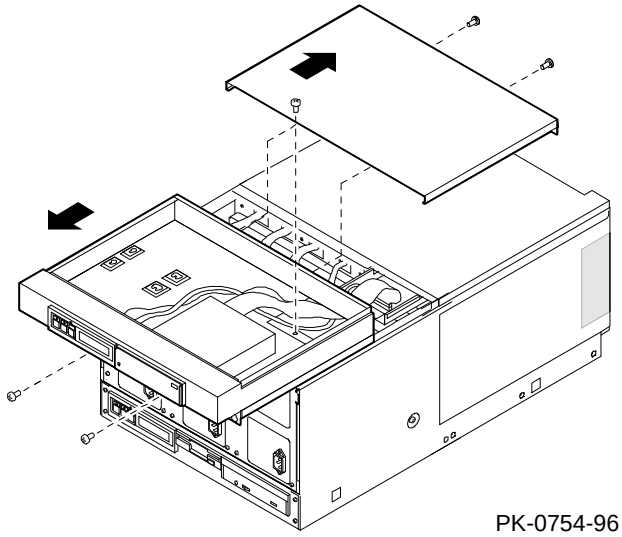
1. Open the front door of the pedestal and remove it by lifting and pulling it away from the system. (See ❶ in Figure 4–1.)
2. Remove the top cover from the pedestal. Remove the two Phillips head retaining screws midway up each side of the pedestal, tilt the cover up, and lift it away from the frame. (See ❷.)
3. If you are installing a power supply, remove the pedestal tray by following the procedure below.

### For power supplies

If you are installing a power supply, remove the front door and the top cover of the pedestal, as explained above. You must also remove the pedestal tray:

1. Remove the tray cover by loosening the retaining screws at the back of the tray. (See Figure 4–2.)
2. Disconnect the cables from the control panel and any optional SCSI device to the bulkhead connector in the rear right corner of the tray.
3. Unscrew the Phillips head retaining screw holding the bulkhead to the tray.
4. Unscrew the two Phillips head retaining screws and slide the tray off the drawer.

## Figure 4–2 Pedestal Tray Removal

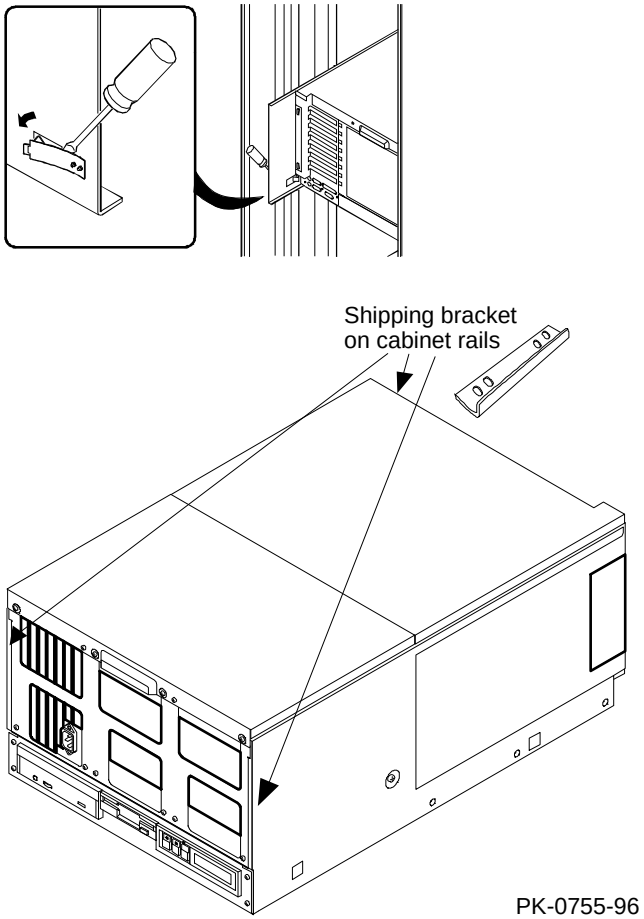


PK-0754-96

## 4.2 Exposing the System Drawer in a Cabinet

**Open both doors, disconnect cables that obstruct movement of the drawer, remove the shipping brackets, and slide the drawer out from the cabinet.**

### Figure 4–3 Exposing System Drawer in Cabinet



*CAUTION: Wear an antistatic wrist strap when working on this system.*

### For CPU and memory modules and PCI/EISA cards

1. Open the front and rear doors of the cabinet.
2. At the front of the cabinet, unplug the power supplies in the drawer.
3. At the rear, remove the two Phillips screws holding the shipping bracket on the mounting rail at the right.
4. Using a flathead screwdriver, disengage the lock mechanism at the lower left corner of the drawer. Slide the drawer out part way and release the lock mechanism by removing the screwdriver.
5. Slide the drawer out until it locks.

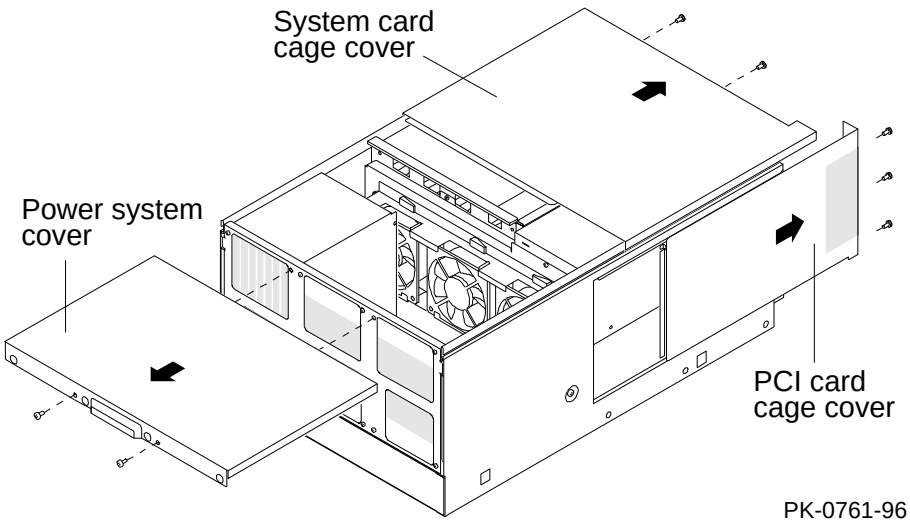
### For power supplies

1. Open the front and rear doors of the cabinet.
2. At the rear of the cabinet, if the cables from any PCI options might interfere with pulling the drawer forward, remove the cables.
3. At the front, remove the shipping brackets on the right and left rails that hold the drawer.
4. Pull out the drawer until it locks.

## 4.3 Removing Covers from the System Drawer

**The system drawer has covers on the system bus, PCI, and power sections. Remove a cover by removing the screws and sliding the cover from the system drawer.**

**Figure 4–4 System Drawer Covers**

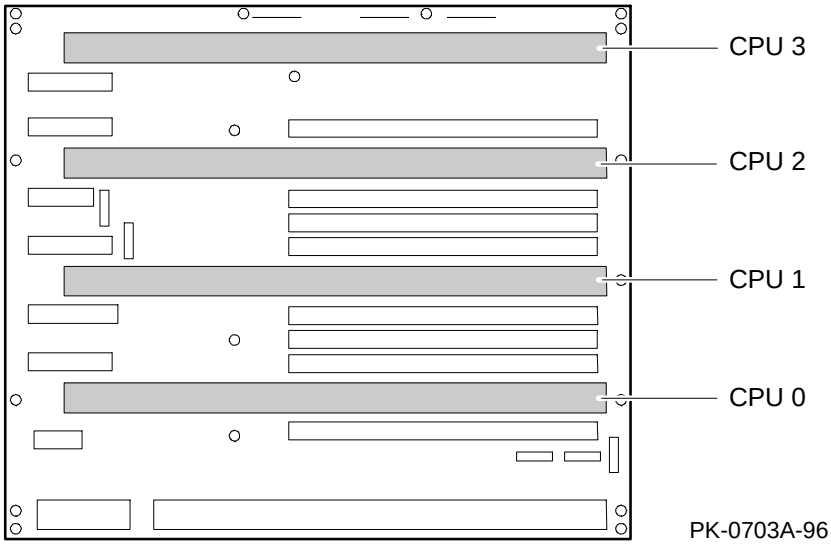


Each of the three covers shown in Figure 4–4 is attached to the system drawer by Phillips head screws. The locations of these screws is shown in the illustration. Detach any of the covers by removing the screws and sliding the cover in the direction shown.

## 4.4 Installing a CPU Module

**Remove the system bus cover from the system drawer and insert the module. To verify the installation, issue the show cpu command from the SRM console or display the system bus configuration from the AlphaBIOS console.**

### Figure 4–5 CPU Module Slots




*WARNING: CPU modules and memory modules have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.*

*CAUTION: Wear an antistatic wrist strap when working on this system.*

#### Installation

1. Shut down the operating system and power down the system.
2. Expose the system drawer by following the instructions in Section 4.1 for a pedestal system or Section 4.2 for a cabinet system.
3. Remove the cover from the system card cage section of the system drawer. See Section 4.3.

4. Insert the new CPU module:
    a. Determine the location of the module. CPU modules are inserted in order, starting at CPU slot 0. See Figure 4–5.
    b. The module handle has a lever at each end. Insert the module in the connector and push down on both levers at the same time. See Figure 4–6.

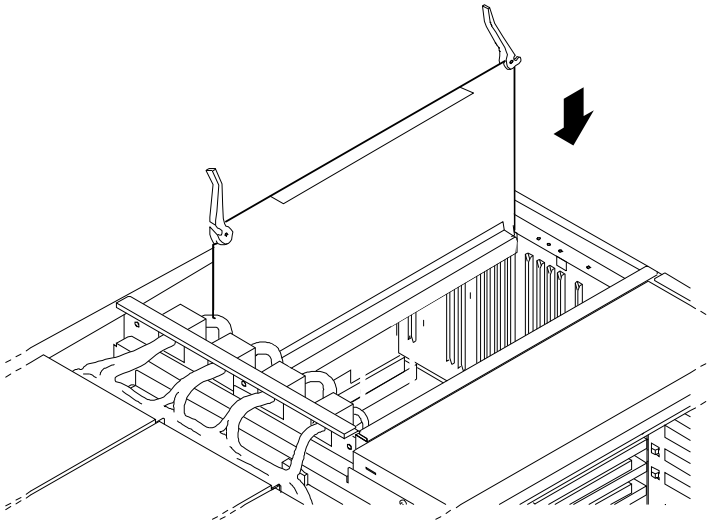
### Verification — Digital UNIX and OpenVMS Systems

- Issue the **show cpu** command to display the status of the new module.

### Verification — Windows NT Systems

1. Start **AlphaBIOS Setup**, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **MC Bus Configuration** to display the status of the new module.

## Figure 4–6 CPU Module Installation

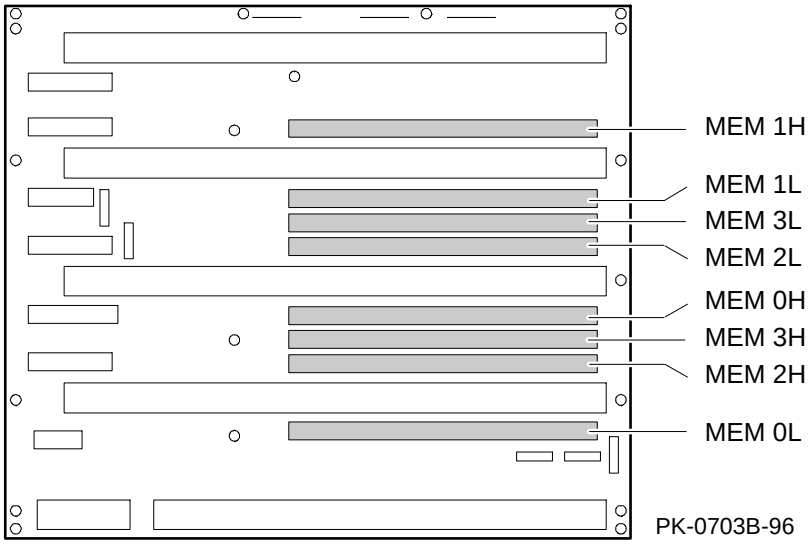


PK-0718-96

## 4.5 Installing a Memory Pair

**Remove the system bus cover from the system drawer and insert the modules. To verify the installation, issue the show memory command from the SRM console or display the memory configuration from the AlphaBIOS console.**

### Figure 4–7 Memory Pair Slots




*WARNING: Memory modules and CPU modules have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.*

*CAUTION: Wear an antistatic wrist strap when working on this system.*

#### Installation

1. Shut down the operating system and power down the system.
2. Expose the system drawer by following the instructions in Section 4.1 for a pedestal system or Section 4.2 for a cabinet system.
3. Remove the cover from the system card cage section of the system drawer. See Section 4.3.

4. Insert the new memory pair in the appropriate slots.
    a. Determine the location of the two modules. See Figure 4–7, which identifies the slots for the high and low halves of each memory pair. Insert the new pair so that all memory in the system drawer is in descending order by size. That is, the largest memory must be in slots 0 Low and 0 High, the next largest in slots 1 Low and 1 High, and so on.
    b. The memory module is held in place by a flathead captive screw attached to the top brace of the module. Insert the module and attach the screw. See Figure 4–8.

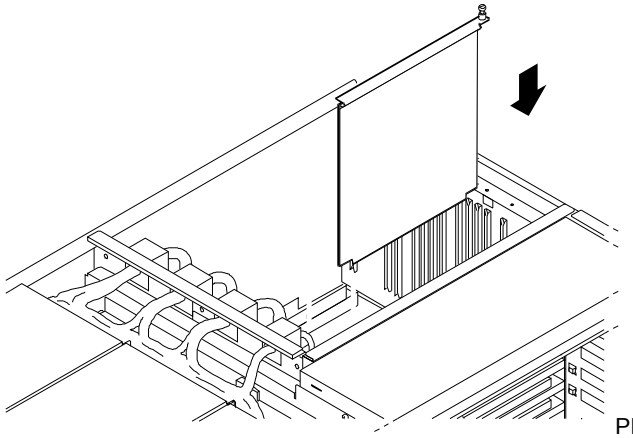
### Verification — Digital UNIX and OpenVMS Systems

1. Issue the **show memory** command to display the status of the new memory.
2. Verify the functioning of the new memory by issuing the command **test mem***n*, where *n* is 0, 1, 2, 3, or *.

### Verification — Windows NT Systems

1. Start **AlphaBIOS Setup**, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **Memory Configuration** to display the status of the new memory.
3. Switch to the SRM console (press the Halt button in so that the LED on the button lights and reset the system). Verify the functioning of the new memory by issuing the command **test mem***n*, where *n* is 0, 1, 2, 3, or *.

## Figure 4–8 Memory Module Installation

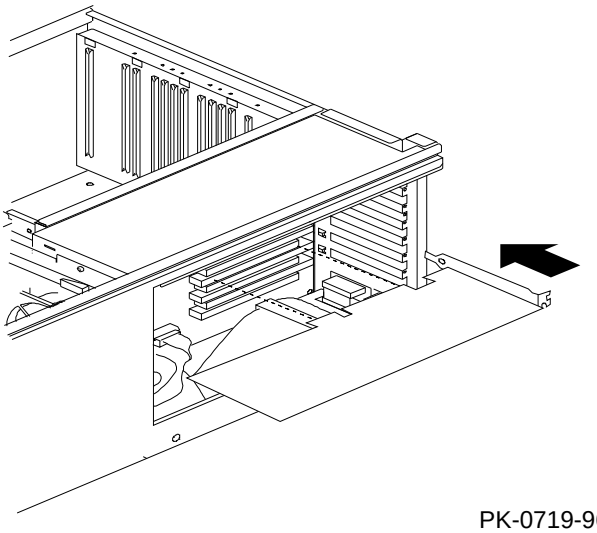


PK-0716A-96

## 4.6 Installing a PCI or EISA Card

**Remove the PCI bus cover from the system drawer and insert the card. If it is an EISA card, run the EISA Configuration Utility. To verify the installation, issue the show config and show device commands from the SRM console or display the PCI or EISA configuration from the AlphaBIOS console.**

### Figure 4–9 PCI/EISA Card Installation



PK-0719-96

*WARNING: To prevent fire, use only modules with current limited outputs. See National Electrical Code NFPA 70 or Safety of Information Technology Equipment, Including Electrical Business Equipment EN 60 950.*

#### Installation

1. Shut down the operating system and power down the system.
2. Expose the system drawer by following the instructions in Section 4.1 for a pedestal system or Section 4.2 for a cabinet system.
3. Remove the cover from the PCI card cage section of the system drawer. See Section 4.3.

4. Install the option card:
   a. Remove and discard the bulkhead filler plate from the PCI slot.
   b. Insert the card into the connector as shown in Figure 4–9. (See Figure 4–10 for the location of PCI and EISA slots.)
   c. Connect cables and secure the module to the card cage with a Phillips head screw.
5. If you installed an EISA card, after powering up the system run the EISA Configuration Utility. See Chapter 7.

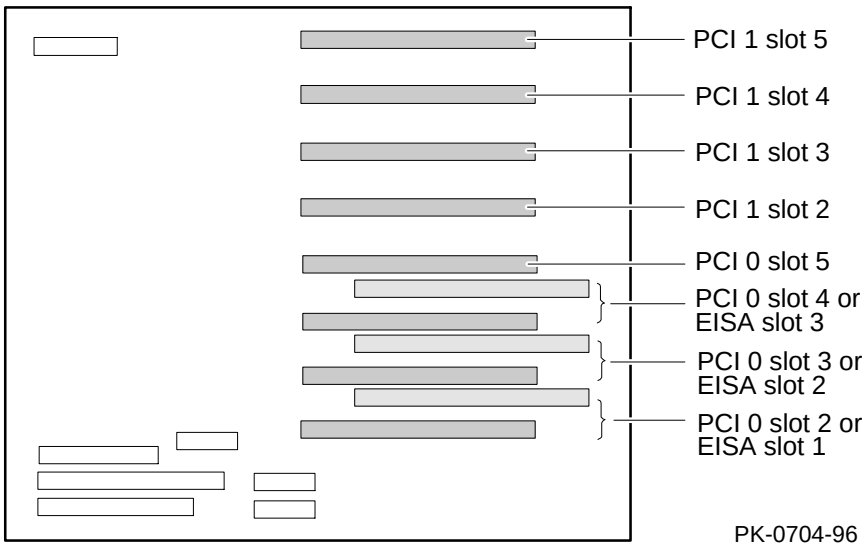
### Verification — Digital UNIX and OpenVMS Systems

- Issue the SRM commands **show configuration** and **show device** to determine that the new option is listed.

### Verification — Windows NT Systems

1. Start **AlphaBIOS Setup**, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **PCI Configuration** or **EISA Configuration** to determine that the new option is listed.

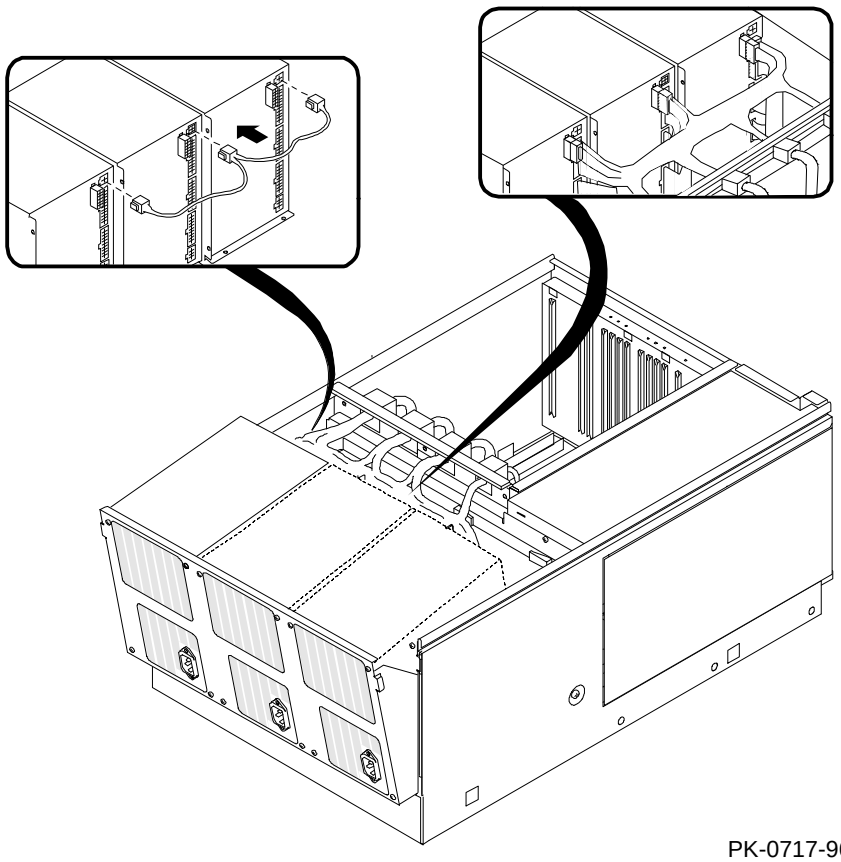
## Figure 4–10 PCI and EISA Slots



## 4.7 Installing a Power Supply

**Remove the power supply cover and install the power supply. To verify the installation, issue the show power command from the SRM console.**

**Figure 4–11 Power System Cabling**



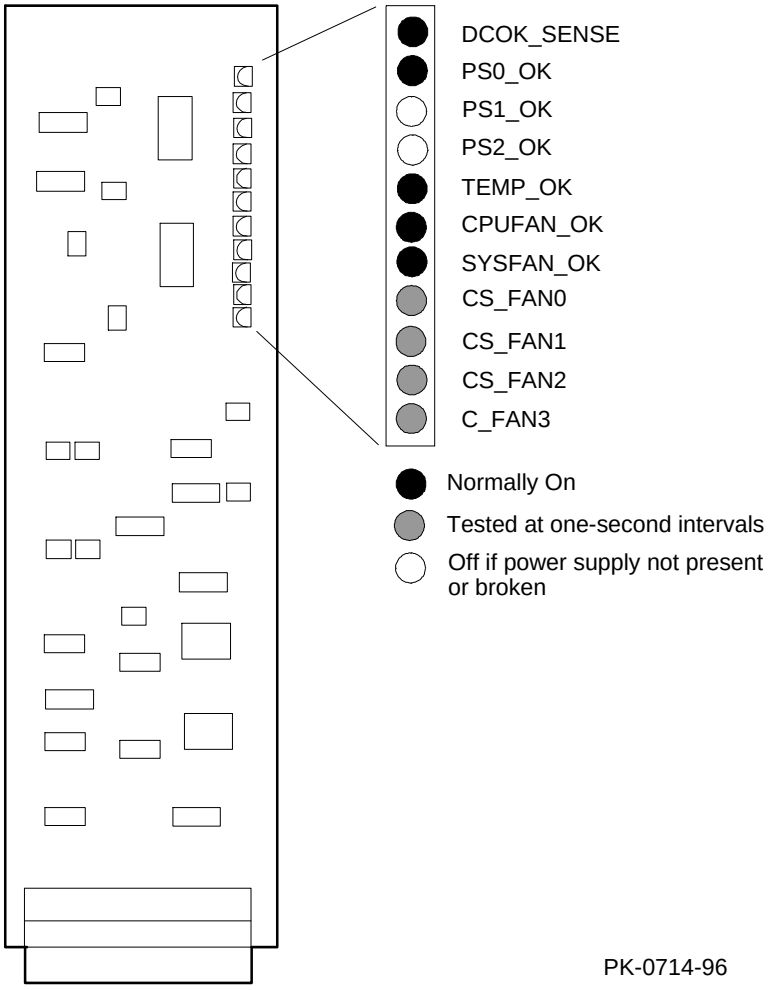
### Installation

1. Shut down the operating system and power down the system.
2. Expose the power system section of the system drawer by following the instructions in Section 4.1 for a pedestal system or Section 4.2 for a cabinet system.
3. Remove the cover from the power system section of the system drawer. See Section 4.3.
4. Install the power supply:
    a. Release the power supply tray by removing the two Phillips head screws on the side of the drawer.
    b. Lift the power supply tray to release it from the sheet metal and slide it out from the drawer until it locks.
    c. Tilt the tray to enable easier access to the back of the power supplies.
    d. Insert the power supply in the appropriate space. (See Section 1.3.9 for configuration rules.)
    e. Insert the four Phillips head screws at the front of the tray to hold the power supply in place.
    f. Attach the connectors at the rear of the power supply.
5. Replace the cover on the power system section and replace the pedestal tray.

### Verification

1. Issue the SRM **show power** command to display the status of the new power supply.
2. Check the LEDs on the power control module to determine that the power supply is functioning properly. See Figure 4–12.

**Figure 4–12 Power Control Module LEDs**



# Part 2

# User Interface

# Chapter 5

# SRM Console

The SRM console is the command line interface that supports the Digital UNIX and OpenVMS operating systems.

This chapter describes typical functions a user performs, along with descriptions of commands and environment variables. Sections in this chapter are:

- Switching from SRM to the AlphaBIOS Console
- Displaying the System Configuration
- Booting the Operating System
- Configuring the System
  - Configuring DSSI
  - Configuring a PCI NVRAM Module
  - Configuring EISA and RAID
- Testing the System
- Stopping and Starting CPUs
- Updating Firmware
- Forcing a System Crash Dump
- Using Environment Variables
- Depositing and Examining Data
- Reading a File
- Initializing the System
- Finding Help
- Command Summary
- Environment Variable Summary

## 5.1 Switching from SRM to the AlphaBIOS Console

**It is necessary to switch to the AlphaBIOS console to run configuration utilities. To switch from SRM to AlphaBIOS, issue the alphabios command.**

### Example 5–1 Switching to the AlphaBIOS Console

```
P00>>> alphabios
```

### alphabios

The **alphabios** command loads and starts the AlphaBIOS console. This is necessary for running AlphaBIOS-based utilities (such as the RAID and EISA configuration utilities). The syntax is:

**alphabios**

To return to the SRM console, do one of the following:

- If the **os_type** environment variable is set to **unix** or **openvms**, reset the system by pressing the Reset button.
- If the **os_type** environment variable is set to **nt**, press the Halt button in (the LED on the button lights) and reset the system by pressing the Reset button. Alternatively, select **Digital UNIX (SRM)** or **OpenVMS (SRM)** from the **Advanced CMOS Setup** screen and reset the system.

To switch permanently to the AlphaBIOS console (that is, if the permanent operating system is changed to Windows NT), do the following:

1. Set the SRM **os_type** environment variable to **nt**.
2. Set the SRM **console** environment variable to **graphics**.
3. Reset the system with the Reset button or by power cycling.
4. Run the EISA Configuration Utility, using the ECU program for Windows NT. See Chapter 7.

## 5.2 Displaying the System Configuration

**Several commands are used to display the system configuration: show config, show cpu, show device, show fru, show memory, show network, show pal, show power, and show version.**

### Example 5–2 Show Config Command

```
P00>>> show config
              Digital Equipment Corporation
                    AlphaServer 4100

Console V2.0-1 OpenVMS PALcode V1.18-8, Digital UNIX PALcode V1.21-12

 Module                          Type     Rev    Name
 System Motherboard              0        0000   mthrbrd0
 Memory  128 MB SYNC             0        0000   mem0
 Memory  128 MB SYNC             0        0000   mem1
 CPU (1MB Cache)                 1        0000   cpu0
 CPU (1MB Cache)                 1        0000   cpu1
 Bridge (IOD0/IOD1)              600      0021   iod0/iod1
 PCI Motherboard                 8        0000   saddle0

 PCI0 (IOD0)
 Slot   Option Name              Type     Rev    Name
 1      PCEB                     4828086  0004   pceb0
 2      DEC PCI MC               181011   008B   mc0
 4      DEC PCI FDDI             f1011    0000   pfi0
 5      PBXGA-AA                 41011    0003   tga0

 eisa0 (EISA connected to pceb0)
 Slot   Option Name              Type     Rev    Name

 PCI1 (IOD1)
 Slot   Option Name              Type     Rev    Name
 1      NCR 53C810               11000    0002   ncr0
 2      DEC KZPSA                81011    0000   kzpsa0
 3      NCR 53C810               11000    0002   ncr1
 4      DECchip 21040-AA         21011    0023   tulip0
P00>>>
```

The **show config** command displays a list of devices found on the system bus and I/O buses. This is the configuration at the most recent initialization. The syntax is:

**show config**

### Example 5–3 Show CPU Command

```
P00>>> show cpu

Primary CPU:      00
Active CPUs:      00    01
Configured CPUs:  00    01
SROM Revision:    V2.0  V2.0

P00>>>
```

The **show cpu** command displays the status of all CPUs in the system drawer. The syntax is:

**show cpu**

### Example 5–4 Show Device Command

```
P00>>> show device
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1      DKa500                        RRD45   1645
polling ncr1 (NCR 53C810) slot 3, bus 0 PCI, hose 1   SCSI Bus ID 7
dkb100.1.0.3.1      DKb100                        RZ29B   0007
dkb200.2.0.3.1      DKb200                        RZ29B   0007
dkb400.4.0.3.1      DKb400                        RZ29B   0007
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0     DVA0                           RX23
polling tulip0 (DECchip 21040-AA) slot 2, bus 0 PCI, hose 1
ewa0.0.0.2.1: 08-00-2B-E5-B4-1A

P00>>>
```

The **show device** command displays status for devices and controllers in the system drawer: SCSI and MSCP devices, the internal floppy drive, and the network. The syntax is:

**show device** [*controller_name*]

*controller_name* The controller name or abbreviation. When abbreviations or wildcards are used, all controllers that match the type are displayed. If no name is given, the display is a list of all devices and controllers in the system.

An example of a device name is **dka500.5.0.1.1**. Table 5–1 shows the interpretation of this device name.

**Table 5–1 Device Naming Convention**

| | Category | Description | | | |
|---|---|---|---|---|---|
| **dk** | Driver ID | Two-letter designator of port or class driver | | | |
| | | dk | SCSI disk | fw | FDDI device |
| | | dr | RAID set device | mk | SCSI tape |
| | | du | DSSI disk | mu | DSSI tape |
| | | dv | Diskette drive | pk | SCSI port |
| | | ew | Ethernet port | pu | DSSI port |
| **a** | Storage adapter ID | One-letter designator of storage adapter (a, b, c...). | | | |
| **500** | Device unit number | Unique number (MSCP unit number). SCSI unit numbers are forced to 100 X node ID. | | | |
| **5** | Bus node number | Bus node ID. | | | |
| **0** | Channel number | Used for multi-channel devices. | | | |
| **1** | Logical slot number | Corresponds to PCI slot number. | | | |
| | | Slot | PCI 0 | PCI 1 | |
| | | 0 | Reserved | Reserved | |
| | | 1 | EISA bridge | Embedded SCSI CD-ROM | |
| | | 2 | PCI device | PCI device | |
| | | 3 | PCI or EISA device | PCI device | |
| | | 4 | PCI or EISA device | PCI device | |
| | | 5 | PCI or EISA device | PCI device | |
| **1** | Hose number | 0 — PCI 0<br>1 — PCI 1 | | | |

### Example 5–5 Show FRU Command

```
P00>>> show fru

                Digital Equipment Corporation
                      AlphaServer 4100

Console V2.0-1 OpenVMS PALcode V1.18-8, Digital UNIX PALcode V1.21-12

 Module                Part #    Type     Rev    Name      Serial #
 System Motherboard    23803-01  0        0000   mthrbrd0  KA01234567
 Memory  128 MB SYNC   N/A       0        0000   mem0      N/A
 Memory   64 MB SYNC   N/A       0        0000   mem1      N/A
 CPU (1MB Cache)       B2000-AA  1        0000   cpu0      AY12341234
 CPU (1MB Cache)       B2001-AA  1        0000   cpu1      AY12341234
 Bridge (IOD0/IOD1)    B3040-AA  600      0021   iod0/iod1 KA01234567
 PCI Motherboard       B3050-AA  8        0000   saddle0   KA01408896


 PCI0 (IOD0)
 Slot   Option Name              Type     Rev    Name
 1      PCEB                     4828086  0004   pceb0
 2      DEC PCI MC               181011   008B   mc0
 4      DEC PCI FDDI             f1011    0000   pfi0
 5      PBXGA-AA                 41011    0003   tga0


 eisa0 (EISA connected to pceb0)
 Slot   Option Name              Type     Rev    Name


 PCI1 (IOD1)
 Slot   Option Name              Type     Rev    Name
 1      NCR 53C810               11000    0002   ncr0
 2      DEC KZPSA                81011    0000   kzpsa0
 3      NCR 53C810               11000    0002   ncr1
 4      DECchip 21040-AA         21011    0023   tulip0
P00>>>
```

The **show fru** command displays information for CPU, memory, PCI, and EISA. This information is based on data in the serial control bus EPROM. The syntax is:

**show fru**

### Example 5–6 Show Memory Command

```
P00>>> show memory

Slot    Type       MB     Base
----    ----      ----    --------
 0       EDO       256    0
 1       EDO       128    10000000

Total              384
P00>>>
```

The **show memory** command displays information about each memory pair: slot number (see Figure 1–6 for memory slots), type (synchronous or EDO), size of the memory pair in megabytes, and the starting address of the pair. The syntax is:

**show memory**

### Example 5–7 Show Network Command

```
P00>>> show network
polling tulip0 (DECchip 21040-AA) slot 2, bus 0 PCI, hose 1
ewa0.0.0.2.1: 08-00-2B-E5-B4-1A
P00>>>
```

The **show network** command displays the assigned network device names and other information about network (FDDI and Ethernet) cards. The syntax is:

**show network**

### Example 5–8 Show PAL Command

```
P00>>> show pal
pal  OpenVMS PALcode V1.17-9, Digital UNIX PALcode V1.21-5
P00>>>
```

The **show pal** command displays the versions of Digital UNIX and OpenVMS PALcode. PALcode is the Alpha Privileged Architecture Library code, written to support Alpha processors. It implements architecturally defined processor behavior. The syntax is:

**show pal**

### Example 5–9 Show Power Command

```
P00>>> show power

                    Status
Power Supply 0       good
Power Supply 1       not present
Power Supply 2       good
System Fans          good
CPU Fans             good
Temperature          good

The system was last reset via a system software reset

0 Environmental events are logged in nvram
P00>>>
```

The **show power** command displays status information about the power supplies, system fans, CPU fans, and temperature. This command is useful for displaying the error state of a Digital UNIX or OpenVMS system that shuts down because of a fan, temperature, or power supply failure. If the system can be restarted, use this command. (If it cannot, use the RCM **status** command; see Section 2.10.3.)

The syntax is:

**show power**

### Example 5–10 Show Version Command

```
P00>>> show version
version                 V2.0-1 1-AUG-1996 13:59:28
P00>>>
```

The **show version** command displays the version of the SRM console program that is installed on the system. The syntax is:

**show version**

## 5.3 Booting the Operating System

**The boot command is used to boot the operating system.**

### Example 5–11 Boot Command

```
P00>>> boot dkb0
 SROM V1.0 on cpu0
 SROM V1.0 on cpu1
 SROM V1.0 on cpu2
 SROM V1.0 on cpu3
XSROM V1.0 on cpu2
XSROM V1.0 on cpu1
XSROM V1.0 on cpu0
XSROM V1.0 on cpu3
mem_pair0 - 256 MB
20..20..20..21..21..21..20..21..23..24..24..24..24..
Memory testing complete on cpu0
Memory testing complete on cpu1
Memory testing complete on cpu3
Memory testing complete on cpu2
starting console on CPU 0
sizing memory
  0    256 MB SYNC
starting console on CPU 1
starting console on CPU 2
starting console on CPU 3
probing IOD1 hose 1
  bus 0 slot 1 - NCR 53C810
  bus 0 slot 2 - DEC KZPSA
  bus 0 slot 3 - NCR 53C810
  bus 0 slot 4 - DECchip 21040-AA
  bus 0 slot 5 - DEC PCI MC
probing IOD0 hose 0
  bus 0 slot 1 - PCEB
  bus 0 slot 4 - DEC PCI FDDI
  bus 0 slot 5 - PBXGA-AA

configuring I/O adapters...
  ncr0, hose 1, slot 1, bus 0
  kzpsa0, hose 1, slot 2, bus 0
  ncr1, hose 1, slot 3, bus 0
  tulip0, hose 1, slot 4, bus 0
  floppy0, hose 0, slot 0, bus 1
  pfi0, hose 0, slot 4, bus 0
```

*Continued on next page*

### Example 5–11 Boot Command (Continued)

```
System temperature is 24 degrees C
AlphaServer 4100 Console V2.0-1, 24-JUL-1996 18:18:26

CPU 0 booting

(boot dkb0.0.0.2.1 -flags 0)
Building FRU table
.
.
.
setting affinity to the primary CPU
jumping to bootstrap code

    OpenVMS (TM) Alpha Operating System, Version 6.2-1H3
```

The **boot** command initializes the processor, loads a program image from the specified boot device, and transfers control to that image. The syntax is:

**boot [-file** *filename***] [-flags [***value***]] [-halt] [-protocols** *enet_protocol***] [***boot_dev***]**

| | |
|---|---|
| **-file** *filename* | The boot file. |
| **-flags** [*value*] | Specifies additional information to the loaded image or operating system. In Digital UNIX, specifies boot flags. In OpenVMS, specifies system root number and boot flags. This qualifier overrides the setting of the **boot_osflags** environment variable. See the **boot_osflags** environment variable for a list of settings and their meanings. |
| **-halt** | Forces the bootstrap operation to halt and invoke the console program once the bootstrap image is loaded and page tables and other data structures are set up. Console device drivers are not shut down. Transfer control to the image by entering the **continue** command. |
| **-protocols** *enet_protocol* | Either **mop** (default) or **bootp**. This qualifier overrides the setting of the **ew*0_protocols** environment variable. |
| *boot_dev* | A device path or list of devices from which the console program attempts to boot, or a saved boot specification in the form of an environment variable. This qualifier overrides the setting of the **bootdef_dev** environment variable. Use the **bootdef_dev** environment variable to define the default boot device string. |

## 5.4 Configuring the System

### 5.4.1 Configuring DSSI

**The set host command is used for system configuration when a DSSI device is in the system.**

#### Example 5–12 Set Host Command

```
P00>>> show device kfpsa0
   dud0.0.0.3.0               R2YQYA$DIA0             RF72
P00>>> set mode diag
P00>>> config kfpsa*
P00>>> set host -dup dud0
    starting DIRECT on pud0.0.0.3.0 (R2YQYA)

    Copyright (C) 1995  Digital Equipment Corporation
    PRFMON V1.0  D  2-NOV-1995 10:30:58
    DKCOPY V1.0  D  2-NOV-1995 10:30:58
    DRVEXR V2.0  D  2-NOV-1995 10:30:58
    DRVTST V2.0  D  2-NOV-1995 10:30:58
    HISTRY V1.0  D  2-NOV-1995 10:30:58
    DIRECT V1.0  D  2-NOV-1995 10:30:58
    ERASE  V2.0  D  2-NOV-1995 10:30:58
    VERIFY V1.0  D  2-NOV-1995 10:30:58
    DKUTIL V1.0  D  2-NOV-1995 10:30:58
    PARAMS V2.0  D  2-NOV-1995 10:30:58
    Total of 10 programs.
    Task?
P00>>> set host -dup -task params dud0
    starting PARAMS on pud0.0.0.3.0 (R2YQYA)

    Copyright (C) 1995  Digital Equipment Corporation

    PARAMS> show allclass

    Parameter   Current       Default      Type   Radix
    --------- ------------- ------------ ------- ------
    ALLCLASS              0            0    Byte    Dec  B

    PARAMS> exit
    Exiting...
P00>>>
```

### set host

The **set host** command connects the console program to another DUP server on an MSCP DSSI device. You can use the DUP protocol to examine and modify parameters of a DSSI device. The syntax is:

**set host -dup** [**-task** *task_name*] *device*

| | |
|---|---|
| **-dup** | Connect to an MSCP DUP server. The DUP service may be used to examine and modify parameters of a DSSI device. |
| **-task** *task_name* | Specifies the DUP utility to invoke. If **-task** is not specified, a directory of utilities displays. |
| *device* | The device to attach to the console program. |

## 5.4.2 Configuring a PCI NVRAM Module

**The prcache command is used for system configuration when a PCI NVRAM module is in the system. Used only with Digital UNIX systems.**

### Example 5–13 Prcache Command

```
P00>>> prcache -f
PCI NVRAM Disk Cache: passed
Size:   4MB
PCI Memory Address: 40000000
System ID: 12000000
State:  - not valid
Battery Status: good (Charging)
Battery Disconnect Circuit Status: enabled

P00>>> prcache -z
This command will zero the PCI NVRAM Disk Cache
Do you really want to continue [Y/N] ? : y
clearing disk cache
P00>>>
```

The **prcache** command is used to check PCI NVRAM configuration information and battery status, to clear data from the NVRAM module, and to set the date of the next battery replacement. This command is used only with Digital UNIX systems. The syntax is:

**prcache -{f,z,b}**

**-f** Checks configuration information and battery status.

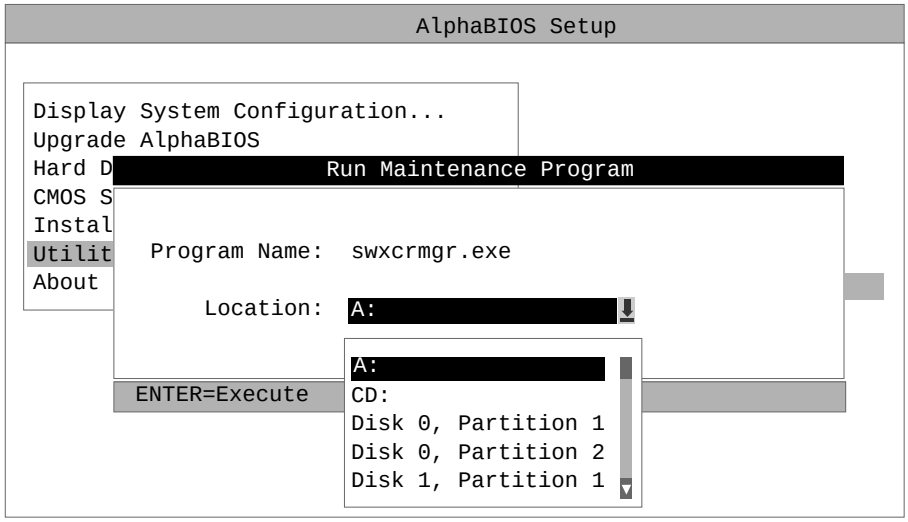
**-z** Clears valid data; writes zeros to memory.

**-b** Sets the date (month and year) for battery replacement.

## 5.4.3 Configuring EISA and RAID

**EISA and RAID devices are configured from the AlphaBIOS console.**

### Figure 5–1 Configuring EISA and RAID Devices




PK-0747-96

Configure EISA and RAID devices from the AlphaBIOS console. Figure 5–1 shows the AlphaBIOS Setup screen, from which you run the configuration utilities. See Chapter 7 for more information about running utility programs.

## 5.5 Testing the System

**The test command runs firmware diagnostics for components of the system drawer. This command is used to test Windows NT systems as well as Digital UNIX and OpenVMS systems.**

### Example 5–14 Test Command

```
P00>>> test
Console is in diagnostic mode
System test, runtime 600 seconds

Type ^C if you wish to abort testing once it has started

Configuring system..
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1     DKa500                   RRD45  1645
polling ncr1 (NCR 53C810) slot 3, bus 0 PCI, hose 1   SCSI Bus ID 7
dkb200.2.0.3.1     DKb200                   RZ29B  0007
dkb400.4.0.3.1     DKb400                   RZ29B  0007
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0    DVA0                      RX23
polling tulip0 (DECchip 21040-AA) slot 2, bus 0 PCI, hose 1
ewa0.0.0.2.1: 08-00-2B-E5-B4-1A

Testing EWA0 network device

Testing VGA (alphanumeric mode only)

Starting background memory test, affinity to all CPUs..
Starting processor/cache thrasher on each CPU..
Starting processor/cache thrasher on each CPU..
Starting processor/cache thrasher on each CPU..
Starting processor/cache thrasher on each CPU..

Testing SCSI disks (read-only)
No CD/ROM present, skipping embedded SCSI test
Testing other SCSI devices (read-only)..

Testing floppy drive (dva0, read-only)

 ID      Program   Device       Pass Hard/Soft Bytes Written  Bytes Read
------- --------- ------------ ---- --------- -------------- ----------
00003047 memtest   memory          1  0    0     134217728     134217728
00003050 memtest   memory        205  0    0     213883392     213883392
00003059 memtest   memory        192  0    0     200253568     200253568
00003062 memtest   memory        192  0    0     200253568     200253568
00003084 memtest   memory         80  0    0      82827392      82827392
000030d8 exer_kid  dkb200.2.0.3   26  0    0             0      13690880
000030d9 exer_kid  dkb400.4.0.3   26  0    0             0      13674496
0000310d exer_kid  dva0.0.0.100    0  0    0             0             0
```

*Continued on next page*

### Example 5–14 Test Command (Continued)

```
 ID      Program   Device       Pass Hard/Soft Bytes Written  Bytes Read
------- --------- ------------- ---- --------- -------------- ----------
00003047 memtest   memory          1  0    0     432013312     432013312
00003050 memtest   memory        635  0    0     664716032     664716032
00003059 memtest   memory        619  0    0     647940864     647940864
00003062 memtest   memory        620  0    0     648989312     648989312
00003084 memtest   memory        263  0    0     274693376     274693376
000030d8 exer_kid  dkb200.2.0.3   90  0    0             0      47572992
000030d9 exer_kid  dkb400.4.0.3   90  0    0             0      47523840
0000310d exer_kid  dva0.0.0.100    0  0    0             0        327680
 ID      Program   Device       Pass Hard/Soft Bytes Written  Bytes Read
------- --------- ------------- ---- --------- -------------- ----------
00003047 memtest   memory          1  0    0     727711744     727711744
00003050 memtest   memory       1054  0    0    1104015744    1104015744
00003059 memtest   memory       1039  0    0    1088289024    1088289024
00003062 memtest   memory       1041  0    0    1090385920    1090385920
00003084 memtest   memory        447  0    0     467607808     467607808
000030d8 exer_kid  dkb200.2.0.3  155  0    0             0      81488896
000030d9 exer_kid  dkb400.4.0.3  155  0    0             0      81472512
0000310d exer_kid  dva0.0.0.100    1  0    0             0        607232
^C
Testing aborted. Shutting down tests.
Please wait..

System test complete

P00>>>
```

The **test** command runs console-based exercisers for devices in the system drawer.

Testing can be aborted by Ctrl/C.

*NOTE: If you are running the Microsoft Windows NT operating system, switch from AlphaBIOS to the SRM console in order to enter the **test** command. From the AlphaBIOS console, press in the Halt button (the LED on the button lights) and reset the system, or select **Digital UNIX (SRM)** or **OpenVMS (SRM)** from the **Advanced CMOS Setup** screen and reset the system.*

The syntax is:

**test** [**-t** *time*] [**-q**] [*option*]

**-t** *time* — Specifies the run time in seconds. The default for system test is 600 seconds (10 minutes).

**-q** — Disables the display of status messages as exerciser processes are started and stopped during testing. Sets the environment variable **d_verbose** to zero.

*option* — Either **cpu***n*, **mem***n*, or **pci***n*, where *n* is 0, 1, 2, 3, or *. If nothing is specified, the entire system is tested.

## 5.6 Stopping and Starting CPUs

**The start, halt, and continue commands are used to stop and start a program on the specified CPU.**

### Example 5–15 Start, Halt, and Continue Commands

```
P00>>> start &p1

P00>>> halt 1

halted CPU 1
CPU 2 is not halted
CPU 3 is not halted

halt code = 1
operator initiated halt
PC = ffffffff8007cc68
P00>>> continue &p1

continuing CPU 1
```

### start

The **start** command starts a program that was previously loaded on the processor specified. If none is specified, the program is started on the primary processor. The syntax is:

**start** [**&p***n*] [*address*]

**&p***n* Specifies the processor. *n* is 0, 1, 2, or 3.

*address* The starting address of the program.

### halt (or stop)

The **halt** (or **stop**) command stops program execution on a secondary CPU that is still running a booted program. The syntax is:

**halt** (or **stop**) *processor_number*

*processor_number* is the logical CPU number displayed by the **show cpu** command.

### continue

The **continue** command resumes program execution on the specified processor or on the primary processor if none is specified. The processor begins executing instructions at the address that is currently in the program counter (PC). The processor is not initialized.

The **continue** command is valid only if you have not disturbed the system state and if you halted the system by pressing the Halt button on the control panel or, for OpenVMS systems only, by entering Ctrl/P on the console terminal. The syntax is:

**continue** [**&p***n*] [*address*]

**&p***n* Specifies the processor. *n* is 0, 1, 2, or 3.

*address* The starting address of the program.

*NOTE: Some console commands, for example, **boot**, can alter the machine state so that program mode cannot be successfully resumed (unless you include **-halt** in the **boot** command). If program mode cannot be resumed, reboot the operating system.*

*Other commands that alter machine state are **lfu**, **show device**, **show network**, **test**, and **start**.*

## 5.7 Updating Firmware

**The lfu command is used to update firmware from the SRM console prompt.**

### Example 5–16 Lfu Command

```
P00>>> lfu

          ***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: cda0

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS4X00FW]: AS4X00CP

Copying AS4X00CP from DKA500.5.0.1.1 .
Copying [as4x00]RHREADME from DKA500.5.0.1.1 .
Copying [as4x00]RHSRMROM from DKA500.5.0.1.1 ......................
Copying [as4x00]RHARCROM from DKA500.5.0.1.1 ............

------------------------------------------------------------------------
 Function    Description
------------------------------------------------------------------------
Display     Displays the system's configuration table.
Exit        Done exit LFU (reset).
List        Lists the device, revision, firmware name, and update
            revision.
Lfu         Restarts LFU.
Readme      Lists important release information.
Update      Replaces current firmware with loadable data image.
Verify      Compares loadable and hardware images.
? or Help   Scrolls this function table.
------------------------------------------------------------------------

UPD> list

Device         Current Revision    Filename            Update
Revision
AlphaBIOS         V5.12-2          arcrom                 V6.40-1
srmflash          V1.0-9           srmrom                 V2.0-3

UPD> update *
```

*Continued on next page*

### Example 5–16 Lfu Command (Continued)

```
WARNING: updates may take several minutes to complete for each
device.

Confirm update on: AlphaBIOS       [Y/(N)] y

                          DO NOT ABORT!
AlphaBIOS       Updating to V6.40-1...  Verifying V6.40-1...  PASSED.

Confirm update on: srmflash        [Y/(N)] y

                          DO NOT ABORT!
srmflash        Updating to V2.0-3...  Verifying V2.0-3...  PASSED.

UPD> exit
```

The **lfu** command starts the Loadable Firmware Update (LFU) Utility. The syntax is:

**lfu**

*NOTE: If the system has been shut down from a booted program (most commonly, the operating system) or in some other way halted back to the SRM console, the system must be reset before running LFU.*

See Chapter 2 for more information about LFU.

## 5.8 Forcing a System Crash Dump

**The crash command forces a crash dump to the selected device on Digital UNIX and OpenVMS systems.**

### Example 5–17 Crash Command

```
P00>>> crash

CPU 0 restarting

DUMP: 401408 blocks available for dumping.
DUMP: 38535 required for a partial dump.
DUMP: 0x805001 is the primary swap with 401407, start our
last 38534
    : of dump at 362873, going to end (real end is one
more, for header)
DUMP.prom: dev SCSI 1 3 0 4 400 0 0, block 131072
DUMP: Header to 0x805001 at 401407 (0x61fff)
DUMP.prom: dev SCSI 1 3 0 4 400 0 0, block 131072
DUMP: Dump to 0x805001: ..................: End 0x805001
DUMP.prom: dev SCSI 1 3 0 4 400 0 0, block 131072
DUMP: Header to 0x805001 at 401407 (0x61fff)
succeeded

halted CPU 0

halt code = 5
HALT instruction executed
PC = fffffc00004e2d64
P00>>>
```

The **crash** command forces a crash dump at the operating system level (Digital UNIX or OpenVMS). This command is used when an error has caused the system to hang and can be halted by the Halt button or the RCM **halt** command. This command restarts the operating system and forces a crash dump to the selected device.The syntax is:

**crash** *[device]*

*device* is the name of the device to which the crash dump is written.

## 5.9 Using Environment Variables

**The set *envar* and show *envar* commands are used to set and view environment variables. The edit command can be used to create a user-defined nonvolatile environment variable.**

### Example 5–18 Set *envar* and Show *envar* commands

```
P00>>> show console
console                  graphics
P00>>> set console serial
P00>>> show console
console                  serial
P00>>>
```

Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. Environment variables are set or changed with the **set *envar*** command and set to default values with the **set -default *envar*** command. Their values are viewed with the **show *envar*** command. User-defined nonvolatile environment variables are created with the **edit** command.

### set *envar*

The **set** command sets or modifies the value of an environment variable. It can also be used to create a new environment variable if the name used is unique. Environment variables are used to pass configuration information between the console and the operating system. The setting of these variables determines how the system powers up, boots the operating system, and operates. The syntax is:

**set** [**-default**] *envar value*

**-default** Restores an environment variable to its default setting.

*envar* The name of the environment variable to be modified.

*value* The new value of the environment variable.

Whenever you modify the value of any of the following environment variables, the new value takes effect only after you reset the system by pressing the Reset button or issuing the **initialize** command:

**console**
**kbd_hardware_type**
**language**
**ocp_text**
**os_type**

### show *envar*

The **show *envar*** command displays the current value (or setting) of an environment variable. The syntax is:

**show** *envar*

*envar* The name of the environment variable to be displayed. The wildcard * displays all environment variables.

### Example 5–19 Edit Command

```
P00>>> edit nvram
editing 'nvram'
0 bytes read in
*10  set mopv3_boot 1
*exit
17 bytes written out to nvram

P00>>>
```

The **edit** command invokes a console editor that behaves much like the line editor for BASIC. Used to add, insert, or delete lines in a RAM file or the "nvram" power-up script or to create a new environment variable.

In Example 5–19 the nvram script is edited so that an environment variable called "mop3_boot" is created and set to 1 on each power-up. By default, MOP boots send four MOP V4 requests before defaulting to MOP V3. This user-created environment variable forces the SRM console to bypass MOP V4 requests. This speeds up MOP booting on networks with MOP V3 software.

The syntax is:

**edit** *file*

*file* is the name of the file to be edited.

The editing commands are:

| | |
|---|---|
| **help** | Displays the brief help file. |
| **list** | Lists the current file prefixed with line numbers. |
| **renumber** | Renumbers the lines of the file in increments of 10. |
| **exit** | Leaves the editor and closes the file, saving all changes. |
| **quit** | Leaves the editor and closes the file without saving changes. |
| *nn* | Deletes line number *nn*. |
| *nn text* | Adds or overwrites line number *nn* with *text*. |

*NOTE: It is possible to disable the system by editing the nvram script. For example, including the command **initialize** in the script will put the system into an endless loop.*

## 5.10 Depositing and Examining Data

**The deposit command stores data in a specified location. The examine command displays the contents of a memory location, a register, or a device.**

### Example 5–20 Deposit Command

```
P00>>> dep -b -n 1ff pmem:0 0 # Clear first 512 bytes of
                              # physical memory.

P00>>> d -l -n 3 vmem:1234 5  # Deposit 5 into four
                              # longwords starting at
                              # virtual memory address
                              # 1234.

P00>>> d -n 8 r0 ffffffff     # Load GPRs R0 through R8
                              # with -1.

P00>>> d -l -n 10 -s 200 pmem:0 8
                              # Deposit 8 in the first
                              # longword of the first 17
                              # pages in physical memory.

P00>>> d -l pmem:0 0          # Deposit 0 to physical
                              # memory address 0.

P00>>> d + ff                 # Deposit FF to physical
                              # memory address 4.

P00>>> d scbb 820000          # Deposit 820000 to SCBB.
```

### Example 5–21 Examine Command

```
P00>>> examine pc             # Examine the program
                              # counter.
PC psr: 0 (    PC)  0000000000001170

P00>>> examine sp             # Examine the stack pointer.
gpr:   F0 (   R30)  0000000000072A60

P00>>> e -n 6 r4              # Examine register R4 and
                              # the next six registers.
```

*Continued on next page*

### Example 5–21 Examine Command (Continued)

```
grp: 20 (    R4) 0000000000005000
grp: 28 (    R5) 000000000FFFE000
grp: 30 (    R6) 00000003F8000C00
grp: 38 (    R7) 0000000053F761AE
grp: 40 (    R8) 0000010000000000
grp: 48 (    R9) 00000003F7800100
grp: 50 (   R10) 00000000000C7FFC

P00>>> examine pmem:400EC     # Examine physical memory.
pmem:  400EC A49D0078A47D0070
```

#### deposit

The **deposit** command stores data in the location specified. If no options are given with a **deposit** command, the system uses the options from the preceding **deposit** command.

If the specified value is too large to fit in the data size listed, the console ignores the command and issues an error response. If the data is smaller than the data size, the higher order bits are filled with zeros. The syntax is shown below.

#### examine

The **examine** command displays the contents of a memory location, a register, or a device.

If no options are given with an **examine** command, the system uses the options from the preceding **examine** command. If conflicting address space or data sizes are specified, the console ignores the command and issues an error.

For data lengths longer than a longword, each longword of data should be separated by a space.

The syntax for both commands is:

**deposit [**-{**b,w,l,q,o,h**}**] [**-{**n** *value*, **s** *value*}**] [***space***:] ***address data***

**examine [**-{**b,w,l,q,o,h**}**] [**-{**n** *value*, **s** *value*}**] [***space***:] ***address***

| | |
|---|---|
| **-b** | Defines data size as byte. |
| **-w** | Defines data size as word. |
| **-l** (default) | Defines data size as longword. |
| **-q** | Defines data size as quadword. |

*Continued on next page*

| | | |
|---|---|---|
| **-o** | Defines data size as octaword. | |
| **-h** | Defines data size as hexword. | |
| **-d** | Instruction decode (**examine** command only) | |
| **-n** *value* | The number of consecutive locations to modify. | |
| **-s** *value* | The address increment size. The default is the data size. | |
| *space***:** | Device name (or address space) of the device to access. | |
| *address* | Offset within a device to which data is deposited. Can be: | |
| | *dev_name* | A device name. |
| | **fpr-** *name* | The floating-point register set; *name* is F0 to F31. |
| | **gpr-** *name* | The general register set; *name* is R0 to R31. |
| | **ipr-** *name* | The internal processor registers. |
| | **pt-** *name* | The PALtemp register set; *name* is PT0 to PT23. |
| | **pmem** | Physical memory (default). |
| | **vmem** | Virtual memory. |
| *data* | Data to be deposited. | |

Symbolic forms can be used for the address. They are:

**pc** The program counter. The address space is set to GPR.

**+** The location immediately following the last location referenced in a **deposit** or **examine** command. For physical and virtual memory, the referenced location is the last location plus the size of the reference (1 for byte, 2 for word, 4 for longword). For other address spaces, the address is the last referenced address plus 1.

**-** The location immediately preceding the last location referenced in a **deposit** or **examine** command. Memory and other address spaces are handled as above.

***** The last location referenced in a **deposit** or **examine** command.

**@** The location addressed by the last location referenced in a **deposit** or **examine** command.

## 5.11 Reading a File

**The more command displays a file one screen at a time.**

### Example 5–22 More Command

```
P00>>> more el          #Display the contents of the
                        #SRM console's event log one
                        # screen at a time.

P00>>> help * | more    #Display the contents of online
                        #help one screen at a time.
```

The **more** command is similar to that used in the MS-DOS and UNIX operating systems. It is useful for displaying output that scrolls too quickly to be viewed. For example, when you power up the system, the system startup messages scroll, and the messages are logged to an event log. When the P00>>> prompt displays, you can use the **more** command to display the contents of the event log file. The syntax is:

**more** [*file*...]

*file* is the name of the file to be displayed.

## 5.12 Initializing the System

**The initialize command resets the system and executes the power-up tests.**

### Example 5–23 Initialize Command

```
P00>>> initialize
Initializing…

 SROM V1.0 on cpu0
 SROM V1.0 on cpu1
 SROM V1.0 on cpu2
 SROM V1.0 on cpu3
XSROM V1.0 on cpu2
XSROM V1.0 on cpu1
XSROM V1.0 on cpu3
XSROM V1.0 on cpu0
BCache testing complete on cpu2
BCache testing complete on cpu0
BCache testing complete on cpu3
BCache testing complete on cpu1
mem_pair0 - 128 MB
mem_pair1 - 128 MB
20..20..21..20..21..20..21..21..23..24..24..24..24..
Memory testing complete on cpu0
Memory testing complete on cpu1
Memory testing complete on cpu3
Memory testing complete on cpu2
starting console on CPU 0
sizing memory
  0    128 MB SYNC
  1    128 MB SYNC
starting console on CPU 1
starting console on CPU 2
starting console on CPU 3
probing IOD1 hose 1
  bus 0 slot 1 - NCR 53C810
  bus 0 slot 2 - DEC KZPSA
  bus 0 slot 3 - NCR 53C810
  bus 0 slot 4 - DECchip 21040-AA
  bus 0 slot 5 - DEC PCI MC
probing IOD0 hose 0
  bus 0 slot 1 - PCEB
  bus 0 slot 4 - DEC PCI FDDI
  bus 0 slot 5 - PBXGA-AA
```

*Continued on next page*

### Example 5–23 Initialize Command (Continued)

```
configuring I/O adapters...
  ncr0, hose 1, slot 1, bus 0
  kzpsa0, hose 1, slot 2, bus 0
  ncr1, hose 1, slot 3, bus 0
  tulip0, hose 1, slot 4, bus 0
  floppy0, hose 0, slot 0, bus 1
  pfi0, hose 0, slot 4, bus 0

System temperature is 24 degrees C
AlphaServer 4100 Console V2.0-1, 24-JUL-1996 18:18:26
P00>>>
```

The **initialize** command resets the system. Issuing this command is equivalent to pressing the Reset button. The syntax is:

**initialize**

After self-tests are executed, the system autoboots unless one of the following is true:

- The Halt button on the control panel is pushed in.
- The **auto_action** environment variable is set to **halt**.

If the **auto_action** environment variable is set to **boot** or **restart** and the Halt button is not pushed in, the system autoboots. In all other cases, the system stops in console mode and does not attempt to boot.

## 5.13 Finding Help

**The help (or man) command displays basic information about a command.**

### Example 5–24 Help (or Man) Command

```
P00>>> help set         # or man set
NAME
      set
FUNCTION
      Set an option or modify the value of an environment
      variable.
SYNOPSIS
      set <option> <value> or <envar> [-] <value>
            where
            <option>={host,mode}
            where
            <envar>={auto_action,bootdef_dev,boot_osflags,...}
            [-default]
```

The **help** (or **man**) command displays basic information about the use of console commands when the system is in console mode. The syntax is:

**help** (or **man**) [*command* **. . .** ]

*command . . .* Command or topic for which help is requested. The options are:

| | |
|---|---|
| *none* | Displays the complete list of commands for which you can receive help. |
| *command_name* | Displays information about the console command. |
| *argument_string* (such as “sh”) | Displays information about all commands that begin with that string. |

## 5.14 Command Summary

**The SRM (*Alpha System Reference Manual*) console is a command line interface used when the operating system is either Digital UNIX or OpenVMS. Console commands enable the user to examine and modify the system state.**

The commands listed in Table 5–2 are described in the previous sections. Table 5–3 lists command syntax, and Table 5–4 shows special characters.

### Table 5–2 Summary of SRM Console Commands

| Command | Function |
|---|---|
| **alphabios** | Loads and starts the AlphaBIOS console. |
| **boot** | Loads and starts the operating system. |
| **clear *envar*** | Resets an environment variable to its default value. |
| **continue** | Resumes program execution. |
| **crash** | Forces a crash dump at the operating system level. |
| **deposit** | Writes data to the specified address. |
| **edit** | Invokes the console line editor on a RAM file or on the nvram file (power-up script). |
| **examine** | Displays the contents of a memory location, register, or device. |
| **halt** | Halts the specified processor. (Same as **stop**.) |
| **help** | Displays information about the specified console command. |
| **initialize** | Resets the system. |
| **lfu** | Runs the Loadable Firmware Update Utility. |
| **man** | Displays information about the specified console command. |
| **more** | Displays a file one screen at a time. |
| **prcache** | Utility that initializes and displays status of the PCI NVRAM. |

*Continued on next page*

**Table 5–2 Summary of SRM Console Commands (Continued)**

| Command | Function |
|---|---|
| **set *envar*** | Sets or modifies the value of an environment variable. |
| **set host** | Connects to a MSCP DUP server on a DSSI device. |
| **show *envar*** | Displays the state of the specified environment variable. |
| **show config** | Displays the configuration at the last system initialization. |
| **show cpu** | Displays the state of each processor in the system. |
| **show device** | Displays a list of controllers and their devices in the system. |
| **show fru** | Displays the serial number and revision level of system bus options. |
| **show memory** | Displays memory module information. |
| **show network** | Displays the state of network devices in the system. |
| **show pal** | Displays the version of the privileged architecture library code (PALcode). |
| **show power** | Displays information about the power supplies, system fans, CPU fans, and temperature. |
| **show version** | Displays the version of the console program. |
| **start** | Starts a program that was previously loaded on the processor specified |
| **stop** | Halts the specified processor. (Same as **halt**.) |
| **test** | Runs firmware diagnostics for the system. |

**Table 5–3 Syntax for SRM Console Commands**

| Parameter | Attribute or Action |
|---|---|
| Length | Up to 255 characters, not including the terminating carriage return or any characters deleted as the command is entered. A command longer than 80 characters and without the backslash character (see Table 5–4) causes display of an error message. |
| Case | Upper- or lowercase characters can be used for input. Characters are displayed in the case in which they are entered. |
| Abbreviation | Only by dropping characters from the end of words. You must enter the minimum number of characters to identify the keyword unambiguously. Abbreviation of environment variables is allowed with the **show** command. |
| Options | You can use command options, to modify the environment, after the command keyword or after any symbol or number in the command. See individual command descriptions for examples. |
| Numbers | Most numbers in console commands are in decimal notation. Two exceptions, both of which use hexadecimal notation, are addresses and numbers used in the **deposit** command. The default radix can be overridden by inserting %d before numbers you want to express in decimal, %b before binary, %o before octal, or %x before hexadecimal. Register names (for example, R0) are not considered numbers and use decimal notation. |
| No characters | A command line with no characters is a null command. The console program takes no action and does not issue an error message; it returns the console prompt. The console supports command line recall and editing. |
| Spaces or tabs | Multiple adjacent spaces and tabs are compressed and treated as a single space. The console program ignores leading and trailing spaces. |

**Table 5–4 Special Characters for SRM Console**

| Character | Function |
|---|---|
| Return or Enter | Terminates a command line. No action is taken on a command until it is terminated. If no characters are entered and this key is pressed, the console just redisplays the prompt. |
| Backslash (\) | Continues a command on the next line. Must be the last character on the line to be continued. |
| Delete | Deletes the previous character. |
| Help | By itself, displays first-level help. When pressed after part of a command, displays options available. |
| Ctrl/A or F14 | Toggles between insert and overstrike modes. The default is overstrike. |
| Ctrl/B or up-arrow | Recalls previous command or commands. The last 16 commands are stored in the recall buffer. |
| Ctrl/C or Ctrl/P | Terminates the process that is running. Clears Ctrl/S; resumes output suspended by Ctrl/O. When entered as part of a command line, deletes the current line. Ctrl/C has no effect as part of a binary data stream. |
| Ctrl/D or left-arrow | Moves the cursor left one position. |
| Ctrl/E | Moves the cursor to the end of the line. |
| Ctrl/F or right-arrow | Moves the cursor right one position. |
| Ctrl/H or Backspace or F12 | Moves the cursor to the beginning of the line. |
| Ctrl/J | Deletes the previous word. |
| Ctrl/O | Stops output to the console terminal for the current command. Toggles between enable and disable. The output can be reenabled by other means as well: when the console prompts for a command, issues an error message, or enters program mode, or when Ctrl/P is entered. |
| Ctrl/Q | Resumes output to the console terminal that was suspended by Ctrl/S. |

**Table 5–4 Special Characters for SRM Console (Continued)**

| Character | Function |
|---|---|
| Ctrl/R | Redisplays the current line. Deleted characters are omitted. This command is useful for hardcopy terminals. |
| Ctrl/S | Suspends output to the console terminal until Ctrl/Q is entered. Cleared by Ctrl/C. |
| Ctrl/U | Deletes the current line. |
| * | Wildcarding for commands such as **show**. |
| " " | Double quotes enable you to denote a string for environment variable assignment. |
| # | Specifies that all text between it and the end of the line is a comment. Control characters are not considered part of a comment. |

## 5.15 Environment Variable Summary

**Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. Environment variables are set or changed with the set *envar* command and returned to their default values with the clear *envar* command. Their values are viewed with the show *envar* command.**

### Table 5–5 Environment Variable Summary

| Environment Variable | Function |
|---|---|
| **auto_action** | Specifies the console's action at power-up, a failure, or a reset. |
| **boot_osflags** | Specifies the default operating system boot flags. |
| **bootdef_dev** | Specifies the default boot device string. |
| **com*_baud** | Changes the default baud rate of the COM 1 or the COM2 serial port. |
| **console** | Specifies the device on which power-up output is displayed (serial terminal or graphics monitor). |
| **cpu_enabled** | Enables or disables a specific secondary CPU. |
| **ew*0_mode** | Specifies the connection type of the default Ethernet controller. |
| **ew*0_protocols** | Specifies network protocols for booting over the Ethernet controller. |
| **kbd_hardware_type** | Specifies the default console keyboard type. |
| **kzpsa*_host_id** | Specifies the default value for the KZPSA host SCSI bus node ID. |
| **language** | Specifies the console keyboard layout. |
| **memory_test** | Specifies the extent to which memory will be tested. For Digital UNIX systems only. |

*Continued on next page*

**Table 5–5 Environment Variable Summary (Continued)**

| Environment Variable | Function |
|---|---|
| **ocp_text** | Overrides the default OCP display text with specified text. |
| **os_type** | Specifies the operating system and sets the appropriate console interface. |
| **pci_parity** | Disables or enables parity checking on the PCI bus. |
| **pk*0_fast** | Enables fast SCSI mode. |
| **pk*0_host_id** | Specifies the default value for a controller host bus node ID. |
| **pk*0_soft_term** | Enables or disables SCSI terminators on systems that use the QLogic ISP1020 SCSI controller. |
| **tga_sync_green** | Specifies the location of the SYNC signal generated by the Digital ZLXp-E PCI graphics accelerator option. |
| **tt_allow_login** | Enables or disables login to the SRM console firmware on other console ports. |

These environment variables are described in the following pages.

*NOTE: These environment variables are specific to the SRM console.*

### auto_action

Specifies the action the console takes any time the system powers up, fails, or resets. When the setting involves autobooting, the system boots from the default boot device specified by the value of the bootdef_dev environment variable. The syntax is:

**set auto_action** *value*

The options for *value* are:

| | |
|---|---|
| **halt** | The system remains in console mode after power-up or a system crash. |
| **boot** | The system boots automatically when it is turned on and halts after a system failure. |
| **restart** | The system boots automatically when it is turned on or after it fails. |

### bootdef_dev

The **bootdef_dev** environment variable specifies one or more devices for booting the operating system. When more than one device is listed, the system searches in the order listed and boots from the first device with operating system software. The syntax is:

**set bootdef_dev** *boot_device*

| | |
|---|---|
| *boot_device* | The name of the device on which the system software has been loaded. To specify more than one device, separate the names with commas. Enter the command **show bootdef_dev** to display the current default boot device. Enter the command **show device** for a list of all devices in the system. |

### boot_osflags

The **boot_osflags** environment variable passes information to the **boot** command. That information is dependent on the operating system to be booted.

When the operating system is Digital UNIX, the syntax is:

**set boot_osflags** *flags_value*

The options for *flags_value* are:

**a** Load operating system software from the specified boot device (autoboot). Boot to multiuser mode.

**i** Prompt for the name of a file to load and other options (boot interactively). Boot to single-user mode.

**s** Stop in single-user mode. Boots /vmunix to single-user mode and stops at the # (root) prompt.

**D** Full dump; implies "**s**" as well. By default, if Digital UNIX crashes, it completes a partial memory dump. Specifying "**D**" forces a full dump at system crash.

When the operating system is OpenVMS, the syntax is:

**set boot_osflags** *root_number,boot_flags*

*root_number* Directory number of the system disk on which OpenVMS files are located. For example:

| *root_number* | Root Directory |
|---|---|
| 0 (default) | [SYS0.SYSEXE] |
| 1 | [SYS1.SYSEXE] |
| 2 | [SYS2.SYSEXE] |
| 3 | [SYS3.SYSEXE] |

*boot_flags* The hexadecimal value of the bit number or numbers set. To specify multiple boot flags, add the flag values (logical OR). See Table 5–6.

**Table 5–6 Settings for boot_osflags Bootflags (OpenVMS)**

| Flags_Value | Bit Number | Meaning |
|---|---|---|
| 1 | 0 | Bootstrap conversationally (enables you to modify SYSGEN parameters in SYSBOOT). |
| 2 | 1 | Map XDELTA to running system. |
| 4 | 2 | Stop at initial system breakpoint. |
| 8 | 3 | Perform diagnostic bootstrap. |
| 10 | 4 | Stop at the bootstrap breakpoints. |
| 20 | 5 | Omit header from secondary bootstrap image. |
| 80 | 7 | Prompt for the name of the secondary bootstrap file. |
| 100 | 8 | Halt before secondary bootstrap. |
| 10000 | 16 | Display debug messages during booting. |
| 20000 | 17 | Display user messages during booting. |

### com*_baud

The default baud rate for the AlphaServer 4100 system is 9600. With the **com*_baud** environment variable, you can set the baud rate to match that of the device connected to the port.

Resetting the baud rate of the COM1 port bypasses the RCM (see Section 2.10). You will be asked to confirm the change, as shown here:

```
P00>>> set com1_baud 19200
Embedded Remote Console only supports 9600 baud. Continue?
(Y/[N]) n
bad value - com1_baud not modified
P00>>>
```

The syntax is:

**set com*_baud** *baud_value*

*baud_value* The new baud rate. A list of possible values is displayed by attempting to set this environment variable to an unacceptable value (for example, **set com2_baud xxx**).

### console

The console terminal can be either a graphics monitor or a serial terminal. The **console** environment variable specifies which is used. The syntax is:

**set console** *output_device*

The options for *output_device* are:

| | |
|---|---|
| **graphics** (default) | The console terminal is a graphics monitor or a device connected to the VGA or TGA module. |
| **serial** | The console terminal is the device connected to the COM1 port. |

Whenever you change the value of **console**, you must reset the system by pressing the Reset button or issuing the **initialize** command.

### cpu_enabled

Sets a bit mask that enables or disables specific CPUs on a multiprocessor system.

Disabled CPUs are prevented from running the console or the operating system. Bit 0 of the mask corresponds to CPU 0, bit 1 to CPU 1, and so on. A zero in the bit mask prevents the corresponding CPU from running; a one allows it to run. The bit mask is expressed as a hexadecimal value.

*NOTE: The primary CPU cannot be disabled.*

The syntax is:

**set cpu_enabled** *hex_digit*

The options for *hex_digit* are 0 – F (hexadecimal).

### ew*0_mode

Sets an Ethernet controller to run an AUI, ThinWire, or twisted-pair Ethernet network. The default is **auto-sense**. For the Fast setting, the device defaults to Fast.

The syntax is:

**set ew*0_mode** *value*

The options for *value* are:

| | |
|---|---|
| **aui** | Device type is AUI. |
| **auto-sense** | Device type is sensed by the console. |
| **bnc** | Device type is ThinWire. |
| **fast** | Device type is fast 100BaseT. |
| **fastfd** | Device type is fast full duplex 100BaseT. |
| **full** | Device type is full duplex twisted pair. |
| **twisted-pair** | Device type is 10BaseT (twisted pair). |

### ew*0_protocols

Enables network prococols for booting and other functions. The syntax is:

**set ew*0_protocols** *protocol_value*

The options for *protocol_value* are:

| | |
|---|---|
| **mop** (default) | Sets the network protocol to mop (Maintenance Operations Protocol), the setting typically used with the OpenVMS operating system. |
| **bootp** | Sets the network protocol to bootp, the setting typically used with the Digital UNIX operating system. |
| **bootp,mop** | When both are listed, the system attempts to use the mop protocol first, regardless of which is listed first. If not successful, it then attempts the bootp protocol. |

### kbd_hardware_type

Used only on systems with the language variant 3C (Français), this environment variable sets the keyboard hardware type as either PCXAL or LK411 and enables the system to interpret the terminal keyboard layout correctly.

Whenever you change the value of **kbd_hardware_type**, you must reset the system by pressing the Reset button or issuing the **initialize** command.

The syntax is:

**set kbd_hardware_type** *keyboard_type*

The options for *keyboard_type* are:

| | |
|---|---|
| **pcxal** (default) | Selects the default keyboard hardware type. |
| **lk411** | Selects the LK411 keyboard layout for use with language variant 3C (Français). |

### kzpsa*_host_id

See **pk*0_host_id.**

### language

Specifies the keyboard layout, which is language dependent. The setting of the **language** environment variable must match the language of the keyboard variant.

Whenever you change the value of **language**, you must reset the system by pressing the Reset button or issuing the **initialize** command.

The syntax is:

**set language** *language_code*

The options for *language_code* are:

| | | | |
|---|---|---|---|
| **0** | No language (cryptic) | **40** | Français (Suisse Romande) |
| **30** | Dansk (Danish) | **42** | Italiano (Italian) |
| **32** | Deutsch (German) | **44** | Nederlands (Netherlands) |
| **34** | Deutsch (Schweiz) (Swiss) | **46** | Norsk (Norwegian) |
| **36** | English (American) | **48** | Portuguese (Portuguese) |
| **38** | English (British/Irish) | **4A** | Suomi (Finnish) |
| **3A** | Español (Spanish) | **4C** | Svenska (Swedish) |
| **3C** | Français (French) | **4E** | Belgisch-Nederlands (Dutch) |
| **3E** | Français (Canadian) | | |

### memory_test

Determines the extent of memory testing on the next reset. The syntax is:

**set memory_test** *value*

The options for *value* are:

| | |
|---|---|
| **full** (default) | Specifies that the full memory test will be run. Systems using the OpenVMS operating system must run the full memory test. |
| **partial** | Specifies that the first 256 MB of memory will be tested. |
| **none** | Specifies that memory will not be tested. (However, tests are always run on the first 32 MB.) |

### ocp_text

Specifies a message to display on the control panel after self-tests and diagnostics have completed.

The value of **ocp_text** takes effect only after you reset the system by pressing the Reset button or issuing the **initialize** command.

The syntax is:

**set ocp_text** *message*

The variable *message* can be up to 16 characters and must be enclosed in quotation marks.

### os_type

The **os_type** environment variable specifies the default operating system. This variable is set at the factory to the setting for the operating system purchased. Use this command to change the factory default setting.

Whenever you change the value of **os_type**, you must reset the system by pressing the Reset button or issuing the **initialize** command.

The syntax is:

**set os_type** *os_type*

The options for *os_type* are:

| | |
|---|---|
| **unix** | Digital UNIX is the default operating system, and the SRM firmware is started during power-up or reset. |
| **openvms** | OpenVMS is the default operating system, and the SRM firmware is started during power-up or reset. |
| **nt** | Windows NT is the default operating system. The SRM firmware is started, and it loads and starts the AlphaBIOS firmware during power-up or reset. (If the Halt button is pushed in so that the LED on the button lights, this environment variable is overridden, and the SRM firmware does not load and start the AlphaBIOS console.) |

### pci_parity

Disables or enables parity checking on the PCI bus.

Some PCI devices do not implement PCI parity checking, and some have a parity-generating scheme in which the parity is sometimes incorrect or is not fully compliant with the PCI specification. A side effect of this aberrant behavior is that superfluous PCI parity errors are reported by the host PCI bridge. In such cases, the device can be used as long as parity is not checked; disabling PCI parity checking prevents false parity errors that can cause system problems.

The syntax is:

**set pci_parity** *value*

The options for *value* are:

| | |
|---|---|
| **on** (default) | Enables PCI parity checking. |
| **off** | Disables PCI parity checking. |

### pk*0_fast

Enables Fast SCSI to perform in either standard or fast mode. If the system has at least one Fast SCSI device, set the default controller speed to Fast SCSI (1). Devices on a controller that connects to both standard and Fast SCSI devices will perform at the appropriate rate for the device. If the system has no Fast SCSI devices, set the default controller speed to standard SCSI (0). If a Fast SCSI device is on a controller set to standard, it will perform in standard mode.

The syntax is:

**set pk*0_fast** *scsi_speed*

The options for *scsi_speed* are:

| | |
|---|---|
| **0** | The controller is in standard SCSI mode. |
| **1** (default) | The controller is in Fast SCSI mode. |

### pk*0_host_id

Sets the controller host bus node ID to a value between 0 and 7.

Each SCSI bus in the system requires a controller. Buses can theoretically support up to eight devices; however, the eighth device must always be a controller. Each device on the bus, including the controller, must have a unique ID, which is a number between 0 and 7. This is the bus node ID number.

On each bus, the default bus node ID for the controller is set to 7. You do not need to change the controller bus node ID unless you place two or more controllers on the same bus.

To list the controllers on your system, enter the command **show device**. SCSI devices begin with the letters "pk" (for example, pka0). The third letter is the adapter ID for the controller. When entering the command **set pk*0_host_id**, replace the asterisk with the adapter ID letter.

The syntax is:

**set pk*_host_id** *scsi_node_id*

The value for scsi_node_id is the bus node ID, a number from 0 to 7.

### pk*0_soft_term

Enables or disables SCSI terminators. This command applies to systems that use the QLogic ISP1020 SCSI controller.

The QLogic ISP1020 SCSI controller implements the 16-bit wide SCSI bus. The QLogic module has two terminators, one for the low eight bits and one for the high eight bits.

The syntax is:

**set pk*0_soft_term** *value*

The options for *value* are:

| | |
|---|---|
| **off** | Disables termination of all 16 bits. |
| **low** (default) | Enables low eight bits and disables high eight bits. |
| **high** | Enables high eight bits and disables low eight bits. |
| **on** | Enables all 16 bits. |
| **diff** | Places the bus in differential mode. |

### tga_sync_green

Sets the location of the SYNC signal generated by the optional ZLXp-E PCI graphics accelerator (PBXGA). The parameter is a bit mask, where the least significant bit sets the vertical SYNC for the first graphics card found, the second for the second found, and so on.

The default for this environment variable is **ff**. The default must be reset to 00 so the graphics monitor on a system with a ZLXp-E PCI graphics accelerator will synchronize.

The syntax is:

**set tga_sync_green** *value*

The options for *value* are:

| | |
|---|---|
| **ff** (default) | Synchronizes the graphics monitor on systems that do not use a ZLXp-E PCI graphics accelerator. |
| **00** | Synchronizes the graphics monitor on systems with a ZLXp-E PCI graphics accelerator. |

### tt_allow_login

Enables or disables login to the SRM console firmware on alternate console ports. If the environment variable **console** is set to serial, the primary console device is the terminal connected through the COM1 port. The command **set tt_allow_login 1** enables logins through either the COM2 port or a graphics monitor.

The syntax is:

**set tt_allow_login** *value*

The options for *value* are:

| | |
|---|---|
| **0** | Disables login through the COM2 port or a graphics monitor. |
| **1** (default) | Enables login through the COM2 port or a graphics monitor. |

# Chapter 6

# AlphaBIOS Console

AlphaBIOS is the graphical interface that supports the Microsoft Windows NT operating system and some utility programs. This chapter explains how to perform common system management tasks with AlphaBIOS, and it provides a reference for the AlphaBIOS screens.

Sections in this chapter are:

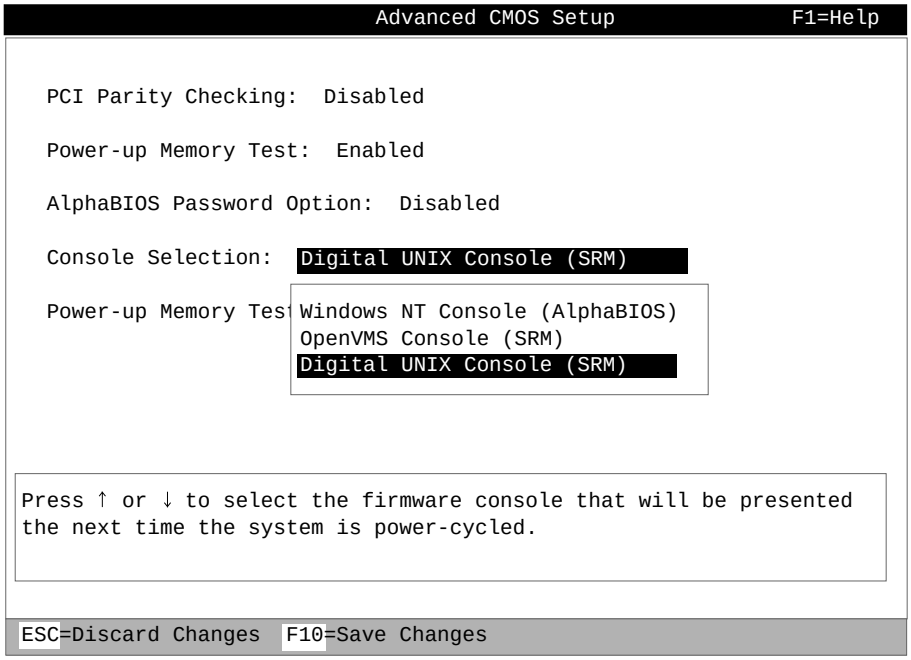
- Switching from AlphaBIOS to the SRM Console
- Starting AlphaBIOS
- Keyboard Conventions and Help
- Displaying the System Configuration
- Updating Firmware
- Setting up the Hard Disk
- Performing Setup Tasks
- Installing Windows NT
- Running Utility Programs
- Selecting the Version of Windows NT

*NOTE: To run firmware tests on a Windows NT system, switch to the SRM console and issue the **test** command.*

## 6.1 Switching from AlphaBIOS to the SRM Console

**It is necessary to switch to the SRM console to boot Digital UNIX or OpenVMS or to run firmware-based diagnostics. To switch from AlphaBIOS to the SRM console, press the Halt button on the control panel (the LED lights) and reset the system, or select Digital UNIX (SRM) or OpenVMS (SRM) from the Advanced CMOS Setup screen and reset the system.**

### Figure 6–1 Switching to the SRM Console




PK-0750B-96

Digital UNIX and OpenVMS are booted and firmware-based diagnostics are run from the SRM console. Follow this procedure to switch from AlphaBIOS to SRM:

1. Select **CMOS Setup** and press Enter.
2. In the **CMOS Setup** screen press F6. The **Advanced CMOS Setup** screen displays.
3. Select **Digital UNIX (SRM)** or **OpenVMS (SRM)** and press F10.
4. The **CMOS Setup** screen displays. Press F10 to save the change.
5. Run the EISA Configuration Utility, using the ECU diskette for Digital UNIX and OpenVMS. See Chapter 7.

   *NOTE: ECU must be run before booting Digital UNIX or OpenVMS.*
6. Reset the system.

Alternatively, you can press the Halt button (the LED on the button lights) and reset the system.

The method you choose depends on how you want the system to function on subsequent resets:

- If you make the selection in the **Advanced CMOS Setup** screen, the system will continue to stop in the SRM console whenever it is powered-up, reset, or initialized until the Windows NT operating system is again specified.
- If you press the Halt button and reset, no system setup information is changed, and AlphaBIOS will start the next time the system is powered-up, reset, or initialized with the Halt button out (the LED is off).

To return to the AlphaBIOS console, issue the command **alphabios** at the SRM console prompt.

## 6.2 Starting AlphaBIOS

**Start AlphaBIOS Setup by pressing F2 from the Boot screen displayed at power-up or reset.**

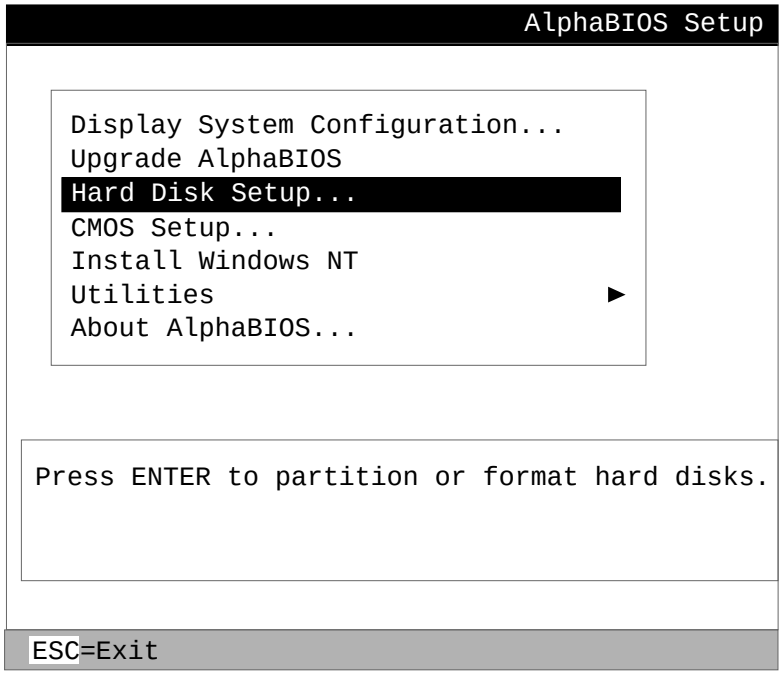
### Figure 6–2 Boot Screen

PK-0728-96

The **Boot** screen shown in Figure 6–2 displays at power-up and reset. Press F2 at this screen to enter the setup program.

The **AlphaBIOS Setup** screen (Figure 6–3) displays. From this screen you can select the tasks to perform. Use the arrow keys to select the menu item you want and press Enter. (Refer to Section 6.3 for information on navigating through the AlphaBIOS screens.)

### Figure 6–3 AlphaBIOS Setup Screen

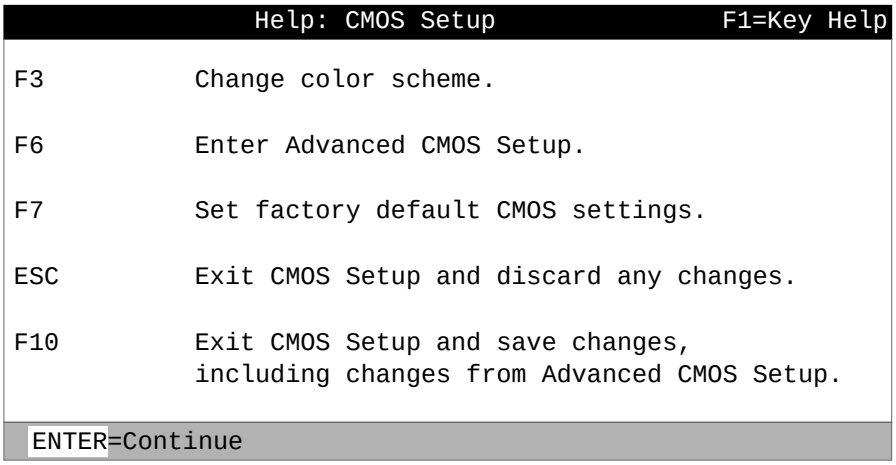



PK-0726-96

## 6.3 Keyboard Conventions and Help

**AlphaBIOS uses universally accepted keys and key combinations for navigating the interface and selecting items.**

### Figure 6–4 Typical First-Level Help Screen



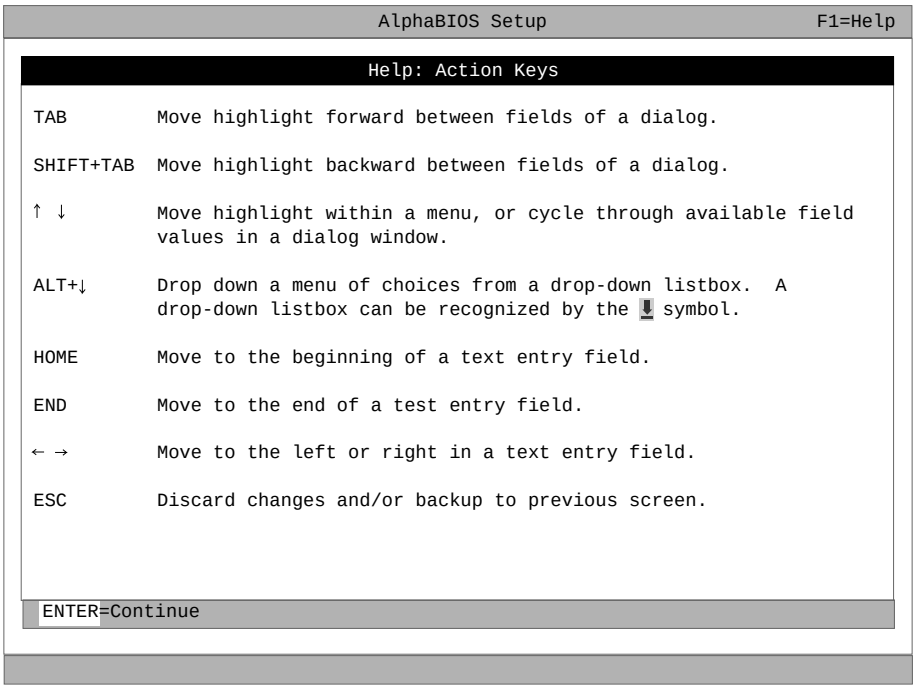

PK-0724-96

AlphaBIOS uses DOS and Windows keyboard conventions for navigating the interface and selecting items. The valid keystrokes are listed in the keyboard help screens.

Two levels of keyboard help are available. The first level, reached by pressing F1 once, shows explanations of the keystrokes available for the specific part of AlphaBIOS currently displayed. An example of the help displayed by pressing F1 once, in this case from the **CMOS Setup** screen, is shown in Figure 6–4.

The second level of keyboard help, reached by pressing F1 from the first help screen, shows explanations of the keystrokes available for navigating the interface throughout AlphaBIOS (see Figure 6–5).

## Figure 6–5 Second-Level Help Screen

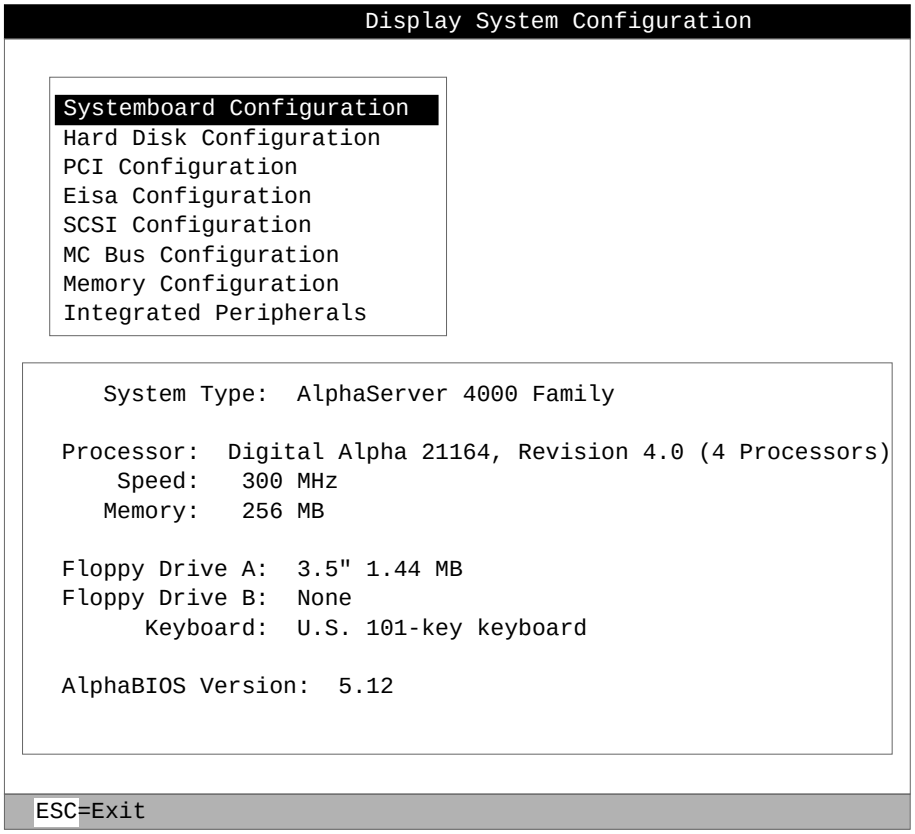



PK-0725-96

## 6.4 Displaying the System Configuration

**The Display System Configuration screen provides information about the system's installed processor, memory, attached devices, and option boards. From the AlphaBIOS Setup screen select Display System Configuration..., then the category for the requisite information.**

### Figure 6–6 Display System Configuration Screen




PK-0731-96

Use this procedure to display the system configuration.

1. Start AlphaBIOS, select **Display System Configuration**, and press Enter.
2. In the **Display System Configuration** screen, use the arrow keys to select the configuration category you want to see.

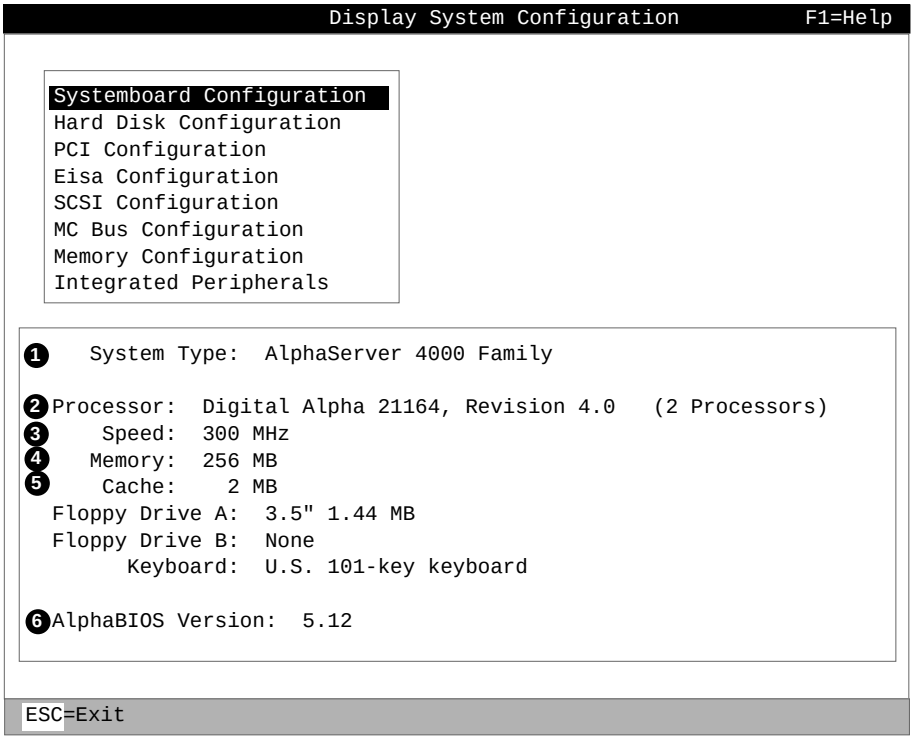
From this screen, you can view configuration information about these system components:

- System motherboard
- Hard disk
- PCI bus
- EISA bus
- SCSI devices
- System bus
- Memory
- Integrated peripherals

The sections that follow explain the display for each component.

### 6.4.1 System Board Configuration

**Figure 6–7 System Board Configuration**




PK-0738A-96

❶ System type — The model number of the system.

❷ Processor — The model and revision of the processor chip. Revision level information can be useful in troubleshooting problems with technical support personnel.

❸ Speed — The speed at which the processor runs.

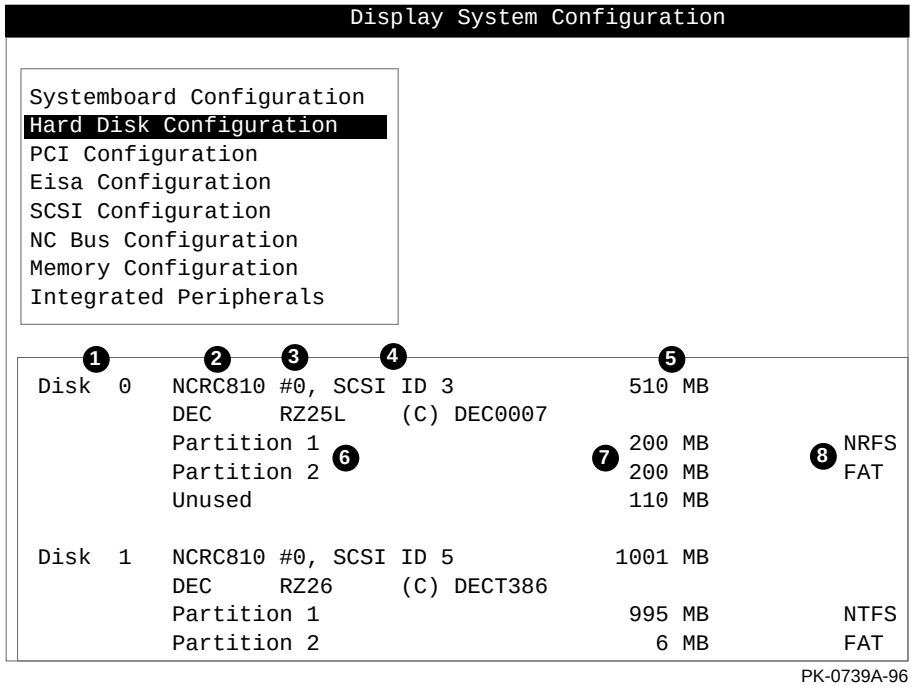
❹ Memory — The amount, in megabytes, of main memory in the system.

❺ Cache — The amount, in kilobytes, of static RAM cache memory installed.

❻ AlphaBIOS version — The version of AlphaBIOS currently running on the system. This field has three parts: major version number is the major revision of the BIOS; minor version number signifies enhancements that have been made since the last major revision; build date shows when the BIOS was compiled.

## 6.4.2 Hard Disk Configuration

### Figure 6–8 Hard Disk Configuration




PK-0739A-96

*NOTE: This screen is for information only; it cannot be edited. To make changes to the hard disk setup, use the **Hard Disk Setup** screen (Section 6.6).*

❶ Physical disk ID — Based on the SCSI ID. The disk with the lowest SCSI ID is disk 0, the disk with the next lowest SCSI ID is disk 1, and so on.

❷ Controller — The brand and model of SCSI chip used on the SCSI controller.

❸ Controller number — Based on how many SCSI controllers of a particular type are installed in the system. The first controller of a type is always numbered 0.

❹ SCSI ID number — A unique number you assign to each SCSI device installed in the system. This is usually done with jumpers or a thumb wheel attached to the drive housing.

❺ Size — The raw capacity of the drive. Formatting the drive with different file systems (for example, FAT and NTFS) may result in different usable sizes because of the differences in how storage is managed under those file systems.

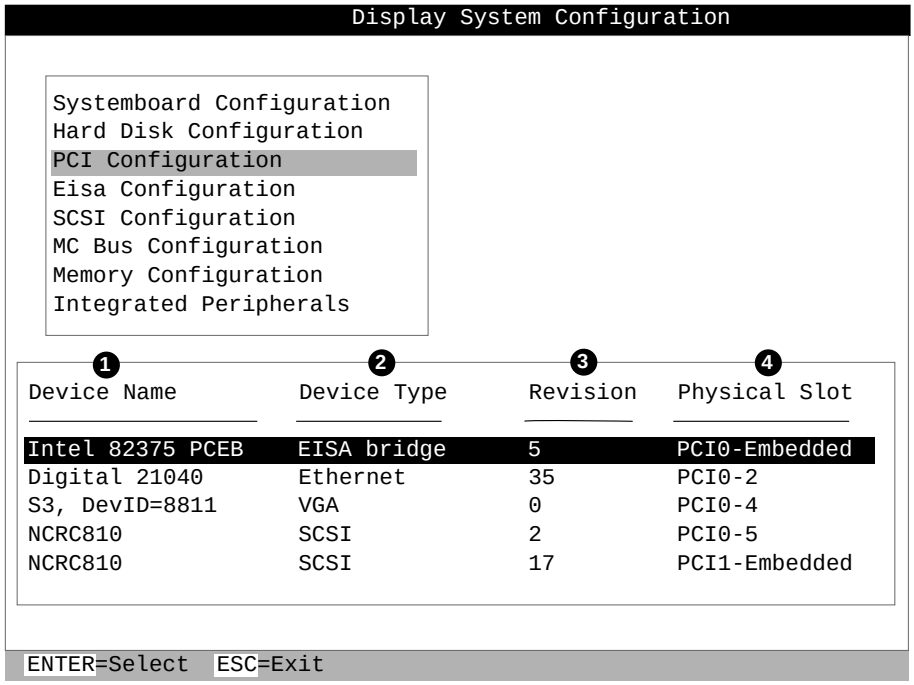
❻ Partition number — Within a single drive, partition numbers are assigned in sequential order: 1, 2, 3, and so on. The partitions populate the drive from the innermost cylinders to the outermost cylinders. If you have a large hard disk (over 800 MB) and plan to use the FAT file system, it is a good idea to break the disk into several smaller partitions because the FAT file system uses disk space more efficiently at smaller partition sizes. This is not a concern for the NTFS file system, however, as it uses disk space very efficiently at all partition sizes.

❼ Partition size — The raw (unformatted) storage capacity of the partition. Actual storage space will differ based on the file system with which the partition is formatted.

❽ Partition format — The file system (if any) used on a partition. This field displays FAT, NTFS, or unknown (if the partition is unformatted).

### 6.4.3 PCI Configuration

**Figure 6–9 PCI Configuration**




PK-0740A-96

❶ Device name — The name and model of the device as recorded in the device's firmware.

❷ Device type — Lists the function of the device in the system.

❸ Physical slot — Lists the actual PCI slot number to which the device is attached. Refer to your system manual for information on the physical location of PCI slot locations on your particular system board.

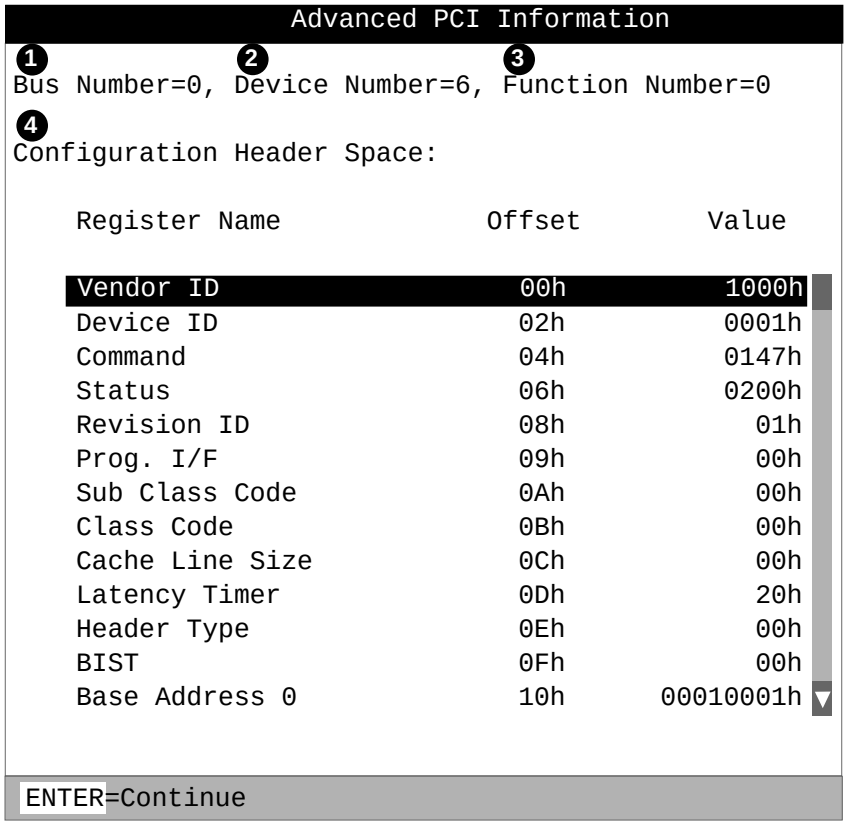
❹ Revision — The revision level of the device signifies the number of times it has been updated by the manufacturer.

You can find additional detail about any of the PCI devices listed in the **Advanced PCI Information** screen. Follow this procedure to view this screen:

1. Press Enter to enable selection in the device list.
2. Use the arrow keys to select the device for which you want additional detail.
3. Press Enter, and the detail is displayed.
4. Press Escape to return to the table of PCI devices.

The **Advanced PCI Information** screen is shown in Figure 6–10 on the next page.

## Figure 6–10 Advanced PCI Information




PK-0741A-96

❶ Bus number — The virtual PCI bus number.

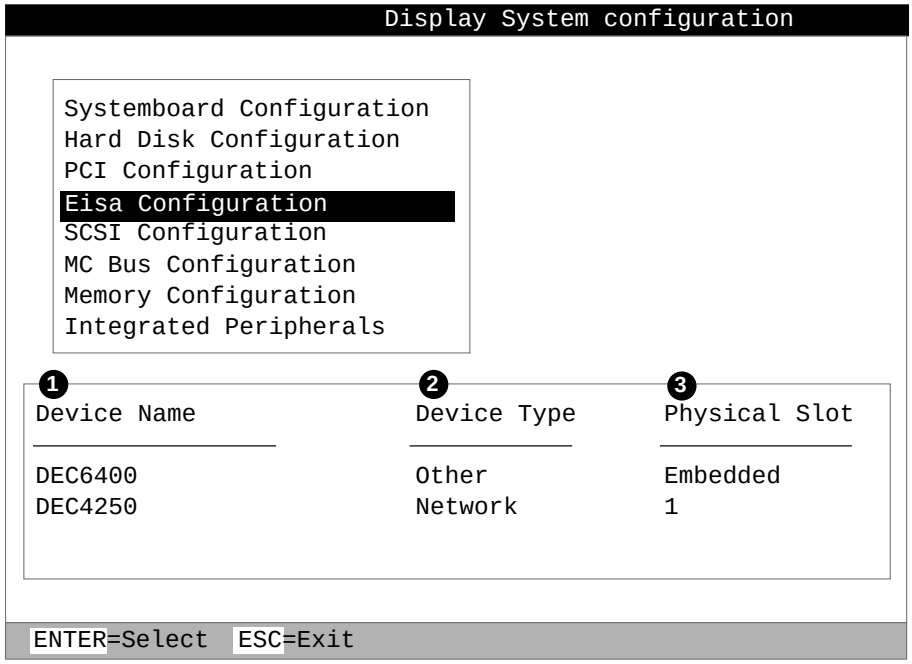
❷ Device number — The PCI bus device number.

❸ Function number — Represents the number assigned to a particular function on a multifunction device. For example, a combination Ethernet/SCSI controller would be listed twice, with the first function listed as 0 and the other as 1.

❹ Configuration header space— Displays the information in the selected device's PCI configuration space.

### 6.4.4 EISA Configuration

**Figure 6–11 EISA Configuration**




PK-0742A-96

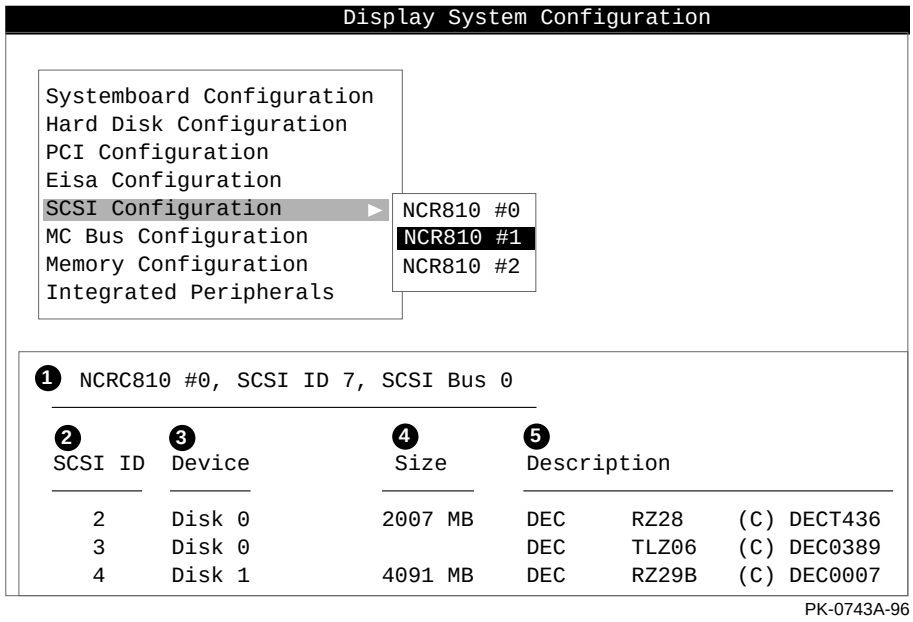
❶ Device name — Includes a three-character manufacturer code, followed by a three-digit board type, followed by a one-digit revision number.

❷ Device type — Identifies the board type (for example, network).

❸ Physical slot — Lists the actual EISA slot number to which the device is attached. The embedded EISA system controller is always first on this list.

## 6.4.5 SCSI Configuration

### Figure 6–12 SCSI Configuration




PK-0743A-96

❶ SCSI controller information — Describes the physical characteristi cs of the selected SCSI controller. This line includes:

Controller — Brand and model of SCSI chip used on the SCSI controller.

Controller number — Based on the number of SCSI controllers of a particular type in the system. The first controller of a type is always numbered 0.

SCSI ID number — A unique number assigned to the SCSI controller. The standard controller numbering scheme is for controllers to be SCSI ID 7.

SCSI bus number — Indicates if the controller is first or second in the system. The first controller is tied to SCSI bus 0, and the second to SCSI bus 1.

❷ SCSI ID — A unique number you assign to each SCSI device in the system. This is usually done with jumpers or a thumb wheel attached to the drive housing.

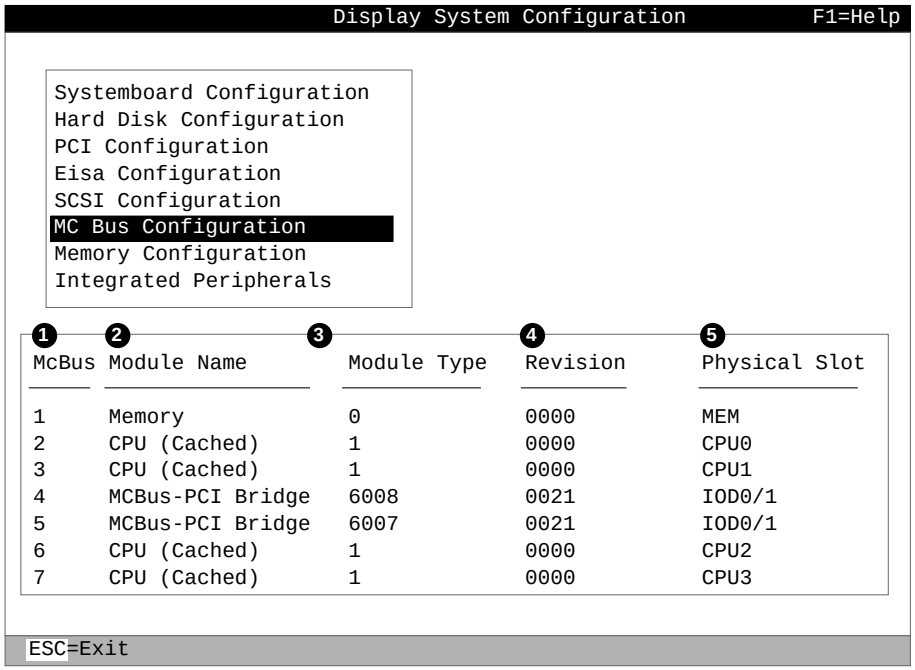
❸ Device type — Displays whether the device is a hard disk, CD-ROM, scanner, or other type of device.

❹ Size — The raw capacity of the drive. Formatting the drive with different file systems (for example, FAT and NTFS) may result in different usable sizes because of differences in the way those file systems manage storage. This is left blank if it is not applicable to a device (for example, a scanner).

❺ Description — The name and model of the device as recorded in the device's firmware.

## 6.4.6 System Bus Configuration

### Figure 6–13 System Bus Configuration




PK-0744A-96

❶ MC bus number — Identifies the location of a component on the system bus.

❷ Module name — The Digital internal identifier for a system component.

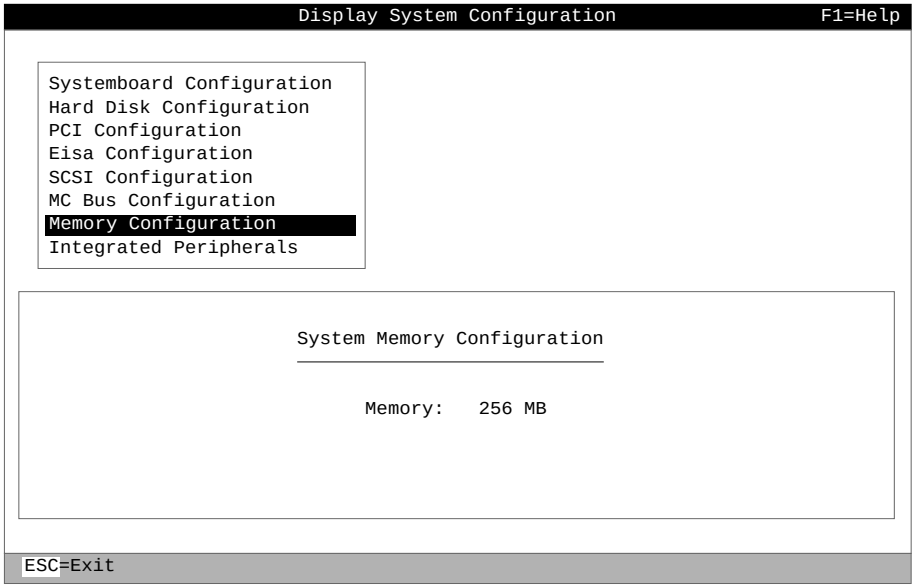
❸ Module type — Identifies the function of the system component.

❹ Revision — A Digital internal number representing the manufacturing revision level of the system component.

❺ Physical slot — Represents the actual location of the system component on the bus. Refer to Chapter 1 for more information about the system bus and module placement.

## 6.4.7 Memory Configuration

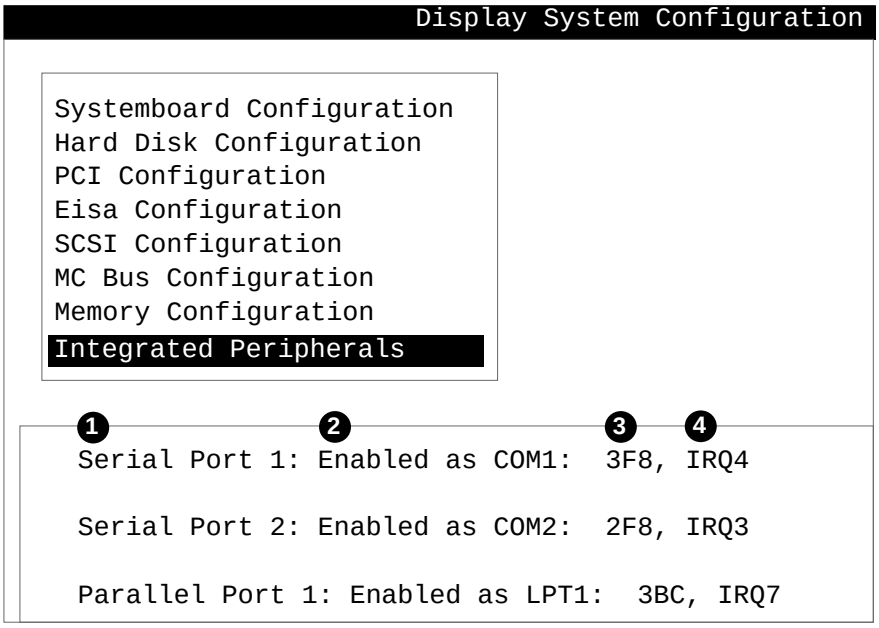
### Figure 6–14 Memory Configuration




PK-0745-96

### 6.4.8 Integrated Peripherals

**Figure 6–15 Integrated Peripherals**




PK-0746A-96

❶ Device type — The physical device as it exists on the system board

❷ MS DOS name — Shows if the device is enabled, and if it is enabled, the addressable MS-DOS name for the device.

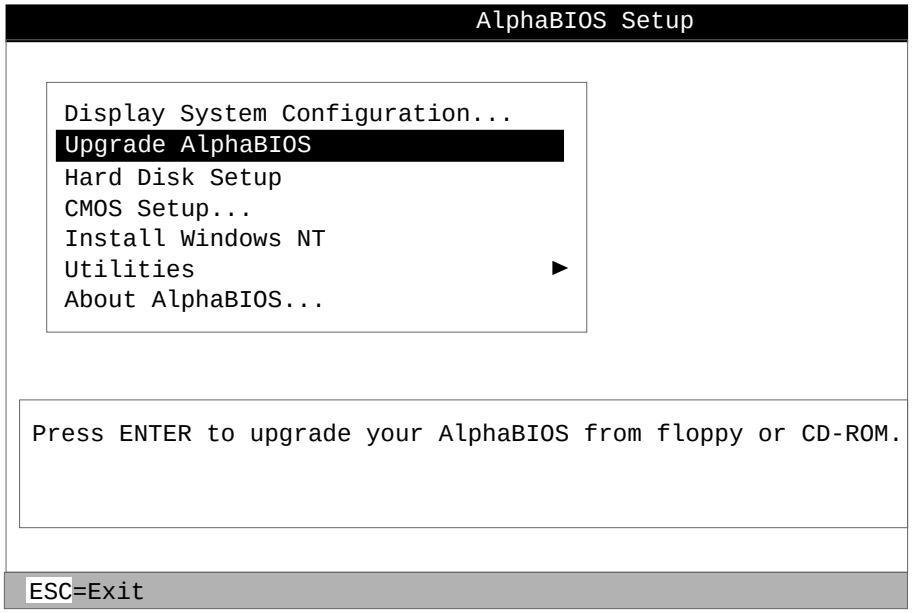
❸ Port address — The physical memory location from and to which data travels as it is received into the device, and sent from the device, respectively.

❹ Interrupt — The interrupt request line (IRQ) used by the device to get the CPU's attention.

## 6.5 Updating Firmware

**Insert the CD-ROM or diskette with the updated firmware and select Upgrade AlphaBIOS from the main AlphaBIOS Setup screen. Use the Loadable Firmware Update (LFU) utility to perform the update. The LFU exit command causes a system reset.**

### Figure 6–16 Updating Firmware




PK-0726A-96

As new versions of Windows NT are released, it might be necessary to upgrade AlphaBIOS to the latest version. Additionally, as improvements are made to AlphaBIOS, it might be desirable to upgrade to take advantage of new AlphaBIOS features.

Use this procedure to upgrade from an earlier version of AlphaBIOS:

1. Insert the diskette or CD-ROM containing the AlphaBIOS upgrade.
2. If you are not already running **AlphaBIOS Setup**, start it by restarting your system and pressing F2 when the **Boot** screen is displayed.
3. In the main **AlphaBIOS Setup** screen, select **Upgrade AlphaBIOS** and press Enter.

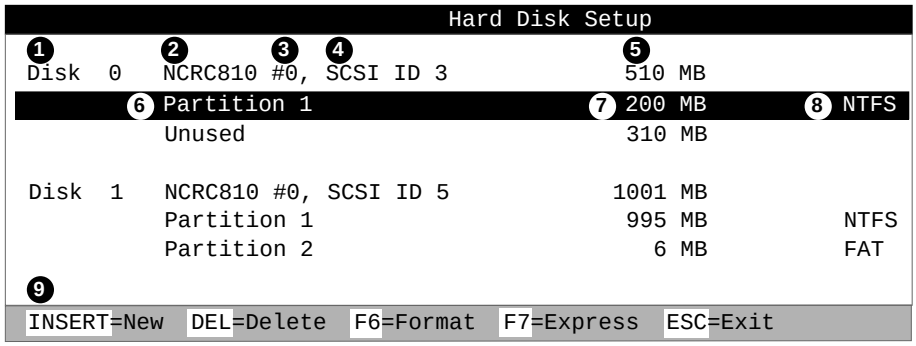
   The system is reset and the Loadable Firmware Update (LFU) utility is started. See Section 2.9.5 for LFU commands.
4. When the upgrade is complete, issue the LFU **exit** command. The system is reset and you are returned to AlphaBIOS.

   If you press the Reset button instead of issuing the LFU **exit** command, the system is reset and you are returned to LFU.

## 6.6 Setting up the Hard Disk

**You can perform either an express or a custom hard disk setup. An express setup, described in this section, creates the recommended partition arrangement on the first hard disk, but it does not map bad sectors. Custom hard disk setup is described in Sections 6.6.1 and 6.6.2.**

### Figure 6–17 Hard Disk Setup Screen




PK-0732A-96

*CAUTION: Hard disk changes are immediate; the changes are made as soon as they are entered. Unintended data loss could occur, so it is important to use care when changing your hard disk arrangement.*

❶ Physical disk ID — Based on the SCSI ID. The disk with the lowest SCSI ID is disk 0, the disk with the next lowest SCSI ID is disk 1, and so on.

❷ Controller — Brand and model of SCSI chip used on the SCSI controller.

❸ Controller number — Based on how many SCSI controllers of a particular type are installed in the system. The first controller of a type is always numbered 0.

❹ SCSI ID number — A unique number you assign to each SCSI device in the system. This is usually done with jumpers or a thumb wheel attached to the drive housing.

❺ Size — The raw capacity of the drive. Formatting the drive with different file systems (for example, FAT and NTFS) may result in different usable sizes because of the differences in how storage is managed under those file systems.

❻ Partition number — Within a single drive, partition numbers are assigned in sequential order: 1, 2, 3, and so on. The partitions populate the drive from the innermost cylinders to the outermost. If you have a large hard disk (over 800 MB) and plan to use the FAT file system, it is a good idea to break the disk into several smaller partitions because the FAT file system uses disk space more efficiently at smaller partition sizes. This is not a concern for the NTFS file system, since it uses disk space very efficiently at all partition sizes.

❼ Partition size — The raw (unformatted) storage capacity of the partition. Actual storage space will differ based on the file system with which the partition is formatted.

❽ Partition format — The file system (if any) used on a partition. This fie ld displays FAT, NTFS, or unknown (if the partition is unformatted).

❾ Disk setup options

Insert partition (Insert key) — Use this option to create new partitions. Before creating a new partition, be sure to select an unpartitioned space. Pressing the Insert key while an already partitioned space is selected causes an informational error to be displayed.

Delete partition (Delete key) — Use this option to delete existing partitions. Before deleting a partition, be sure any data you want to save on the partition has been backed up. Deleting a partition deletes all data on that partition but leaves the rest of the disk unaffected. Pressing the Delete key while an unpartitioned space is selected displays an informational error.

Format partition (F6 key) — Format a partitioned space with the FAT file system.

Express setup (F7 key) — Create the default recommended partition arrangement for Windows NT.

Exit (Escape key) — Return to the **AlphaBIOS Setup** screen.

An express hard disk setup creates the recommended partition arrangement on the first hard disk (disk 0). It does not, however, format the large partition with NTFS, and it does not map bad sectors on the disk. The NTFS formatting can be done during Windows NT installation.

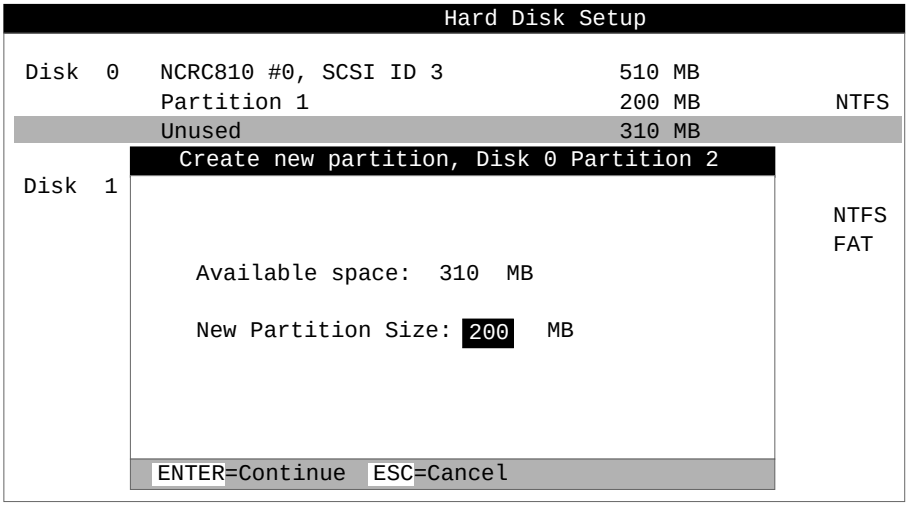
### To perform an express hard disk setup

1. If you have any needed information on your disk, back it up before proceeding.
2. Start **AlphaBIOS Setup**, select **Hard Disk Setup**, and press Enter.
3. Press F7 to enter **Express Setup**.
4. Press F10 to continue with the setup.

## 6.6.1 Manually Creating and Deleting Partitions

**Use the Create Partition and Delete Partition options if you need to create a custom hard disk partition arrangement or otherwise manually manage your hard disk partitions.**

### Figure 6–18 Create New Partition Dialog Box




PK-0733-96

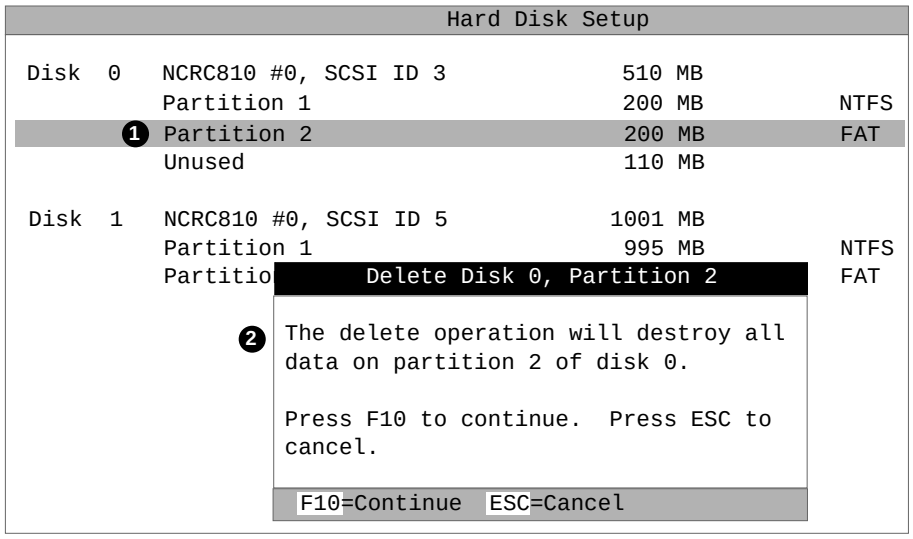
#### To create a partition

1. Start AlphaBIOS and select **Hard Disk Setup**. Press Enter.
2. Select the disk on which to create the partition.
3. If one or more partitions already exist on the disk, select the unpartitioned space.
4. Press Insert. A dialog box displays, similar to Figure 6–18.
5. Type the size of the partition to create and press Enter.

### To delete a partition

1. Start AlphaBIOS and select **Hard Disk Setup**. Press Enter.
2. Select the partition to be deleted (see ❶ in Figure 6–19).
3. Press Delete. A dialog box displays (see ❷).
4. Press F10 to confirm the deletion.

## Figure 6–19 Delete Partition Dialog Box

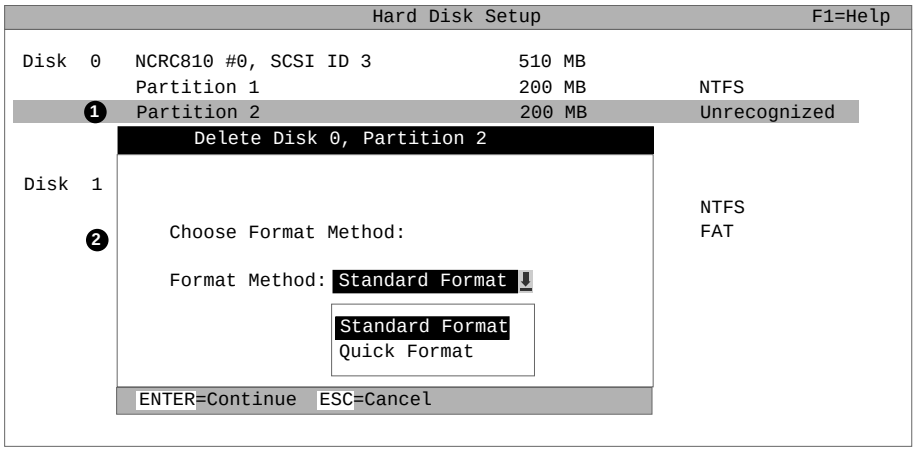



PK-0734A-96

## 6.6.2 Formatting a FAT Partition

**AlphaBIOS can format partitions with the FAT file system. Use Windows NT to format a partition using NTFS.**

### Figure 6–20 Formatting a FAT Partition

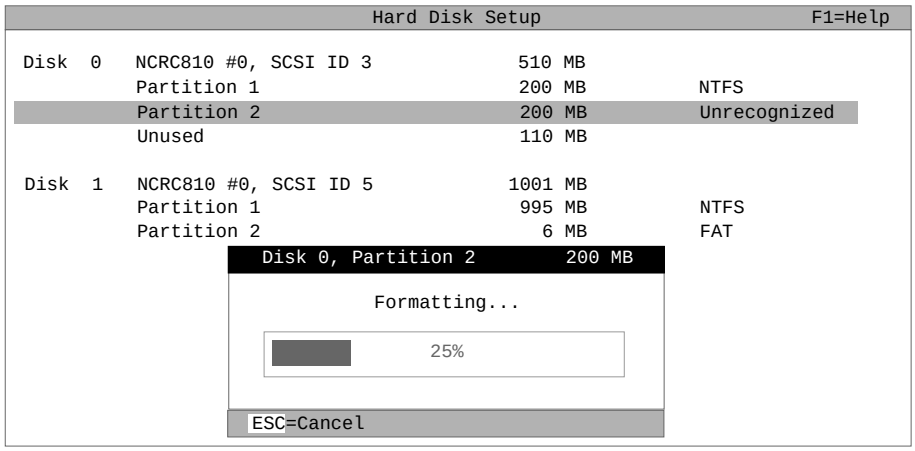



PK-0735A-96

### To format a FAT partition

1. Start AlphaBIOS and select **Hard Disk Setup**. Press Enter.
2. Select the partition to be formatted (see ❶ in Figure 6–20).
3. Press F6. A dialog box displays, asking whether to perform a quick or standard format (see ❷). If you select **Quick Format**, the formatting is completed immediately, but no bad sectors are mapped. If you select **Standard Format**, a dialog box similar to that in Figure 6–21 displays while the drive is formatted, showing the progress of the formatting. Standard formatting maps bad sectors.
4. Select a format method, and press Enter.

## Figure 6–21 Standard Formatting

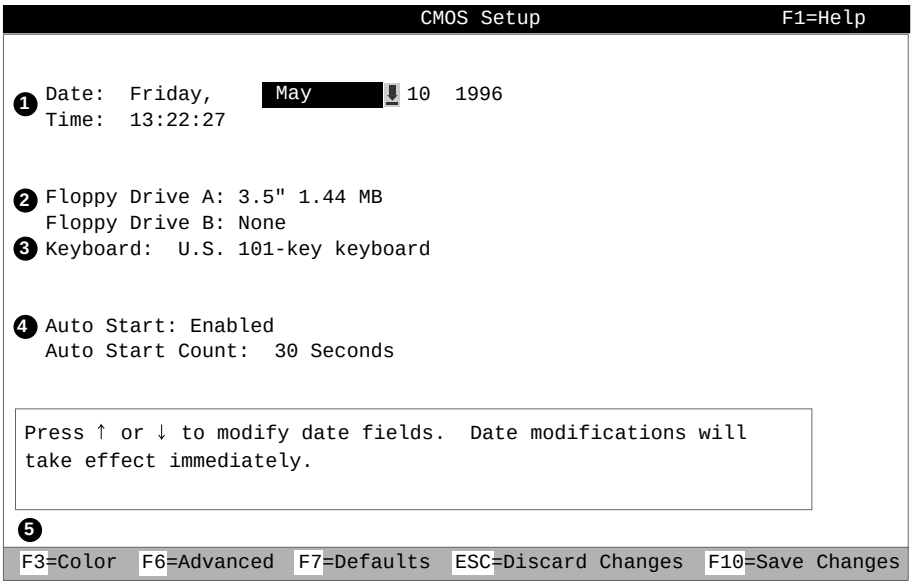



PK-0736-96

## 6.7 Performing Setup Tasks

**CMOS Setup is used to configure several system parameters. CMOS Setup has two modes: Standard CMOS Setup is used to configure basic system parameters; Advanced CMOS Setup is used for system-specific parameters and password protection.**

### Figure 6–22 Standard CMOS Setup Screen




PK-0749A-96

#### To enter Standard CMOS Setup

- Start AlphaBIOS, select **CMOS Setup**, and press Enter.

❶ Date and time — When setting the time, use the 24-hour format. (For example, 10:00 p.m. is 22:00:00.)

❷ Floppy drive — The only drive type supported is 3.5 inch, 1.44 MB.

❸ Keyboard — The keyboard setting makes it possible to use most language keyboards. To ensure correct character mappings, the language of your keyboard, Windows NT, and the keyboard language selection in **CMOS Setup** should all match.

❹ Auto start and auto start count — The **Auto Start** setting determines whether the primary operating system is automatically started after the system is reset or power-cycled. The Auto Start Count setting is the amount of time the **Boot** screen is displayed before the default system is automatically started. This delay gives you the opportunity, after resetting or power-cycling the system, to select another version of Windows NT to start or to enter **AlphaBIOS Setup**.

❺ **Standard CMOS Setup** options

Color (F3 key) — Repeatedly cycles through the available AlphaBIOS color schemes.

Advanced (F6 key) — Displays the **Advanced CMOS Setup** screen.

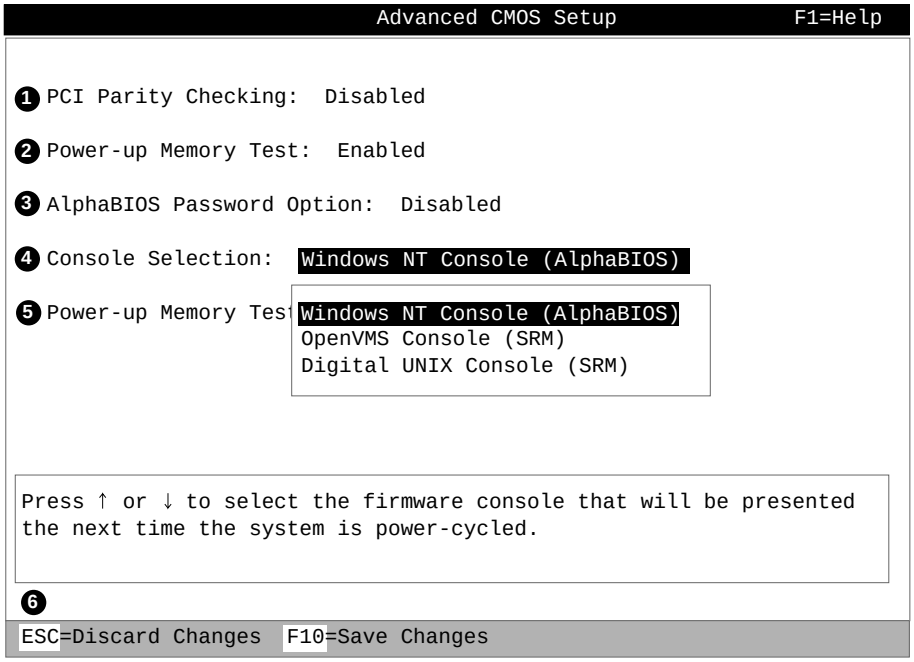
Defaults (F7 key) — Restores the default **Standard CMOS Setup** values without affecting the **Advanced CMOS Setup** values.

Discard changes (Escape key) — Restores the settings in effect when you started **CMOS Setup**. This option also discards changes made in **Advanced CMOS Setup**.

Save changes (F10) — Saves changes made in both the standard and advanced modes of **CMOS Setup**.

*Continued on next page*

## Figure 6–23 Advanced CMOS Setup Screen




PK-0750A-96

### To enter Advanced CMOS Setup

1. Start AlphaBIOS, select **CMOS Setup**, and press Enter.
2. In the **Standard CMOS Setup** screen, press F6.

❶ PCI parity checking — Enables and disables settings for PCI parity checking, which ensures data integrity across the PCI bus. Because some third-party PCI options do not correctly implement PCI parity generation, the default is Disabled.

❷ Power-up memory test — Enables and disables settings for the power-up memory test.

❸ Password setup — Enables and disables settings for password setup.

Password protection provides two levels of security for your system: setup, which requires a password to start **AlphaBIOS Setup**, and startup, which requires a password before the system initializes. Startup protection provides more comprehensive protection than setup protection because the system cannot be used at all until the correct password is entered.

❹ Console selection — Switches from AlphaBIOS to SRM firmware. If you select OpenVMS (SRM) or Digital UNIX (SRM), the next time you reset your system, the SRM firmware will be loaded instead of AlphaBIOS. This menu selection changes the setting of the **os_type** environment variable in the SRM console.

❺ Memory test length — Selects the length of memory testing that will be done the next time the system is power-cycled.

❻ **Advanced CMOS Setup** options

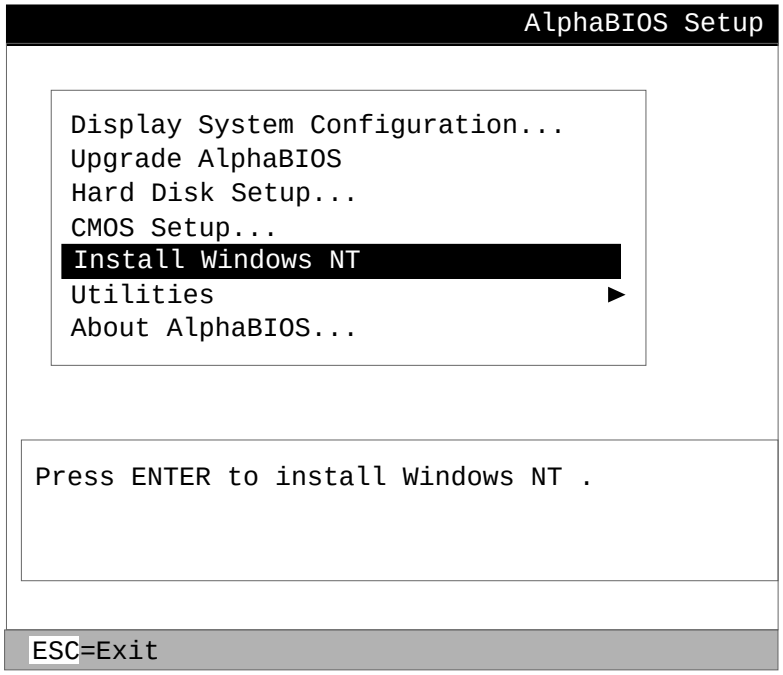
Discard changes (Escape key) —Restores those settings in effect when you started **Advanced CMOS Setup**. This does not discard changes made to **Standard CMOS Setup**.

Save Changes (F10 key) — Saves changes made in **Advanced CMOS Setup**. Note that, when exiting **CMOS Setup**, you must also save your changes at the **Standard CMOS Setup** screen for the changes to be stored permanently.

## 6.8 Installing Windows NT

**Install Windows NT from CD-ROM. Up to four versions of the operating system can be resident in the system at one time.**

### Figure 6–24 Installing Windows NT




PK-0726B-96

If Windows NT was installed at the factory, Windows NT setup will start automatically the first time the system powers up. If it was not installed, or if you are installing another version, you must have a CD-ROM drive attached to your system.

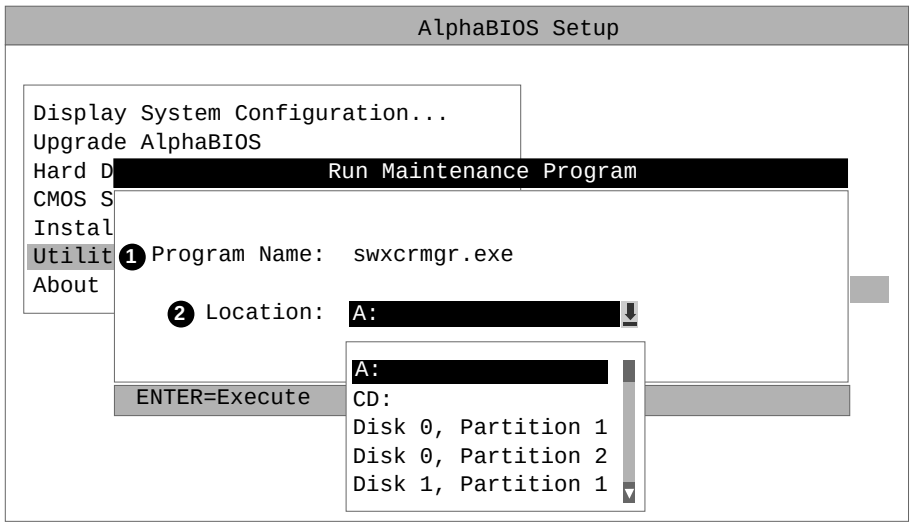
*NOTE: Steps 1 and 2 in the following procedure are necessary only when you are first setting up your system. On subsequent installations and upgrades, begin at step 3.*

1. Use **CMOS Setup** to set the system date and time: start **AlphaBIOS Setup**, select **CMOS Setup**, and press Enter.
2. Perform an express hard disk setup: return to the main **AlphaBIOS Setup** screen, select **Hard Disk Setup**, and press Enter.
3. Put the Windows NT CD into the CD-ROM drive.
4. Start **AlphaBIOS Setup**, select **Install Windows NT**, and press Enter.
5. Follow the prompts to complete the installation. For more information on installing Windows NT, refer to the *Installation Guide* in your Windows NT software package.

## 6.9 Running Utility Programs

**Maintenance programs such as RAID and EISA configuration utilities are run from the AlphaBIOS Utility menu.**

### Figure 6–25 Run Maintenance Program Dialog Box




PK-0747A-96

❶ Program name — The program to be run. It must be an executable program with an .EXE filename extension. However, when entering the program name, it is not necessary to type the extension.

Programs run from AlphaBIOS must be written as ARC compatible images.

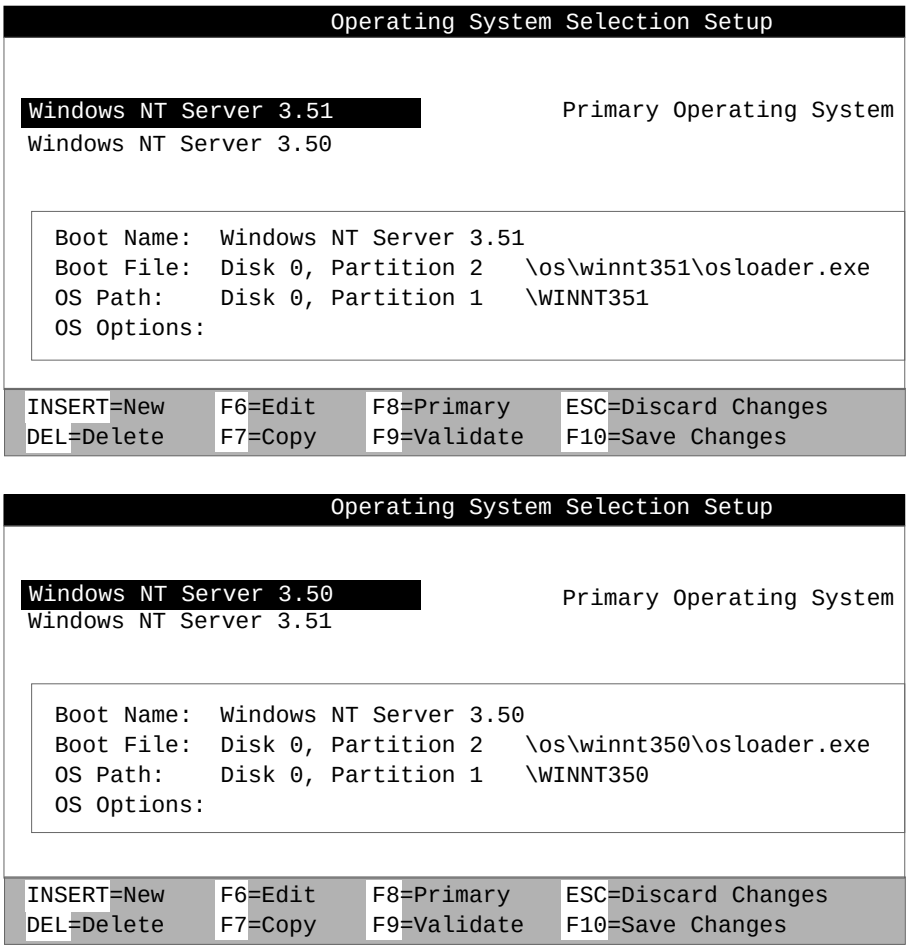
❷ Location — The location from which the program in the Program Name field will be run if no path is entered along with the program name. To display a list of all available disks and partitions, press the Alt and down arrow keys at the same time. Once the list is displayed, use the plus arrow keys to cycle through the selections.

See Chapter 7 for directions for running utilities.

## 6.10 Selecting the Version of Windows NT

**Up to four versions of Windows NT can be resident on the system at one time. It is necessary to select the version that will be started.**

### Figure 6–26 Operating System Selections




PK-0723-96

*NOTE: The term "operating system selection," as it is used in this context, refers to a version of Windows NT. It does not pertain to the Digital UNIX and OpenVMS operating systems.*

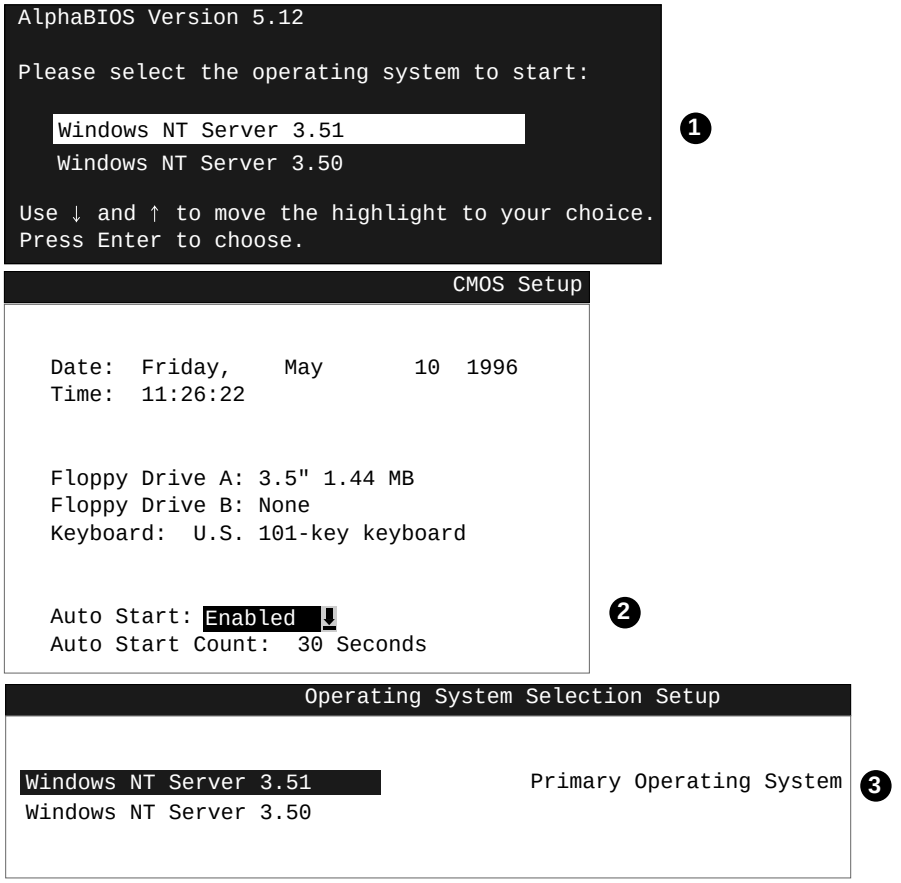
Each operating system selection is a set of information for a version of Windows NT. It describes the disk and partition containing the OSLOADER.EXE file associated with a particular operating system installation, as well as the path to the operating system itself. In addition, the operating system selection contains any options passed to the operating system during boot.

By default, one operating system selection displays on the **AlphaBIOS Boot** screen for each operating system installed on the system. It is not normally necessary to modify operating system selections, as the Windows NT installation creates and modifies operating system selections as needed.

Figure 6–26 shows how operating system selections appear in AlphaBIOS.

### 6.10.1 Designating a Primary Operating System

**Figure 6–27 Primary Operating System**




PK-0720A-96

Multiple versions of Windows NT can be installed at the same time. This can be very useful in a variety of circumstances—for example, when testing application compatibility across different versions of Windows NT.

Each time you install a separate version of Windows NT, a new operating system selection is created. Although you can start any of the installed versions of Windows NT, one of them must be the primary operating system. The version of Windows NT you select as the primary operating system is the one that automatically starts if the **Auto Start** option is enabled in AlphaBIOS.

Figure 6–27 illustrates the relationship between multiple-operating-system installations, **Auto Start**, and the primary operating system.

❶ The primary operating system is listed first on the **Boot** screen.

❷ The primary operating system starts automatically if the **Auto Start** option is enabled in **CMOS Setup**.

❸ The primary operating system can be selected in the **Operating System Selection Setup** screen.

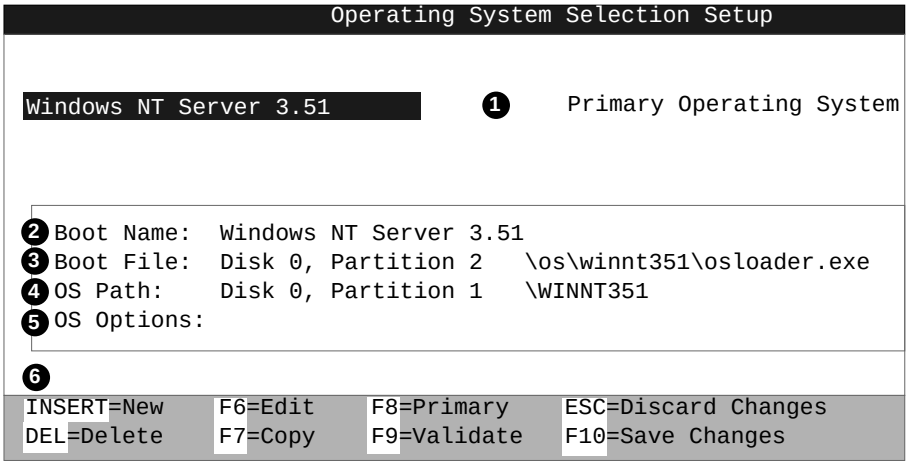
Follow this procedure to designate a new primary operating system:

1. From the **AlphaBIOS Setup** screen select **Utilities**. In the selection box that displays, choose **OS Selection Setup**.
2. The **Operating System Selections Setup** screen displays. Select the primary operating system from the list displayed.

## 6.10.2 Primary Operating System and the Auto Start Option

**The process of setting up your operating system selections is similar to using an editor. You can make changes to your operating system selections and then either save your changes or exit without saving.**

### Figure 6–28 Operating System Selection Setup




PK-0737A-96

AlphaBIOS boots Windows NT in two stages. The first stage involves reading an OS Loader program from a disk. As of Windows NT version 3.51, the name of the OS Loader program is OSLOADER.EXE. The selection for the operating system must describe the path where AlphaBIOS will find the OS Loader program. Note that the OS Loader program *must* be located on a partition formatted with the FAT file system.

Once the OS Loader program executes, it uses the services provided by AlphaBIOS to load the operating system components. After the operating system is loaded, the OS Loader starts execution of the operating system.

❶ Primary operating system — The OS that appears first on the **AlphaBIOS Boot** screen. It is also the version of the OS that automatically starts if **Auto Start** is selected. Any of the operating system selections can be the primary operating system.

❷ Boot name — Each boot name is associated with an operating system selection. Windows NT setup automatically creates a boot name each time you install the operating system. Because the boot name is only a name, you can modify it at any time without affecting the rest of the operating system selection. Note that the boot name must have at least one character.

❸ Boot file — Describes the disk, partition, path, and name of the file that AlphaBIOS passes control to during the process of starting the operating system. This setting is created along with the operating system selection during Windows NT setup, and it is usually not modified by the user. However, this setting can be modified if necessary. For example, a developer testing different versions of OSLOADER.EXE can store the different versions in different locations and modify this line to start the operating system with the different versions as needed.

During **Operating System Selection Setup**, you can select the disk and partition for the location of the boot file from a list of choices presented in a list box. To open a list of values for the field, press the Alt and down arrow keys together. When you select a boot file location and name, AlphaBIOS searches for the specified program on the specified partition. If the search fails, a warning displays, saying that the file does not exist. You are given the choice to continue with the changes anyway or to cancel the operation and fix the problem.

Note that because the boot file must be located in a FAT partition, only FAT partitions are available as the boot partition within the list box.

❹ OS path — Describes the disk, partition, and path to the operating system root directory for an operating system selection.

During **Operating System Selection Setup**, you can select the disk and partition for the location of the boot file from a list of choices presented in a list box. To open a list of possible values for the field, press the Alt and down arrow keys at the same time. When you select or enter an OS path, AlphaBIOS searches for the directory. If the search fails, a warning displays, saying that the directory does not exist. You are given the choice to continue with the changes anyway or to cancel the operation and fix the problem.

*Continued on next page*

❺ OS options — Lists the startup parameters passed to the operating system for an operating system selection. One example of a startup parameter is whether to start the operating system in debug mode. By default, Windows NT does not add any entries to this field. This field can be modified.

❻ **Operating System Selection Setup** options — You can use the options listed at the bottom of the screen to edit operating system selections. When you edit an OS selection, the fields of the OS selection are validated when you exit the **OS Selection Setup** screen. Depending on the option you choose, one of the following dialog boxes will display.

New OS selection (Insert key) — Displays the **Insert New Operating System Selection** dialog box, with default values for the new OS selection already filled in. You can change these values as necessary.

Delete OS selection (Delete key) — Tags the currently selected OS selection for deletion. Although the OS selection is removed from the screen, it is not actually deleted until you save changes.

You can also delete all of your OS selections at once by pressing the Control and Delete keys at the same time while in the **Operating System Selection Setup** screen.

*NOTE: If you delete all the OS selections, a dialog box displays informing you that no OS selections exist and offering three options. You can create a new OS selection, exit without saving changes, or exit and save changes.*

Edit OS selection (F6 key) — Edit all values of an OS selection by selecting the OS selection to edit and pressing F6. A dialog box displays with current information. You can then edit the OS selection fields.

Copy OS selection (F7 key) — Create a new OS selection by using an existing OS selection as a template. To do this, select the OS selection you want to copy and press F7. A dialog box displays with the values of your OS selection. You can then edit the OS selection fields. Note that if you do not make any changes, a duplicate copy of the OS selection you copied is made.

Primary operating system (F8 key) — Sets the selected OS selection as the primary operating system. When you make an OS selection primary, it is displayed first on the **Operating System Selection Setup** screen with the text "Primary Operating System." In addition, the primary operating system is displayed first on the **AlphaBIOS Boot** screen and is automatically started if Auto Start is enabled.

Validate OS selection (F9 key) — Validates the fields in the currently selected OS selection. The validation routine checks that the OS loader file and OS directory fields contain valid paths, and that the OSLOADER.EXE file exists in the directory specified. At the end of the validation, a dialog box displays describing the results of the validation. If there is an error in an OS selection, the validation routine displays a dialog box describing the component of the OS selection that is in error. At this point you can choose to edit the OS selection to correct the error, or delete the OS selection altogether.

You can also validate all OS selections at once by pressing the Control and F9 keys at the same time. All OS selections are validated in the order they are listed on screen.

Discard changes (Escape key) — Returns to the **AlphaBIOS Setup** screen without saving changes.

Save changes (F10 key) — Saves changes and returns to the **AlphaBIOS Setup** screen.

# Chapter 7

# Utilities

Utilities that configure EISA options and RAID devices are run from AlphaBIOS. This chapter gives directions for running utilities from both a graphics monitor and a serial terminal.

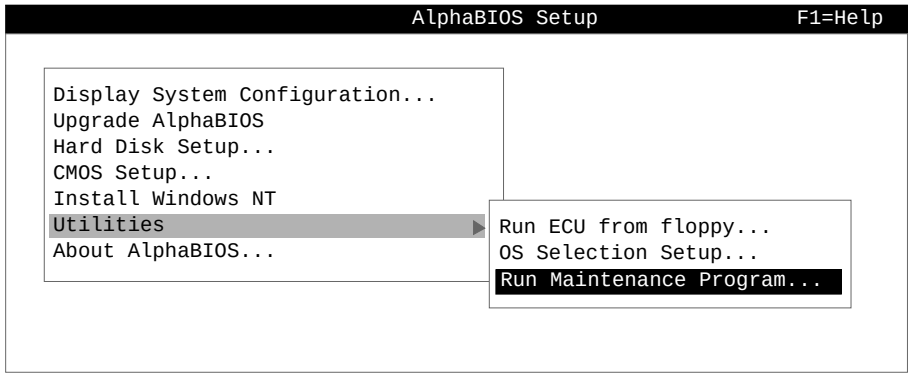
Sections in this chapter are:

- Running Utilities from a Graphics Monitor
- Running Utilities from a Serial Terminal

## 7.1 Running Utilities from a Graphics Monitor

**Start AlphaBIOS and select Utilities from the menu. Then select Run Maintenance Program.**

### Figure 7–1 Running a Utility from a Graphics Monitor



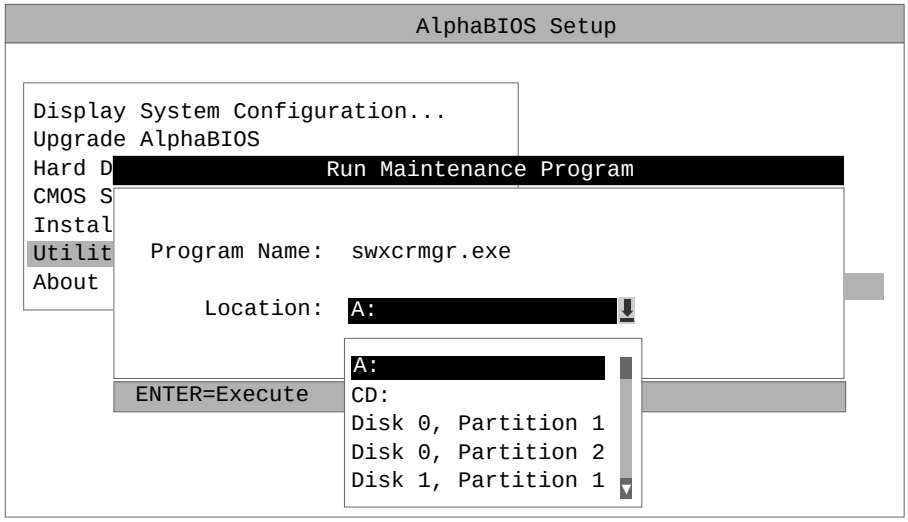

PK-0729-96

1. Start AlphaBIOS Setup. If the system is in the SRM console, set the SRM **console** environment variable to **graphics** and issue the command **alphabios**.
2. From **AlphaBIOS Setup**, select **Utilities**, then select **Run Maintenance Program** from the submenu that displays, and press Enter.
3. In the **Run Maintenance Program** dialog box, type the name of the program to be run in the Program Name field. Then Tab to the Location list box, and select the hard disk partition, floppy disk, or CD-ROM drive from which to run the program.
4. Press Enter to execute the program.

*NOTE: If you are running the EISA Configuration Utility, use the appropriate program for the operating system you intend to boot. If you run the wrong program, your system will behave unpredictably during booting.*

See Section 6.3 for information about navigating through AlphaBIOS screens.

## Figure 7–2 Run Maintenance Program Dialog Box




PK-0747-96

## 7.2 Running Utilities from a Serial Terminal

**Utilities are run from a serial terminal in the same way as from a graphics monitor. The menus are the same, but some keys are different.**

### Table 7–1 AlphaBIOS Option Key Mapping

| AlphaBIOS Key | VTxxx Key |
|---|---|
| F1 | Ctrl/A |
| F2 | Ctrl/B |
| F3 | Ctrl/C |
| F4 | Ctrl/D |
| F5 | Ctrl/E |
| F6 | Ctrl/F |
| F7 | Ctrl/P |
| F8 | Ctrl/R |
| F9 | Ctrl/T |
| F10 | Ctrl/U |
| Insert | Ctrl/V |
| Delete | Ctrl/W |
| Backspace | Ctrl/H |
| Escape | Ctrl/[ |

1. Start **AlphaBIOS Setup** by issuing the command **alphabios**.
2. Press F2 in the **AlphaBIOS Boot** screen.
3. From **AlphaBIOS Setup**, select **Utilities**, and select **Run Maintenance Program** from the submenu that displays. Press Enter.
4. In the **Run Maintenance Program** dialog box, type the name of the program to be run in the Program Name field. Then tab to the Location list box, and select the hard disk partition, floppy disk, or CD-ROM drive from which to run the program.
5. Press Enter to execute the program.

*NOTE: If you are running the EISA Configuration Utility, use the appropriate program for the operating system.*

# Appendix A

# How to Find Firmware Updates

Firmware for all AlphaServer systems is available for download from the Internet. You can access the firmware as follows.

- Using a World Wide Web browser (such as Mosaic or Netscape), follow links to AlphaServer 4100 systems, then technical resources from URL:
  http://www.digital.com/info/alphaserver/products.html
- Using ftp, copy the file:
  ftp://ftp.digital.com/pub/Digital/Alpha/firmware/

The files are structured as those on the firmware CD and are separated by CD release. For example, the contents of the V3.6 firmware CD are located at:
ftp://ftp.digital.com/pub/Digital/Alpha/firmware/v3.6/

The latest firmware (if released since the last firmware CD) is located at:
ftp://ftp.digital.com/pub/Digital/Alpha/firmware/interim/

# Appendix B

# Hard Disk Partitioning for AlphaBIOS and Windows NT

The recommended hard disk partition arrangement on the first hard disk in your system is:

- Partition 1 — Six megabytes less than the total size of the drive. This large partition holds the operating system and the application and data files.
- Partition 2 — The remaining six megabytes. This small partition holds only the few files necessary for your computer to boot.

This arrangement provides two benefits.

- Windows NT requires that a boot partition be formatted with the FAT file system. However, the Windows NT file system (NTFS) provides advantages over FAT, such as additional security and more efficient use of disk space. By keeping the FAT boot partition as small as possible, the maximum amount of space is left available for use as an NTFS partition.
- Most applications install themselves onto drive C by default. Although the program installation drive is usually configurable by the user, many people accept the default of drive C. By making the first partition large, drive C becomes the larger drive. This arrangement makes program installation easier and avoids time-consuming insufficient disk space mistakes.

## B.1 Hard Disk Error Conditions

### B.1.1 Disk Initialization Failed

When you start hard disk setup, if you receive an "Internal error occurred" message, it means that a disk was found, but there was an error in communicating with the disk. The likely conditions that can cause this error are:

- Incompatible or failed disk cables. You may not have the correct cable installed, or the cable might have a broken lead or connector. Try another cable known to be good.
- Disk controller not configured. You may have to run a configuration utility to set up your hard disk controller. Check your controller documentation.

- Disk controller malfunction. Most controllers come with a diagnostic utility to test controller functioning. If a controller error is found, call the manufacturer for a replacement.
- Improper SCSI termination. Many SCSI controllers require that the terminating resistor packs be removed from all drives between the controller and last drive. Only the controller itself, and the last drive connected to the controller (the ends of the "chain"), should have terminating resistor packs left on.

### B.1.2 No Hard Disks Found

When you start hard disk setup, if you receive a "No hard drives were found connected to your computer" message, it means that AlphaBIOS could not locate a hard drive. The likely conditions that cause this error are:

- Cable not connected to either the disk or controller. The cable may have worked loose from the connector on the controller or disk drive. Check the cable connections, making sure the cable connectors are fully seated.
- No power to the drive. The power connector may have worked loose from the receptacle on the drive, or the power cable itself may be malfunctioning. Check the cable connections, making sure the cable connectors are fully seated. Try connecting the drive to a different power connector.
- Disk drive malfunction. The disk drive itself may be malfunctioning and not responding to requests from the controller. If this is the problem, then it would appear as though the disk were absent. Replace the drive with a drive known to be good. If the known good drive is correctly detected, then your hard drive is defective. Contact the manufacturer for a replacement.

### B.1.3 No Partitions on Disk

If hard disk 0 does not have any partitions defined, then a message will appear when you start hard disk setup, asking if you want to perform an express disk setup. Express disk setup automatically creates the default disk partition arrangement on hard disk 0.

For more information on express setup, see Chapter 6.

## B.2 System Partitions

To install Windows NT, a system partition must be defined. If no system partition is found, the user is asked to choose from the available FAT partitions.

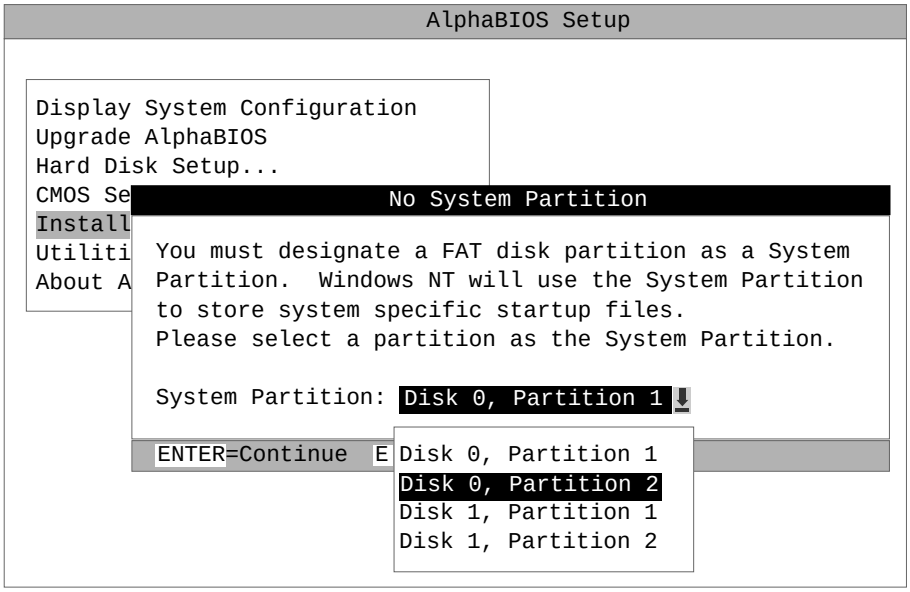
The purpose of the system partition is twofold. First, it tells the Windows NT installation program where to place the OS Loader and hardware support files. Second, upon subsequent restarts of Windows NT, the system partition definition tells AlphaBIOS where the OSLOADER.EXE file is so it can successfully hand off control to the OS Loader and continue the boot process.

The system partition can be the same partition into which Windows NT is installed, or it can be separate. However, the system partition must be formatted with the FAT file system. Because the default recommended partition arrangement on Alpha calls for the partition into which Windows NT is installed to be an NTFS partition, the system partition will almost always be a small FAT partition on the same disk onto which Windows NT is installed.

## B.3 How AlphaBIOS Works with System Partitions

If you are installing Windows NT for the first time, AlphaBIOS will determine that a system partition has not been defined when you select **Install Windows NT** in the **AlphaBIOS Setup** screen (see Figure B–1). When this occurs, AlphaBIOS searches for all FAT partitions on the system. If only one FAT partition exists, AlphaBIOS designates that FAT partition as the system partition and continues with the Windows NT installation. If more than one FAT partition exists on your system, AlphaBIOS displays the list of FAT partitions from which you can choose the system partition. After choosing the system partition, the installation process continues.

### Figure B–1 System Partition Not Defined




PK-0748-96

# Appendix C

# Differences Between AlphaBIOS and ARC Firmware

This appendix is for users of Alpha systems who are experienced with ARC firmware, and who want an overview of the major differences between ARC and AlphaBIOS firmware.

## C.1 Revamped User Interface

AlphaBIOS has a graphical-style windowed interface. Many of the navigational and selection keystrokes implemented in Microsoft Windows are present in AlphaBIOS as well. If you are familiar with navigating Microsoft Windows using a keyboard, AlphaBIOS navigation will also be simple. Additionally, information and selections are grouped more logically. Groups of related tasks can often be performed from a single screen.

## C.2 Hard Disk Setup

Using ARC firmware, initial hard disk partitioning and formatting was accomplished with a utility called ARCINST.EXE. This utility is distributed on the Windows NT CD-ROM. This utility was run in a manner similar to how a maintenance program is run with AlphaBIOS; that is, from a command line. Under AlphaBIOS, hard disk setup functionality is integrated into the menu-based interface.

The following list summarizes the new hard disk setup functionality in AlphaBIOS but not found in the ARC program:

- With one keystroke, express disk setup configures your first hard disk with the recommended partition arrangement.
- AlphaBIOS uses the same path descriptions as Windows NT Disk Administrator, rather than the ARC path names used in ARCINST. For example:
- ARC Path Name = scsi(0)disk(0)rdisk(0)partition(1)
- AlphaBIOS Path Name = Disk 0, Partition 1
- All partitions of all disks can be seen at the same time.
- Disk administration is integrated into the BIOS, rather than being a maintenance program.
- No menu hierarchy to navigate.

- Detailed display of disk and partition configuration.
- Windowed interface.

## C.3 Changing CMOS and NVRAM Values

With ARC firmware, a number of settings stored in CMOS and NVRAM are set using several separate ARC firmware selections. With AlphaBIOS, these settings are more logically arranged by being integrated into the CMOS Setup program. Table C–1 summarizes the differences between AlphaBIOS and ARC firmware in this regard.

### Table C–1 Differences Between AlphaBIOS and ARC Menu Selections

| Function | AlphaBIOS Menu | ARC Menu |
|---|---|---|
| Set date/time | CMOS Setup | Set System Time |
| SCSI termination and PCI parity checking | CMOS Setup | Machine specific setup |
| Floppy and keyboard | CMOS Setup | Set Default Configuration |
| Default system partition | N/A (handled automatically by hard disk setup) | Set Default Environment Variables |
| SCSI host ID | N/A | Set Default Configuration |
| Automatic OS startup | CMOS Setup | Setup AutoBoot |
| Edit environment variables | N/A (handled automatically by AlphaBIOS) | Edit Environment Variables |
| Set monitor resolution | N/A | Set Default Configuration |

## C.4 Working with Operating Systems

Because Alpha systems can have multiple versions of Windows NT installed simultaneously, you can examine, verify, and modify the values associated with each installed version of Windows NT. With ARC firmware, the group of values associated with a particular Windows NT installation are collectively known as a "boot selection." With AlphaBIOS, these values are known as an "operating system selection."

With ARC firmware, working with boot selections is difficult because the various pieces of information necessary are located on separate screens displayed by using the Manage Boot Selections menu. In addition, the interface provides no way to easily see the alternatives available for a given item. For example, if you want to customize a boot selection to use an OSLOADER.EXE file on a different partition or disk, there is no way to see the different choices available while in the menu where you enter that information. This makes it necessary to navigate back and forth between the separate menus these pieces of information are located under and to remember the information for entry into the correct menu.

With AlphaBIOS, working with operating system selections has been greatly simplified by virtue of the revamped user interface. All the functionality available under ARC firmware has been integrated into a single Operating System Selection Setup screen. Alternative choices for values, such as disk and partition numbers, are available with drop-down list boxes, and only valid values are displayed. This makes processes that were once error-prone and clumsy relatively smooth and error-free.

In addition, a new feature, copying operating system selections, makes creating new operating system selections even easier by making it possible to use an existing selection as a template for a new selection.

## C.5 Running a Program from the BIOS

The procedure for running configuration programs remains essentially unchanged with AlphaBIOS. With ARC firmware, the menu selection to run a program was located on the Boot menu, whereas, with AlphaBIOS, you can select **Utilities** from the main AlphaBIOS Setup screen to run configuration programs.

Note that the selection name has changed from "Run a program" under ARC, to "Run a maintenance program" under AlphaBIOS.

## C.6 Differences in System Configuration Display

The system configuration display in ARC firmware provides general information about system setup. In contrast, the system configuration display in AlphaBIOS displays detailed information categorized by major subsystem. In addition, for many

of the elements listed, you can select the element and obtain additional detailed information.

## C.7 Resetting to Factory Defaults

With ARC firmware, the option to reset the system to factory defaults is located on the Setup screen. With AlphaBIOS, the equivalent function is performed by pressing **F7** in the CMOS Setup screen.

## C.8 Multilingual Support

Support for multiple languages, which was recently added to ARC firmware, is not currently a feature of AlphaBIOS.

# Appendix D

# Acoustical Data

**Table D–1 Acoustics - Declared Values per ISO 9296 and ISO 7779**

| | Sound Power Level $L_{WAd}$, B | | Sound Pressure Level $L_{pAm}$, dBA (bystander positions) | |
|---|---|---|---|---|
| Product | Idle | Operate | Idle | Operate |
| BA30A-AA (AlphaServer 4100 drawer as sub-assembly, 1-4 cpu, 1-2 psu) | 6.1 | 6.1 | 42 | 42 |
| BA30P-AA/AB (BA30A with 1-4 cpu, 1-2 psu, no drives, in pedestal cab) | 6.0 | 6.0 | 41 | 41 |
| BA30A with 1-4 cpu, 1-2 psu + 2xBA356 with 2xRZ29B each, in pedestal cab | 6.3 | 6.4 | 44 | 44 |
| H9A10-EB/EC (diskless rack) | 6.3 | 6.3 | 39 | 39 |
| BA30A with 1-4 cpu, 1-2 psu, in H9A10 | 6.1 | 6.1 | 42 | 42 |
| BA356 with 2xRZ29B, in H9A10 | 6.0 | 6.1 | 40 | 41 |

[Current values for specific configurations are available from Digital representatives 1 B = 10 dBA.]

**Table D–2 Schallemissionswerte - Werteangaben nach ISO 9296 und ISO 7779/DIN EN27779**

| | Schalleistungspegel $L_{WAd}$, B | | Schalldruckpegel $L_{pAm}$, dBA (Zuschauerpositionen) | |
|---|---|---|---|---|
| Gerät | Leerlauf | Betrieb | Leerlauf | Betrieb |
| BA30A-AA (AlphaServer 4100 Einschubbau-gruppe, 1-4 cpu, 1-2 psu) | 6,1 | 6,1 | 42 | 42 |
| BA30P-AA/AB (BA30A mit 1-4 cpu, 1-2 psu, ohne Laufwerke, in Standfußge-häuse) | 6,0 | 6,0 | 41 | 41 |
| BA30A mit 1-4 cpu, 1-2 psu + 2xBA356 jeweils mit 2xRZ29B, in Standfußge-häuse | 6,3 | 6,4 | 44 | 44 |
| H9A10-EB/EC (Gestell ohne Laufwerke) | 6,3 | 6,3 | 39 | 39 |
| BA30A mit 1-4 cpu, 1-2 psu, in H9A10 | 6,1 | 6,1 | 42 | 42 |
| BA356 mit 2xRZ29B in H9A10) | 6,0 | 6,1 | 40 | 41 |

[Aktuelle Werte für spezielle Ausrüstungsstufen sind über die Digital Equipment Vertretungen erhältlich. 1 B = 10 dBA.]

# Index


## ?

? command (RCM), 2-51

## A

Advanced CMOS setup screen, AlphaBIOS, 6-34
alert_clr command (RCM), 2-49
alert_dis command (RCM), 2-49
alert_ena command (RCM), 2-49
alphabios command (SRM), 5-2, 5-3
AlphaBIOS console, 2-8, 6-1
  boot screen, 2-18, 6-4
  checking PCI parity, 6-35
  differences from ARC firmware, C-1
  displaying system configuration, 6-8
  enabling and disabling settings for memory test, 6-35
  finding help, 6-6
  first-level help screen, 6-6
  key conventions, 6-6
  partitioning hard disk, B-1
  running in serial mode, 7-4
  second-level help screen, 6-7
  setting date and time, 6-33
  setting keyboard language, 6-33
  setting password, 6-35
  setup screen, 6-5
  starting, 6-4
  switching from SRM, 5-2
  switching to SRM, 6-2, 6-35
Architecture, 1-4
auto_action environment variable (SRM), 5-42
auto_action environment variable (SRM), 2-8

## B

BA30A. *See* System drawer
Baud rate, setting for COM2 port, 5-45
B-cache, 2-5, 2-8
Block diagram, 1-4
boot command (SRM), 2-9, 2-13, 5-12
  passing information to command, 5-43
Boot screen, AlphaBIOS, 2-18, 6-4
boot_osflags environment variable (SRM), 2-10, 5-43
  Digital UNIX, 5-43
  OpenVMS, 5-43, 5-44
Booting
  Digital UNIX, 2-9
  OpenVMS, 2-13
  passing information to command, 5-43
  Windows NT, 2-18
Bridge module, 1-5, 1-16

## C

CMOS setup screen, AlphaBIOS, 6-32, 6-34
COM2 port, setting baud rate, 5-45
com2_baud environment variable (SRM), 5-45
Command syntax, SRM console, 5-37
Configuring parameters, AlphaBIOS console, 6-32
Configuring the system
  Digital UNIX system, 5-14

OpenVMS system, 5-14
Console
AlphaBIOS, 2-8
specifying, 6-35
SRM, 2-8, 5-1
switching from AlphaBIOS to SRM, 6-2
switching from SRM to AlphaBIOS, 5-2
Console commands (SRM), 5-35
console environment variable (SRM), 2-5, 2-8, 2-19, 5-26, 5-45
Console terminal, specifying type, 5-45
continue command (SRM), 5-20, 5-21
Control panel, 1-22
display, 1-22, 2-2, 2-3, 2-5, 3-4
halt button, 1-22, 1-23
messages in display, 3-7
power button, 1-22, 1-23, 2-2
reset button, 1-22, 1-23
specifying display message (SRM), 5-48
Controls
halt button, 1-23
power button, 1-23, 2-2
reset button, 1-23
Cover interlocks, 3-2
CPU module, 1-5, 1-10
enabling and disabling specific processors, 5-45
installing, 4-6
slot numbers, 4-7
cpu_enabled environment variable (SRM), 5-45

## D

Date, setting in AlphaBIOS console, 6-33
deposit command (SRM), 5-28, 5-29
Depositing data, 5-28
Device naming conventions (SRM), 5-7
Digital UNIX
booting, 2-9
specifying as default operating system, 5-49
disable command (RCM), 2-50
display command (LFU), 2-38, 2-39
Display system configuration screen, AlphaBIOS console, 6-8
Displaying system configuration, AlphaBIOS console, 6-8
Displaying the value or setting of an environment variable, 5-26

## E

edit command (SRM), 5-27
EISA configuration screen, AlphaBIOS, 6-17
EISA options
installing, 4-10
slot numbers, 4-11
enable command (RCM), 2-50
Environment variables (SRM), 5-25
auto_action, 2-8
console, 2-5, 2-8, 2-19
displaying the value or setting, 5-26
modifying, 5-26
os_type, 2-8, 2-19
setting, 5-26
summary, 5-40
Ethernet, specifying controller type, 5-46
ew*0_mode environment variable (SRM), 5-46
ew*0_protocols environment variable (SRM), 5-46
examine command (SRM), 5-28, 5-29
Examing data, 5-28
exit command (LFU), 2-27, 2-33, 2-37, 2-38, 2-39

## F

Finding help
AlphaBIOS console, 6-6
SRM console, 5-34
Firmware
AlphaBIOS console, 6-1

RCM, 2-47
SRM console, 5-1
updates, downloading from Internet, A-1
updating, 2-23
updating from AlphaBIOS, 6-24
updating from CD-ROM, 2-24
updating from floppy disk, 2-28, 2-30
updating from network device, 2-34
updating from SRM, 5-22
updating, AlphaBIOS selection, 2-22
updating, SRM command, 2-22
Formatting hard disk, AlphaBIOS, 6-26, 6-28, 6-30

## G

Graphics accelerator, setting location of SYNC signal (SRM), 5-51
Graphics monitor, specifying as console terminal (SRM), 5-45

## H

halt command (SRM), 5-20, 5-21
halt command (RCM), 2-51
hangup command (RCM), 2-50
Hard disk configuration screen, AlphaBIOS, 6-12
Hard disk setup screen, AlphaBIOS, 6-26
Hard disk, AlphaBIOS
creating and deleting partitions, 6-28
error conditions, B-1
formatting a FAT partition, 6-30
partitioning for AlphaBIOS and Windows NT, B-1
setting up, 6-26
Help
AlphaBIOS console, 6-6
AlphaBIOS first-level help screen, 6-6
AlphaBIOS second-level help screen, 6-7
help command (LFU), 2-38, 2-39
help command (SRM), 5-34
help command (RCM), 2-51

## I

Initialization and answer strings
modifying for modem, 2-66
substitutions, 2-67
initialize command (SRM), 5-32, 5-33
Initializing the system, 5-32
Installing Windows NT, 2-20, 6-36
Integrated peripherals screen, AlphaBIOS, 6-22
IOD, 2-8

## K

kbd_hardware_type environment variable (SRM), 5-26, 5-47
Key conventions, AlphaBIOS console, 6-6
Key mapping, AlphaBIOS in serial mode, 7-4
Keyboard
setting language (AlphaBIOS), 6-33
specifying hardware type (SRM), 5-47
specifying layout (SRM), 5-47
kzpsa*_host_id environment variable (SRM), 5-47

## L

language environment variable (SRM), 5-26, 5-47
LEDs, CPU and bridge modules, 3-4
LFU
starting, 2-22, 2-23
starting the utility, 2-22
typical update procedure, 2-23
update command, 2-40
updating firmware from CD-ROM, 2-24

updating firmware from floppy disk, 2-28, 2-30
updating firmware from network device, 2-34
lfu command (LFU), 2-31, 2-33, 2-38, 2-39
lfu command (SRM), 5-22
LFU commands
display, 2-38, 2-39
exit, 2-27, 2-33, 2-37, 2-38, 2-39
help, 2-38, 2-39
lfu, 2-31, 2-33, 2-38, 2-39
list, 2-25, 2-31, 2-33, 2-35, 2-37, 2-38, 2-40
readme, 2-38, 2-40
summary, 2-38
update, 2-27, 2-38, 2-40
verify, 2-38, 2-40
list command (LFU), 2-25, 2-31, 2-35, 2-38, 2-40
Loadable Firmware Update utility. *See* LFU

## M

Maintenance programs, AlphaBIOS, 6-38
man command (SRM), 5-34
Memory configuration screen, AlphaBIOS, 6-21
Memory pair, 1-5, 1-12
installing, 4-8
slot numbers, 4-9
Memory test
determining extent of testing (SRM), 5-48
enabling and disabling settings in AlphaBIOS, 6-35
Memory tests, 2-5
memory_test environment variable (SRM), 5-48
Modem
answer, 2-65
dial-in procedure, 2-44
hangup, 2-65
phases of operation, 2-64
ring detection, 2-65
Modem (RCM), 2-42
Modifying an environment variable (SRM), 5-26
Modules
bridge, 1-5, 1-16
CPU, 1-5, 1-10, 4-6
EISA, 4-10
memory pair, 1-5, 1-12, 4-8
PCI, 4-10
PCI bridge, 1-5, 1-16
PCI motherboard, 1-18
power control, 1-14
processor, 1-5, 1-10, 4-6
server control, 1-20
system bus to PCI bus bridge, 1-5, 1-16
system motherboard, 1-8
more command (SRM), 5-31

## N

Naming conventions (SRM), 5-7
Network protocols, enabling for booting and other functions (SRM), 5-46

## O

OCP, 1-22
ocp_text environment variable (SRM), 5-26, 5-48
OpenVMS
booting, 2-13
installing, 2-16
specifying as default operating system, 5-49
Operating system
booting Digital UNIX, 2-9
booting OpenVMS, 2-13
booting Windows NT, 2-18
installing Digital UNIX, 2-11
installing OpenVMS, 2-16
installing Windows NT, 2-20
specifying default, 5-49
Operating system selection, defined, 6-41

Operating system selections screen, AlphaBIOS, 6-40, 6-44
Operation, system drawer, 2-1
Options
  EISA, 4-10
  PCI, 4-10
os_type environment variable (SRM), 2-8, 2-19, 5-26, 5-49, 6-35

## P

PALcode, 2-8
Parameters, configuring in AlphaBIOS console, 6-32
Password, setting in AlphaBIOS console, 6-35
PCI
  enabling and disabling parity checking on bus (SRM), 5-49
  finding device information, AlphaBIOS, 6-16
  parity checking in AlphaBIOS, 6-35
PCI bridge module, 1-5, 1-16
PCI configuration screen, AlphaBIOS, 6-14
PCI motherboard, 1-18
PCI options
  installing, 4-10
  slot numbers, 4-11
pci_parity environment variable (SRM), 5-49
PCM, 1-14
pk*0_fast environment variable (SRM), 5-50
pk*0_host_id environment variable (SRM), 5-50
pk*0_soft_term environment variable (SRM), 5-51
Power, 3-2
Power button, 2-2
Power control module, 1-14
Power supply, 1-24
  installing, 4-12
Powering up the system drawer, 2-2
poweroff command (RCM), 2-51
poweron command (RCM), 2-52
Power-up display, 2-4
Power-up, specifying console action (SRM), 5-42
prcache command (SRM), 5-16
Primary operating system, AlphaBIOS, 6-42
Processor, enabling and disabling specific CPUs, 5-45
Processor module, 1-5, 1-10
  installing, 4-6
  slot numbers, 4-7

## Q

quit command (RCM), 2-52

## R

RCM, 1-20, 2-41
  command summary, 2-47
  dial-out alerts, 2-57
  entering and leaving command mode, 2-46
  modem usage, 2-42
  resetting to factory defaults, 2-60
  troubleshooting, 2-61
  typical dialout command, 2-57
RCM commands
  ?, 2-51
  alert_clr, 2-49
  alert_dis, 2-49
  alert_ena, 2-49
  disable, 2-50
  enable, 2-50
  halt, 2-51
  hangup, 2-50
  help, 2-51
  poweroff, 2-51
  poweron, 2-52
  quit, 2-52
  reset, 2-53
  setesc, 2-53
  setpass, 2-54
  status, 2-55
Reading a file (SRM), 5-31

readme command (LFU), 2-38, 2-40
Remote console monitor. *See* RCM
reset command (RCM), 2-53
Reset, specifying console action (SRM), 5-42
Running maintenance programs, AlphaBIOS, 6-38
Running utility programs, 6-38

## S

S-cache, 2-5
SCSI
  enabling and disabling terminators (SRM), 5-51
  enabling Fast SCSI (SRM), 5-50
  Setting controller host bus node ID (SRM), 5-50
SCSI configuration screen, AlphaBIOS, 6-18
Serial terminal, specifying as console terminal (SRM), 5-45
Server control module, 1-20
set *envar* command (SRM), 5-26
set host command (SRM), 5-14, 5-15
setescape command (RCM), 2-53
setpass command (RCM), 2-54
Setting an environment variable, 5-26
Setting up the hard disk, AlphaBIOS, 6-26
Setup screen, AlphaBIOS, 6-5
show config command (SRM), 5-4
show cpu command (SRM), 5-5
show device command (SRM), 2-9, 2-13, 5-6
show *envar* command (SRM), 2-13, 5-26
show fru command (SRM), 5-8
show memory command (SRM), 5-9
show network command (SRM), 5-9
show pal command (SRM), 5-10
show power command (SRM), 1-25, 5-10
show version command (SRM), 5-11
Special characters, SRM console, 5-38
SRM console, 2-8, 5-1
  command syntax, 5-37
  device naming conventions, 5-7
  special characters, 5-38
  summary of commands, 5-35
  summary of environment variables, 5-40
  switching from AlphaBIOS, 6-2, 6-35
  switching to AlphaBIOS, 5-2
SRM console commands
  alphabios, 5-2, 5-3
  boot, 5-12
  continue, 5-20, 5-21
  deposit, 5-28, 5-29
  edit, 5-27
  examine, 5-28
  halt, 5-20, 5-21
  help, 5-34
  initialize, 5-32, 5-33
  lfu, 5-22
  man, 5-34
  more, 5-31
  prcache, 5-16
  set *envar*, 5-26
  set host, 5-14, 5-15
  show config, 5-4
  show cpu, 5-5
  show device, 5-6
  show *envar*, 5-26
  show fru, 5-8
  show memory, 5-9
  show network, 5-9
  show pal, 5-10
  show power, 1-25, 5-10
  show version, 5-11
  start, 5-20, 5-21
  stop, 5-21
  summary of, 5-35
  test, 5-18, 5-19
SRM console environment variables
  auto_action, 5-42
  boot_osflags, 5-43, 5-44
  boot_osflags (Digital UNIX), 5-43

boot_osflags (OpenVMS), 5-43
com2_baud, 5-45
commands for displaying the value or setting, 5-26
commands for modifying, 5-26
commands for setting, 5-26
commands for using, 5-25
console, 5-45
cpu_enabled, 5-45
ew*0_mode, 5-46
ew*0_protocols, 5-46
kbd_hardware_type, 5-47
kzpsa*_host_id, 5-47
language, 5-47
memory_test, 5-48
ocp_text, 5-48
os_type, 5-49, 6-35
pci_parity, 5-49
pk*0_fast, 5-50
pk*0_host_id, 5-50
pk*0_soft_term, 5-51
tga_sync_green, 5-51
tt_allow_login, 5-52
SRM firmware, enabling and disabling login to, 5-52
SROM, 2-5
Standard CMOS setup screen, AlphaBIOS, 6-32
Standard I/O, 1-20
start command (SRM), 5-20, 5-21
Starting AlphaBIOS, 6-4
Starting CPUs, 5-20
status command (RCM), 2-55
stop command (SRM), 5-21
Stopping CPUs, 5-20
Syntax, SRM console commands, 5-37
System board configuration screen, AlphaBIOS, 6-10
System bus configuration screen, AlphaBIOS, 6-20
System bus to PCI bus bridge module, 1-5, 1-16
System configuration
displaying in AlphaBIOS console, 6-8
System drawer
architecture, 1-4
characteristics, 1-2
does not power up, 3-2
operation, 2-1
overview, 1-1
parts of, 1-6
powering up, 2-2
power-up display, 2-4
remote operation, 2-41
System failure, specifying console action (SRM), 5-42
System motherboard, 1-8

## T

Test command, 5-18, 5-19
Testing the system (all operating systems), 5-18
tga_sync_green environment variable (SRM), 5-51
Time, setting in AlphaBIOS console, 6-33
Troubleshooting, 3-1
tt_allow_login environment variable (SRM), 5-52

## U

update command (LFU), 2-27, 2-33, 2-37, 2-38, 2-40
Updating firmware
AlphaBIOS console, 6-24
from AlphaBIOS console, 2-22
from SRM console, 2-22
SRM console, 5-22
Utility programs, 6-38
running from graphics monitor, 7-2
running from serial terminal, 7-4

## V

verify command (LFU), 2-38, 2-40
Version selection, Windows NT, 6-40

## W

Windows NT
- auto start, 6-44
- booting, 2-18
- designating primary operating system, 6-42
- installing, 2-20, 6-36
- partitioning hard disk, B-1
- selecting version, 6-40
- specifying as default operating system, 5-49
- starting, 6-44
- starting automatically, 6-33

## X

XSROM, 2-5