AlphaServer 8200/8400

Operations Manual

Order Number: EK–T8030–OP. B01

This manual is intended for the system manager or system operator and covers the basic operation of the DIGITAL AlphaServer 8200 and 8400 systems.

Digital Equipment Corporation
Maynard, Massachusetts

Second Printing, March 1998

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 © 1998 by Digital Equipment Corporation. All rights reserved.

The following are trademarks of Digital Equipment Corporation: AlphaServer, OpenVMS, PowerStorm, StorageWorks, VAX, 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. Microsoft and Windows NT are registered trademarks of Microsoft Corporation. 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.

Avertissement! 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 xiii

Chapter 1 Introduction

1.1 AlphaServer 8200/8400 System Overview 1-2
1.2 Console Firmware and Utilities Overview 1-4
1.3 System Architecture 1-6

Chapter 2 AlphaServer 8200 System

2.1 System Characteristics 2-2
2.2 Sample 8200 System 2-4
2.3 System Front View 2-6
2.4 System Rear View 2-8
2.5 System Components- 2-10
2.5.1 Processor System Unit 2-10
2.5.2 Cabinet Control Logic Panel 2-12
2.5.3 Console Load Device 2-14
2.5.4 Power System 2-16
2.6 Controls and Indicators 2-18
2.6.1 AC Power Circuit Breaker 2-20
2.7 Options 2-22

Chapter 3 AlphaServer 8400 System

3.1 System Characteristics 3-2
3.2 Sample 8400 System 3-4
3.3 System Front View 3-6
3.4 System Rear View 3-8
3.5 System Components 3-10
3.5.1 Console Load Device 3-10
3.5.2 Power System 3-12
3.5.3 TLSB Card Cage 3-14
3.5.4 Control/Status and I/O Connections 3-16
3.5.5 Cooling System 3-18

3.5.6 System Options....3-20
3.6 Controls and Indicators....3-22
3.6.1 Control Panel Keyswitch....3-22
3.6.2 Control Panel Indicator Lights....3-24
3.6.3 Circuit Breaker and AC Power Indicators....3-26

Chapter 4 I/O Subsystems

4.1 I/O Subsystem Overview....4-2
4.2 I/O Port Modules....4-4
4.3 System Configuration Information....4-6
4.4 PCI Adapter....4-16
4.5 KFE70 and KFE72 Adapters....4-18
4.6 EISA Configuration Utility....4-20

Chapter 5 Booting an Operating System

5.1 Preparation....5-2
5.1.1 Set os_type Environment Variable....5-2
5.1.2 Set console Environment Variable....5-3
5.1.3 Set auto_action Environment Variable....5-4
5.2 Selecting a Boot Device....5-6
5.2.1 Show Config and Show Device Commands (for OpenVMS and DIGITAL UNIX Systems....5-8
5.2.2 Boot Environment Variables (for OpenVMS and DIGITAL UNIX Systems)....5-11
5.2.3 Boot Path for Windows NT Systems....5-12
5.3 Booting OpenVMS....5-14
5.4 Booting DIGITAL Unix....5-16
5.5 Booting Windows NT....5-18
5.5.1 Effect of SRM Console auto_action Environment Variable....5-18
5.5.2 Effect of AlphaBIOS Auto Start and Auto Start Count Options....5-20
5.5.3 Effect of AlphaBIOS Password Option....5-22

Chapter 6 System Troubleshooting

6.1 Troubleshooting During Power-Up....6-2
6.2 Troubleshooting During Booting....6-4
6.3 Troubleshooting a PCI Shelf....6-6
6.4 Troubleshooting an XMI Plug-In Unit....6-8
6.5 Troubleshooting a Futurebus+ Plug-In Unit....6-10
6.6 Troubleshooting a BA655 Plug-In Unit....6-12
6.7 Troubleshooting a Battery Plug-In Unit....6-14
6.8 Self-Test Overview....6-16
6.9 Testing Sequence....6-18

6.10 Sample Self-Test Display 6-20
6.11 Self-Test Lines NODE# and TYP 6-22
6.12 Self-Test Lines ST and BPD....6-24
6.13 Self-Test Lines C0, C1, C2,…,Cn 6-26
6.14 Self-Test Lines ILV and MB 6-28
6.15 Self-Test Identification Line....6-30
6.16 Show Commands....6-32
6.16.1 Show Configuration....6-32
6.16.2 Show Network 6-34
6.16.3 Show Device....6-36
6.17 Test Command....6-38
6.17.1 Testing the System....6-40
6.17.2 Testing a Subsystem 6-42
6.17.3 Testing a Module or Devices 6-44
6.18 Error Reports 6-50

Chapter 7 SRM Console Commands

7.1 Overview 7-2
7.2 SRM Command Syntax 7-4
7.3 SRM Console Special Characters 7-6
7.4 SRM Console Environment Variables....7-9
7.5 SRM Console Commands....7-14
7.5.1 AlphaBIOS (for Windows NT Only)....7-14
7.5.2 Boot (for OpenVMS or DIGITAL UNIX Systems Only) 7-15
7.5.3 Building the EEPROM 7-16
7.5.4 Building the Nonvolatile RAM....7-17
7.5.5 Building the SEEPROM....7-18
7.5.6 Clear EEPROM 7-19
7.5.7 Clear <envar> 7-20
7.5.8 Clear Screen 7-21
7.5.9 Continue 7-22
7.5.10 Crash....7-24
7.5.11 Create 7-25
7.5.12 Date 7-26
7.5.13 Deposit 7-27
7.5.14 Examine....7-30
7.5.15 Halt....7-33
7.5.16 Help or Man 7-34
7.5.17 Init 7-36
7.5.18 Prcache 7-37
7.5.19 Run (for OpenVMS or DIGITAL UNIX Systems Only)....7-38
7.5.20 Runecu (for OpenVMS or DIGITAL UNIX Systems Only) 7-41
7.5.21 Set EEPROM....7-43
7.5.22 Set <envar> 7-44

7.5.23 Set Host 7-45
7.5.24 Set Power 7-47
7.5.25 Set SEEPROM 7-48
7.5.26 Show Configuration 7-49
7.5.27 Show CPU 7-51
7.5.28 Show Device 7-52
7.5.29 Show EEPROM 7-53
7.5.30 Show <envar> 7-54
7.5.31 Show Memory 7-55
7.5.32 Show Network 7-56
7.5.33 Show Power 7-57
7.5.34 Show SEEPROM 7-58
7.5.35 Start 7-59
7.5.36 Stop 7-60
7.5.37 Test 7-61
7.5.38 Type 7-63
7.5.39 Vga 7-64
7.5.40 Comment (#) 7-65

Chapter 8 AlphaBIOS Firmware

8.1 Introduction 8-2
8.2 Switching Between Windows NT, AlphaBIOS Setup and the SRM Console 8-4
8.3 Keyboard Conventions and Help 8-8
8.4 Starting AlphaBIOS Setup 8-10
8.5 Displaying the System Configuration 8-12
8.5.1 System Board Configuration 8-14
8.5.2 Hard Disk Configuration 8-16
8.5.3 PCI Configuration 8-19
8.5.4 EISA Configuration 8-21
8.5.5 SCSI Configuration 8-22
8.5.6 Memory Configuration 8-24
8.5.7 Integrated Peripherals 8-25
8.6 Updating Firmware 8-26
8.7 Setting up the Hard Disk 8-28
8.7.1 Manually Creating and Deleting Partitions 8-30
8.7.2 Formatting a FAT Partition 8-32
8.8 Performing Setup Tasks 8-34
8.9 Installing Windows NT 8-38
8.10 Running Utility Programs 8-40
8.11 Selecting the Version of Windows NT 8-42
8.11.1 Designating a Primary Operating System 8-44
8.11.2 Primary Operating System and the Auto Start Option 8-46

Appendix A OpenVMS and DIGITAL UNIX Boot Options

Appendix B Updating Firmware

B.1 Booting LFU with OpenVMS and DIGITAL UNIX Systems B-2
B.2 List B-4
B.3 Update B-6
B.4 Exit B-10
B.5 Display and Verify Commands B-12
B.6 How to Update Corrupted Firmware B-14
B.7 How to Modify Device Attributes B-18

Appendix C Running Configuration Utilities from the SRM Console

C.1 Configuring a RAID Storage Array C-2
C.2 ISP1020 Configuration Utility C-11

Glossary

Index

Examples

4–1 System Self-Test Display 4-6
4–2 Sample Show Configuration Command with a KFE70 4-8
4-3 Sample Show Configuration Command with a KFE72 4-12
4–4 Show Device Command 4-14
5–1 Setting os_type for OpenVMS 5-2
5–2 Setting os_type for DIGITAL UNIX 5-2
5–3 Setting os_type for Windows NT 5-2
5–4 Set Console to Graphics for DIGITAL UNIX Systems 5-3
5–5 Set Console to Both for Windows NT Systems 5-3
5–6 Setting the auto_action Environment Variable 5-4
5–7 Show Configuration and Show Device for OpenVMS and DIGITAL UNIX Systems 5-8
5–8 Viewing and Setting Boot Environment Variables for OpenVMS and DIGITAL UNIX Systems 5-11
5–9 OpenVMS Boot 5-14
5–10 DIGITAL UNIX Boot 5-16
6–1 Sample Self-Test Display, Failing DWLMA Adapter 6-9
6–2 Show Power Command 6-15

6–3 Testing Sequence 6-18
6–4 Self-Test Results 6-20
6–5 Self-Test Results: Node# and TYP 6-22
6–6 Self-Test Results: ST and BPD 6-24
6–7 Self-Test Results: C0, C1, C2,…,Cn 6-26
6–8 Self-Test Results: ILV and MB 6-28
6–9 Self-Test Results: Identification Line 6-30
6–10 Sample System Hardware Configuration 6-32
6–11 Sample Output of Show Network Command 6-34
6–12 Sample Output of Show Device Command 6-36
6–13 Sample Test Commands 6-38
6–14 Sample Test Command, System Test 6-40
6–15 Sample Test Command, I/O Subsystem Test 6-42
6–16 Sample Test Command, I/O Adapter Test 6-44
6–17 Sample Test Command, Memory Module Test 6-45
6–18 Sample Test Command, Testing Devices 6-46
6–19 Testing Network Adapters 6-48
6–20 Sample Summary Error Report 6-50
6–21 Sample Full Error Report 6-52
7–1 AlphaBIOS Command 7-14
7–2 Boot Command 7-15
7–3 Building the EEPROM 7-16
7–4 Building the Nonvolatile RAM 7-17
7–5 Building the SEEPROM 7-18
7–6 Clear EEPROM Command 7-19
7–7 Clear <envar> 7-20
7–8 Clear Screen Command 7-21
7–9 Continue Command 7-22
7–10 Crash Command 7-24
7–11 Create Command 7-25
7–12 Date Command 7-26
7–13 Deposit Command 7-27
7–14 Examine Command 7-30
7–15 Halt Comand 7-33
7–16 Help Command 7-34
7–17 Initialize Command 7-36
7–18 Prcache Command 7-38
7–19 Run Command 7-41
7–20 Runecu Command 7-43
7–21 Set EEPROM Command 7-44
7–22 Set <envar> 7-45
7–23 Set Host Command 7-47
7–24 Set Power Command 7-48
7–25 Set SEEPROM Command 7-49

7–26 Show Configuration Command....7-50
7–27 Show CPU Command....7-51
7–28 Show Device Command....7-52
7–29 Show EEPROM Command....7-53
7–30 Show <envar> Command....7-54
7–31 Show Memory Command....7-55
7–32 Show Network Command....7-56
7–33 Show Power Command....7-57
7–34 Show SEEPROM Command....7-58
7–35 Start Command....7-59
7–36 Stop Command....7-60
7–37 Test Command....7-61
7–38 Type Command....7-63
7–39 Vga Command....7-64
7–40 Comment (#) Command....7-65
B–1 Booting LFU from CD-ROM....B-2
B–2 List Command....B-4
B–3 Update Command....B-6
B–4 Exit Command....B-10
B–5 Display and Verify Commands....B-12
B–6 Updating an “Unknown” Device....B-14
B–7 Modify Command....B-18

Figures

1–1 AlphaServer 8200 and 8400 Systems....1-2
1–2 Accessing Firmware at the Console Device....1-4
1–3 Sample System Architecture....1-6
2–1 Sample System Footprint....2-2
2–2 Sample 8200 System....2-4
2–3 System Front View....2-6
2–4 System Rear View....2-8
2–5 Processor System Unit....2-10
2–6 TLSB Card Cage....2-11
2–7 Cabinet Control Logic Panel....2-12
2–8 Accessing the Console Load Device....2-14
2–9 Power System....2-16
2–10 Control Panel....2-18
2–11 Circuit Breaker....2-20
2–12 System Options....2-22
3–1 Sample System Footprint....3-2
3–2 Sample System....3-4
3–3 System Front View....3-6
3–4 System Rear View....3-8

3–5 Accessing the Console Load Device....3-10
3–6 Power System....3-12
3–7 TLSB Card Cage....3-14
3–8 Control/Status and I/O Connections....3-16
3–9 Cabinet Airflow....3-18
3–10 System Options....3-20
3–11 Control Panel Keyswitch....3-22
3–12 Control Panel Indicator Lights....3-24
3–13 Circuit Breaker and AC Power Indicators....3-26
4–1 I/O Subsystem....4-2
4–2 I/O Port Modules....4-4
4–3 Hose Numbering Scheme for KFTIA and KFTHA....4-7
4–4 PCI/EISA Slot Configuration with the KFE70....4-16
4–5 PCI Slot Configuration with the KFE72....4-17
4–6 KFE70 Modules....4-18
4–7 KFE72 Modules....4-19
5–1 Boot Devices....5-6
5–2 AlphaBIOS Boot Screen with Auto Start Enabled....5-12
5–3 AlphaBIOS Boot Screen with Auto Start Disabled....5-18
5–4 AlphaBIOS Boot Screen with Auto Start Enabled....5-20
5–5 AlphaBIOS Boot Screen with Startup Password Selected....5-22
6–1 Power-Up Troubleshooting Flowchart....6-2
6–2 Power-Up Troubleshooting Steps....6-3
6–3 Booting Troubleshooting Flowchart (OpenVMS and DIGITAL UNIX)....6-4
6–4 Troubleshooting Steps During Booting (OpenVMS and DIGITAL UNIX)....6-5
6–5 PCI Shelf in a BA655 PIU....6-6
6–6 PCI Shelf in an 8200....6-7
6–7 Troubleshooting Steps in PCI Shelf....6-7
6–8 Troubleshooting an XMI Plug-In Unit....6-8
6–9 FBUS+ PIU Troubleshooting – 48V LED Off....6-10
6–10 FBUS+ PIU Troubleshooting – MOD OK LED Off....6-11
6–11 SCSI Indicator LEDs....6-12
6–12 Battery Plug-In Unit....6-14
6–13 Determining Self-Test Results....6-16
8–1 AlphaBIOS Functions....8-2
8–2 8200 Control Panel....8-4
8–3 8400 Control Panel....8-5
8–4 Typical First-Level Help Screen....8-8
8–5 Second-Level Help Screen....8-9
8–6 Boot Screen....8-10
8–7 AlphaBIOS Setup Screen....8-11
8–8 Display System Configuration Screen....8-12
8–9 System Board Configuration....8-14
8–10 Hard Disk Configuration....8-16

8–11 PCI Configuration....8-18
8–12 Advanced PCI Information....8-20
8–13 EISA Configuration....8-21
8–14 SCSI Configuration....8-22
8–15 Memory Configuration....8-24
8–16 Integrated Peripherals....8-25
8–17 Updating Firmware....8-26
8–18 LFU Load Screen....8-27
8–19 Hard Disk Setup Screen....8-28
8–20 Create New Partition Dialog Box....8-30
8–21 Delete Partition Dialog Box....8-31
8–22 Formatting a FAT Partition....8-32
8–23 Standard Formatting....8-33
8–24 Standard CMOS Setup Screen....8-34
8–25 Advanced CMOS Setup Screen....8-36
8–26 Installing Windows NT....8-38
8–27 Utilities Selection....8-40
8–28 Run Maintenance Program Dialog Box....8-41
8–29 Operating System Selections....8-42
8–30 Primary Operating System....8-44
8–31 Operating System Selection Setup....8-46

Tables

1 AlphaServer 8200 and 8400 Documentation....xii
2–1 Electrical Characteristics....2-3
2–2 Environmental Characteristics....2-3
2–3 Control/Status and I/O Connections....2-13
2–4 Control Panel Pushbuttons....2-18
2–5 Control Panel Indicator Lights....2-19
3–1 Electrical Characteristics....3-3
3–2 Environmental Characteristics....3-3
3–3 Keyswitch Positions....3-23
3–4 Control Panel Indicator Lights....3-25
4–1 EISA Bus Configuration Procedure Summary....4-18
5–1 Boot Devices....5-3
6–1 SCSI Disk Drive....6-13
6–2 SCSI Power Supply....6-13
6–3 System Configuration for Example 6-4....6-21
6–4 I/O Subsystem Configuration for Example 6–4....6-21
6–5 Test Command Options....6-39
6–6 Test Command Environment Variables....6-39
7–1 SRM Console Command Language Syntax....7-4
7–2 SRM Console Special Characters....7-6

7–3 Environment Variables 7-10
7–4 Deposit Command Options 7-28
7–5 Device Name and Address Space Options 7-29
7–6 Examine Command Options 7-31
7–7 Device Name and Address Space Options 7-32
7–8 Test Command Options 7-59
A–1 DIGITAL UNIX Boot Options A-1
A–2 OpenVMS Alpha Boot Options A-2
C–1 Number of Drives You Can Use in a Drive Group for Each RAID Level ... C-8
C–2 How the Capacity of Each Drive Affects the Capacity of the Drive Group C-8
C–3 Host Adapter Parameters C-13

Preface

Intended Audience

This manual is written for the system manager or system operator who has training in systems management and is running a DIGITAL AlphaServer 8200 or 8400 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 eight chapters and three appendixes, as follows:

- **Chapter 1, Introduction,** provides a brief overview of the AlphaServer 8200 and AlphaServer 8400 hardware, firmware, and system architecture.
- **Chapter 2, AlphaServer 8200 System,** and **Chapter 3, AlphaServer 8400 System,** give a basic introduction to your system and its parts.
- **Chapter 4, I/O Subsystems,** describes the AlphaServer 8200 and AlphaServer 8400 systems' I/O design.
- **Chapter 5, Booting an Operating System,** tells how to start running the Windows NT, OpenVMS, and DIGITAL UNIX operating systems.
- **Chapter 6, System Troubleshooting,** provides basic troubleshooting procedures.
- **Chapter 7, SRM Console Commands,** lists the SRM console commands with an example of each command.
- **Chapter 8, AlphaBIOS Firmware**, describes the AlphaBIOS menu selections and boot screen.
- **Appendix A, OpenVMS and DIGITAL UNIX Boot Options**, lists the options used with the boot command for OpenVMS and DIGITAL UNIX to control various phases of booting.
- **Appendix B, Updating Firmware,** explains how to run the Loadable Firmware Update (LFU) utility.

- **Appendix C, Running Configuration Utilities from the SRM Console,** explains how to run the configuration utilities required when installing some options.

Documentation Titles

Table 1 lists the books in the AlphaServer 8200 and AlphaServer 8400 documentation set.

Table 1 AlphaServer 8200 and 8400 Documentation

Title	Order Number
Hardware User Information and Installation	
Operations Manual	EK–T8030–OP
Site Preparation Guide	EK–T8030–SP
AlphaServer 8200 Installation Guide	EK–T8230–IN
AlphaServer 8400 Installation Guide	EK–T8430–IN
KFE72 Installation Guide	EK–KFE72–IN
AlphaServer 8200/8400 Windows NT Administrator's Guide and Release Notes	EK–T8WNT–RN
Service Information Kit	QZ–00RAC–GC
Service Manual (hard copy)	EK–T8030–SV
Service Manual (diskette)	AK–QKNFB–CA AK–QUW7B–CA AK–QUW6B–CA

Table 1 AlphaServer 8200 and 8400 Documentation (Continued)

Title	Order Number
Reference Manuals	
System Technical Manual	EK–T8030–TM
System Technical Manual Supplement: CPU	EK–T8030–TS
System Technical Manual Supplement: Memory	EK–MS7CC–TS
DWLPA/DWLPB PCI Adapter Technical Manual	EK–DWLPA–TM
Upgrade Manuals for Both Systems	
KN7CC CPU Module Installation Card	EK–KN7CC–IN
KN7CD CPU Module Installation Card	EK–KN7CD–IN
KN7CE CPU Module Installation Card	EK–KN7CE–IN
KN7CF CPU Module Installation Card	EK–KN7CF–IN
MS7CC Memory Installation Card	EK–MS7CC–IN
KFTHA System I/O Module Installation Guide	EK–KFTHA–IN
KFTIA Integrated I/O Module Installation Guide	EK–KFTIA–IN
Upgrade Manuals: 8400 System Only	
AlphaServer 8400 Upgrade Manual	EK–T8430–UI
BA654 DSSI Disk PIU Installation Guide	EK–BA654–IN
BA655 SCSI Disk and Tape PIU Installation Guide	EK–BA655–IN
DWLAA Futurebus+ PIU Installation Guide	EK–DWLAA–IN
DWLMA XMI PIU Installation Guide	EK–DWLMA–IN
DWLPA/DWLPB PCI PIU Installation Guide	EK–DWL84–IN
H7237 Battery PIU Installation Guide	EK–H7237–IN
H7263 Power Regulator Installation Card	EK–H7263–IN
KFMSB Adapter Installation Guide	EK–KFMSB–IN
KZMSA Adapter Installation Guide	EK–KXMSX–IN
RRDCD Installation Guide	EK–RRDRX–IN

Table 1 AlphaServer 8200 and 8400 Documentation (Continued)

Title	Order Number
Upgrade Manuals: 8200 System Only	
DWLPA/DWLPB PCI Shelf Installation Guide	EK–DWL82–IN
H7266 Power Regulator Installation Card	EK–H7266–IN
H7267 Battery Backup Installation Card	EK–H7267–IN

Chapter 1

Introduction

The AlphaServer 8200 and 8400 systems are high-performance, symmetric multiprocessing systems that are suitable for office and datacenter environments. They offer access to multiple high-bandwidth I/O buses, very large memory capacities, up to 14 high-performance Alpha CPUs, and many other features normally associated with mainframe systems.

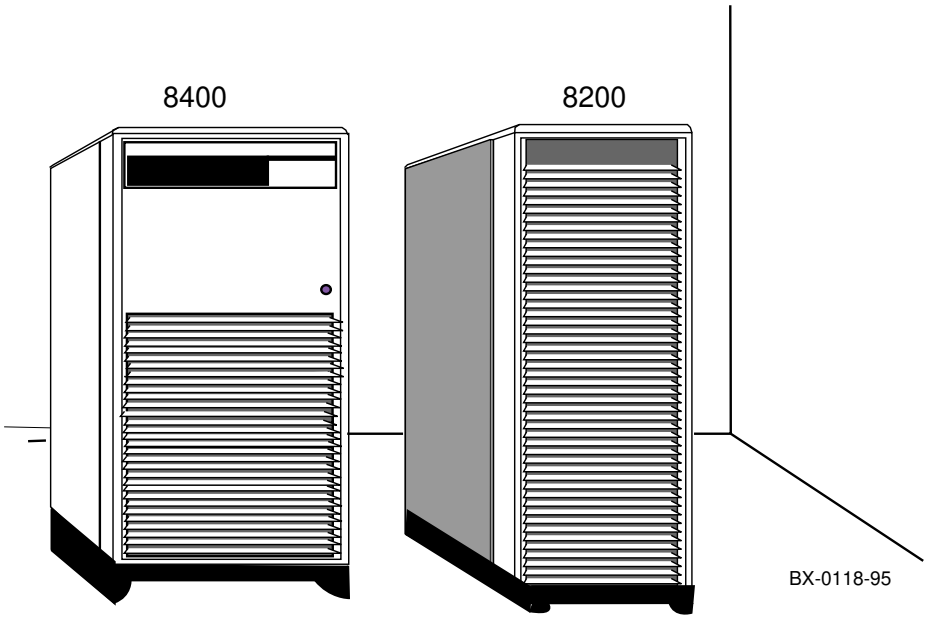
This chapter introduces the AlphaServer 8200 and AlphaServer 8400 systems. There are three sections:

- AlphaServer 8200/8400 System Overview
- Firmware and Utilities Overview
- System Architecture

1.1 AlphaServer 8200/8400 System Overview

The AlphaServer 8200 and 8400 systems are separate, but related, systems that use the same system bus, the TLSB. The processor, memory, and I/O adapter units that can be configured on this bus are also the same. The cabinets, and some of their components, vary.

Figure 1-1 AlphaServer 8200 and 8400 Systems



AlphaServer 8200 System

The AlphaServer 8200 system main cabinet contains the processor system unit (PSU) including a five-slot card cage, power regulators, and space for PCI I/O shelves or StorageWorks shelves. The 8200 system can have up to two expander cabinets, containing additional PCI I/O shelves and StorageWorks shelves.

AlphaServer 8400 System

The AlphaServer 8400 system main cabinet contains the nine-slot TLSB card cage with processor, memory, and I/O modules, power regulators, and one or more plug-in units for I/O, disks, and batteries. The 8400 system can have up to two expander cabinets and additional plug-in units for I/O, disks, and batteries. The 8400 system can also have up to two battery cabinets to provide battery backup. Chapter 3 covers the AlphaServer 8400 system.

Chapter 4 describes the I/O subsystem for both 8200 and 8400 systems. Booting an operating system is discussed in Chapter 5, basic troubleshooting in Chapter 6, SRM console commands in Chapter 7, and AlphaBIOS firmware operations in Chapter 8.

AlphaServer 8200/8400 Options

The *DIGITAL Systems and Options Catalog* describes all options for AlphaServer 8200 and AlphaServer 8400 systems. In addition, DIGITAL maintains a list of the latest supported options on the Internet, which you can access as follows:

Using ftp, copy the file:

ftp.digital.com/pub/Digital/Alpha/systems/as8400/docs/8400-options.txt

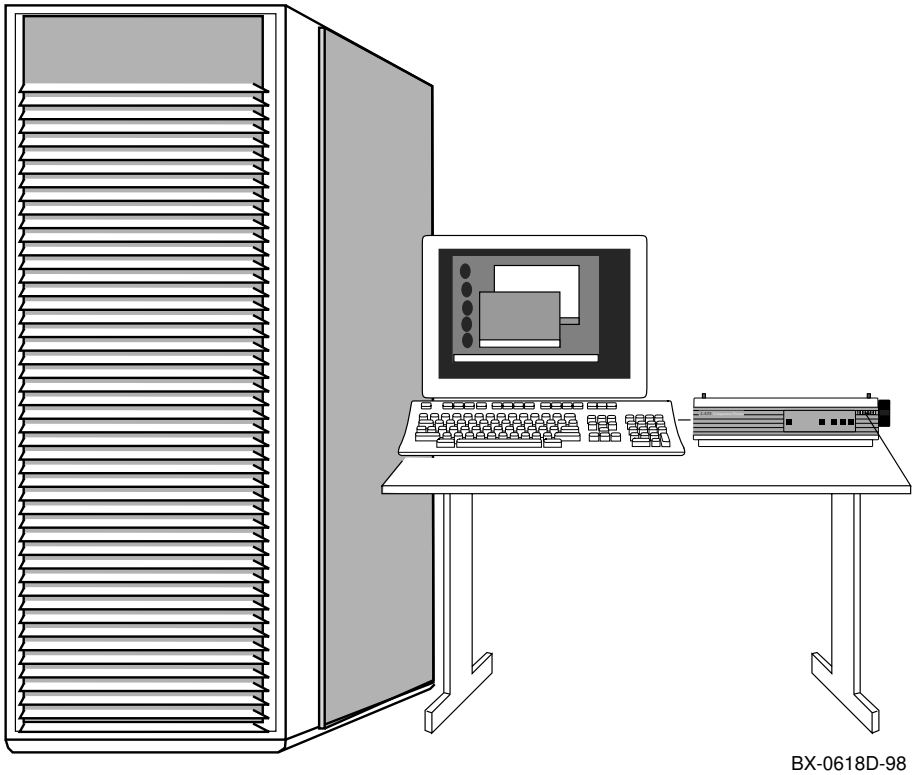
Using a Web browser, follow links from the URL:

http://www.digital.com/info/alphaserver/products.html

1.2 Firmware and Utilities Overview

Firmware residing in flash ROM on CPU and other modules in the system provides commands for booting the operating system, testing devices and I/O adapters, and other tasks useful in operating and maintaining a running system. You type the commands or select from menus at the console device.

Figure 1-2 Accessing Firmware at the Console Device



SRM Console

The SRM console firmware is the first to be executed after system self-test when the system is powered up. If you have requested automatic boot (see Chapter 5), the operating system is booted automatically. Otherwise, the system halts at the SRM console prompt. The SRM console provides commands for booting the OpenVMS and DIGITAL UNIX operating systems, and for testing adapters and I/O devices on all operating systems. The SRM console command **alphabios** transfers control to AlphaBIOS firmware to boot Windows NT.

You can update firmware from the CD-ROM disk using either SRM console commands or AlphaBIOS Setup menus. The SRM console firmware resides in flash ROM on the CPU modules.

AlphaBIOS Firmware

The AlphaBIOS firmware allows you to boot the Windows NT operating system, view and change configuration information, and perform maintenance tasks in the Windows NT environment. You can also update firmware, including the SRM console, AlphaBIOS firmware, and I/O device firmware. The AlphaBIOS firmware is on the same CD-ROM as the SRM console firmware, which is updated periodically. When loaded, the firmware resides in flash ROM in the CPU modules.

LFU (Loadable Firmware Update Utility)

Boot this utility (with the SRM **boot** command, or, for Windows NT systems, from the **Updating AlphaBIOS** menu item) whenever you need to update the SRM console firmware, the AlphaBIOS firmware, or I/O device firmware. The CD is updated periodically.

SCSI Configuration Utilities

The SRM console **run** command lets you run four SCSI-related utilities:

- RCU—RAID Configuration Utility
- SWXCRFW—Updates firmware on the RAID controller
- EEROMCFG—ISP1020 EEPROM configuration utility
- UTIL_CLI—KZPSA configuration utility

The first two utilities are on a floppy diskette; the latter two, on a CD-ROM.

EISA Configuration Utility

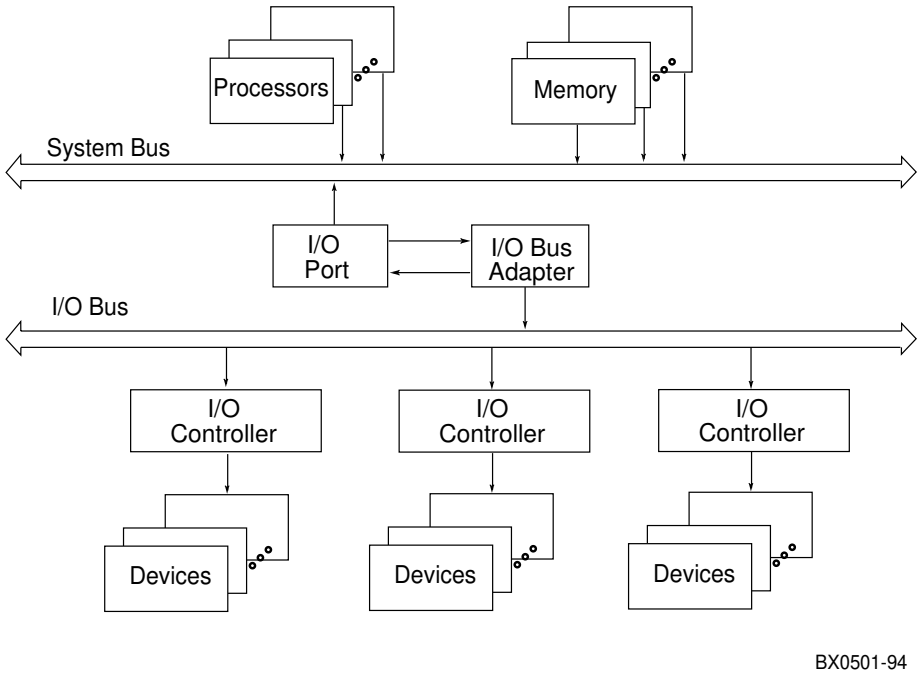
The EISA Configuration Utility (ECU) can be run from:

- The SRM command **runecu,** to configure EISA options in the DWLPA/DWLPB card cage when either the KFE70 or KFE72 adapter option is installed. This command can be used with the OpenVMS or DIGITAL UNIX operating systems.
- One of the AlphaBIOS “Utilities” menu items, used as part of the KFE72 installation procedure for Windows NT systems.

1.3 System Architecture

The high-speed TLSB system bus is used to interconnect processors, memory modules, and I/O port modules.

Figure 1-3 Sample System Architecture



The TLSB bus is a synchronous bus (with a 256-bit data bus and a 40-bit command/address bus) that interconnects processors, memory modules, and the I/O port. The I/O port (KFTHA or KFTIA) module connects the TLSB bus to I/O buses through separate I/O adapter modules.

The TLSB bus uses the concept of a **node**. The TLSB bus has three types of nodes: processors, memories, and I/O port controllers.

A **processor node** is a single module. It consists of one or two scalar processors, the shared TLSB bus interface, separate cache, and support logic.

In a multiprocessing system, one processor becomes the boot processor during power-up, and that boot processor loads the operating system and handles communication with the operator console. The other processors become secondary processors and receive system information from the boot processor. The AlphaServer 8200 can have up to three processor modules for a total of six CPUs. The AlphaServer 8400 can have up to seven processor modules for a total of 14 CPUs.

A **memory node** is one memory module. Memory is a global resource equally accessible by all processors on the TLSB. Memory modules can have 128, 256, or 512 Mbytes or 1, 2, or 4 Gbytes of memory with ECC and associated control logic. The memories are automatically interleaved when the system is configured with multiple memory banks. The 8200 system supports up to three memory modules; the 8400 supports up to seven.

The **I/O port module** (KFTHA) or integrated I/O module (KFTIA) provides the interface between the TLSB and the I/O subsystem. The KFTHA provides connections for up to four I/O buses using cables called hoses. For OpenVMS and DIGITAL UNIX systems, these buses include the PCI/EISA, Futurebus+, or XMI buses. For Windows NT, only PCI buses are supported. The KFTIA provides a connection to one I/O subsystem, with the same possibilities for the operating system as were noted for a KFTHA. In an 8200, only PCI buses are supported, regardless of operating system.

In Figure 1–3, the I/O bus adapter can be the DWLPA/DWLPB module for the PCI, the DWLAA module for the Futurebus+, and the DWLMA module for the XMI.

NOTE: The DWLPA is not supported with the Windows NT operating system; you must have a DWLPB in this case.

The PCI I/O bus adapter module connects to various interconnects such as SCSI, FDDI, Ethernet, NVRAM, and, with OpenVMS and DIGITAL UNIX systems, EISA bus interfaces.

The Futurebus+ I/O bus adapter module connects to various interconnects such as SCSI and FDDI.

The XMI I/O bus adapter module connects to various interconnects such as CI, SDI/STI, SCSI, FDDI, and Ethernet..

Chapter 2

AlphaServer 8200 System

The DIGITAL AlphaServer 8200 system, designed for use in an office environment, can support many users in a time-sharing environment. The 8200 system:

- Supports the full range of system applications of OpenVMS, DIGITAL UNIX, and Windows NT operating systems
- Allows for expansion of processors, memory, and I/O
- Uses a high-speed system interconnect bus (TLSB bus), which has a peak bandwidth of 2.4 Gbytes/sec.
- Supports up to 12 Gbytes of physical memory
- Provides optional self-contained uninterruptible power system (UPS) capability that supports the system in case of power failure
- Performs automatic self-test on power-up, reset, reboot, or system initialization
- Operates as a standalone system, a member of a cluster, or as a boot node of a local area cluster

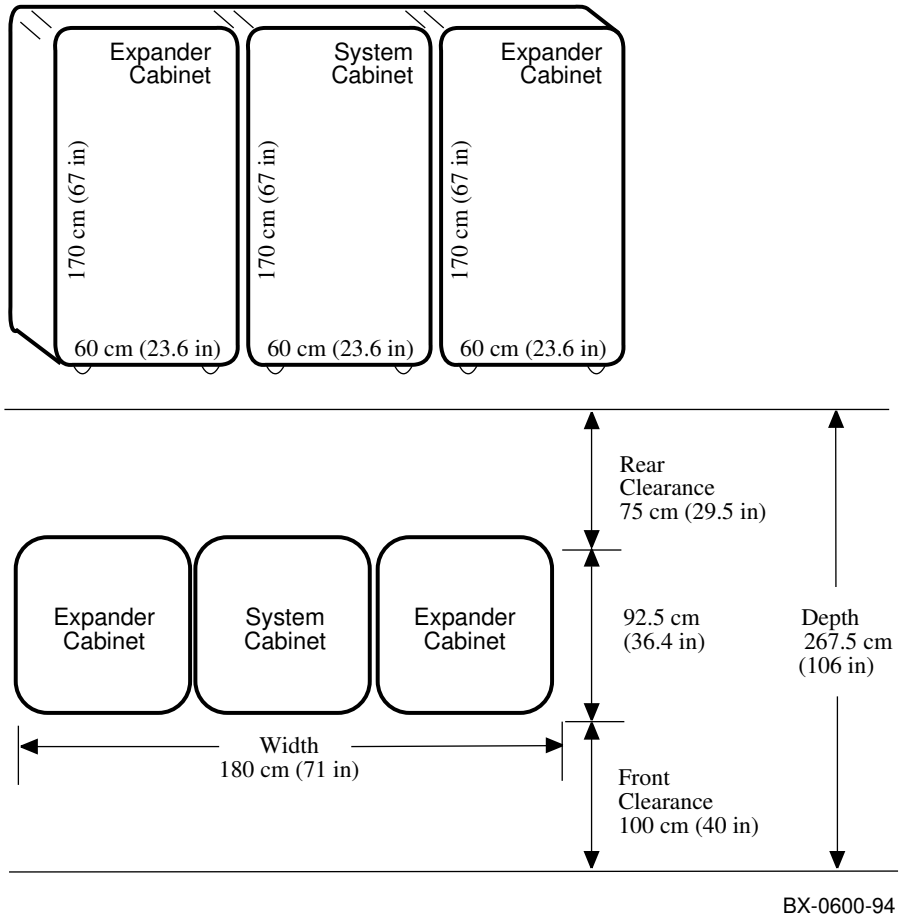
This chapter describes the system package and the location of components in the cabinet. Sections include:

- System Characteristics
- Sample 8200 System
- System Front View
- System Rear View
- System Components (Processor System Unit, Cabinet Control Logic Panel, Console Load Device, and Power System)
- Controls and Indicators
  - Control Panel
  - AC Power Circuit Breaker
- Options

2.1 System Characteristics

Figure 2-1 shows the cabinet dimensions and the required clearance space. The tables list the electrical and environmental characteristics.

Figure 2-1 Sample System Footprint



The values in Table 2–1 and Table 2–2 apply to the system cabinet only. The values are configuration dependent.

Table 2-1 Electrical Characteristics

Electrical	Specification
Single-phase AC input voltage (nominal)	202–240 (208) – North America 202–240 (230) – Europe/AP 202–240 (202) – Japan
Nominal frequency	50–60 Hz
AC current (nominal)	16 A (202 V)
AC current (maximum)	30 A – North America 32 A – Europe/AP 30 A – Japan
AC power consumption (maximum)	2.6 KW

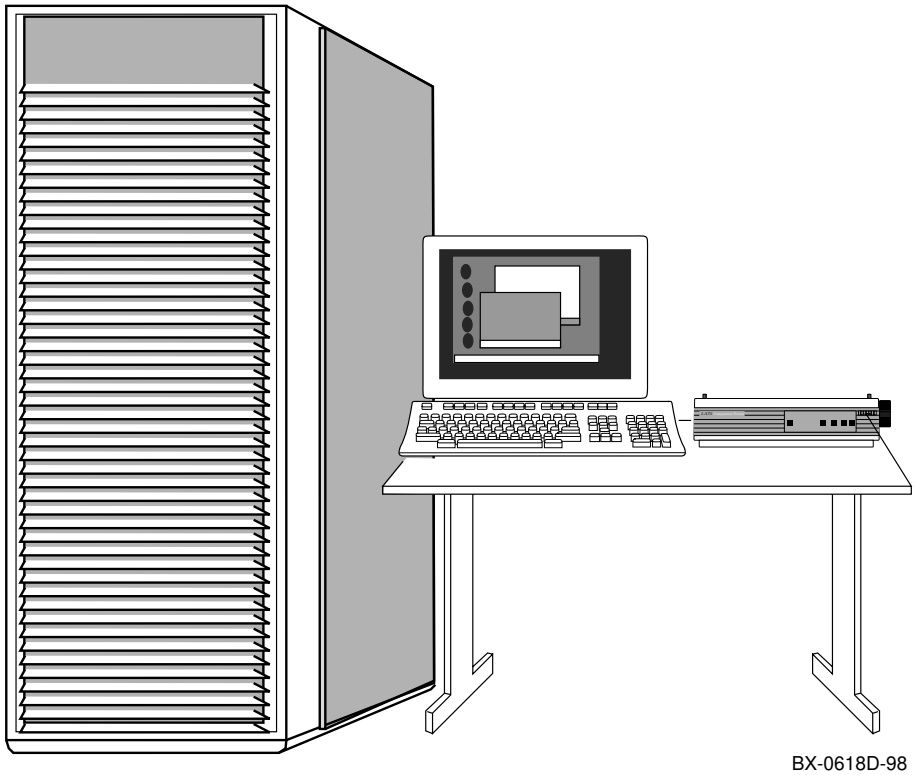
Table 2-2 Environmental Characteristics

Environmental	Operating	Storage
Heat dissipation (maximum)	9,100 Btu/hr	
Temperature	10°C–35°C (50°F–95°F)	-40°C–66°C (-40°F to 151°F)
Relative humidity	10–90%	10–95%
Altitude	0–2.4 km (0–8000 ft)	0–9.1 km (0–30,000 ft)

2.2 Sample 8200 System

Figure 2-2 shows a sample system. The system includes a console terminal and printer, an accessories kit, and a documentation set, which includes this manual. The system can have up to two optional expander cabinets, additional disk drives, and optional battery backup.

Figure 2-2 Sample 8200 System



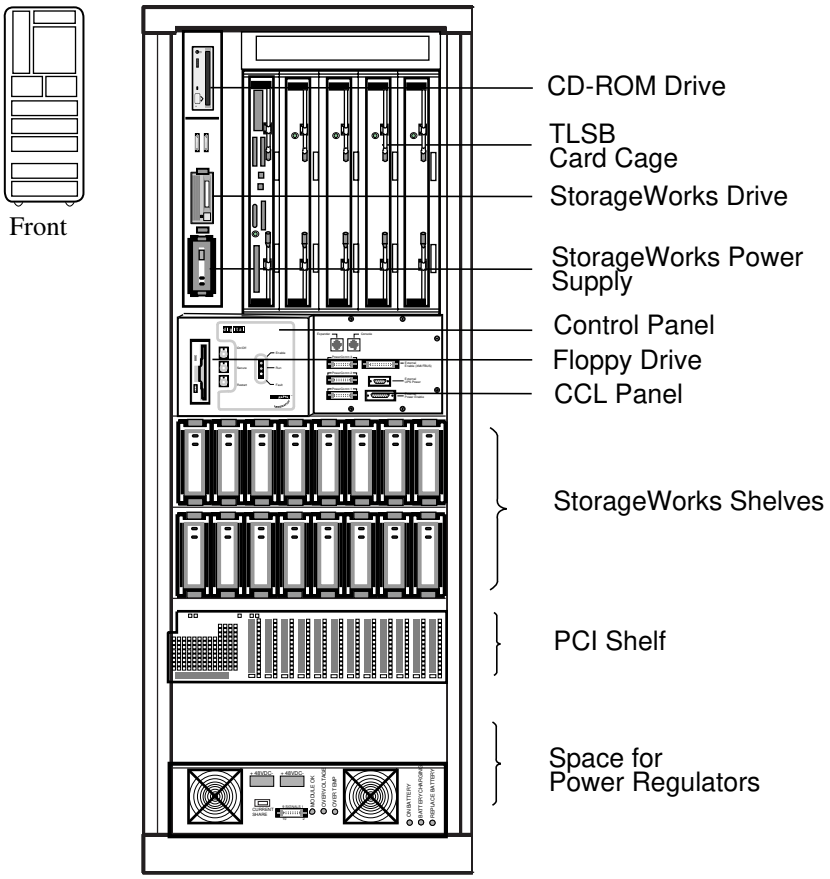
Your DIGITAL customer service engineer has installed your system and verified that it is running properly. Before you turn on the system, familiarize yourself with its components:

- The **system cabinet** houses the power system (with optional battery backup) and the processor system unit (PSU) which contains a storage drawer, the TLSB card cage, control panel, the cabinet control logic panel, and a CD-ROM drive. Optional hardware includes StorageWorks shelves and PCI shelves.
- The **console load device** is used for installing operating systems and software.
- The **console device** is used for booting and for system management operations. The console device must include a serial console monitor, and, for Windows NT systems, must also include a graphics monitor (shown in Figure 2-2).
- The **console printer** provides a hardcopy record of system operations.
- **Optional I/O components** include PCI shelves and StorageWorks shelves. These shelves are installed in the system or expander cabinets to provide space for I/O and disk options.
- **Optional expander cabinets** provide additional space for PCI I/O devices and disk drives.
- A system documentation kit.

2.3 System Front View

With the front door open, you can see the control panel, the TLSB card cage, blower, PCI shelves, StorageWorks shelves, and power regulators.

Figure 2-3 System Front View



BX-0604B-97

The following components are visible from the inside front of the cabinet:

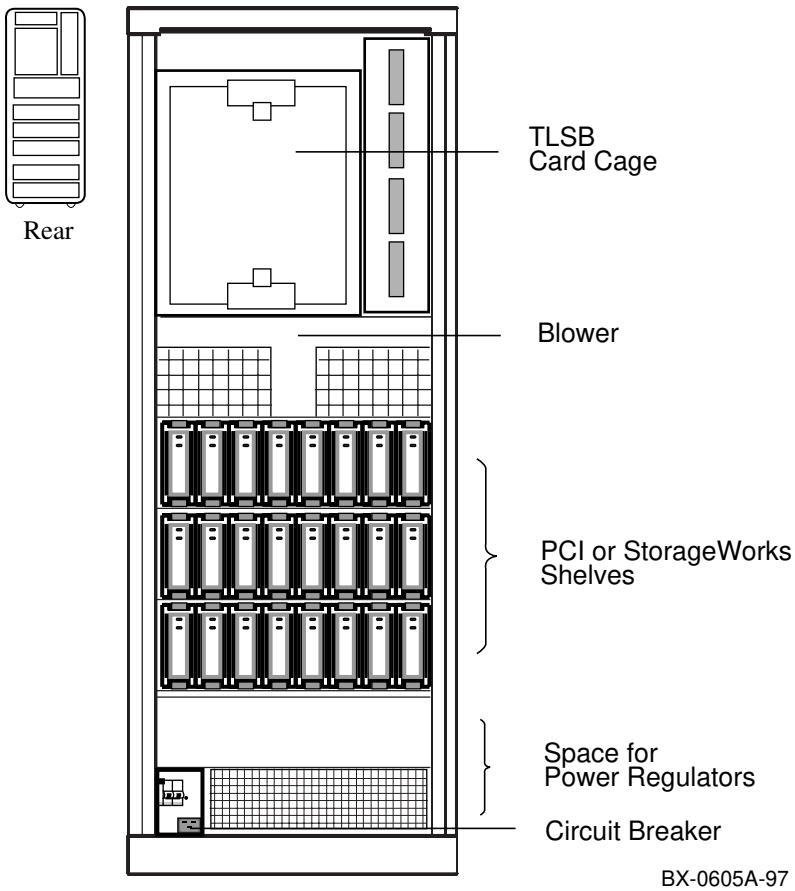
- TLSB card cage
- CD-ROM drive
- Floppy drive
- Control panel
- Cabinet control logic (CCL) panel
- PCI or StorageWorks shelves
- Power regulators

NOTE: Four optional storage devices installed in the processor system unit are not visible. You access these devices from the front of the cabinet by sliding out the removable storage drawer from the processor system unit.

2.4 System Rear View

With the rear door open, DIGITAL customer service engineers can access the circuit breaker and AC power cord.

Figure 2-4 System Rear View



The following components are visible from the inside rear of the cabinet:

- TLSB card cage
- Blower
- PCI or StorageWorks shelves
- Power regulators
- Circuit breaker

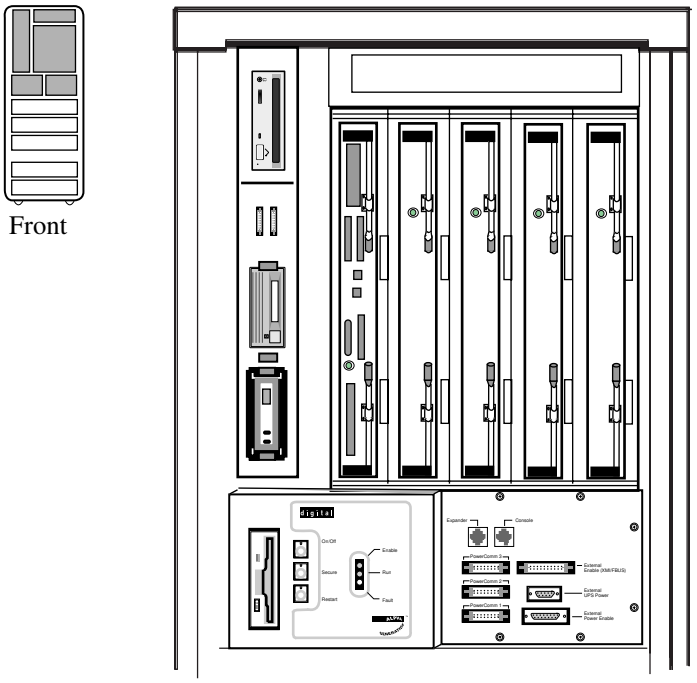
Optional components visible from the inside rear (and front) of the cabinet include PCI shelves, StorageWorks shelves, and an additional power regulator.

2.5 System Components

2.5.1 Processor System Unit

The processor system unit (PSU) contains the 5-slot TLSB card cage and blower, a storage drawer housing integrated I/O devices, an optional floppy drive, a cabinet control logic (CCL) panel, and the control panel.

Figure 2-5 Processor System Unit



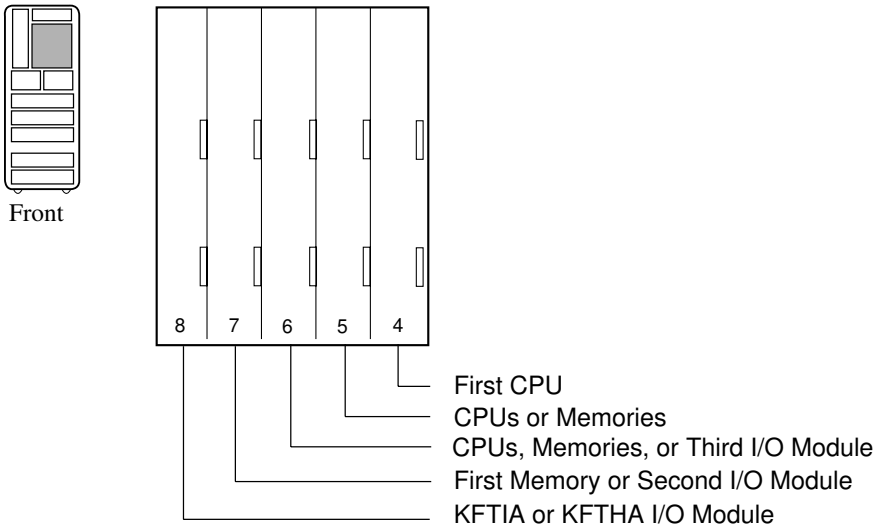
BX-0606-94

The PSU is located in the upper half of the system cabinet, as viewed from the front. The PSU storage drawer can house up to six optional SCSI devices, including one 5.25-inch removable media device and five 3.5-inch devices. The 5.25-inch device and one 3.5-inch device are accessible from the front of the cabinet; the other four 3.5-inch devices are accessible from the rear of the cabinet.

The TLSB card cage slots are numbered 4 through 8 from right to left in the front of the cabinet. See Figure 2-6. The card cage contains one KFTIA or KFTHA module, one CPU module, and one memory module as a minimum configuration. A KFTIA or KFTHA module is always installed in slot 8.

The blower cools the card cage.

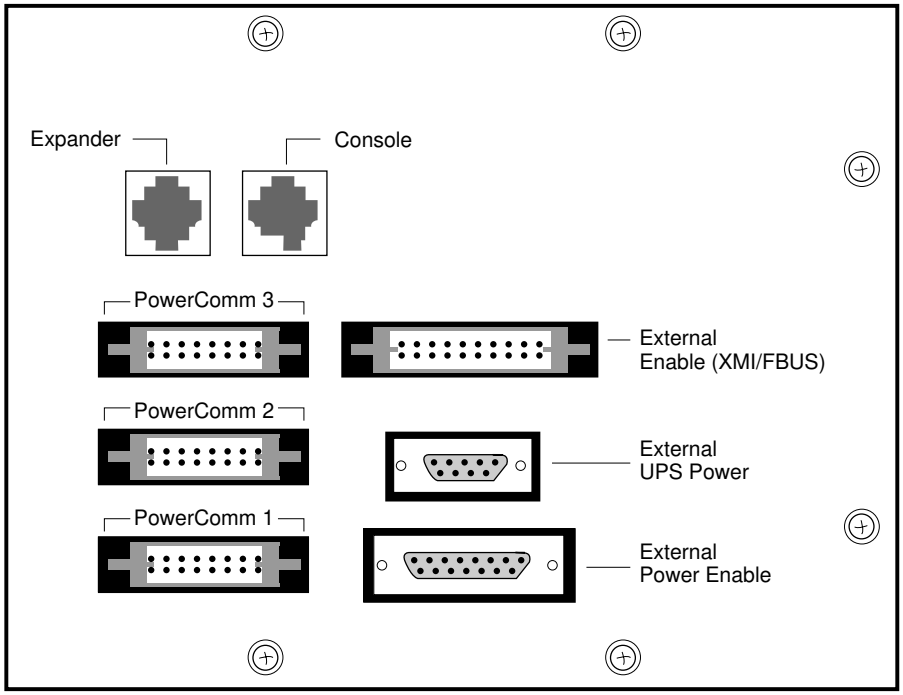
Figure 2-6 TLSB Card Cage



2.5.2 Cabinet Control Logic Panel

Console terminal I/O and expander cabinet remote power control/status connections are located on the cabinet control logic (CCL) panel to the right of the control panel. See Table 2-3 for a list of the other connections shown in Figure 2-7.

Figure 2-7 Cabinet Control Logic Panel



BX-0000-94

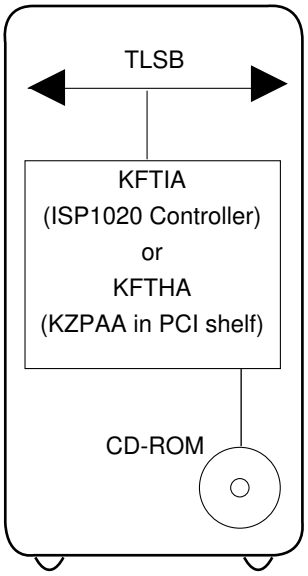
Table 2-3 Control/Status and I/O Connections

Connector Name	Function
Console	Serial console device connection for OpenVMS, DIGITAL UNIX, and Windows NT systems. (For Windows NT systems, the graphics monitor is connected through the KFE72 adapter installed in the primary DWLPB adapter (connected to hose 1 on the KFTHA or KFTIA in slot 8 of the TLSB).
Expander	Expander cabinet power supply control connection. NOTE: The expander cabinet connector is not intended to be connected to a public telecommunications network.
Power Comm3	Reserved for future use.
Power Comm2	Power supply 2 signal and control connection.
Power Comm1	Power supply 1 signal and control connection.
External Enable XMI /FBUS+	Reserved for future use.
External UPS Power	Battery backup option connection.
External Power Enable	Enables power to PCI and StorageWorks shelves.

2.5.3 Console Load Device

The CD-ROM drive is the in-cabinet console load device.

Figure 2-8 Accessing the Console Load Device



BX-0601A-97

The console load device is used for:

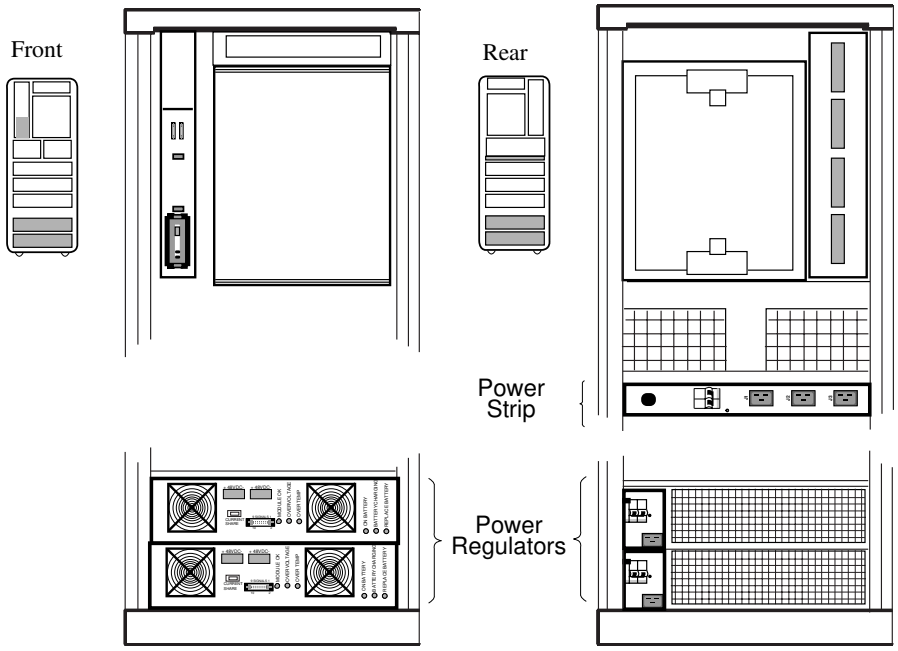
- Installing or updating firmware or software
- Loading a backup utility program
- Interchanging user data
- Updating module firmware

The CD-ROM drive is the console load device. It is installed in the system cabinet and is used to access software and online documentation. Access to the CD-ROM is provided directly through a KFTIA module or through the PCI subsystem through a KFTHA module.

2.5.4 Power System

The power system consists of one or two power regulators (with optional battery backup), a cabinet control logic (CCL) module, and power distribution and signal interconnect cables. The AC circuit breaker controls power to the entire system.

Figure 2-9 Power System



BX-0603-94

The power regulator is located in the lower third of the cabinet. The CCL panel is located in the processor system unit (PSU), next to the control panel.

The system can have up to two power regulators. In this configuration an optional power strip is installed at the rear of the cabinet so that only one AC input connection is required. In a dual power supply system the regulators are used in parallel, one for the required load plus an additional power regulator for backup in case of failure.

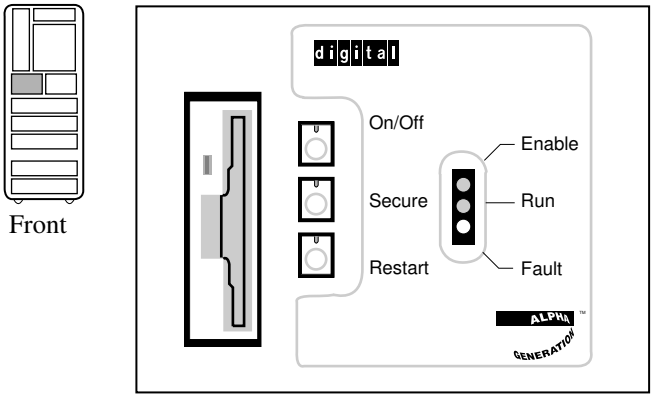
Each power regulator has an AC input assembly, a 48 VDC power regulator, two cooling fans, indicator lights, and optional battery backup (charger module and battery packs for UPS operation).

Each power regulator has a circuit breaker; access is from the rear of the cabinet.

2.6 Controls and Indicators

This section describes the control panel and the AC power circuit breaker.

Figure 2-10 Control Panel



BX-0607-98

Table 2-4 Control Panel Pushbuttons

Push-button	Position	Effect
On/Off	In	Supplies power to the PSU. When this button is pressed and the Secure button is not pressed, OpenVMS or DIGITAL UNIX users can interrupt operating system program execution and enter console mode by typing Ctrl/P at the console device.
	Out	Removes 48 VDC power from the system. This position is useful to field service when they wish to power down the system in an orderly way, prior to switching the system off completely while replacing or installing a new piece of hardware.

Table 2-4 Control Panel Pushbuttons (Continued)

Push-button	Position	Effect
Secure	In	Prevents input from the console device.
		On DIGITAL UNIX or OpenVMS systems, used to protect inadvertent or inadvisable entry into SRM console mode by typing Ctrl/P at the console device. For example, you might push this button in when a critical program is running, in case someone might unknowingly try to enter SRM console mode to load new console code or otherwise use the console device.
		For Windows NT systems, the Secure pushbutton is used in controlling transitions from the Windows NT operating system to the SRM console, as described in Section 8.2.
	Out	Allows input from the console device. See above.
Restart	In	Momentary switch used to reinitialize the system. Registers are reinitialized and system self-test starts running. You can thus restart the system without removing power from the system, which can cause wear and tear on the machine.

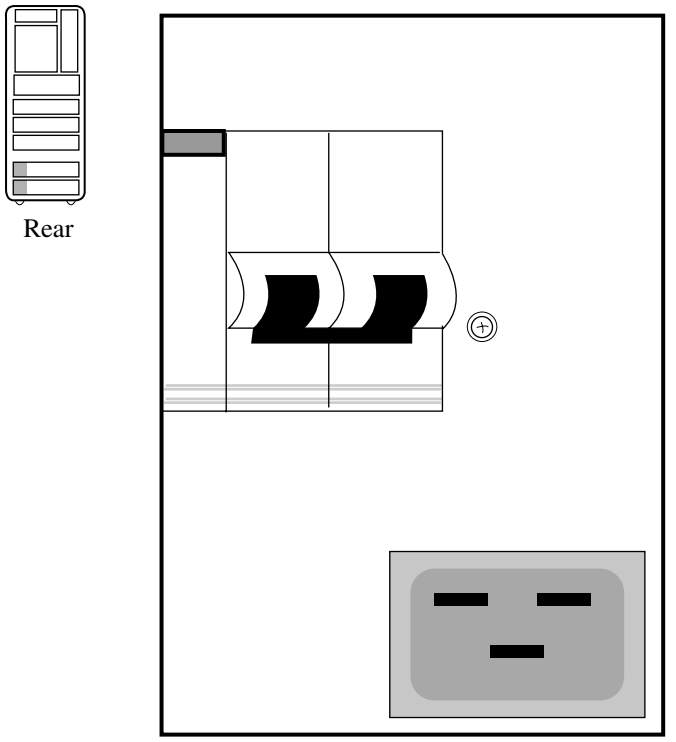
Table 2-5 Control Panel Indicator Lights

Light	Color	State	Meaning
Enable	Green	On	Power is supplied to entire system.
		Off	Power is removed from the system.
Run	Green	On	Console firmware has passed control to the operating system.
		Off	System is in console mode or powered off.
Fault	Yellow	On	Fault on system bus.
		Slow Flash	Power sequencing is in progress or airflow error is detected.
		Fast Flash	Power system error.
		Off	No faults were found.

2.6.1 AC Power Circuit Breaker

The circuit breaker is located on the power regulator at the rear of the cabinet.

Figure 2-11 Circuit Breaker



Up Position: On
Down Position: Off

BX-0608-94

Each power regulator has a circuit breaker.

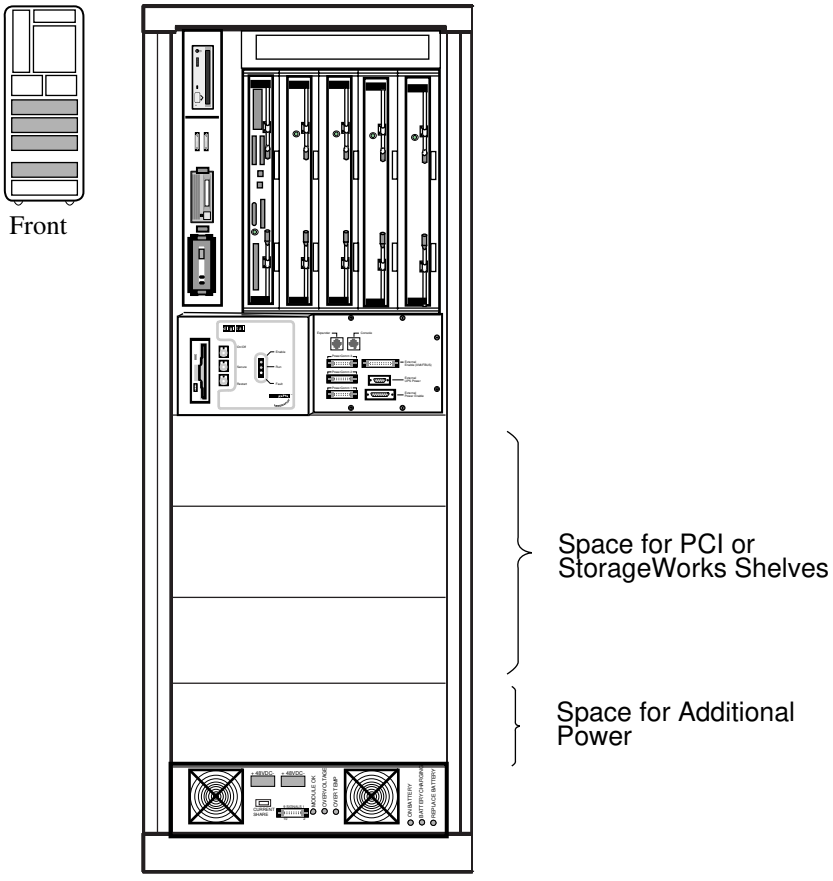
The circuit breaker controls power to the entire system.

For normal operation, the circuit breaker must be in the on position, in which the handle is pushed up. To shut the circuit breaker off, push the handle down.

2.7 Options

System options include a floppy drive, PCI shelves, StorageWorks shelves, an additional power regulator, and optional battery backup.

Figure 2-12 System Options



PCI I/O

PCI I/O is used in the 8200 system. The PCI shelf has 12 slots, a PCI adapter module, a hose interface to the TLSB bus, and a power supply. The KFE70 adapter provides a bridge module for access to EISA I/O. The KFE72 adapter provides a graphics and keyboard and mouse ports for a system using a graphics console monitor. A maximum of three PCI shelves can be installed in the main cabinet.

StorageWorks Shelf

A maximum of six StorageWorks shelves can be installed in the main cabinet. Two shelves can be installed in the same vertical space; one shelf at the front of the cabinet and one shelf at the rear of the cabinet.

Additional Power Regulator

An additional power regulator may be installed for backup in case the other power regulator fails.

Battery Backup Option

A power regulator can be equipped with the battery backup option (a charger module and battery packs) to provide uninterrupted power in case of a power failure.

Chapter 3

AlphaServer 8400 System

The DIGITAL AlphaServer 8400 system is designed for growth, offering configuration flexibility, an outstanding I/O subsystem, and expansion capability in a single or multi-cabinet environment. Functionally, this system is identical to the AlphaServer 8200 system. The 8400 system, however, can have:

- Up to six or seven processor modules for a total of 12 or 14 CPUs (see details in the supported options catalog).
- Up to seven memory modules for a total of 28 Gbytes of memory.
- A three-phase power system with optional battery backup.
- For OpenVMS and DIGITAL UNIX systems, options include XMI, FBUS+, and PCI/EISA I/O bus plug-in units (PIUs), and StorageWorks PIUs. For Windows NT systems, options include PCI I/O bus plug-in units.

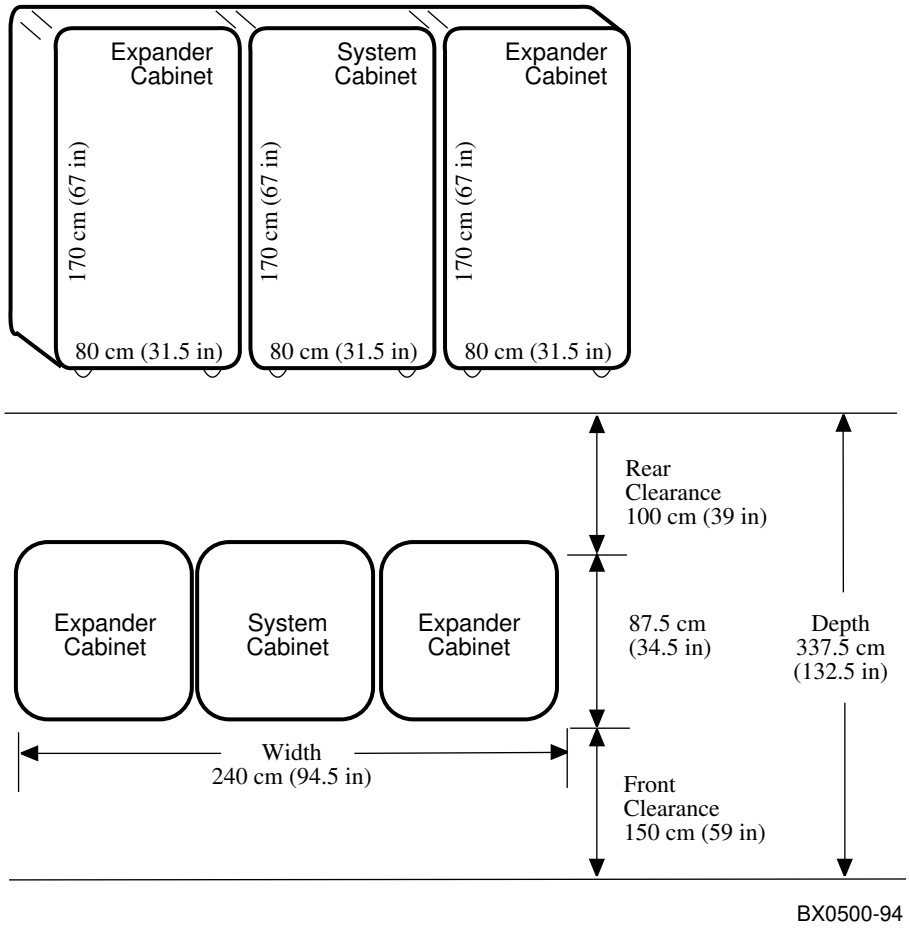
This chapter describes the AlphaServer 8400 system package, introduces the location of components in the cabinet—both front and rear views; and describes the system controls and indicators. Sections include:

- System Characteristics
- Sample 8400 System
- System Front View
- System Rear View
- System Components
- Controls and Indicators

3.1 System Characteristics

DIGITAL AlphaServer 8400 characteristics are shown in Table 3-1 and Table 3-2. Figure 3-1 shows a system footprint.

Figure 3-1 Sample System Footprint



The values in Table 3-1 apply to the 8400 system cabinet only. The values are configuration dependent. Additional options will increase electrical requirements so that an additional power regulator may be needed.

Table 3-1 Electrical Characteristics

Electrical	Specification
3-phase AC input voltage (current, maximum)	120/208 V Wye (30A) – North America 380–415 V Wye (30 A) – Europe/AP 202 V Delta (16 A) – Japan
Nominal Frequency	50–60 Hz
AC power consumption (maximum)	4.6 KW

Table 3-2 Environmental Characteristics

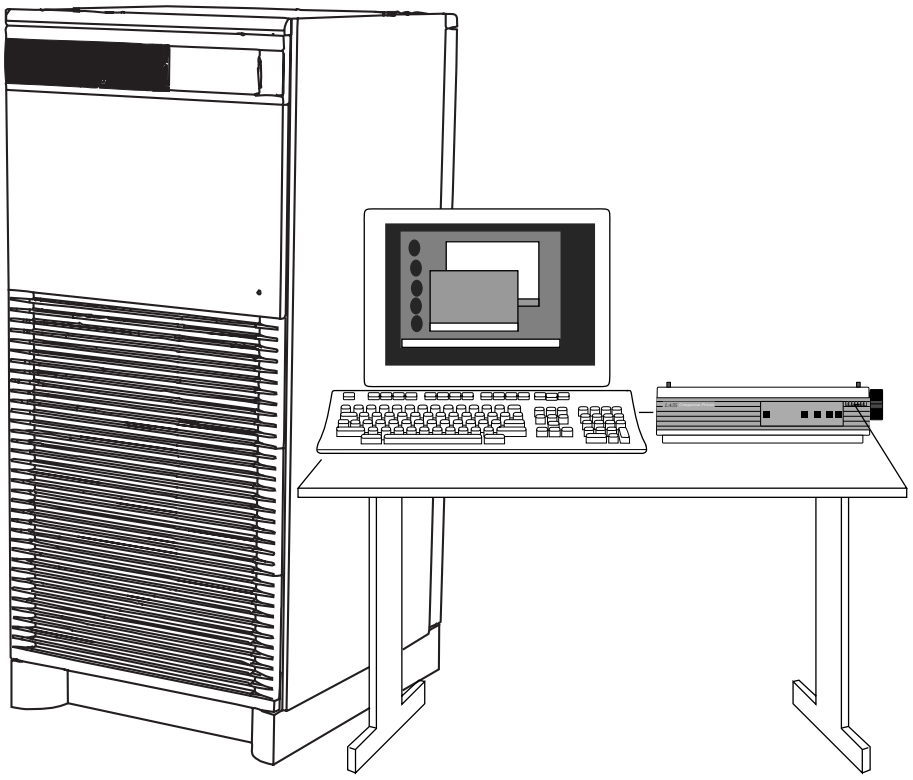
Environmental	Operating	Storage
Heat dissipation	15,700 Btu/hr	–
Temperature[1]	15º–28º C (59º–82º F)	-40º–66º C (-40º–151º F)
Relative humidity[1]	20–80%	10–95%
Altitude	0–2.4 km (0–8200 ft)	0–9.1 km (0–30,000 ft)

[1]Recommended operating temperature is 18º–24º C (65º–75º F) and 40–60% relative humidity.

3.2 Sample 8400 System

Figure 3-2 shows a sample AlphaServer 8400. The system includes a CD-ROM drive, a console device and printer, an accessories kit, and a documentation set. The system options include battery backup PIUs, three different types of I/O bus PIUs (depending on the operating system) and StorageWorks disk drive. Expander cabinets can be added for options.

Figure 3-2 Sample System



BX0502A-98

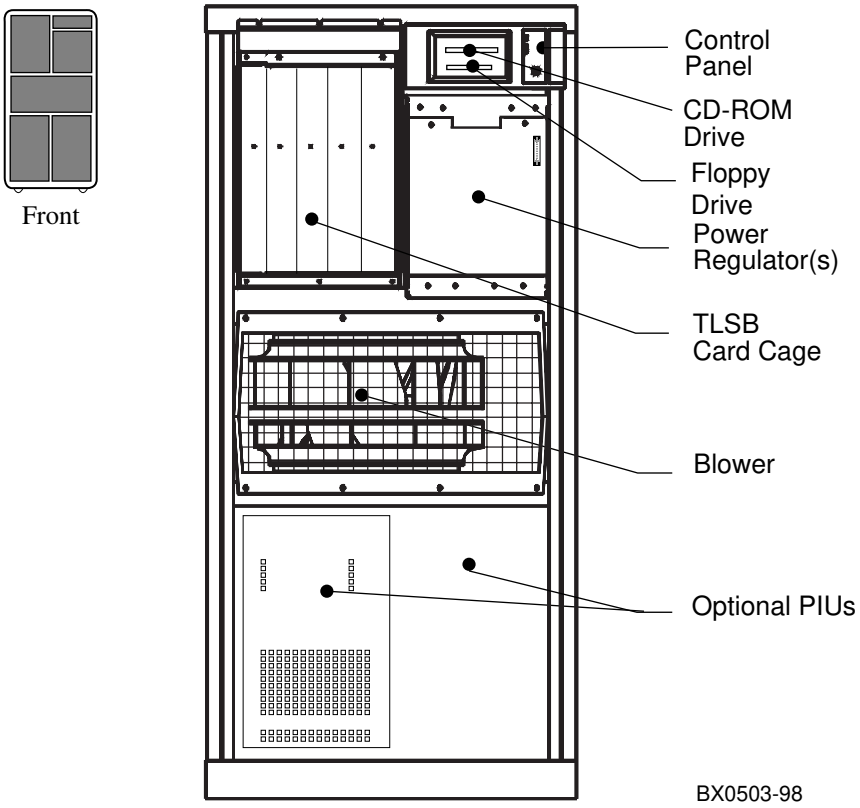
Your DIGITAL customer service engineer has installed your system and verified that it is running properly. Before you turn on the system, familiarize yourself with its components:

- The **system cabinet** houses the TLSB card cage, power system, cooling system, the control panel with status indicators, and a CD-ROM drive. Optional hardware includes disk plug-in units (PIUs), battery PIUs, and I/O PIUs.
- The CD-ROM drive is the **console load device** and is used for installing operating systems and software.
- The **console device** is used for booting and for system management operations. The console device must include a serial console monitor, and also a graphics monitor (for Windows NT systems).
- The **console printer** provides a hardcopy record of system operations.
- **Optional PIUs** include, for OpenVMS and DIGITAL UNIX systems, the XMI PIU, Futurebus+ PIU, PCI PIU, SCSI PIU, and the battery PIU. For Windows NT systems, a PCI PIU is included. These plug-in units are installed in the system or expander cabinets to provide space for I/O, disk, and battery options.
- **Optional expander cabinets** provide additional space for battery, I/O, and disk options.
- A **system documentation kit.**

3.3 System Front View

The control panel, plug-in unit panels, and CD-ROM and optional floppy drive are on the front of the system cabinet. With the front door open, DIGITAL customer service engineers can access the TLSB card cage, the power regulator, cooling system, and optional plug-in units.

Figure 3-3 System Front View



These components are visible from the inside front of the cabinet (see Figure 3-3 for their location):

- Control panel
- CD-ROM drive
- Optional floppy drive
- Power regulator(s) (48 VDC)
- TLSB card cage (holds CPU/memory; slots 0–3)
- Cooling system (blower)

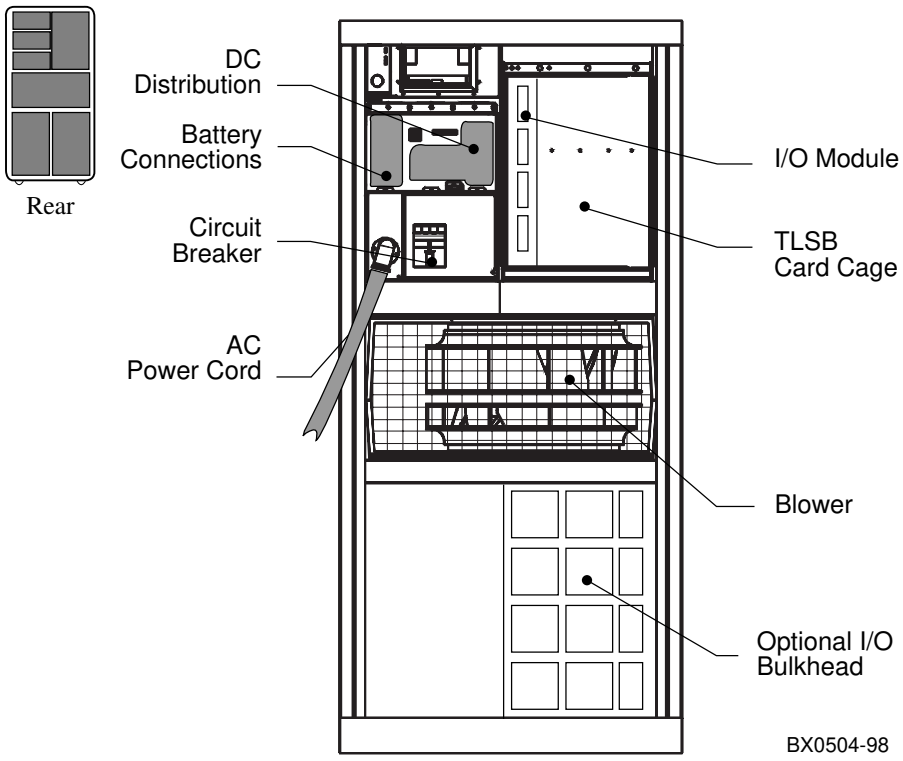
Optional components visible from the inside front include:

- I/O devices
- I/O plug-in unit (PIU)
- Battery PIU

3.4 System Rear View

With the rear door open, DIGITAL customer service engineers can access the TLSB card cage, DC distribution box, battery connections, AC power cord, circuit breaker, blower, and I/O PIU area.

Figure 3-4 System Rear View



The following components are visible from the rear of the cabinet (see Figure 3-4):

- TLSB card cage (slots 4–8)
- I/O port module (slot 8)
- DC distribution box
- Battery PIU connections
- AC power cord and connector
- Circuit breaker
- Blower
- I/O PIU area

Optional components visible from the inside rear include:

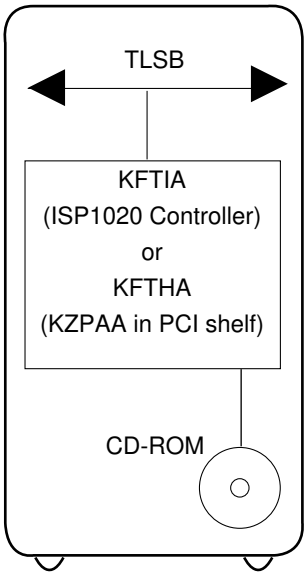
- I/O PIUs
- Battery PIU

3.5 System Components

3.5.1 Console Load Device

The CD-ROM drive is the in-cabinet console load device.

Figure 3-5 Accessing the Console Load Device



BX-0601A-97

The console load device is used for:

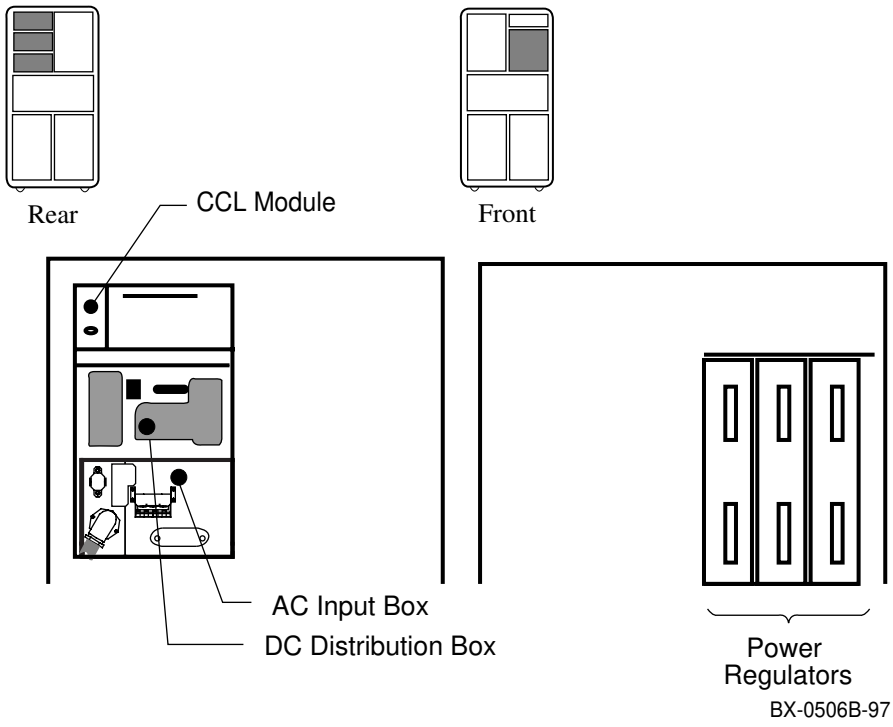
- Installing or updating software
- Loading a backup utility program
- Interchanging user data
- Updating module firmware

The CD-ROM drive is the console load device. It is installed in the system cabinet and is used to access software and online documentation. Access to the CD-ROM is provided directly through a KFTIA module or through the PCI subsystem through a KFTHA module.

3.5.2 Power System

The power system includes an AC input box, DC distribution box, power regulator(s), cabinet control logic module, power distribution cables, signal interconnect cables, and an optional battery PIU (with H7263-AC/AD regulators only).

Figure 3-6 Power System



The power system uses H7263 power regulators (see Figure 3-6), requiring three-phase AC. Two types of H7263 regulators are available. The H7263–AA/AB regulators (optional) have the ability to use and charge optional battery packs. The H7263–AC/AD regulators, shipped with a standard 8400 system, do not allow battery backup. Both types of power regulator allow switching between regulators in the event one fails. Power regulator filler modules are used in unused slots to help direct airflow.

The DC distribution box and AC input box are located on the upper left of the system cabinet (when viewing the system cabinet from the rear). The 48 VDC power regulators are located at the upper right side (when viewing the system cabinet from the front).

The AC input box provides the interface for the system to the AC utility power. The main input circuit breaker, on the AC input box, contains a circuit breaker trip indicator to indicate an open circuit breaker. The DC distribution box connects the AC input box and power regulators. It distributes the 48 VDC power.

NOTE: Additional options can increase the power requirements so that an additional power regulator may be needed.

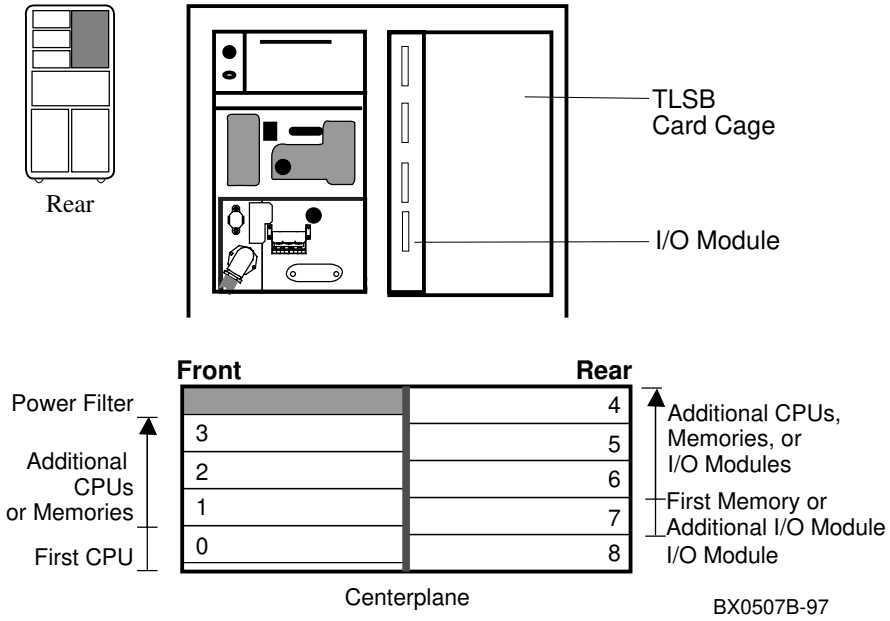
The cabinet control logic (CCL) module has a yellow power LED. When this LED is on, the CCL module is receiving 48 VDC power from the power regulators.

Battery backup capability can be provided in systems with H7263–AA/AB regulators by the addition of the optional battery PIU. The battery PIU is mounted in the bottom of the system cabinet and provides approximately 11 minutes of operating time in N+1 configured systems.

3.5.3 TLSB Card Cage

The TLSB card cage is a 9-slot card cage that contains slots for up to six or seven (KN7CF only) CPU modules, up to seven memory array modules, and up to three I/O modules. The TLSB bus interconnects the CPU, memory, and I/O modules.

Figure 3-7 TLSB Card Cage



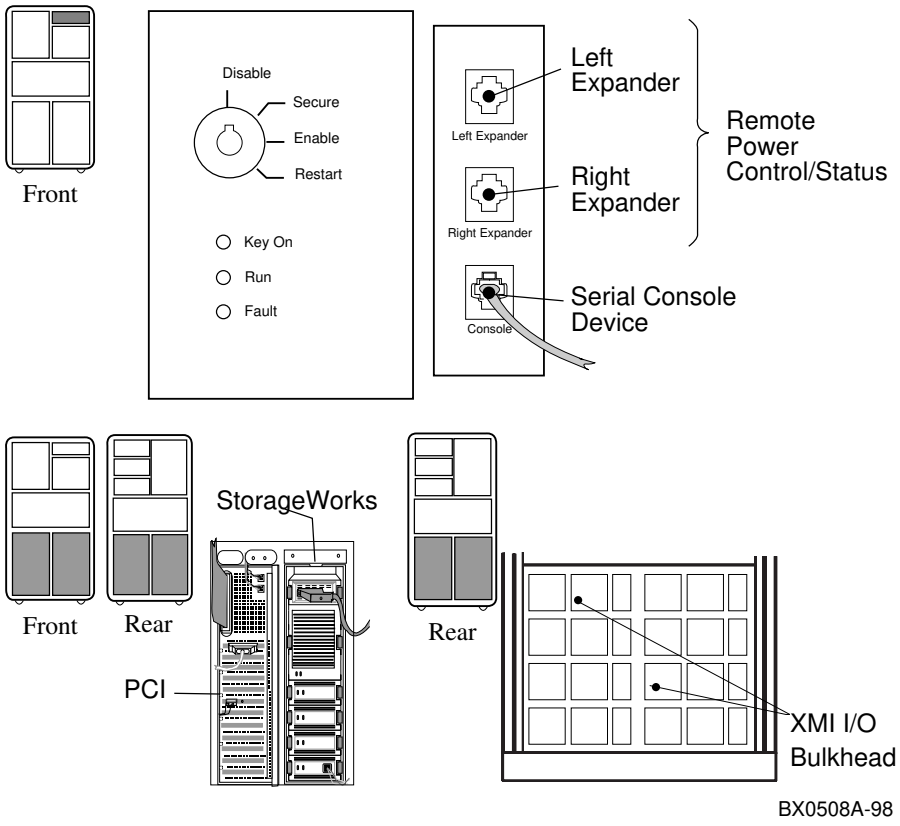
The 9-slot TLSB card cage is located in the upper left (front and rear) of the system cabinet, as viewed from the front. The TLSB card cage must contain one I/O port module, which is always installed in slot 8. The other eight slots contain a combination of KFTHA/KFTIA I/O modules (rear only), memory, and CPU modules.

The TLSB card cage slots are numbered 0 through 3 from right to left in the front of the cabinet and slots 4 through 8 right to left in the rear of the cabinet.

3.5.4 Control/Status and I/O Connections

An I/O connection for the serial console device, and expander cabinet remote power control/status connections are located to the right of the control panel. Ethernet and other I/O connections are located off the I/O modules. For the PCI and FBUS+ PIUs, these connections are directly off the modules. For the XMI PIU, I/O connections are made off the I/O bulkhead in the lower rear of the cabinet.

Figure 3-8 Control/Status and I/O Connections



Three modular jacks to the right of the control panel allow power control/status connections to the left expander cabinet, right expander cabinet, and I/O connections to a serial console terminal. This serial console terminal modified modular jack is keyed so that an expander cabinet connector cannot be plugged into its jack.

The system cabinet has four quadrants in the bottom of the cabinet.

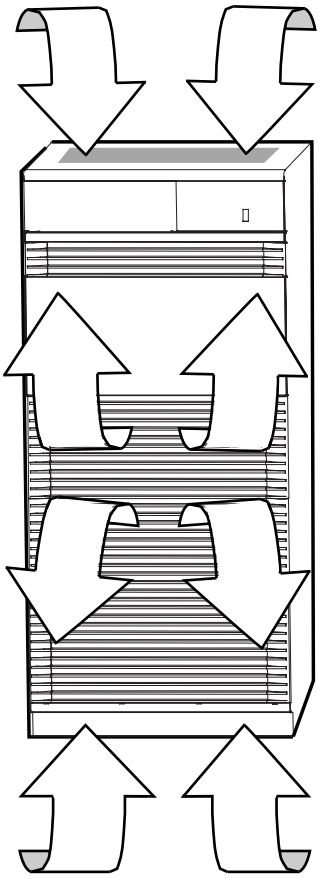
For OpenVMS and DIGITAL UNIX systems, a PCI or Futurebus+ PIU can be installed in any of the four quadrants. An XMI PIU requires two quadrants, with the I/O modules in front and the I/O bulkhead supplying I/O connectors in the back (see Figure 3-8).

For Windows NT systems, a PCI PIU can be installed in any of the quadrants. A PCI PIU (DWLPB only) with a KFE72 adapter *must* be installed off of hose 1 (not hose 0) of the KFTHA or KFTIA installed in slot 8 of the TLSB. (A graphics monitor, keyboard, and mouse, required for Windows NT systems, are connected to the KFE72. Connectors for two serial ports and a parallel port are also supplied with the KFE72.)

3.5.5 Cooling System

The cooling system cools the power system, the TLSB card cage, control logic, and PIUs.

Figure 3-9 Cabinet Airflow



BX0509-94

The cooling system is designed to keep system components at an optimal operating temperature. It is important to keep the front and rear doors free of obstructions, leaving a minimum clearance space of 1.5 meters (59 inches) in the front and 1 meter (39 inches) in the rear between cabinets (see Figure 3-9) to maximize airflow.

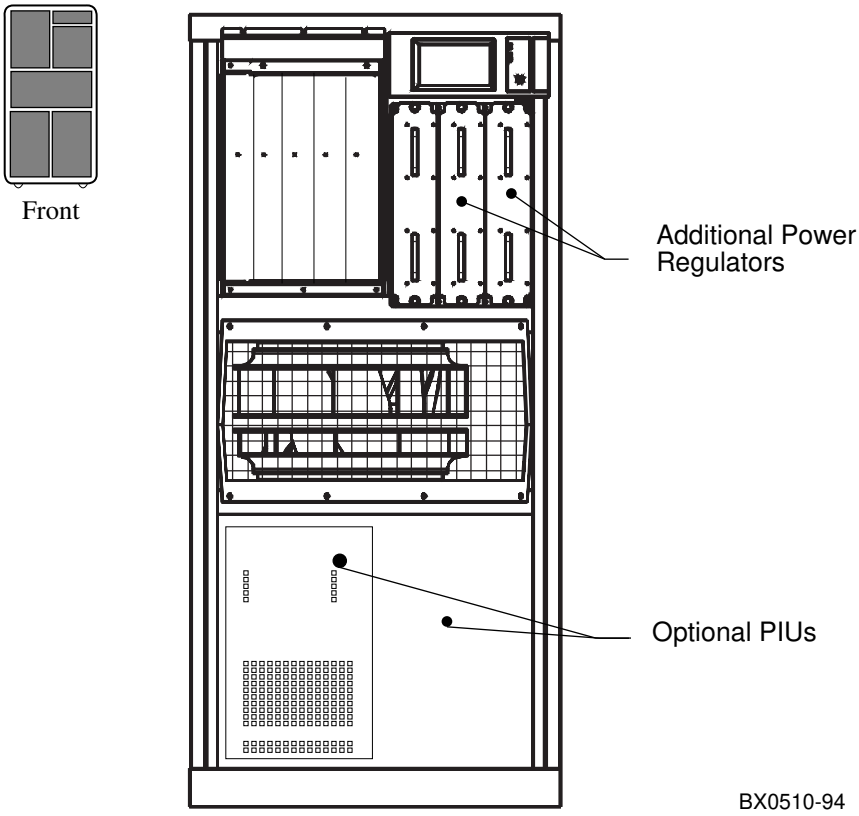
The blower, located in the center of the cabinet, draws air downward through the power regulators and TLSB card cage. It draws air upward through the PIUs. Filler modules, located in the TLSB card cage, help to direct airflow. Air is exhausted at the middle of the cabinet front and rear. The blower speed varies based on the system's ambient temperature.

CAUTION: Anything placed on top of the cabinet could restrict airflow. This will cause the system to power down.

3.5.6 System Options

System options include a floppy drive beside the CD-ROM, additional power regulators and additional PIUs for I/O, disks, tapes, and batteries.

Figure 3-10 System Options



SCSI Disk and Tape PIUs

A maximum of three SCSI PIUs can be installed in the system cabinet or six in an expander cabinet. The PIU can occupy any quadrant in the main or expander cabinet. The PIU can contain up to seven 3.5-inch or two 5.25-inch disks. Supported devices include SCSI disk drives, CD-ROM, and tape drives.

PCI PIUs

Up to two PCI PIUs can be installed in the system or expander cabinet. A PCI PIU occupies one quadrant, either Q2 or Q4. A PCI PIU can contain two PCI shelves, each containing a 12–slot card cage, configured as three 4-slot PCI buses, or one PCI and one SCSI shelf. EISA options can be used on systems with a KFE70 adapter. A system with a KFE72 adapter can only contain PCI options (see Chapter 4).

XMI PIUs

A maximum of two XMI PIUs can be installed in the system or expander cabinet on OpenVMS or DIGITAL UNIX systems. Each XMI PIU occupies two quadrants and has 14 slots. Twelve slots can contain the following modules: CIXCD, DEMFA, DEMNA, KZMSA, and KFMSB. One module must be installed in slots 1 or 14. Slot 7 contains the clock module, and slot 8 contains the DWLMA module.

Futurebus+ PIUs

One Futurebus+ PIU can be installed in the system or expander cabinet on OpenVMS or DIGITAL UNIX systems. The Futurebus+ PIU occupies one quadrant and has 10 slots. The DWLAA module is standard and is installed in slot 5.

Battery PIUs

The system can be equipped with an optional battery PIU to provide power in case of a power failure. Each regulator requires a battery pack mounted in the bottom of the system cabinet. The battery PIUs provide a minimum of 8 minutes of full system operation when fully charged.

Additional Power Regulators

AlphaServer 8400 systems using H7263–AC/AD power regulators, which require 3-phase input can have a second or third power regulator installed as backup should a regulator fail. Systems requiring battery backup must have the H7263–AA/AB regulators installed.

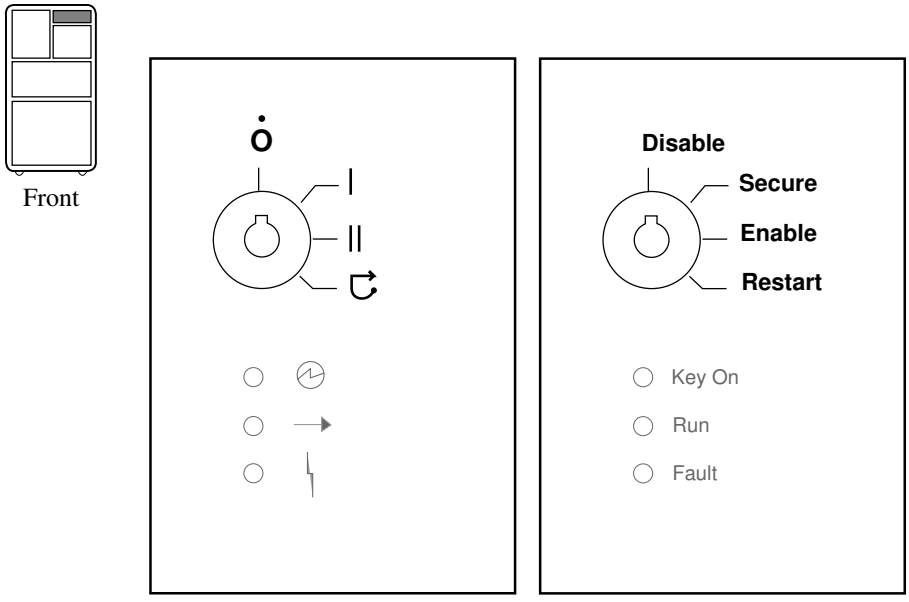
Older AlphaServer 8400 systems may have one H7264 power regulator, using single-phase AC, and a second may be required depending on the system configuration. This system has no support for battery backup operation or for N+1 redundancy.

3.6 Controls and Indicators

This section introduces the system controls and indicators. The system control panel, located in the upper right front of the cabinet, contains a keyswitch and status lights. The keyswitch regulates power going into the system, secures the console device, and controls system operation.

3.6.1 Control Panel Keyswitch

Figure 3-11 Control Panel Keyswitch



The keyswitch labels can be in English or international versions as shown in Figure 3-11.

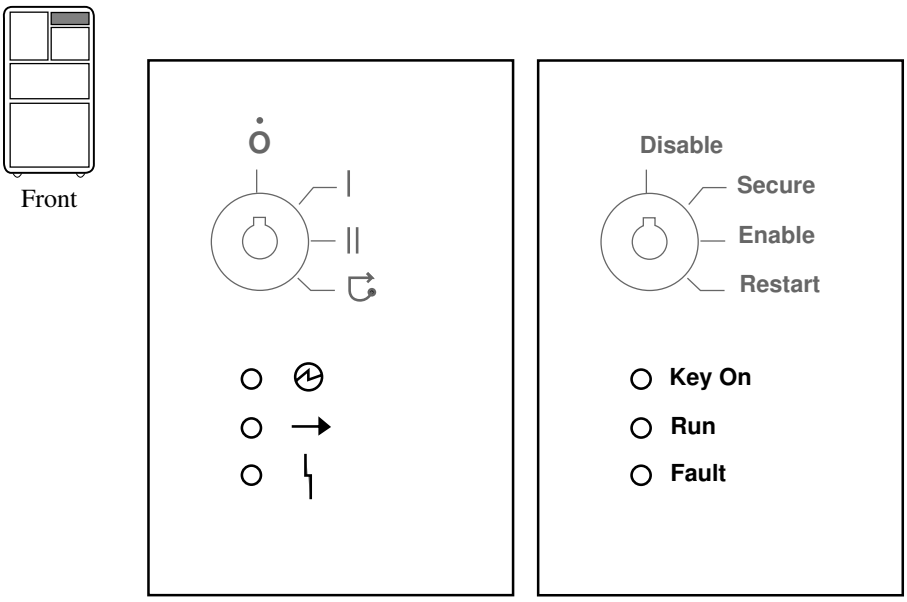
Table 3–3 Keyswitch Positions

Position	Effect
Disable	Removes 48 VDC power from the system. Power is still supplied to the CCL module. This switch is useful for field service when they wish to power down the system in an orderly way, prior to switching the power off completely while replacing or installing a new piece of hardware.
Secure	Prevents input from the console device. On DIGITAL UNIX or OpenVMS systems, used to protect against inadvertent or inadvisable entry into SRM console mode by typing Ctrl/P at the console device. For example, you might switch the system to Secure when a critical program is running, to prevent someone from unknowingly trying to enter SRM console mode. For Windows NT systems, the Secure keyswitch position is used in controlling transitions from Windows NT to SRM console mode, as described in Section 8.2.
Enable	Position used while the machine executes programs. On DIGITAL UNIX or OpenVMS systems, you can interrupt program execution and enter SRM console mode by typing Ctrl/P at the console device when the switch is in this position. Ctrl/P does not work on Windows NT systems. There are several ways to interrupt program execution on these systems. See Section 8.2 for instructions on how to switch between Windows NT, AlphaBIOS Setup, and SRM console mode.
Restart	A momentary switch position, used to reinitialize the system. Registers are reinitialized and system self-test starts running. You can thus restart the system without removing power from the system, which can cause wear and tear on the machine.

3.6.2 Control Panel Indicator Lights

The control panel has three status indicator lights: Key On, Run, and Fault. These lights indicate the operating status of the system.

Figure 3-12 Control Panel Indicator Lights



Three status indicator lights (see Figure 3-12) show the state of the system: (Key On) DC power supplied, (Run) execution, and (Fault) errors. Table 3–4 describes the conditions indicated by the lights.

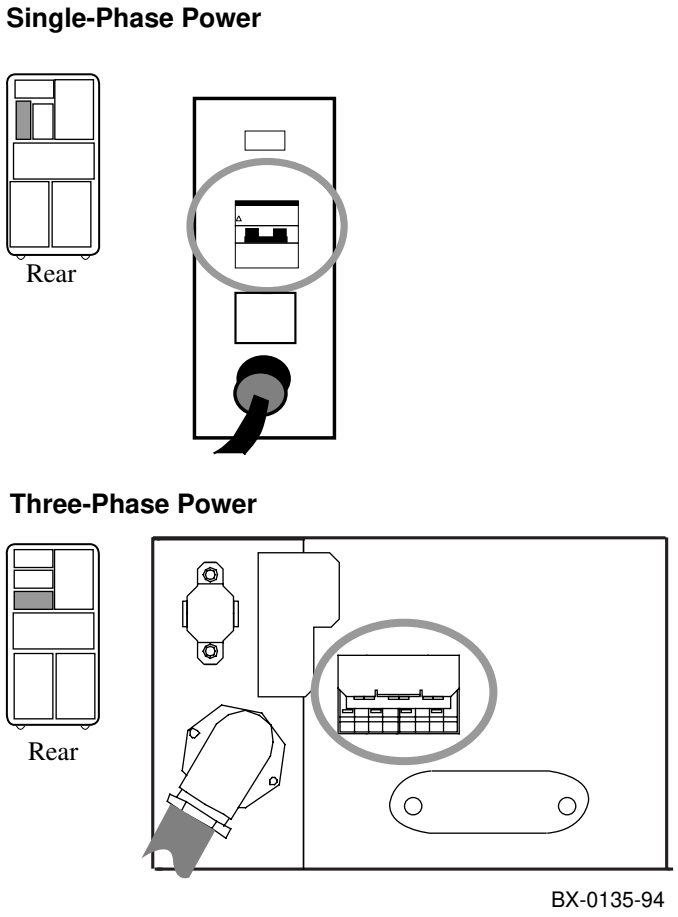
Table 3–4 Control Panel Indicator Lights

Light	Color	State	Meaning
Key On	Green	On	Power is supplied to entire system; the blower is running.
		Off	Power is supplied only to the cabinet control logic module.
Run	Green	On	System is executing operating programs or certain power-up tests.
		Off	System is in console mode, operating system is not running, or the system is turned off.
Fault	Yellow	On	Fault on system bus.
		Slow Flash	Power sequencing is in progress or airflow error is detected.
		Fast Flash	Power system error, airflow error, or keyswitch in Disable position transition detected.
		Off	No faults were found.

3.6.3 Circuit Breaker and AC Power Indicators

The circuit breaker is located on the left side of the rear of the system cabinet, just above the blower assembly. With three–phase power, the circuit breaker can be secured in the off position with a lock.

Figure 3-13 Circuit Breaker and AC Power Indicators



The circuit breaker and power indicators are at the rear of the cabinet.

Circuit Breaker

The circuit breaker controls power to the entire system, including the power regulators, blower, battery backup, and in-cabinet options. Current overload causes the breaker to trip to the off position, so that power to the system is turned off.

For normal operation, the circuit breaker must be in the on position, in which the handle is pushed up. To shut the circuit breaker off, push the handle down.

AC Power Indicators

With three-phase power, the power indicators are located below the circuit breaker handle. When the system is powered on, the power indicators are red. When the circuit breaker is off, tripped, or open, the power indicators change to green. When one phase has tripped, the power indicator for that phase will change to green.

NOTE: The power indicators in the 202V version are different. If one phase trips, all power indicators trip, so that all indicators are green.

Circuit Breaker Lockout

With three-phase power, the circuit breaker lockout secures the circuit breaker in the off position. The lockout consists of a hinged plate that is placed over the circuit breaker handle. A padlock can be placed on the right or left side of the lockout, so that no one can turn the power on.

Single–Phase Power

On older systems with single-phase power, there are no power indicators with single-phase power, nor is there circuit breaker lockout.

Chapter 4

I/O Subsystems

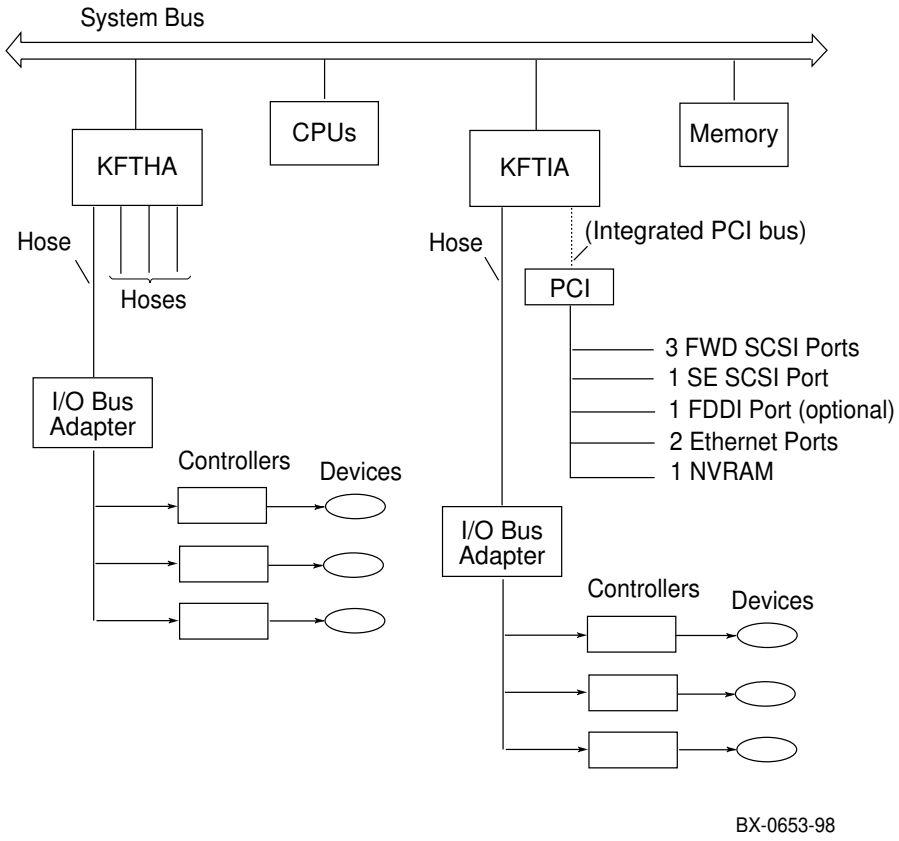
This chapter describes the AlphaServer 8200 and AlphaServer 8400 I/O subsystems. Sections include:

- I/O Subsystem Overview
- I/O Port Modules
- System Configuration Information
- PCI Adapter
- KFE70 and KFE72 Adapters
- EISA Configuration Utility

4.1 I/O Subsystem Overview

Figure 4-1 illustrates CPU, memory, and I/O port module (KFTHA and KFTIA) interfaces to the system bus.

Figure 4-1 I/O Subsystem

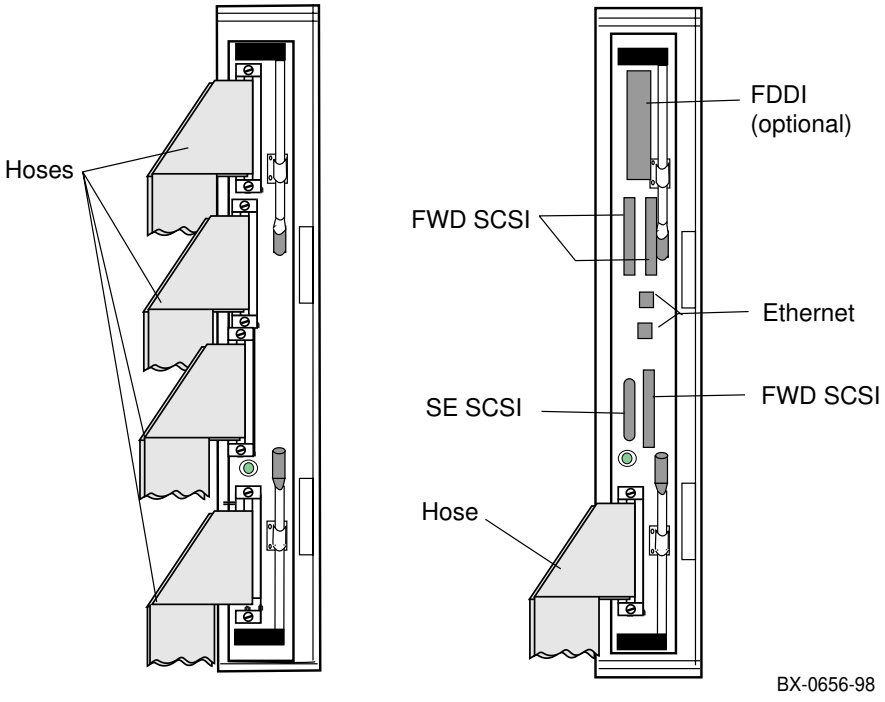


The interface from the AlphaServer 8200 and 8400 system bus to I/O is provided by two types of I/O adapter modules, the KFTHA and the KFTIA. The KFTHA has four channels, or hoses, to connect to I/O devices (see Figure 4-1). The KFTIA has one internal hose (an integrated PCI bus) and one hose to connect to external I/O.

4.2 I/O Port Modules

The KFTHA module has four channels, or "hoses," to external I/O subsystems such as the PCI bus (for any 8200 system and 8400 systems with Windows NT), or the PCI, Futurebus+, and XMI buses (for 8400 systems with OpenVMS or DIGITAL UNIX). The KFTIA module has one channel, or "hose," to external I/O such as a PCI (for any 8200 system and 8400 systems with Windows NT) or PCI, Futurebus+, or XMI bus (for 8400 systems with OpenVMS or DIGITAL UNIX). It has one internal or *integrated* PCI bus.

Figure 4-2 I/O Port Modules



KFTHA

The KFTHA module is designed for high-speed, high-volume data transfers. The KFTHA has four channels (called hoses) connecting (see Figure 4-2) to external I/O buses. Each I/O bus may be a PCI bus (for 8200 systems and 8400 systems with Windows NT systems), or a PCI, Futurebus+, or XMI bus (for 8400 systems with OpenVMS or DIGITAL UNIX).

KFTIA

The KFTIA module has an internal PCI bus that connects to various kinds of I/O devices at the front of the module. This internal *integrated* bus has, as shown in Figure 4-2:

- An FDDI connector. Provides access to a local area network (up to 2 km) or an office-type local area network (up to 100 m), depending on the optional daughter card installed.
- Three FWD (fast wide differential) SCSI connectors. Can be combined with DIGITAL StorageWorks RAID controllers, storage cabinets and devices to access large amounts of SCSI disk storage.
- One single-ended (SE) SCSI connector. Can be used to connect to a CD-ROM drive.
- PrestoServe NVRAM. Optional daughter card for DIGITAL UNIX systems only.
- Two Ethernet connectors

The KFTIA module also has one hose to connect to an external bus. The bus may be a PCI bus (8200 systems and 8400 systems with the Windows NT operating system) or a PCI, XMI, or Futurebus+ bus (8400 systems with OpenVMS or DIGITAL UNIX systems).

4.3 System Configuration Information

Basic information on the system and I/O subsystem configuration is displayed on power-up. Example 4-1 shows an AlphaServer 8400 system self-test display.

Example 4-1 System Self-Test Display

```
                        ❶  ❷
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1  0  NODE #
                    A  A  .  .  .  .  .  M  P  TYP
                    o  o  .  .  .  .  .  +  ++ ST1
                    .  .  .  .  .  .  .  .  EB BPD
                    o  o  .  .  .  .  .  +  ++ ST2
                    .  .  .  .  .  .  .  .  EB BPD
                    +  +  .  .  .  .  .  +  ++ ST3
                    .  .  .  .  .  .  .  .  EB BPD
                       +  +  +  +  .  +  +  +  C0 PCI+ ❸
                 .  .  .  .  .  +  .  .  .  .     C1 FBUS+❹

   +  .  .  .  .  .  .  +  .  .  .  .  +  .  +  .  C4 XMI+❺
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5❻
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6❼
            .  +  .  +  +  .  .  .  .  .  .  .  +  C7 PCI+❽
                    .  .  .  .  +  .  .  .  .        EISA+

                    .  .  .  .  .  .  .  . A0  .  ILV
                    .  .  .  .  .  .  .  .256     256MB
AlphaServer 8400 Console V5.1-30, 14-JAN-1998 SROM V3.1
P00>>>
```

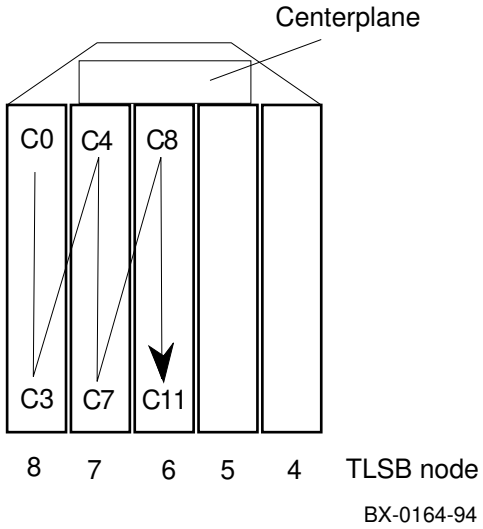
On power-up, the console displays the self-test results. (Chapter 6 describes the system self-test in detail.) The AlphaServer 8400 system shown in Example 4-1 has one dual-CPU module in node (slot) 0 of the TLSB card cage, one 256-megabyte memory module in slot 1, and two I/O adapters, located in slots 7 and 8. The I/O adapter in slot 7 ❷ is a KFTHA and in slot 8 ❶ , a KFTIA.

The KFTIA adapter's *internal* PCI is indicated by C0 ❸. The KFTIA's external hose connector, C1 ❹, is connected to a Futurebus+ PIU. The internal PCI, C0, of the KFTIA module has devices in "slots" 0 through 7, excluding slot 3 which is empty. The KFTIA hose (C1) is connected to a Futurebus+ PIU, which has a DWLAA in slot 5.

As shown in Example 4-1, the KFTHA adapter has two of its four connectors used. The first hose, C4 ❺, is connected to an XMI PIU. The second and third hoses, C5 ❻ and C6 ❼, are not used. The fourth hose, C7 ❽, is connected to a PCI PIU. The XMI PIU, connected to C4 of the KFTHA, has modules installed in slots 1, 3, 8, and E (14) and the PCI PIU, connected to C7 of the KFTHA, has modules installed in slots 0, 7, 8, and A (10).

Figure 4-3 shows the connector numbering scheme for KFTIA and KFTHA modules. Each slot has four connector numbers associated with it, as shown. A KFTIA's internal PCI bus will take the topmost number for a given slot, and its hose connector will take the next number. The bottom two numbers are associated with the slot, even though nothing will ever occupy them on a KFTIA. The KFTHA has four hose connectors, numbered in increasing order from top to bottom.

Figure 4-3 Hose Numbering Scheme for KFTIA and KFTHA



Example 4-2 Sample Show Configuration Command for a DIGITAL UNIX or OpenVMS System with a KFE70

```
P00>>> show config
                                    ❸    ❹      ❺
  ❶      Name ❷                  Type Rev   Mnemonic

TLSB
0++      KN7CF-AB                8014 0000  kn7cf-ab0
1+       MS7CC                   5000 0000  ms7cc0
7+       KFTHA                   2000 0D02  kftha0
8+       KFTIA                   2020 0000  kftia0

C0 Internal PCI connected to kftia0    pci0    ❻
0+     ISP1020         10201077  0001  isp0
1+     ISP1020         10201077  0001  isp1
2+     DECchip 21040-AA   21011  0023  tulip0
4+     ISP1020         10201077  0001  isp2
5+     ISP1020         10201077  0001  isp3
6+     DECchip 21040-AA   21011  0023  tulip1
7+     PCI NVRAM          71011  0000  pci_nvram0

C1 FBUS connected to kftia0            fbus0   ❼
5+     DWLAA               2003  0000  dwlaa0
```

(Continued)

The SRM console command **show configuration** displays system configuration information detailing the I/O adapters connected to your system and their status. Example 4-2 shows a display for an 8400 running either OpenVMS or DIGITAL UNIX (since XMI and Futurebus+ buses are included). The information is displayed in five columns:

❶ module slot number

❷ module name

❸ module type

❹ module revision

❺ module mnemonic

TLSB information is shown first. There are four modules in the TLSB card cage:

- Slot 0 - dual-processor CPU module
- Slot 1 - MS7CC memory module
- Slot 7 - KFTHA I/O port module
- Slot 8 - KFTIA I/O port module

When there are multiple modules of a single type, the mnemonics are numbered consecutively (for example; isp0, isp1, isp2, and isp3).

Information for the KFTIA I/O adapter (slot 8) devices is displayed first. C0 ❻ illustrates the internal PCI devices, located in "slots" 0 through 7, excepting slot 3, as follows:

- Slots 0 and 1 - two ISP1020s (isp0 and isp1) for FWD (fast wide differential) SCSI
- Slots 2 and 6 - two DECchip 21040 (tulip0 and tulip1) twisted-pair Ethernet
- Slots 4 and 5 - two ISP1020s (isp2 and isp3) for SE (single-ended) SCSI and FWD SCSI
- Slot 7 - KFTIA NVRAM daughter card (pci_nvram0)

The Futurebus+ PIU, connected to C1 on the KFTIA, has a DWLAA installed in slot 5. ❼

Example 4-2 Sample Show Configuration Command for a DIGITAL UNIX or OpenVMS System (Continued)

```
C4 XMI connected to kftha0                xmi0     ❽
1+     DEMNA                  C03  0803   demna0
3+     DEMFA                  823  0514   demfa0
8+     DWLMA                 102A  020A   dwlma0
E+     KZMSA                  C36  5256   kzmsa0

C7 PCI connected to kftha0                pci1     ❾
0+     SIO                4828086  0003   sio
7+     KZPSA                 8101  0000   kzpsa0
8+     KZPSA                 8101  0000   kzpsa1
A+     DAC960               11069  0000   dac0

Controllers on SIO                        sio0     ❿
0+     DECchip 21040-AA   21011  0000   tulip3
1+     FLOPPY                 2  0000   floppy0
2+     KBD                    3  0000   kbd0
3+     MOUSE                  4  0000   mouse0

 EISA connected to pci1 through sio0      eisa0    ⑤
```

The KFTHA module in TLSB slot 8 has an XMI PIU attached to its first hose (C4) ❽. The are four modules in the XMI card cage:

- Slot 1 - DEMNA - Ethernet controller
- Slot 3 - DEMFA - FDDI controller
- Slot 8 - DWLMA - TLSB to XMI interface
- Slot E - KZMSA - SCSI controller

The KFTHA's second and third hoses (C5 and C6) are not used, so they do not appear on the show configuration display. The fourth hose, C7, is connected to a PCI adapter containing a KFE70 adapter ❾. There are four modules in the PCI adapter:

- Slot 0 - SIO (standard I/O module) of the KFE70 adapter (Note that the connector module is not shown.)
- Slot 7 - KZPSA (PCI SCSI)
- Slot 8 - KZPSA (PCI SCSI)
- Slot A - DAC960 - KZPSC (SCSI RAID)

Any PCI bus containing the standard I/O module (SIO) will show the information shown in ❿. This information is not meaningful, however. The ports shown are not currently supported.

The EISA support provided by the KFE70 is reported in ⑤.

Example 4-3 Sample Show Configuration Command for a System with a KFE72

```
P08> show config
        Name                    Type   Rev  Mnemonic
TLSB
4++    KN7CD-AB                8014  0000  kn7cd-ab0     ❶
7+     MS7CC                   5000  0000  ms7cc0
8+     KFTHA                   2000  0000  kftha0

C0 PCI connected to kftha0                 pci0
3+     DEC PCI FDDI          F1011  0000  pfi0
5+     Mylex DAC960          11069  0000  dac0
9+     KZPSA                 81011  0000  kzpsa0

C1 PCI connected to kftha0                 pci1      ❷
0+     SIO                 4828086  0004  sio0
4+     VGA                   D1011  0022  vga0
7+     KZPSA                 81011  0000  kzpsa1
9+     KZPSA                 81011  0000  kzpsa2
B+     DECchip 21041-AA     141011  0011  tulip1

  Controllers on SIO                       sio0      ❸
0+     DECchip 21040-AA      21011  0023  tulip0
1+     FLOPPY                    2  0000  floppy0
2+     KBD                       3  0000  kbd0
3+     MOUSE                     4  0000  mouse0

  EISA connected to pci1 through sio0   eisa0

C3 PCI connected to kftha0                 pci2
6+     QLogic ISP1020     10201077  0005  isp0
8+     KZPAA                 11000  0002  kzpaa0
9+     KZPSA                 81011  0000  kzpsa3
```

Example 4-3 shows a sample **show config** SRM command for an AlphaServer 8200 configured with a KFE72 adapter, providing support for a graphics monitor, keyboard and mouse, serial and parallel ports.

❶ The processor module is located in TLSB slot 4, the lowest-numbered slot on an 8200.

❷ All buses off the KFTHA are PCI buses, as required for an AlphaServer 8200 (or an 8400 running Windows NT). Note that hose 1 is connected to a KFE72 adapter, taking the first five slots in the PCI bus. Slot 0 contains the standard I/O module, and slot 4 contains the PowerStorm 3D30 graphics module. Note that the KFE72 must always connect to hose 1 of the KFTHA or KFTIA (in slot 8 of the TLSB).

❸ This section is shown for any PCI bus containing a standard I/O module (SIO). It is not meaningful, however. The keyboard and mouse connectors on a KFE72 adapter occupy slot 2 of a PCI card cage. Slot 1 contains two serial ports, and slot 3 is allocated for a parallel port assembly.

Example 4-4 Sample Show Device Command

```
P00>>> show dev
polling for units on kzmsa0, slot 14, bus 0, xmi0...
dkb100.1.1.14.0    DKB100                RZ26L  440C
dkb200.2.1.14.0    DKB200                RZ26L  440C
dkb300.3.1.14.0    DKB300                RZ26L  440C
                        ❶       ❷      ❸     ❹
polling for units on floppy0, slot 0, bus 1, hose3...
dva0.0.0.1100.3    DVA0                     RX26
polling for units on isp0, slot 7, bus 0, hose3...
polling for units on isp1, slot 8, bus 0, hose3...
polling for units on dac0, slot 10, bus 0, hose3...
P00>>>
```

The SRM command **show device** is useful for locating the boot device for the Loadable Firmware Utility, or the boot device for the operating system for OpenVMS or DIGITAL UNIX systems. The **show device** command is also helpful in isolating non-functioning I/O devices detected by a **test** command. Note that, for Windows NT systems, the device mnemonics shown by **show device** are not the same as the device mnemonics used by AlphaBIOS, which in turn are different from the mnemonics used by the Windows NT operating system.

The **show device** command provides the following device information:

❶ Device mnemonic

❷ Slot number

❸ Bus number

❹ Hose number

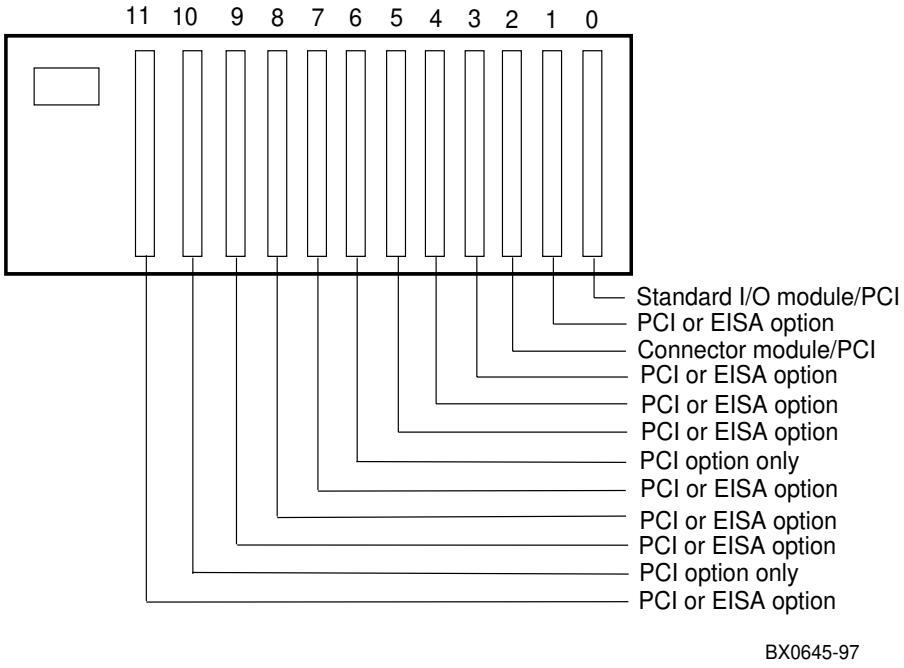
The KZMSA, in slot E (14) of the XMI PIU, is a SCSI controller with three RZ26L SCSI disk drives.

Hose 3 of the KFTHA connects to a PCI bus, which has devices in slots 7, 8, and 10.

4.4 PCI Adapter

The PCI adapter, DWLPA/DWLPB, provides a complete PCI bus subsystem for use with the AlphaServer 8200 and 8400 systems. The KFE70 adapter provides PCI/EISA capability on the DWLPA/DWLPB. The KFE72 adapter provides I/O output needed for systems using a graphics device and must connect to a DWLPB adapter. At least one PCI shelf connected to the system must contain a KFE70 or KFE72, since these adapters provide I/O ports for the console device.

Figure 4-4 PCI/EISA Slot Configuration with the KFE70



If a PCI shelf has neither a KFE70 or KFE72, it contains 12 slots for PCI adapters. numbered right-to-left from 0 to 11.

Configurations with a KFE70

The KFE70 consists of two modules. With OpenVMS and DIGITAL UNIX, the KFE70 adapter provides I/O for the floppy device in the system cabinet, and provides the capacity for EISA options.

If no EISA options are to be used, ten slots are available for PCI options (slots 0 and 2 are used by the KFE70 standard I/O and connector modules).

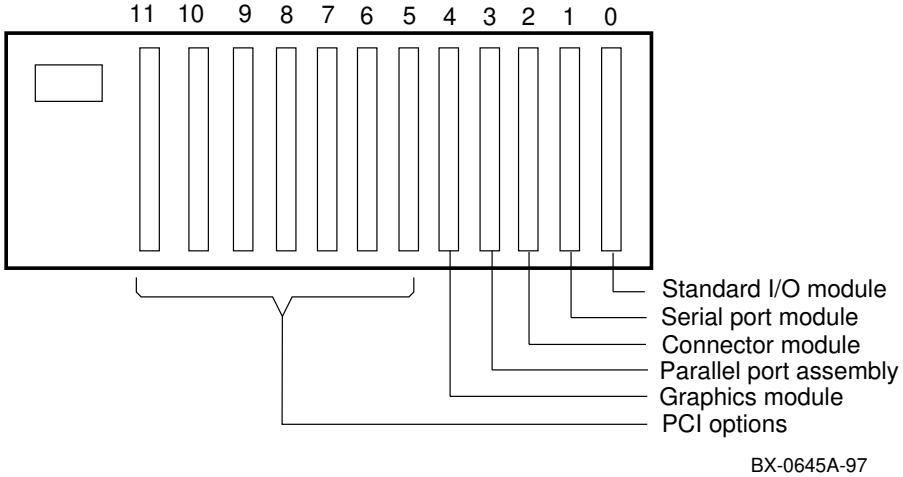
If EISA options are to be used, Figure 4-4 shows the allowable configuration of PCI and EISA options.

- With no PCI options, eight slots are available for EISA options. Slots 0, 2, 6, and 10 cannot be used because slots 0 and 2 are occupied by the modules of the KFE70 and because the alignment of the connectors for EISA options does not allow connection of EISA modules through slots 6 and 10.
- With a mixture of PCI and EISA options, the number of available slots is nine (if there is only one PCI or EISA option) or 10 (if there are at least two PCI or EISA options).

Configurations with a KFE72

The KFE72 consists of four modules, and is supported by a DWLPB (DWLPA does not support the KFE72). With Windows NT systems, one shelf (on hose 1) *must* contain a KFE72 adapter for system I/O devices. Figure 4-5 shows that the KFE72 uses slots 0–4 for modules. Slots 5 –11 are available for PCI devices.

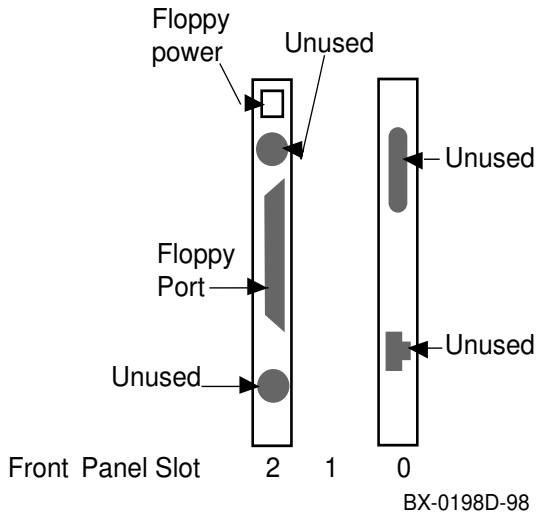
Figure 4-5 PCI Slot Configuration Used with the KFE72



4.5 KFE70 and KFE72 Adapters

The KFE70 adapter contains two modules and supplies EISA capability and a floppy I/O port for the in-cabinet load device (see Figure 4-6). The KFE72 provides I/O support for systems with a graphics console device (see Figure 4-7).

Figure 4-6 KFE70 Modules



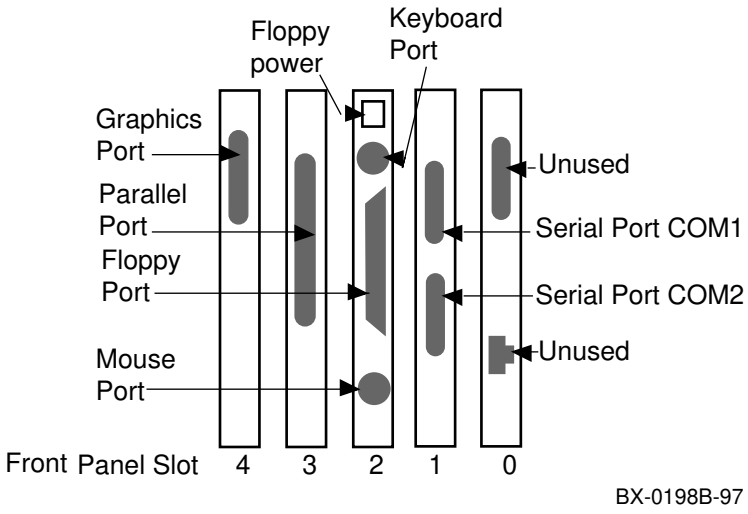
KFE70 Adapter

The KFE70 adapter contains two modules: the standard I/O module (part number B2110-AA) and a connector module (part number 54-23491-01) supplying floppy, keyboard, and mouse ports (see Figure 4-6). The keyboard and mouse ports are not supported. The KFE70 also provides EISA support for DIGITAL UNIX and OpenVMS systems, as described in Section 4.4.

KFE72 Adapter

The KFE72 contains four modules and a parallel port assembly. The four modules are the standard I/O module (part number B2110-AA), a connector module (part number 54-25133-01) supplying floppy, mouse, and keyboard ports, a serial port module (part number 54-25082-01) supplying two serial ports, and the PowerStorm 3D30 graphics module (part number 54-23481-01). The parallel port assembly is cabled to the connector module and occupies a slot position as shown in Figure 4-7.

Figure 4-7 KFE72 and Graphics Modules



4.6 EISA Configuration Utility

There are two reasons for running ECU. On OpenVMS or DIGITAL UNIX systems, run ECU from the SRM console whenever you add, remove, or move an EISA board in your system. For Windows NT, you run ECU from AlphaBIOS Setup to initialize NVRAM on the standard I/O module after installation of a KFE72.

Table 4-1 EISA Bus Configuration Procedure Summary (for OpenVMS or DIGITAL UNIX Only)

Step	Explanation
1. Install EISA option	Use the instructions supplied with the EISA option.
2. Power up and run ECU	If ECU locates the required CFG configuration files, it displays the main menu. The CFG file for the option may reside on a configuration diskette packaged with the option or may be included on the system configuration diskette. Some options designated for ARC-compliant systems include the CFG as part of Option Module Firmware (OMF). OMF is stored on the EISA board in read-only memory, or on an option diskette.
3. View or edit details (optional)	The "View or edit details" option is used to change user-selectable settings, such as COM1 or COM2 for a serial port, or to change the resources allocated for these functions (IRQs, DMA channels, I/O ports, and so on).
4. Save your configuration and restart the system	The "Save and Exit" ECU option saves your configuration information to the system's nonvolatile memory.

The EISA Configuration Utility (ECU) is supplied on the system configuration diskette shipped with the KFE70 and KFE72 adapters.

How ECU Is Used with OpenVMS or DIGITAL UNIX Systems

You need to run ECU from the SRM console whenever you add, remove, or move an EISA board in your system. ECU uses the board's corresponding configuration (CFG) file, which describes the characteristics and the required system resources for that option, to create a conflict-free configuration.

NOTE: The CFG files supplied with the option you want to install may not work on this system. These files may call overlay files that are not required on this system or may reference inappropriate system resources. Contact your DIGITAL representative if you want to use a configuration file that is not supplied on the ECU system diskette.

ECU saves your system configuration to the system configuration diskette. You should make a backup copy of the system configuration diskette and keep the original in a safe place. Use the backup copy when you are configuring the system. Refer to Chapter 7 for a description of the **runecu** command used to invoke ECU.

How ECU Is Used with Windows NT Systems

You need to run ECU from AlphaBIOS to initialize the NVRAM on the standard I/O module after installation of a KFE72. This procedure is described in the *KFE72 Installation Guide* (EK-KFE72-IN). Windows NT on AlphaServer 8200/8400 systems does not support EISA options.

Chapter 5

Booting an Operating System

This chapter describes how to boot the OpenVMS, DIGITAL UNIX, and Windows NT operating systems on AlphaServer 8200/8400 systems. Sections include:

- Preparation
- Selecting a Boot Device
- Booting OpenVMS
- Booting DIGITAL UNIX
- Booting Windows NT

5.1 Preparation

There are some steps you must take preparatory to booting and other steps that can make booting thereafter easier.

5.1.1 Set os_type Environment Variable

For factory-installed software (FIS), a default operating system is defined. If you do not have factory-installed software, or you wish to change the default operating system, you use the set os_type command.

Example 5-1 Setting os_type for OpenVMS

```
P00>>> set os_type vms
```

Example 5-2 Setting os_type for DIGITAL UNIX

```
P00>>> set os_type unix
```

Example 5-3 Setting os_type for Windows NT

```
P00>>> set os_type nt
```

These examples show the SRM command used to define the operating system as OpenVMS, DIGITAL UNIX, and Windows NT, respectively.

5.1.2 Set console Environment Variable

For Windows NT systems, you must set the console environment variable to both. For OpenVMS systems and DIGITAL UNIX systems, the console environment variable should be set to serial (the default).

Example 5-4 Set Console to Serial for DIGITAL UNIX and OpenVMS Systems

```
P00>>> set console serial
```

Example 5-5 Set Console to Both for Windows NT Systems

```
P00>>> set console both
```

The **set console** command is needed for systems having a graphics monitor as part of their console interface. The default setting for the **console** environment variable is **serial.**

By specifying **set console graphics** or **set console both** the SRM console will build the proper parameters in an a data structure called the HWRPB, or Hardware Restart Parameter Block, so that an operating system will build with graphics enabled.

All operating systems require a serial console terminal. The Windows NT operating system also requires a graphics monitor.

5.1.3 Set auto_action Environment Variable

You can direct the SRM console to automatically boot the operating system upon system power-up, Restart, or init. The default is to halt the system after system self-test and leave the console device in SRM console mode.

Example 5-6 Setting the auto_action Environment Variable

```
P00>>> show auto_action
auto_action   halt
P00>>> set auto_action boot
```

The **auto_action** environment variable is set to **halt** by default. In this case, at system restart (power-up, reset, or **init** command), the SRM console software will display the system self-test and display the SRM console prompt to accept commands.

If you set the **auto_action** environment variable to **boot**, at system restart, the SRM console software will display the system self-test and then automatically boot the operating system according to the default boot parameters you defined.

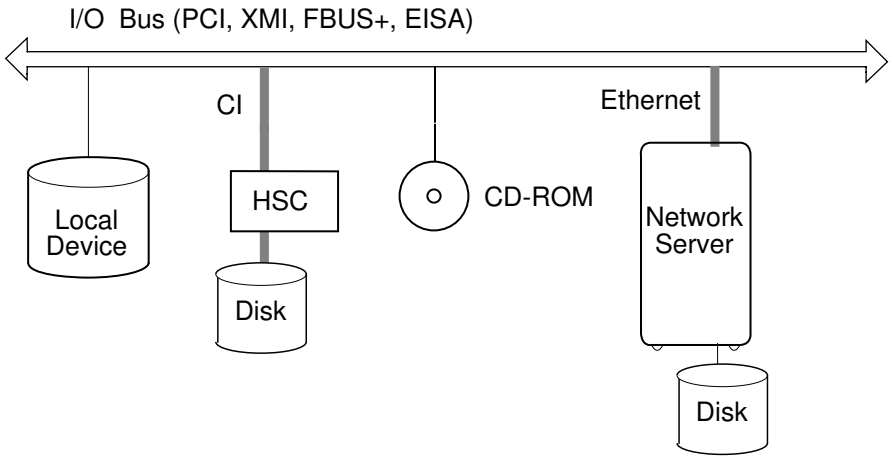
For OpenVMS and DIGITAL UNIX systems, you can return to the SRM console once the system has booted by typing Ctrl/P. This halts the operating system on the boot processor and displays the SRM console prompt.

For Windows NT systems, setting **auto_action** to **boot** causes the SRM console to display the system self-test and start the AlphaBIOS firmware, which will proceed to the boot screen, as described in Section 5.5.1. (See directions in Section 8.2 for switching back to the SRM console from the AlphaBIOS firmware or from Windows NT.)

5.2 Selecting a Boot Device

The OpenVMS, DIGITAL UNIX, and Windows NT operating systems can be booted from a number of devices: the CD-ROM drive, a local system disk, a disk connected to the system through a CIXCD/CIPCA adapter, or by Ethernet from a remote disk on another system. Certain operating systems use a subset of these devices; check the DIGITAL Systems and Options Catalog for details. Note that factory-installed software has a default boot device selected.

Figure 5-1 Boot Devices



For systems with factory-installed software (FIS), a default boot device (OpenVMS or DIGITAL UNIX) or boot path (Windows NT) has already been selected for you. If you want to change this default, follow the instructions here.

NOTE: The default boot device or boot path selected applies to the operating system. If you wish to boot the Loadable Firmware Update (LFU) utility, you must specify the CD-ROM device on your system, if you use the SRM console ***boot*** *command. If you are booting LFU from the* ***AlphaBIOS Setup*** *menu* ***Updating AlphaBIOS****, you must follow the instructions given for that menu in Chapter 8.*

For OpenVMS and DIGITAL UNIX systems, you can set or change the default boot device with the **set bootdef_dev** command, as described in Sections 5.2.1 and 5.2.2. You can also specify a particular device in the SRM console command **boot.**

For Windows NT systems, you do not use either of these commands, either to select a boot device, or to boot the system. Multiple versions of Windows NT can be available for booting, and you can select from one of these installed versions at boot time, as described in Section 5.2.3. Windows NT is booted by the AlphaBIOS firmware as described in Section 5.5.

See Figure 5-1 and Table 5-1 for the types of devices available for your operating system for booting.

Table 5-1 Boot Devices

Device	Location
CD-ROM	In-cabinet compact disk drive, used for booting the Loadable Firmware Update (LFU) Utility. See Appendix B for OpenVMS and DIGITAL UNIX Systems, and Chapter 8 for Windows NT systems.
Local device	Disk connected to the system through an adapter on an I/O bus.
CI disk	Disk located on the system's HSC controller connected to the system by a CIXCD adapter on the XMI bus or by a CIPCA adapter on the PCI bus.
Remote disk	Disk connected to another system on the Ethernet, through the Ethernet port interface or the adapter.

5.2.1 Show Config and Show Device Commands (for Booting OpenVMS and DIGITAL UNIX Systems)

If you do not wish to use the default boot device selected for factory-installed OpenVMS or DIGITAL UNIX systems, the show config and show device SRM console commands display your system configuration, allowing you to identify and select an appropriate boot device. (For Windows NT systems, see Section 5.2.3.)

Example 5-7 Show Config and Show Device Commands

```
P08> show config
        Name                    Type   Rev  Mnemonic
TLSB
4++   KN7CD-AB               8014  0000  kn7cd-ab0
7+    MS7CC                  5000  0000  ms7cc0
8+    KFTHA                  2000  0000  kftha0

C0 PCI connected to kftha0                pci0
3+    DEC PCI FDDI          F1011  0000  pfi0
5+    Mylex DAC960          11069  0000  dac0
9+    KZPSA                 81011  0000  kzpsa0

C1 PCI connected to kftha0                pci1
0+    SIO                 4828086  0004  sio0
4+    VGA                   D1011  0022  vga0
7+    KZPSA                 81011  0000  kzpsa1
9+    KZPSA                 81011  0000  kzpsa2
B+    DECchip 21041-AA     141011  0011  tulip1

  Controllers on SIO                      sio0
0+    DECchip 21040-AA      21011  0023  tulip0
1+    FLOPPY                    2  0000  floppy0
2+    KBD                       3  0000  kbd0
3+    MOUSE                     4  0000  mouse0

  EISA connected to pci1 through sio0   eisa0

C3 PCI connected to kftha0                pci2
6+    QLogic ISP1020    10201077  0005  isp0
8+    KZPAA                 11000  0002  kzpaa0
9+    KZPSA                 81011  0000  kzpsa3
P08>>>sho device
polling for units on dac0, slot 5, bus 0, hose0...
dra0.0.0.5.0       DRA0           2 Member RAID 0
dra1.0.0.5.0       DRA1           2 Member RAID 0
polling for units on kzpsa0, slot 9, bus 0, hose0...
kzpsa0.7.0.9.0     dka     TPwr 1 Fast 1 Bus ID 7   E01  A11
dka400.4.0.9.0     DKA400                  RZ26L  442D
polling for units on floppy0, slot 0, bus 1, hose1...
dva0.0.0.1100.1    DVA0                    RX26/RX23
polling for units on kzpsa1, slot 7, bus 0, hose1...
kzpsa1.7.0.7.1     dkb     TPwr 1 Fast 1 Bus ID 7   P01  A10
dkb100.1.0.7.1     DKB100                  RZ28M  0568
dkb200.2.0.7.1     DKB200                  RZ26L  442D
```

Example 5-7 Show Config and Show Device Commands (Continued)

```
dkb300.3.0.7.1     DKB300                   RZ28M  0568
dkb400.4.0.7.1     DKB400                   RZ28M  0568
dkb600.6.0.7.1     DKB600                   RZ28M  0568
polling for units on kzpsa2, slot 9, bus 0, hose1...
kzpsa2.7.0.9.1     dkc     TPwr 1 Fast 1 Bus ID 7   P01  A10
dkc200.2.0.9.1     DKC200                   RZ28M  0568
dkc400.4.0.9.1     DKC400                   RZ28M  0568
dkc600.6.0.9.1     DKC600                   RZ28M  0568
polling for units on isp0, slot 6, bus 0, hose3...
dkd300.3.0.6.3     DKD300                   RZ1DB-CA  LYJ0
dkd500.5.0.6.3     DKD500                   RZ28M  0568
polling for units on kzpaa0, slot 8, bus 0, hose3...
pke0.7.0.8.3       kzpaa4              SCSI Bus ID 7
dke500.5.0.8.3     DKE500                   RRD46  0557   ❶
polling for units on kzpsa3, slot 9, bus 0, hose3...
kzpsa3.7.0.9.3     dkf     TPwr 1 Fast 1 Bus ID 7   F01  A11
dkf200.2.0.9.3     DKF200                   RZ29B  0016
dkf300.3.0.9.3     DKF300                    RZ28  T436
dkf500.5.0.9.3     DKF500                   RZ28M  0568
P08>>>
```

❶ The RRD46 is a CD-ROM disk drive, where you load firmware update CD-ROM disks. You can use the **boot** command to boot LFU from the disk. In this example, the command would be:

```
P08>>> boot dke500
```

If you are not planning to use the default boot device selected for your factory-installed software (FIS), or if your OpenVMS or DIGITAL UNIX operating system is not factory installed, you select an appropriate boot device for your system. Use the **show config** and **show device** SRM console commands to display all the I/O adapters and devices on your system. The format for SRM console device names is:

ddau.n.c.s.h

where the fields are as defined in Table 5-2.

Table 5-2 SRM Console Device Naming Conventions

Field	Meaning	Description
dd	Device driver	Two-letter designator for a port or class device driver. Usually one of: dk SCSI disk dr RAID set disk du DSSI disk dv Floppy disk ew Ethernet network fw FDDI network mk SCSI tape mu DSSI tape pk SCSI class pu DSSI class
a	Adapter ID	Specifies the one-letter designator for adapter ID. One of a, b, c,…
u	Unit number	Specifies the device unit number. For DIGITAL MSCP devices, this is a unique monotonically increasing number. For SCSI devices, this is 100 times the bus node number, the next designator.
n	Node number	Specifies the bus node ID. Only meaningful for storage devices.
c	Channel number	Specifies the channel or PCI function number. Only meaningful for multi-channel or multi-function devices.
s	Slot number	Specifies the device's logical slot number. The output of a show config command indicates the logical slot numbers for devices in a system.
h	Hose number	Specifies the device's hose number. The output of a show config command shows the KFTIA or KFTHA modules on the TLSB of your system.

Ideally, your system manager labeled the I/O devices on your system, so that you can identify a particular unit by its name. If not, you will have to follow cables back to the device adapter and then back to the hose to identify a particular unit.

In either case, select a device on your system that is of the type appropriate for the media your operating system software is on. It is useful to reserve one device on your system for this use, for consistency. This is done by setting the **bootdef_dev** environment variable, as described next.

5.2.2 Boot Environment Variables (for OpenVMS and DIGITAL UNIX)

With OpenVMS or DIGITAL UNIX systems, you can change the factory-installed default boot device, or, if your operating system is not factory-installed, define a default boot device, to be used automatically if no device is specified in the boot command itself.

Example 5-8 Viewing and Setting Boot Environment Variables (for OpenVMS and DIGITAL UNIX)

```
P00>>> show boot*    ❶
boot_dev
boot_file
boot_osflags            0
boot_reset              OFF
bootdef_dev
booted_dev
booted_file
booted_osflags
P00>>> set bootdef_dev dkb100    ❷
P00>>> show boot*
boot_dev
boot_file
boot_osflags            0
boot_reset              OFF
bootdef_dev             dkb100.1.0.6.1
booted_dev
booted_file
booted_osflags
```

For OpenVMS and DIGITAL UNIX systems, some boot environment variables can be set by the user using the **set** SRM console command. Others are set by values you may specify in a **boot** command. For Windows NT systems, the boot path can be set with the **alphabios** firmware.

❶ The **show boot*** command displays the current settings of the boot environment variables. These are described in detail following.

❷ In Example 5-8, the user sets the default boot device environment variable **bootdef_dev** to device dkb100.

5.2.3 Boot Path for Windows NT Systems

Multiple versions of the Windows NT operating system may be available for booting on your system. One is defined as the primary operating system, which is booted by default. You can select a different operating system for booting from the boot screen, displayed after AlphaBIOS is started.

Figure 5-2 AlphaBIOS Boot Screen with Auto Start Enabled

BX-0568A-98

You use the AlphaBIOS firmware to boot a Windows NT system (the SRM command **boot** does not apply to Windows NT systems).

Figure 5-2 shows an AlphaBIOS boot screen where the primary operating system will be booted within a specified amount of time. You can use the arrow keys to select another of the installed operating systems, if any, during this time. The effect of various AlphaBIOS boot options are described in greater detail in Section 5.5.

Information on how to use AlphaBIOS to install a new Windows NT operating system is given in Chapter 8. During installation, a boot path is defined for the bootstrap loader and the operating system file, using AlphaBIOS and the Windows NT software itself.

For information on how to define the boot path for a newly installed Windows NT system, see the description of the AlphaBIOS Utilities selection from the **AlphaBIOS Setup** screen, and the description of the **Operating System Selection Setup** screen in Chapter 8.

5.3 Booting OpenVMS

Check the environment variables for default boot parameters and make any changes needed. Issue the boot command.

Example 5-9 OpenVMS Boot

```
P00>>> show boot*    ❶
boot_dev
boot_file
boot_osflags            0
boot_reset              OFF
bootdef_dev
booted_dev
booted_file
booted_osflags
P00>>> set bootdef_dev dkb100    ❷
P00>>> show boot*
boot_dev
boot_file
boot_osflags            0
boot_reset              OFF
bootdef_dev             dkb100.1.0.6.1
booted_dev
booted_file
booted_osflags

P00>>> b    ❸
(boot dkb100.1.0.6.1 -flags 0)❹
SRM boot identifier: scsi 1 6 0 1 100 ef00 81011
boot adapter: kzpsa0  rev 0 in bus slot 6 off of kftha0 in
TLSB slot 8
block 0 of dkb100.1.0.6.1 is a valid boot block
reading 904 blocks from dkb100.1.0.6.1
bootstrap code read in
Building FRU table.......
FRU table size = 0x2ff8
base = 200000, image_start = 0, image_bytes = 71000
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
```

Continued on next page

Example 5-1 OpenVMS Boot (Continued)

```
6-JAN-1998 11:36:18.27

OpenVMS (TM) Alpha Operating System, Version V7.1    ❺

      .
      .
      .
   [more informational messages specific to the site]
      .
      .
      .                    ❻
 Accounting information:
 Buffered I/O count:    79521      Peak working set size:       6080
 Direct I/O count:       4042      Peak virtual size:         173104
 Page faults:            4845      Mounted volumes:               23
 Charged CPU time:  0 00:00:32.62  Elapsed time:       0 00:05:26.03¨  ❼
```

❶ The **show boot*** command shows the values of all the environment variables beginning with the characters "boot."

❷ Noting that no default boot device has been set, the user in this example issues the command **set bootdef_dev** to the desired disk. This disk will be used in subsequent boot commands that specify no device.

❸ The boot command is issued and the SRM console firmware begins the process of booting the operating system.

❹ The SRM console displays information regarding the boot device and proceeds to load the bootstrap code. Once this process is finished, the SRM console transfers control to the bootstrap code.

❺ The date and time and operating system banner display.

❻ Information specific to the site is displayed (sometimes many lines). The display ends with accounting information about the boot operation.

❼ The operating system display stops. Press Enter to proceed to system login.

5.4 Booting DIGITAL UNIX

Use the show boot* command to ensure that the device from which to boot the operating system, and other boot parameters, are correct. Issue the boot command.

Example 5-10 DIGITAL UNIX Boot

```
P00>>> show boot*    ❶
boot_dev                dkc300.3.0.7.1
boot_file
boot_osflags            a
boot_reset              OFF
bootdef_dev             dkc300.3.0.7.1
booted_dev
booted_file
booted_osflags
P00>>> boot    ❷
 (boot dkc300.3.0.7.1 -flags a)
SRM boot identifier: scsi 1 7 0 3 300 ef00 81011
boot adapter: kzpsa0  rev 0 in bus slot 7 off of kftha0 in
TLSB slot 8
block 0 of dkc300.3.0.7.1 is a valid boot block
reading 16 blocks from dkc300.3.0.7.1
bootstrap code read in
Building FRU table.......
FRU table size = 0x2632
    .
    .
    .
[initialization and configuration information]
    .
    .
    .
Digital UNIX Version V4.0 (thenut.Eng.PKO.DEC.Com) console

login:    ❸
```

❶ Type the **show boot*** command to show the current settings for boot environment variables. If you wish to change anything, use the appropriate **set** command to make the change.

❷ Type the **boot** command.

❸ The DIGITAL UNIX operating system boots, displays the system banner, and prompts the user for login information.

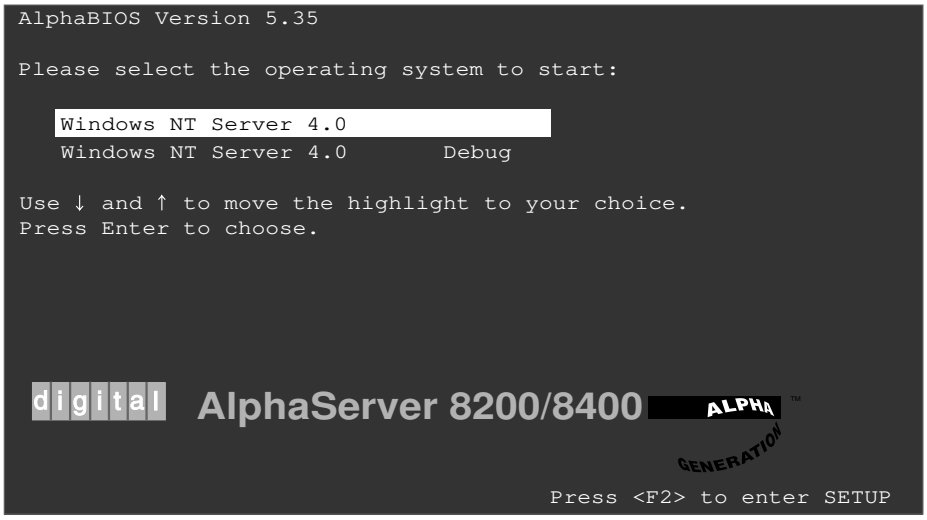
5.5 Booting Windows NT

Microsoft Windows NT is started from the AlphaBIOS Boot screen.

5.5.1 Effect of SRM Console auto_action Environment Variable

The way you boot Windows NT depends upon how the SRM console auto_action environment variable is set.

Figure 5-3 AlphaBIOS Boot Screen with Auto Start Disabled



BX-0568-98

Once you have installed a Windows NT system (see Chapter 8), the Windows NT operating system is booted from the AlphaBIOS firmware. How this firmware is started depends on the setting of the SRM console's **auto_action** environment variable (see Table 5-3) and the setting of the **Auto Start** and **Auto Start Count** options in the **CMOS Setup** screen of **AlphaBIOS Setup**. Figure 5-3 shows the boot screen displayed when the **Auto Start** option from the **CMOS Setup** screen of AlphaBIOS is disabled.

*NOTE: The **os_type** environment variable must be set to **nt** before Windows NT can be booted.*

Table 5-3 SRM Console auto_action Environment Variable

auto_action Setting	Action Taken

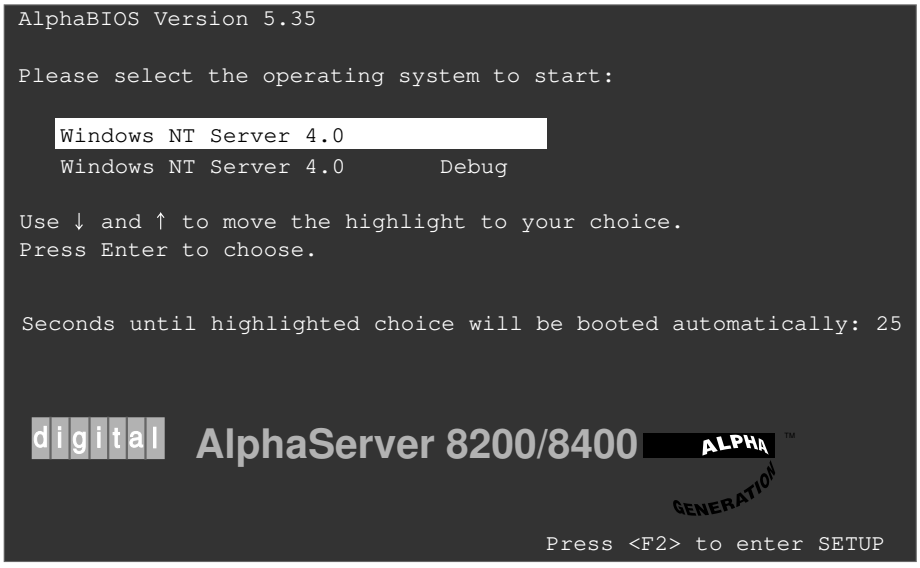
halt (default)	At system restart (power-up, reset, or **init** command), the system self-test displays on the serial console terminal, along with lines showing initialization of certain devices on both the serial and graphics console terminals. The display stops with the SRM console prompt. At this point, to boot Windows NT, make sure that the **os_type** environment variable is set to **nt** and type the **alphabios** command. For example:
	
	`P08>>> show os_type`
	`os_type  nt`
	`P08>>> alphabios`
	
	AlphaBIOS starts, displaying screens showing the initialization of devices and terminating with the Windows NT boot screen. What happens at this point depends on the setting of AlphaBIOS options (see Sections 5.5.2 and 5.5.3).

boot	At system restart (power-up, reset, or **init** command), the system self-test displays on the serial console terminal, along with lines showing initialization of certain devices on both the serial and graphics console terminals. AlphaBIOS then starts automatically, displaying screens showing the initialization of devices and terminating with the Windows NT boot screen. What happens at this point depends on the setting of AlphaBIOS options (see Section 5.5.2 and 5.5.3.) *NOTE: Once **auto_action** is set to **boot**, AlphaBIOS will be started automatically and proceed to the boot screen. To reach the SRM console, you must follow the directions described under the Ctrl/Alt/Delete option described in Table 5-4.*

5.5.2 Effect of AlphaBIOS Auto Start and Auto Start Count Options

Booting Windows NT also depends upon the Auto Start, Auto Start Count, and Password options of the AlphaBIOS CMOS Setup screens.

Figure 5-4 AlphaBIOS Boot Screen with Auto Start Enabled



BX-0568A-98

AlphaBIOS allows multiple versions of the Windows NT operating system to be installed. A primary operating system is designated during installation (see Section 8.9) as the default. The action taken before booting proceeds depends upon the setting of three AlphaBIOS options from the **CMOS Setup** and **Advanced CMOS Setup** screens.

Table 5-4 Auto Start and Auto Start Count Options

Auto Start Setting	Action Taken
Disabled	The AlphaBIOS firmware stops at the boot screen until you type something. There are four possibilities: • Enter — The primary operating system is booted, or a screen asking for Startup Password is displayed (see Section 5.5.3.) • Arrow key and Enter — You can select another operating system to be booted using the arrow keys, and then press Enter to boot that operating system. A screen asking for Startup Password may be displayed first, as described in Section 5.5.3. • F2 — Enter **AlphaBIOS Setup** Mode (see Chapter 8), or a screen asking for Setup or Startup Password is displayed (see Section 5.5.3.) • Ctrl/Alt/Delete — Pressing these keys simultaneously or pressing the Reset button (on 8200 systems) or turning the keyswitch to Reset (on 8400 systems) restarts the system. *NOTE: What happens at this point depends on the setting of the **auto_action** environment variable, as discussed in Section 5.5.1. If **auto_action** is set to **boot,** you can reach the SRM console by pressing the Secure button in (8200) or turning the keyswitch to Secure (8400) before AlphaBIOS is automatically started. The system will then display the SRM console prompt. At this point, restore the Secure button to out (8200) or return the keyswitch to Enable (8400) so that you can enter an SRM console command.*
Enabled	The AlphaBIOS firmware displays the boot screen for the number of seconds specified with the **Auto Start Count** option. During this time, you can perform any of the actions described above, or simply wait until the time specified by the **Auto Start Count** option expires and the system boots.

5.5.3 Effect of the AlphaBIOS Password Option

If Startup Password in the AlphaBIOS Password Option in the Advanced CMOS Setup has been selected, you will have to enter a password before Windows NT will boot. If Setup Password in the AlphaBIOS Password Option has been selected, Windows NT will boot according to the SRM console and AlphaBIOS Auto Start options described in Section 5.5.1 and 5.5.2, but you will be asked for a Setup password if you press F2 to select AlphaBIOS Setup.

Figure 5-5 AlphaBIOS Boot Screen with Startup Password Selected

BX-0568C-98

You can enable the **Setup Password** option of the **Advanced CMOS Setup** screen of AlphaBIOS to limit access to **AlphaBIOS Setup** mode, but allow booting of the operating system. You can enable the **Startup Password** option of the **Advanced CMOS Setup** screen of AlphaBIOS to limit access to **AlphaBIOS Setup** and operating system booting.

If the **Startup Password** has been selected from the **Advanced CMOS Setup** screen of AlphaBIOS, then you will be prompted for a password before the Windows NT system is booted or before you can enter **AlphaBIOS Setup**, after you press the F2 key.

Figure 5-5 shows the Windows NT boot screen with the password authentication window superimposed. In this case, the Startup Password option has been selected and the user must type a password before the operating system will boot or before the user can enter **AlphaBIOS Setup** mode. A similar screen would prompt for a **Setup Password** if that has been selected from the **Advanced CMOS Setup** screen of AlphaBIOS and you pressed F2 to enter **AlphaBIOS Setup** mode before the operating system was booted.

Chapter 6

System Troubleshooting

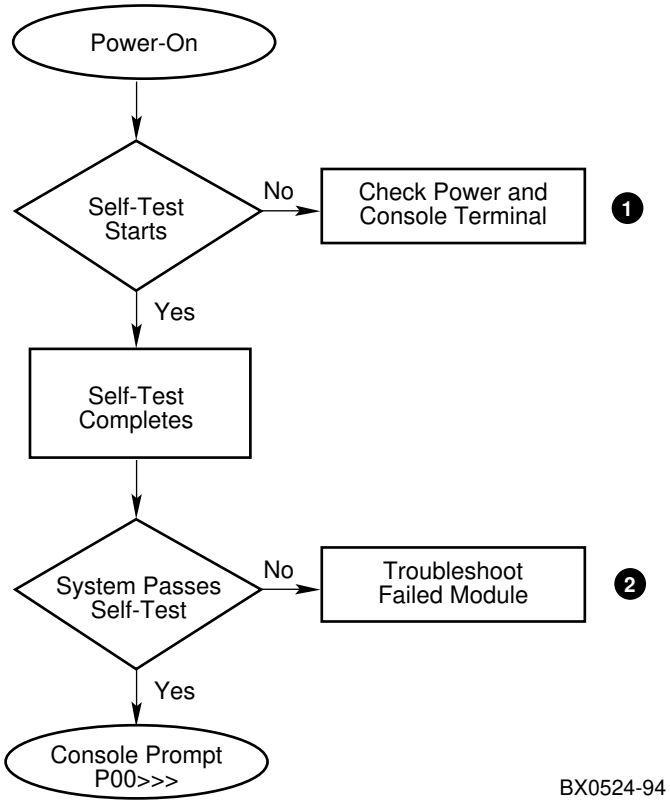
This chapter describes system problems and their symptoms during power-up, booting, and normal operation. Advanced troubleshooting is discussed in the *Service Manual*. Topics discussed include:

- Troubleshooting During Power-Up
- Troubleshooting During Booting
- Troubleshooting a PCI Shelf
- Troubleshooting an XMI Plug-In Unit
- Troubleshooting a Futurebus+ Plug-In Unit
- Troubleshooting a BA655 Plug-In Unit
- Troubleshooting a Battery Plug-In Unit
- Self-Test Overview
- Testing Sequence
- Sample Self-Test Display
- Self-Test Lines NODE # and TYP
- Self-Test Lines ST and BPD
- Self-Test Lines C0, C1, C2, ... Cn
- Self-Test Identification Line
- Self-Test Lines ILV and MB
- Show Commands
- Test Command
- Error Reports

6.1 Troubleshooting During Power-Up

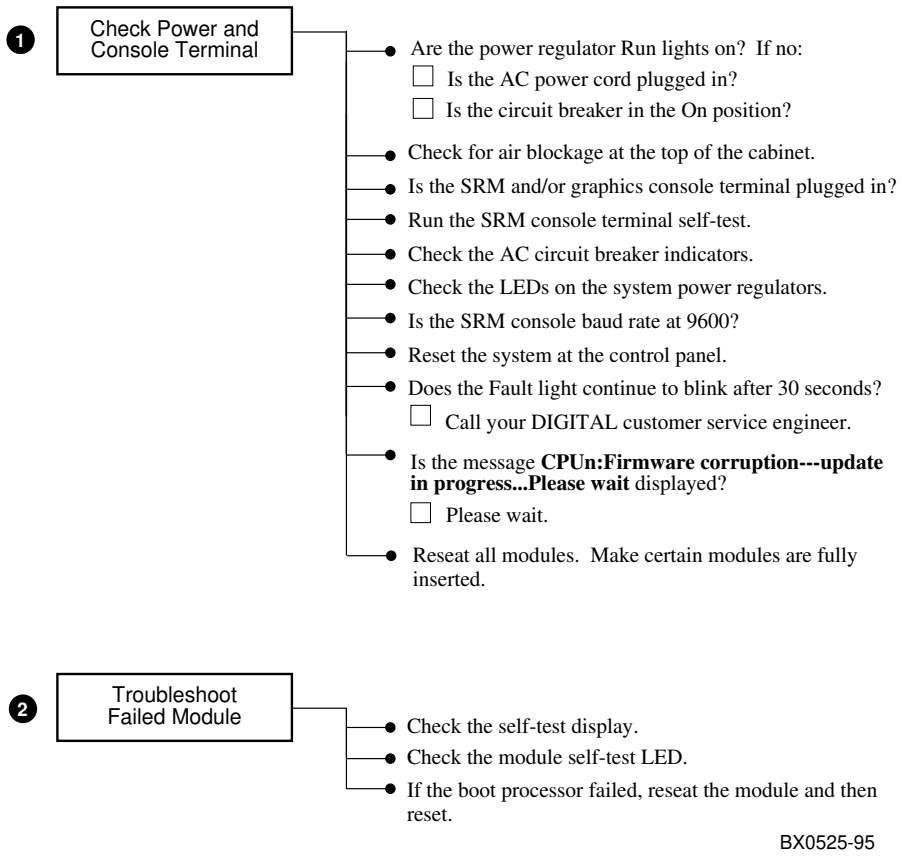
Figure 6-1 shows the power-up sequence and steps to take if a problem occurs.

Figure 6-1 Power-Up Troubleshooting Flowchart



Troubleshooting steps during power-up are described in Figure 6-2.

Figure 6-2 Power-Up Troubleshooting Steps



6.2 Troubleshooting During Booting (OpenVMS and DIGITAL UNIX)

When booting fails for an OpenVMS or DIGITAL UNIX system, you can check several parameters. Figure 6-3 shows the boot sequence. Figure 6-4 shows the steps to take if a problem occurs during booting. If you are unable to correct the problem, call your DIGITAL customer service engineer.

Figure 6-3 Booting Troubleshooting Flowchart

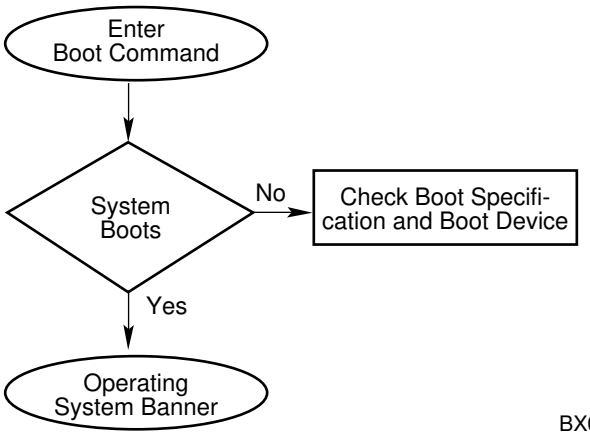
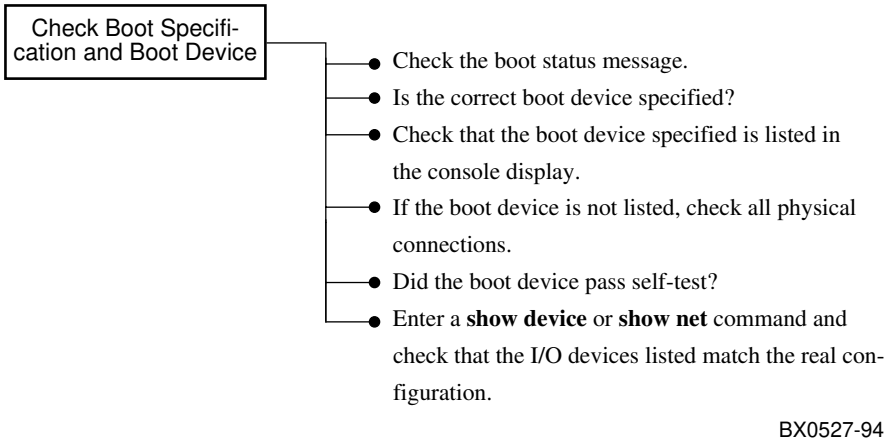


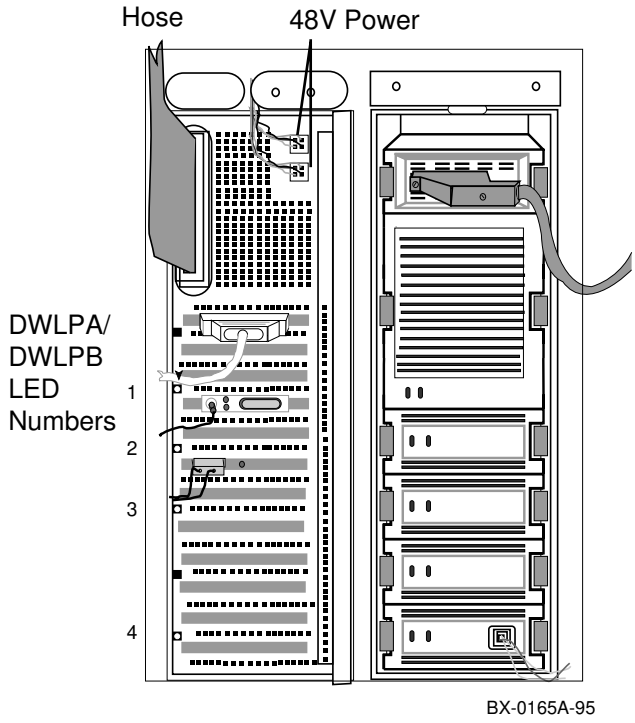
Figure 6-4 Troubleshooting Steps During Booting



6.3 Troubleshooting a PCI Shelf

LEDs show the status of power supplies as well as the adapter self-test results in the PCI shelf, mounted vertically in the AlphaServer 8400 and horizontally in the AlphaServer 8200.

Figure 6-5 PCI Shelf in a BA655 PIU



LED Status in PCI Shelf

LED 1 - On-board power system OK

LED 2 - Motherboard self-test passed

LED 3 - 48 VDC power supply OK

LED 4 - Hose Error

Figure 6-6 PCI Shelf in an 8200

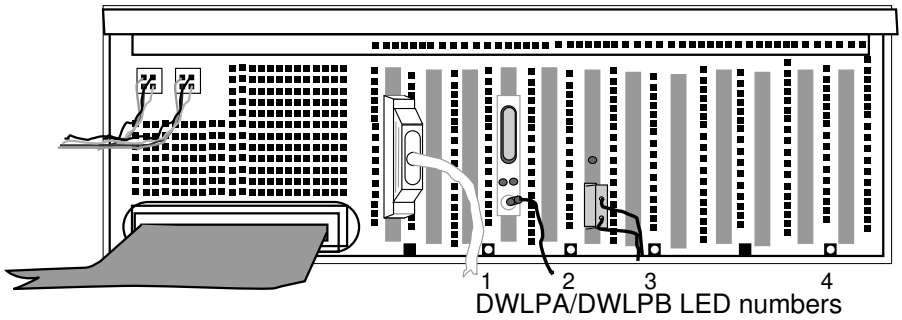
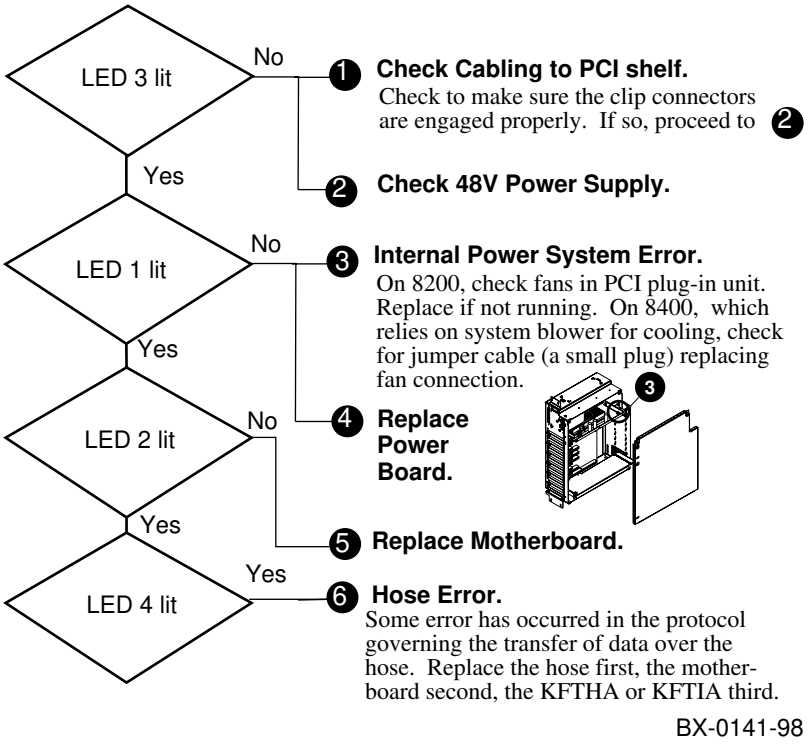


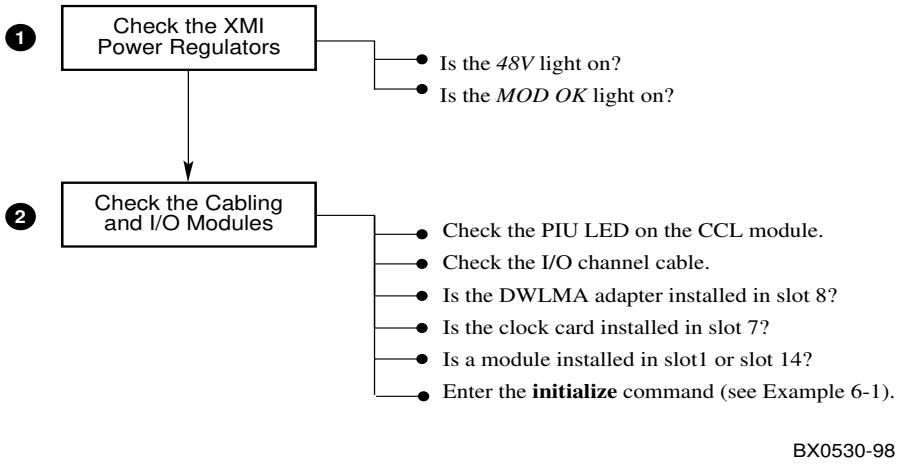
Figure 6-7 Troubleshooting Steps for PCI Shelf



6.4 Troubleshooting an XMI Plug-In Unit

You can perform several checks if you suspect a problem with an XMI plug-in unit. See Figure 6-8. If you are unable to correct the problem, call your DIGITAL customer service engineer.

Figure 6-8 Troubleshooting an XMI Plug-In Unit



In Example 6-1, an **initialize** command is issued, causing a system reset and self-test. Self-test results indicate a failing DWLMA adapter. See Section 6.10 for more information on the self-test display.

Example 6-1 Sample Self-Test Display, Failing DWLMA Adapter

```
P00>>> initialize        # Resets the entire system.
Initializing...

F   E   D  C  B  A  9  8  7  6  5  4  3  2  1  0  NODE #
                      A  M  .  .  .  .  .  M  P  TYP
                      o  +  .  .  .  .  .  +  ++ ST1
                      .  .  .  .  .  .  .  .  EB BPD
                      o  +  .  .  .  .  .  +  ++ ST2
                      .  .  .  .  .  .  .  .  EB BPD
                      +❶+  .  .  .  .  .  +  ++ ST3
                      .  .  .  .  .  .  .  .  EB BPD
    .   .  .  .  .  .  .  .  .  .  .  .  .  .     C0 XMI-❷
.   .   .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1
.   .   .  .  .  .  .  .  .  .  .  .  .  .  .  .  C2
              .  .  .  .  .  .  .  .  .  .  .  +  C3 PCI+
                    .  .  .  .  .  .  +  +     EISA +
                    . B0  .  .  .  .  . A0  .  ILV
                    .128  .  .  .  .  .256  .  384MB

AlphaServer 8400 Console V4.3, SROM 3.9,Jan 7 1997 00:00
P00>>>
```

❶ In Example 6-1, the KFTHA module passes self-test, as indicated by the plus sign **(+)** at node 8 on self-test line ST3.

❷ The DWLMA adapter fails self-test, as indicated by the minus sign **(–)** displayed on the **C0 XMI** line. When a DWLMA fails self-test, the failure is reported, and the results of the I/O adapter self-tests are not displayed.

6.5 Troubleshooting a Futurebus+ Plug-In Unit

You can perform several checks if you suspect a problem with a Futurebus+ plug-in unit. If the 48V LED indicates a power problem, refer to Figure 6-9. If the MOD OK LED indicates a problem, see Figure 6-10. If you are unable to correct the problem, call your DIGITAL customer service engineer.

Figure 6-9 FBUS+ PIU Troubleshooting – 48V LED Off

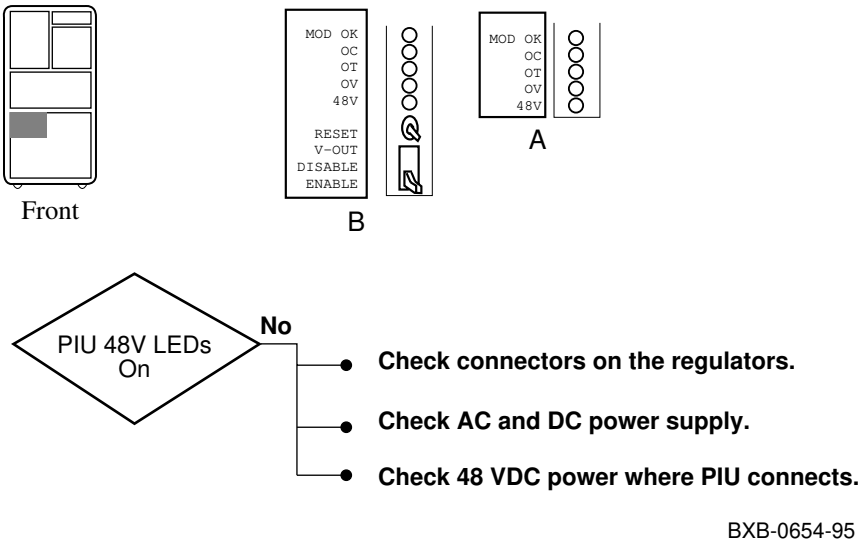
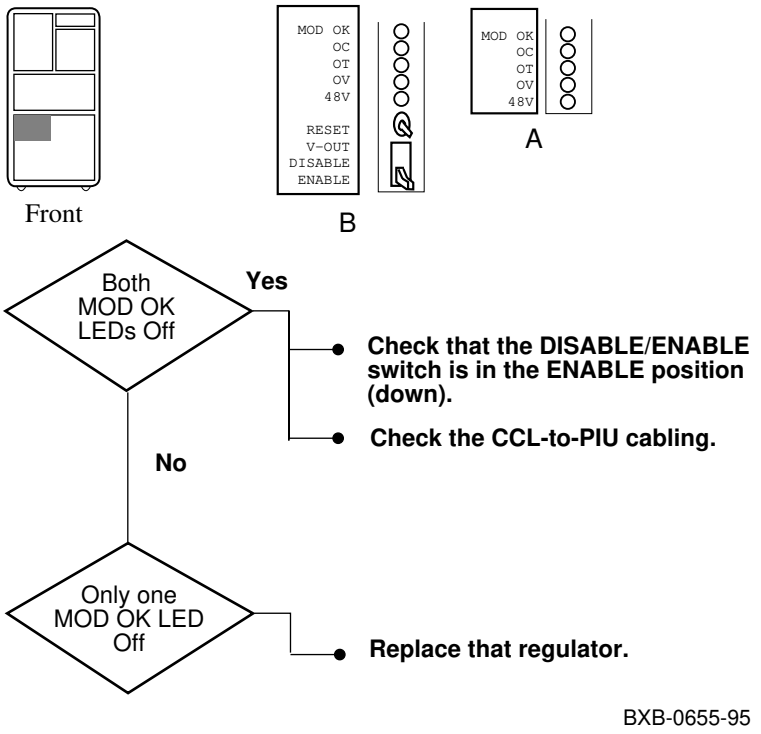


Figure 6-10 FBUS+ PIU Troubleshooting – MOD OK LED Off



Regulator A and B in the FBUS+ PIU are dependent on each other. For the MOD OK LEDs to be on, the following sequence of power applies: 3 volts from A, followed by 5 volts from B, followed by 2 volts from A. Therefore, if any one of the voltages fails, both regulators will turn the MOD OK LED off.

6.6 Troubleshooting a BA655 Plug-In Unit

SCSI indicator LEDs are located on each disk drive and power supply. Table 6-1 and Table 6-2 list the functions of the LEDs shown in Figure 6-11.

Figure 6-11 SCSI Indicator LEDS

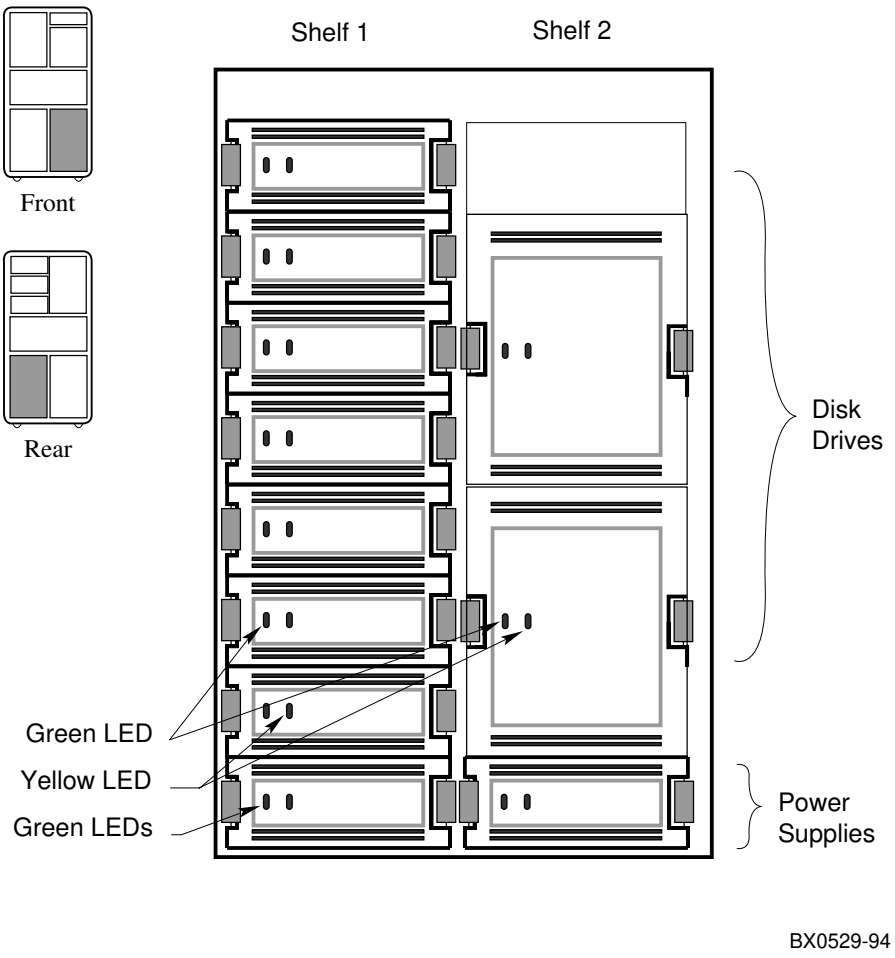


Table 6-1 SCSI Disk Drive LEDs

Indicator LED	LED State	Meaning
Green	Off	No activity
	Flashing	Activity
	On	Activity
Yellow	Off	Normal
	Flashing	Spinup/spin down
	On	Not used

Table 6-2 SCSI Power Supply LEDs

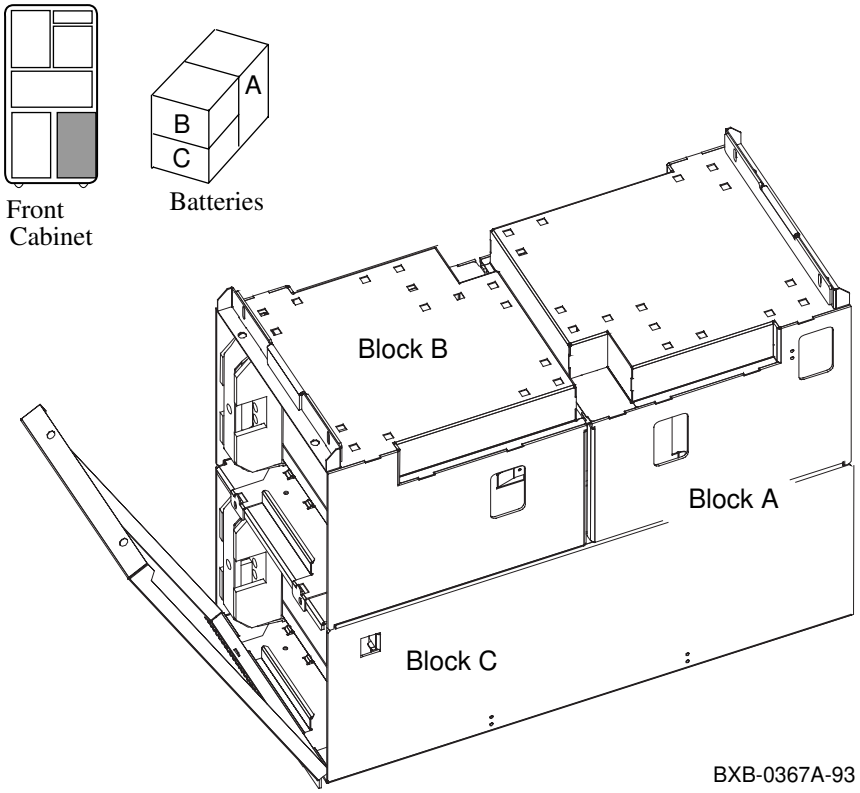
Indicator LED	LED State	Meaning
Green (left)	Off	Shelf fault
	On	Shelf OK
Green (right)	Off	Power fault
	On	Power OK

If the power supply LEDs indicate a problem, call your DIGITAL customer service engineer.

6.7 Troubleshooting a Battery Plug-In Unit

The H7237 battery PIU houses one to three battery blocks, each providing backup power to one H7263-AA/AB power regulator. The battery PIU is installed in the bottom of the system or expander cabinet (8400 with three-phase power and H7263-AA/AB power regulators only).

Figure 6-12 Battery Plug-In Unit



The **show power** command is used to see the status of the power system (AlphaServer 8400 system with three-phase power only).

Example 6-2 Show Power Command

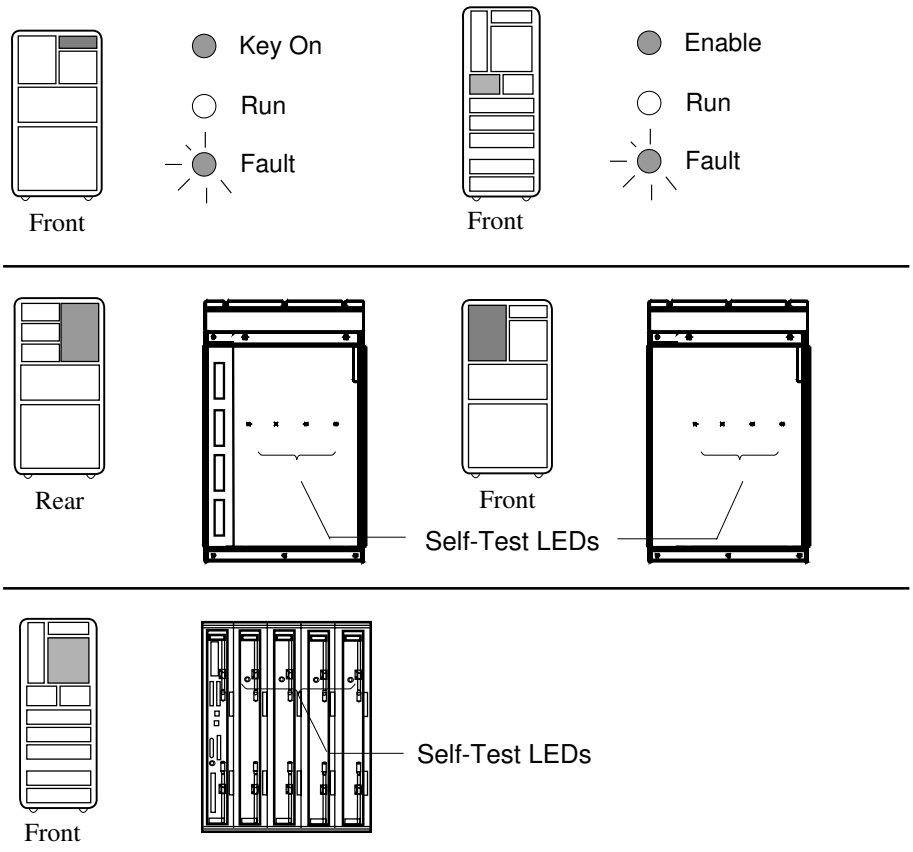
```
P00>>> show power                                  ❶
Cabinet: Main                Regulator:           A      B      C
-------------------------------------------------------------------------
          Primary Micro Firmware Rev :         2.0    2.0    2.0
        Secondary Micro Firmware Rev :         2.0    2.0    2.0
                  Power Supply State :       NORMAL NORMAL NORMAL
              AC Line Voltage (V RMS) :      113.71 114.35 115.93
               DC Bulk Voltage (VDC) :       227.02 227.02 227.02
            48V DC Bus Voltage (VDC) :        47.57  47.57  47.57
            48V DC Bus Current (ADC) :        30.17  29.68  29.58
      48V Battery Pack Voltage (VDC) :        50.85  50.72  50.70
      24V Battery Pack Voltage (VDC) :        25.56  25.56  23.95
   Battery Pack Charge Current (IDC) :         2.91   2.90   2.90
      Ambient Temperature (Degree C) :        26.22  24.80  24.75
                Elapsed Time (Hours) :       290.00 290.00 290.00
Remaining Battery Capacity (Minutes) :         8.00   8.00   8.00
     Battery Cutoff Counter (Cycles) :         0      1.00   1.00
               Battery Configuration : 4 Batteries  4 Batteries  4 Batteries
                     Heatsink Status :       NORMAL  NORMAL    NORMAL
                Battery Pack Status :        CHARGING CHARGING CHARGING
                Last UPS Test Status :       PASSED  PASSED  PASSED
```

❶ The user enters a **show power** command (Example 6-2). The main cabinet has three power regulators.

6.8 Self-Test Overview

The system provides a record of its testing in the self-test display. The control panel Fault light and the module self-test LEDs also indicate success or failure.

Figure 6-13 Determining Self-Test Results



Following power-up and system reset, the system performs testing. Self-test results are indicated by the following:

- Fault light on the control panel
- Self-test LEDs on the modules
- Self-test display

During system self-test, the yellow Fault light on the control panel lights. If a module fails self-test, the Fault light remains lit. If all modules pass self-test, the Fault light goes off.

Each CPU, memory, and I/O adapter module has a green LED that lights when the module passes self-test. These LEDs can be viewed through the module enclosure from the front and rear (AlphaServer 8400 only) of the cabinet when the doors are open. If a module fails self-test, its green LED does not light. Although a dual-processor module's green LED will light only when both processors pass self-test, it is still possible to use the module if there is a – in the self-test display for one of the CPUs.

Some PIU modules have LEDs: the XMI DWLMA and clock card, the Futurebus+ DWLAA, and the PCI DWLPA/DWLPB I/O module. The *AlphaServer 8200/8400 Service Manual* describes their use.

The self-test display is discussed in detail in this chapter.

6.9 Testing Sequence

The self-test display shows the results of system self-test. The pass (+) or fail (–) status of each module is indicated. Dual-processor modules report the status of each CPU. The first self-test display in the following example is for an AlphaServer 8400; the second for an AlphaServer 8200.

Example 6-3 Testing Sequence

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1   0  NODE #
                      A  M  .  .  .  .  M  P   P  TYP
                      o  +  .  .  .  .  + ++  ++  ST1
                      .  .  .  .  .  .  . EE  EB  BPD ❶
                      o  +  .  .  .  .  + ++  +-  ST2 ❷
                      .  .  .  .  .  .  . EE  BE  BPD
                      +  +  .  .  .  .  + ++  +-  ST3
                      .  .  .  .  .  .  . EE  BE  BPD ❸
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .      C0 PCI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .      C1
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .      C2
            .  .  .  .  .  .  .  .  .  .  .  .  +   C3 PCI +
                      .  .  .  .  .  .  +  +         EISA +
                      . B0  .  .  .  . A0  .  .   ILV
                      .128  .  .  .  .256  .  .   384MB
AlphaServer 8400 Console V5.1-30, 14-JAN-1998, SROM V3.1
------------------------------------------------------------
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1  0  NODE #
                      A  M  .  .  P  .  .  .  .  TYP
                      o  +  .  . -+  .  .  .  .  ST1
                      .  .  .  . EB  .  .  .  .  BPD
                      o  +  .  . -+  .  .  .  .  ST2
                      .  .  .  . EB  .  .  .  .  BPD
                      +  +  .  . -+  .  .  .  .  ST3
                      .  .  .  . EB  .  .  .  .  BPD
                         .  +  +  +  .  +  +  +  C0 PCI +
            .  .  +  .  .  .  .  .  .  .  .  .     C1 PCI +
                      . A0  .  .  .  .  .  .  .  ILV
                      .128  .  .  .  .  .  .  .  128MB
AlphaServer 8200 Console V5.1-30, 14-JAN-1998, SROM V3.1❹
```

During the first round of testing each module runs its own self-test and the results are shown on the ST1 line. The boot processor is then determined (indicated by a B on the first BPD line). This processor then generates the results of testing to this point (❶).

If one processor on a dual-CPU module fails self-test (ST1), the failing CPU is displayed as a minus under the node # (for example, – +). The processor failure is also indicated by the control panel Fault light remaining lit after power-up. If both processors fail, the module is disconnected from the backplane to prevent faulty system operation. If a uni-processor (available with KN7CC and KN7CD processor modules) fails self-test (ST1) in a multiprocessor system, there is no indication of its failure in the self-test display. The failing processor is logically disconnected from the backplane and the processor failure is indicated by the control panel Fault light remaining lit after power-up. In both cases, the module self-test LED will be out.

Next, the processors run a second round of tests using the memory modules. In this testing the processor that had been designated as boot processor could fail (as shown in Example 6-3), so the boot processor is again determined. Results are displayed on the ST2 line (❷).

Finally, the processors run a third round of tests, the multiprocessing tests. Depending on configuration, there may be some delay at the ST3 line (around 20–60 seconds). Once again the boot processor is determined. The status of the boot processor and secondary processors is then displayed on the third BPD line (❸). The I/O adapter tests are then run.

Results of the I/O adapter self-test are displayed next.

The boot processor next configures memory and displays the configuration. Note that it is the boot processor determined at (❸) that displays the lines after the third BPD line. The final line before the console prompt contains the boot processor's SRM console version number, the date, and the SROM revision (❹).

Each line of the self-test display is described in detail in the following sections of this chapter.

6.10 Sample Self-Test Display

The primary processor reports the results of self-test. Results are displayed on the console terminal, as shown in Example 6-4, for an AlphaServer 8400. With an AlphaServer 8200, however, nodes 0 through 3 on the self-test display do not apply (see Section 2.5.1 and the second part of Example 6-3).

Example 6-4 Self-Test Results

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1   0 NODE #   ❶
                      A  A  M  .  .  .  M  P   P TYP      ❷
                      o  o  +  .  .  .  + ++  ++ ST1      ❸
                      .  .  .  .  .  .  . EE  EB BPD      ❹
                      o  o  +  .  .  .  + ++  +- ST2      ❺
                      .  .  .  .  .  .  . EE  BE BPD      ❻
                      +  +  +  .  .  .  + ++  +- ST3      ❼
                      .  .  .  .  .  .  . EE  BE BPD      ❽

                         +  +  +  +  .  +  +  +  C0 PCI +❾
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-❿
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -①
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5      ②
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6      ③
            .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +④
                      .  .  .  .  .  .  +  +        EISA+⑤

                      .  . B0  .  .  . A0  .  .  ILV      ⑥
                      .  .128  .  .  .256  .  .  384MB    ⑦
AlphaServer 8400 Console V5.1-30, 14-JAN-1998, SROM V3.1
     ⑧
     .
     .
     .
[lines showing configuring of I/O adapters]
    .
    .
    .
P01>>>
```

The self-test display reflects the system configuration listed in Table 6-3 and

Table 6-4. Each numbered item in the example is explained in Section 6.11 through Section 6.15. These sections assume the same system configuration (see Section 6.16.1 for the results of the **show configuration** command for this example).

Table 6-3 System Configuration for Example 6-4

Module	TLSB Node #	Module Type
KN7CD-AB	0	Dual processor; CPU0 fails ST2 and ST3 testing. CPU1 becomes boot processor.
KN7CD-AB	1	Dual processor; passes ST1, ST2, and ST3 testing.
MS7CC	2, 6	Memory (384 Mbytes).
KFTHA	7	I/O adapter leading to I/O channels.
KFTIA	8	I/O adapter leading to one I/O channel.

Table 6-4 I/O Subsystem Configuration for Example 6-4

Module	Node #	Module Type
KFTIA	8	I/O adapter
I/O Channel C0		Internal PCI; passes self-test.
I/O Channel C1		FBUS I/O adapter; fails self-test, no devices reported.
KFTHA	7	I/O adapter
I/O Channel C4		XMI I/O adapter; fails self-test, no devices reported.
I/O Channels C5, C6		(no devices attached here)
I/O Channel C7		PCI I/O adapter; passes self-test.
PCI/EISA	1	I/O adapter; passes self-test.

6.11 Self-Test Lines NODE# and TYP

The first two lines of the self-test printout provide the node number identification (NODE #) and the type of module (TYP).

Example 6-5 Self-Test Results: Node # and TYP

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1   0 NODE # ❶
                   A  A  M  .  .  .  M  P   P TYP     ❷
                   o  o  +  .  .  .  + ++  ++ ST1
                   .  .  .  .  .  .  . EE  EB BPD
                   o  o  +  .  .  .  + ++  +- ST2
                   .  .  .  .  .  .  . EE  BE BPD
                   +  +  +  .  .  .  + ++  +- ST3
                   .  .  .  .  .  .  . EE  BE BPD

                      +  +  +  +  .  +  +  +  C0 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6
            .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +
                   .  .  .  .  .  .  +  +        EISA+

                   .  . B0  .  .  . A0  .  .  ILV
                   .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V5.1-30, 14-JAN-1998, SROM V3.1

P01>>>
```

❶ The NODE # line lists the node numbers on the TLSB and I/O buses.

The nodes on this line are numbered in hexadecimal. Note that TLSB entries use nodes 0 through 8 (the 8400 TLSB has 9 slots). Nodes 4 through 8 are used for the AlphaServer 8200. The TLSB slot numbers and node numbers are identical. For the 8400, nodes 0 through 3 reflect the right-to-left position of the TLSB slots as you view the TLSB from the front of the cabinet and nodes 4 through 8 reflect the right-to-left position of the TLSB slots as you view the TLSB from the rear of the cabinet. For the 8200, nodes 4 through 8 reflect the right-to-left position of the TLSB slots as you view the TLSB from the front of the cabinet.

Each XMI bus has 14 slots. The XMI slot and node numbers are identical. Node numbers are 1 through E on the XMI.

Each PCI bus has 12 slots, numbered 0 through B.

Each EISA bus has 8 slots, numbered 1 through 8.

❷ The TYP line in the printout indicates the type of module at each TLSB node:

- An adapter: the KFTHA or KFTIA port module (A)
- A memory module (M)
- A processor (P) module

A period (.) indicates that the slot is not populated or that the module is not reporting.

6.12 Self-Test Lines ST and BPD

The next six lines of the self-test display provide test information on the processors (ST1, ST2, and ST3) and boot processor designation (BPD).

Example 6-6 Self-Test Results: ST and BPD

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1   0 NODE #
                      A  A  M  .  .  .  M  P   P TYP
                      o  o  +  .  .  .  + ++  ++ ST1    ❸
                      .  .  .  .  .  .  . EE  EB BPD    ❹
                      o  o  +  .  .  .  + ++  +- ST2    ❺
                      .  .  .  .  .  .  . EE  BE BPD    ❻
                      +  +  +  .  .  .  + ++  +- ST3    ❼
                      .  .  .  .  .  .  . EE  BE BPD    ❽

                         +  +  +  +  .  +  +  +  C0 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6
               .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +
                      .  .  .  .  .  .  +  +        EISA+

                      .  . B0  .  .  . A0  .  .  ILV
                      .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V5.1-30, 14-JAN-1998 SROM V3.1

P01>>>
```

❸ The ST1 line shows the results of self-test. The entries are:

- ++ (dual processor)
- + (pass)
- – (fail)
- o (does not apply)

NOTE: If a uniprocessor fails self-test (ST1) in a multiprocessor system or if both CPUs fail on a dual-processor module, there is no indication of the failure in the self-test display. The failing processor is logically disconnected from the backplane to prevent faulty system operation. The processor failure is indicated by the control panel Fault light remaining lit after power-up and the module's self-test LED being off.

Since the I/O port module does not have a module-resident self-test, its entry for the ST1 line is always "o".

❹ The BPD line indicates boot processor designation. When the system goes through self-test, the processor with the lowest ID number that passes self-test (ST1 line is +) becomes the boot processor, unless you intervene.

The results on the BPD line indicate:

- The boot processor (B)
- Processors eligible to become the boot processor (E)
- Processors ineligible tobecome the boot processor (D)

This BPD line is printed three times. After the first determination of the boot processor, the processors go through two more rounds of testing. Since it is possible for a processor to pass self-test (at line ST1) and fail ST2 or ST3 testing, the processors again determine the boot processor following each round of tests.

In Example 6-6, the first processor to pass self-test is chosen as the boot processor.

❺ During the second round of testing (ST2) all processors run additional CPU tests involving memory. In Example 6-6, results printed on the ST2 line indicate that the processor 1 at node 0 failed ST2, while processor 2 at node 0 passed ST2 (+ –).

❼ During the third round of testing (ST3) all processors run multi-processor tests, and the status of each processor is once again reported on the BPD line.

❽ The primary CPU also tests the I/O port module at this time.

6.13 Self-Test Lines C0, C1, C2, ... Cn

The I/O channel lines (C0, C1, C2, ...) of the self-test display provide information on the node numbers and self-test status for modules in the I/O subsystems, which are connected to the system bus.

Example 6-7 Self-Test Results: C0, C1. C2,...,Cn

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1   0 NODE #
                    A  A  M  .  .  .  M  P   P TYP
                    o  o  +  .  .  .  + ++  ++ ST1
                    .  .  .  .  .  .  . EE  EB BPD
                    o  o  +  .  .  .  + ++  +- ST2
                    .  .  .  .  .  .  . EE  BE BPD
                    +  +  +  .  .  .  + ++  +- ST3
                    .  .  .  .  .  .  . EE  BE BPD

                       +  +  +  +  .  +  +  +  C0 PCI +❾
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-❿
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -①
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5       ②
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6       ②
            .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +③
                    .  .  .  .  .  .  +  +        EISA+④

                    .  . B0  .  .  . A0  .  .  ILV
                    .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V5.1-30, 14-JAN-1998, SROM V3.1

P01>>>
```

The I/O channel lines (C0, C1, C2, ...) indicate the:

- I/O channel that connects the I/O adapter bus to the system bus
- DWLMA adapter self-test results
- XMI adapters' self-test results
- PCI adapters' self-test results
- Internal PCI self-test results
- FBUS+ adapters' self-test results

A + indicates an adapter passed self-test, a – indicates a failure, and a period (.) indicates that that node number is not used.

❾ In Example 6-7, the internal PCI (channel C0) and its options at nodes 0, 1, 2, 4, 5, 6, and 7 passed self-test as indicated by the + symbols.

❿ The DWLAA adapter accessed through C1 failed self-test (C2 FBUS–).

① The DWLMA adapter accessed through C4 failed self-test (C0 XMI –).

② I/O channels C5 and C6 are not used in this system configuration.

③ The PCI connected to channel C7 passed self-test as indicated by the + symbol.

④ The two EISA adapters at nodes 1 and 2 passed self-test.

The **show configuration** command gives additional information on I/O subsystems and adapters (see Section 6.16 and Section 7.5.25).

6.14 Self-Test Lines ILV and MB

The ILV line details the interleaving of the memories, and the MB line gives the Mbytes of each memory module and the total size of the system memory.

Example 6-8 Self Test Results: ILV and MB

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1   0 NODE #
                      A  A  M  .  .  .  M  P   P TYP
                      o  o  +  .  .  .  + ++  ++ ST1
                      .  .  .  .  .  .  . EE  EB BPD
                      o  o  +  .  .  .  + ++  +- ST2
                      .  .  .  .  .  .  . EE  BE BPD
                      +  +  +  .  .  .  + ++  +- ST3
                      .  .  .  .  .  .  . EE  BE BPD

                         +  +  +  +  .  +  +  +  C0 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6
               .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +
                      .  .  .  .  .  .  +  +        EISA+

                      .  . B0  .  .  . A0  .  .  ILV     ⑤
                      .  .128  .  .  .256  .  .  384MB   ⑥
AlphaServer 8400 Console V5.1-30, 14-JAN-1998, SROM V3.1

P01>>>
```

⑤ The ILV line contains a memory interleave value (ILV) for each memory. The default memory configuration algorithm attempts to maximize memory interleaving; arrays on a single memory module are interleaved by default. In Example 6-8, the memory modules at nodes 2 and 6 are each in a two-way system internal interleave (they are not interleaved with each other). There are two interleave sets; set A and set B. Information on memory interleaving can be obtained by entering a **show memory** command:

```
P00>>>  show memory
Set   Node   Size      Base Address       Intlv     Position
---   ----   ----    -------- --------   -----     --------
A      2     256 Mb  00000000 00000000  2-Way          0
B      6     128 MB  00000000 10000000  2-Way          0
```

A system with four memory modules is shown in the following example:

```
F  E  D  C  B  A  9  8  7  6   5   4  3  2  1  0  NODE #
                   .  B0  A3 A1  A0  .  .  .  .  ILV
                   . 128 128 128 256 .  .  .  . 640MB
```

In this example, there is one memory interleave set designated by the letter A. Note that there is no A2 designator on the ILV line; the memory at node 4 provides on-board interleaving, and so supplies both the A0 memory word and the A2 memory word (which is not reported on the ILV line). Also note that different size memory arrays can be interleaved into a single set by "stacking" the smaller arrays to interleave with their larger counterparts.

⑥ The line after the ILV line displays the size of each configured memory module in the system and gives the total size of system memory. In Example 6-8, the total size is 384 Mbytes.

6.15 Self-Test Identification Line

The last line of the self-test display gives the firmware revision numbers, the SROM revision numbers, and the date and time stamp of the console.

Example 6-9 Self-Test Results: Identification Line

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1   0 NODE #
                   A  A  M  .  .  .  M  P   P TYP
                   o  o  +  .  .  .  + ++  ++ ST1
                   .  .  .  .  .  .  . EE  EB BPD
                   o  o  +  .  .  .  + ++  +- ST2
                   .  .  .  .  .  .  . EE  BE BPD
                   +  +  +  .  .  .  + ++  +- ST3
                   .  .  .  .  .  .  . EE  BE BPD

                      +  +  +  +  .  +  +  +  C0 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6
            .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +
                   .  .  .  .  .  .  +  +        EISA+

                   .  . B0  .  .  . A0  .  .  ILV
                   .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V5-1.30, 14-JAN-1998 SROM V3.1
          ⑦                        ⑧           ⑨
P01>>>    ⑩
```

⑦ In Example 6-9, the primary processor indicates the version of the console firmware.

⑧ The date and time stamp indicates the production date of the running console.

⑨ The SROM Rev information indicates the primary processor's serial ROM version. The serial ROM contains the first level of console, diagnostic, and bootstrap code. This code initializes the CPU programmable features and diagnoses any faults detected along with the bootstrap path and bootstrapping code execution out to the main console program (the second level of console, diagnostic, and bootstrap code).

⑩ The console prompt indicates the number of the primary processor. In this example used to describe the self-test results, there are four CPUs, two on the module in node 0, and two on the module in node 1. Therefore, possible console prompts that could appear are P00>>>, P01>>>, P02>>>, and P03>>>. In this example, the second CPU on the first module was the first CPU to pass all three rounds of testing, so it became the boot processor.

6.16 Show Commands

To get system information for OpenVMS or DIGITAL UNIX systems, you can use the SRM commands show configuration, show network, or show device. The show command output identifies the subsystem, module, or device you may want to test. The following sections explain the show command output.

6.16.1 Show Configuration

Enter a show configuration command to display the system hardware configuration.

Example 6-10 Sample System Hardware Configuration

```
P00>>> show configuration  ❶

    Name                Type     Rev     Mnemonic    ❷
TLSB
0++ KN7CD-AB            8011     0000    kn7cd-ab0
1++ KN7CD-AB            8011     0000    kn7cd-ab1
2+  MS7CC               5000     0000    ms7cc0
6+  MS7CC               5000     0000    ms7cc1
7+  KFTHA               2000     0D02    kftha0
8+  KFTIA               2020     0000    kftia0

C0 Internal PCI connected to kftia      pci0        ❸
0+ ISP1020          10201077     0001    isp0
1+ ISP1020          10201077     0001    isp1
2+ DECchip 21040-AA   21011      0023    tulip0      ❹
4+ ISP1020          10201077     0001    isp2
5+ ISP1020          10201077     0001    isp3
6+ DECchip 21040-AA   21011      0023    tulip1
7+ PCI NVRAM          71011      0000    pci_nvram0

C1 FBUS connected to kftia0             fbus0
5- DWLAA               2003     0000    dwlaa0 XMI
```

Example 6-10 Sample System Hardware Configuration (Continued)

```
C4 XMI connected to kftha0                xmi0          ❸
1-   DWLMA            102A      020A   dwlma0

C7 PCI connected to kftha0                pci1
0+  SIO             4828086     0003    sio0
7+  KZPSA              8101     0000    kzpsa0
8+  KZPSA              8101     0000    kzpsa1
A+  DAC960            11069     0000    dac0
  Controllers on SIO                     sio0
0+  DECchip 21040-AA  21011     0000    tulip2
1+  FLOPPY                2     0000    floppy0
2+  KBD                   3     0000    kbd0
3+  MOUSE                 4     0000    mouse0

  EISA connected to PCI0 through sio0   eisa0
```

❶ The operator enters a **show configuration** command to show the system hardware configuration.

❷ The mnemonic for each processor, memory module, I/O adapter, and I/O subsystem is displayed in this column. You use mnemonics to identify the subsystem, module, or device you wish to test.

❸ This system has one XMI I/O subsystem and two PCI subsystems; in the mnemonic column they are identified as **xmi0**, **pci0**, and **pci1**. To test the xmi0 subsystem, you would issue the command **test xmi0**.

❹ To test an I/O adapter, such as the Ethernet adapter, you would enter the command **test tulip1**.

6.16.2 Show Network

Enter a show network command to display network devices.

Example 6-11 Sample Output of Show Network Command

```
P00>>> show network    ❶
polling for units on demna0  ❷,slot 1, xmi0...
exa0.0.0.1.0: 08-00-2B-27-D7-96  ❸
polling for units on demna1  ❹,slot 2, xmi1...
exb0.0.0.2.1: 08-00-2B-27-D7-BB  ❺
polling for units on tulip0, slot 2, bus 0, hose0...
ewa0.0.0.2.0: 08-00-2B-E2-B8-52
polling for units on tulip1, slot 6, bus 0, hose0...
ewb0.0.0.6.0: 08-00-2B-E2-B8-5F
P00>>>
```

❶ In Example 6-11, the operator enters a **show network** command to show the network adapters in the system.

❷ The first DEMNA adapter is identified by the demna0 mnemonic. This DEMNA adapter is in slot 1 of the I/O card cage connected to I/O channel 0.

❸ The unit number for demna0 is exa0.0.0.1.0. The hardware address is 08-00-2B-27-D7-96.

❹ The second DEMNA adapter is demna1. This adapter is in slot 2 of the I/O card cage connected to I/O channel 1.

❺ The unit number for demna1 is exb0.0.0.2.1. The hardware address is 08-00-2B-27-D7-BB.

The same information is given for the two Ethernet adapters (tulip0 and tulip1).

6.16.3 Show Device

Enter a show device command to display system devices.

Example 6-12 Sample Output of Show Device Command

```
P00>>> show device    ❶
polling for units on kzmsa0, slot 14, bus 0, xmi0...❷
dkb100.1.1.14.0 ❸   DKB100     ❹       RZ26L  ❺  440C ❻
dkb100.2.1.14.0      DKB200            RZ26L      440C
dkb100.3.1.14.0      DKB300            RZ26L      440C
polling for units on floppy 0, slot 0, bus 1, hose 3...
dva0.0.1100.3       DVA0                        RX26
polling for units on kfesb0, slot 4, bus 1, hose 3...
duc5.0.1004.3       RF3111$DIA5                 RF31
polling for units on isp0, slot 7, bus 0, hose 3...
polling for units on isp1, slot 8, bus 0, hose 3...
polling for units on dac0, slot 10, bus 0, hose 3...
polling for units on isp2, slot 0, bus 0, hose 4...
polling for units on isp3, slot 1, bus 0, hose 4...
polling for units on isp4, slot 4, bus 0, hose 4...
polling for units on isp5, slot 5, bus 0, hose 4...
P00>>>
```

❶ In Example 6-12, the operator enters a **show device** command to show all disks and tapes supported by the system.

❷ The first adapter polled is the KZMSA with the mnemonic kzmsa0. This KZMSA is located in slot 14 of the XMI card cage. The bus number is 0.

❸ Device information is displayed. Device mnemonics are listed in the first column. Device mnemonics are used with the **boot** command.

❹ The name of each disk as presented by the SCSI controller is displayed in the second column.

❺ The third column lists the device type.

❻ The fourth column lists the device firmware revision, if applicable. Not all devices report this.

6.17 Test Command

You can use the test command to test the entire system, an I/O subsystem, a module, a group of devices, or a specific device. Enter a show configuration command to see a list of the subsystems and devices that you may want to test. Examples of the test command are shown in the following sections.

Example 6-13 Sample Test Commands

```
1. P00>>> test -q     # Runs a system test.  Since a test
                      # run time was not specified, the
                      # entire system will be tested
                      # provided that testing does not
                      # exceed 10 minutes.  Status messages
                      # will not be displayed.

2. P00>>> test ms7cc*     # Tests all memory modules in the
                          # system.

3. P00>>> test xmi0       # Tests the XMI0 I/O subsystem.

4. P00>>> test dua80.0.0.2.1 # Tests the disk unit 80 on
                           # controller A.  The disk adapter
                           # located in slot 2 of the XMI1 card
                           # cage.
```

Two helpful **test** command options are outlined in Table 6-5. Environment variables commonly used with **test** are listed in Table 6-6.

Table 6-5 Test Command Options

Option	Meaning
-t <value>	Specifies the test run time in seconds. For a system test the default run time is 600 seconds (10 minutes). The run time for a device test is dependent upon the number and type of devices selected for testing. The **-t** option takes any value between 30 and 99999999
-q	Disables the status messages displayed by default as exerciser processes are started and stopped during testing. **-q** sets **d_verbose** to zero.

Table 6-6 Test Command Environment Variables

Environment Variable	Meaning
d_report	Specifies the type of error report displayed. Values are **summary** and **full**. **Full** is the default value.
d_harderr	Specifies the action taken when a hard error occurs. Values are **halt** or **continue**. **Halt** is the default value.
d_softerr	Specifies the action taken when a soft error occurs. Values are **halt** or **continue**. **Continue** is the default value.

6.17.1 Testing the System

To test the entire system (except for memory), enter the test command.

Example 6-14 Sample Test Command, System Test

```
P00>>> test -t 180    ❶
Configuring system...
Testing system...
Type Ctrl/C to abort     ❷


Starting network exerciser on exa0.0.0.4.0 in external  ❸
 mode (id #62) Starting network exerciser on exb0.0.0.5.0
in external mode (id #63) Starting network exerciser on
exc0.0.0.4.1 in external mode (id #64) Starting device
exerciser on dua10.14.0.1.0 (id# 67)
Stopping memory exerciser (id #59)
Stopping memory exerciser (id #60)
Test time has expired... ❹

Stopping network exerciser on exa0.0.0.4.0 (id #62)  ❺
Stopping network exerciser on exb0.0.0.5.0 (id #63)
Stopping network exerciser on exc0.0.0.4.1 (id #64)
Stopping device exerciser on dua10.14.0.1.0 (id# 67)
 -------------Testing done ---------------
Shutting down drivers...
Shutting down units on tulip1, slot 2, bus 0, hose 12...
Shutting down units on tulip2, slot 6, bus 0, hose 12...
Shutting down units on demna0, slot 1, bus 0, hose 0...
Shutting down units on demfa0, slot 3, bus 0, hose 0...
Shutting down units on kzmsa0, slot 14, bus 0, hose 0...
Shutting down units on tulip0, slot 12, bus 0, hose 3...
P00>>>   ❻
```

In Example 6-14:

❶ At the console prompt, the operator enters a **test -t 180** command. The option **-t** specifies a system test run time of 180 seconds.

❷ To stop execution of the test command before normal completion, use Ctrl/C (^C). Termination using ^C may take a number of seconds depending upon the particular configuration being tested.

❸ Status messages indicate the start of the console-based exercisers.

❹ The exercisers run for 180 seconds; not enough time to test the entire system, so this status message is displayed.

❺ All exercisers are stopped, as indicated by the status messages.

❻ The console prompt returns.

6.17.2 Testing a Subsystem

To test a portion of the system, such as an I/O subsystem, enter the test command and the I/O subsystem mnemonic. I/O subsystem mnemonics are displayed when you enter a show configuration command.

Example 6-15 Sample Test Command, I/O Subsystem Test

```
P00>>> test xmi0     ❶
Configuring xmi0 subsystem...
Testing xmi0
Starting DWLMA exerciser on dwlma0
Stopping DWLMA exerciser on dwlma0    ❷
Initializing DEMNA0
Initializing CIXCD0
Initializing DEMFA0
DEMNA0 self-test passed  ❸
CIXCD0 self-test passed
DEMFA0 self-test passed
Starting network exerciser on exa0.0.0.4.0 (id #31)in internal mode ❹
Starting network exerciser on fxa0.0.0.e.0 (id #33)in internal mode
Starting device exerciser on dua0.0.0.6.0 (id #34) in READ-ONLY mode
Stopping device exerciser on dua0.0.0.6.0 (id #34) Starting device
exerciser on dub0.0.0.a.0 (id #35) in READ-ONLY mode
Stopping device exerciser on dub0.0.0.a.0 (id #35) Starting device
exerciser on dua1.1.0.6.0 (id #36) in READ-ONLY mode
Stopping device exerciser on dua1.1.0.6.0 (id #36)
Starting device exerciser on dub1.1.0.a.0 (id #37) in READ-ONLY mode
Stopping device exerciser on dub1.1.0.a.0 (id #37)
Starting device exerciser on dua2.2.0.6.0 (id #38) in READ-ONLY mode
Stopping device exerciser on dua2.2.0.6.0 (id #38) Stopping all testing
Stopping network exerciser on exa0.0.0.4.0 (id #31) Stopping network
exerciser on fxa0.0.0.e.0 (id #33)
--------------Testing done -----------
Shutting down drivers...
Shutting down units on tulip1, slot 2, bus 0, hose 12...
Shutting down units on tulip2, slot 6, bus 0, hose 12...
Shutting down units on demna0, slot 1, bus 0, hose 0...
Shutting down units on demfa0, slot 3, bus 0, hose 0...
P00>>>   ❺
```

❶ At the console prompt, the operator enters a **test xmi0** command to test all adapters and I/O devices on the XMI I/O bus.

❷ The status message indicates that the DWLMA adapter passes all tests. Next, the adapters in the XMI card cage are initialized.

❸ All XMI adapters pass self-test.

❹ The status messages report that all exercisers started.

❺ The console prompt returns.

6.17.3 Testing a Module or Devices

To test a processor, memory module, or an I/O adapter and its associated devices, enter the test command and the correct mnemonic. Mnemonics are displayed when you enter a show configuration or a show device command.

Example 6-16 Sample Test Command, I/O Adapter Test

```
P00>>> set d_report full
P00>>> test demna0❶
Console is in diagnostic mode
Network adapter test selected for runtime of 120 seconds
Initializing demna0
Waiting for self tests to complete...
Self-test passed on device demna0❷
Configuring demna0
polling for units on demna0, slot 2, bus 0, xmi0...
exa0.0.0.2.1: 08-00-2B-2A-CF-BE
Starting network exerciser on exa0.0.0.2.1 (id #41da) in internal
loopback mode
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Test time has expired...

Stopping all testing... please wait

Stopping network exerciser on exa0.0.0.2.1 (id #41da)

 ---------Testing done ------------

 Shutting down drivers...
Shutting down units on demna0, slot 2, bus 0, hose 1...❸

P00>>>
```

In Example 6-16:

❶ At the console prompt, the operator enters **test demna0**. The device mnemonic, **demna0**, was identified by issuing a **show configuration** command.

❷ The DEMNA0 adapter is initialized and passes self-test.

❸ The status messages report that the network exercisers started and completed.

Example 6-17 Sample Test Command, Memory Module Test

```
P00>>> test ms7cc0 ❶
Testing ms7cc0
Type Ctrl/C to abort
Starting memory exerciser, running on kn7ce0 (id #77)   ❷
Stopping memory exerciser on kn7ce0 (id #77)
Done testing...
P00>>>
```

In Example 6-17:

❶ Enter **test ms7cc0**.

❷ The ms7cc0 memory module is tested by the memory exerciser, a series of tests executed from the processor module.

NOTE: To test all memories on your system, type:

test ms7c*

Example 6-18 Sample Test Command, Testing Devices

```
P00>>> test kz*          ❶
Device adapter test selected for runtime of 120 seconds
Type ^C to stop testing
Self-test passed on device kzmsa0
Configuring kzmsa0
polling for units on kzmsa0, slot 14, bus 0, xmi0...
dkb0.0.1.14.0     DKB0                      RZ26L    440C
dkb100.1.1.14.0   DKB100                    RZ26L    440C
dkb200.2.1.14.0   DKB200                    RZ29B    0006
dkb300.3.1.14.0   DKB300                    RZ26L    440C
Starting device exerciser on dkb0.0.1.14.0 (id #15b) in READ-ONLY mode
Stopping device exerciser on dkb0.0.1.14.0 (id #15b)
Starting device exerciser on dkb100.1.1.14.0 (id #1c2) in READ-ONLY mode
Stopping device exerciser on dkb100.1.1.14.0 (id #1c2)
Starting device exerciser on dkb200.2.1.14.0 (id #226) in READ-ONLY mode
Stopping device exerciser on dkb200.2.1.14.0 (id #226)
Starting device exerciser on dkb300.3.1.14.0 (id #27f) in READ-ONLY mode
Stopping device exerciser on dkb300.3.1.14.0 (id #27f)
Starting device exerciser on dkb0.0.1.14.0 (id #2f1) in READ-ONLY mode
Stopping device  exerciser on dkb0.0.1.14.0 (id #2f1)
Starting device exerciser on dkb100.1.1.14.0 (id #355) in READ-ONLY mode
Stopping device exerciser on dkb100.1.1.14.0 (id #355)
Starting device exerciser on dkb200.2.1.14.0 (id #3bc) in READ-ONLY mode
Stopping device exerciser on dkb200.2.1.14.0 (id #3bc)
Starting device exerciser on dkb300.3.1.14.0 (id #412) in READ-ONLY mode
Stopping device exerciser on dkb300.3.1.14.0 (id #412)
Starting device exerciser on dkb0.0.1.14.0 (id #487) in READ-ONLY mode
Stopping device exerciser on dkb0.0.1.14.0 (id #487)
Starting device exerciser on dkb100.1.1.14.0 (id #4f3) in READ-ONLY mode
Stopping device exerciser on dkb100.1.1.14.0 (id #4f3)
Starting device exerciser on dkb200.2.1.14.0 (id #55a) in READ-ONLY mode
Stopping device exerciser on dkb200.2.1.14.0 (id #55a)
Starting device exerciser on dkb300.3.1.14.0 (id #5a8) in READ-ONLY mode
Stopping device exerciser on dkb300.3.1.14.0 (id #5a8)
Starting device exerciser on dkb0.0.1.14.0 (id #622) in READ-ONLY mode
Stopping device exerciser on dkb0.0.1.14.0 (id #622)
Starting device exerciser on dkb100.1.1.14.0 (id #689) in READ-ONLY mode
Stopping device exerciser on dkb100.1.1.14.0 (id #689)
Starting device exerciser on dkb200.2.1.14.0 (id #6ed) in READ-ONLY mode
Stopping device exerciser on dkb200.2.1.14.0 (id #6ed)
Starting device exerciser on dkb300.3.1.14.0 (id #746) in READ-ONLY mode
Stopping device exerciser on dkb300.3.1.14.0 (id #746)
Starting device exerciser on dkb0.0.1.14.0 (id #7b8) in READ-ONLY mode
Time has expired...❷
Stopping device exerciser on dkb0.0.14.0 (id #7b8)
Stopping all testing
---------Testing done -----------
Shutting down drivers...
Shutting down units on kzmsa0, slot 14, bus 0, hose 0...
P00>>>  ❸
```

In Example 6-18:

❶ At the console prompt, the operator enters **test kz***. The wildcard entered will test all "**kz**" devices.

❷ Not enough time to test devices, so this status message is displayed.

❸ Testing is stopped, drivers halted, and the console prompt returns.

Example 6-19 Testing Network Adapters

```
P00>>> set d_report full  ❶
P00>>> test demna0  ❷
Console is in diagnostic mode
Network adapter test selected for runtime of 120 seconds
Initializing demna0
Waiting for self tests to complete...
Self-test passed on device demna0
Configuring demna0
polling for units on demna0, slot 2, bus 0, xmi0...
exa0.0.0.2.1: 08-00-2B-2A-CF-BE
Starting network exerciser on exa0.0.0.2.1 (id #41da) in
internal loopback mode
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Still testing...network
Test time has expired...

Stopping all testing... please wait

Stopping network exerciser on exa0.0.0.2.1 (id #41da)

 ---------Testing done ------------

 Shutting down drivers...
Shutting down units on demna0, slot 2, bus 0, hose 1...

P00>>> set d_trace off   ❸
P00>>> test ewa0  ❹
Console is in diagnostic mode
Network adapter test selected for runtime of 120 seconds
Starting network exerciser on ewa0.0.0.12.1 (id#334e) in
internal loopback mode
Stopping network exerciser on ewa0.0.0.12.1 (id#334e)
P00>>>
```

❶ This command requests an extended report format for following tests.

❷ The demna is the network adapter for an XMI bus. The full test is conducted and displayed. When the time allotted expires, testing stops.

❸ This command requests that full tracing for the test be omitted. Note that the listing following at ❹ does not "trace" the testing of the adapter, as is done for the demna, above.

❹ This command requests that ewa0, a network adapter on the internal PCI bus of a KFTIA, be tested. A brief description of the testing conducted follows.

6.18 Error Reports

In the event of an error, either a summary or a full error report is displayed at the console. Error reports are specified by setting the d_report environment variable. See Example 6-20 and Example 6-21.

Example 6-20 Sample Summary Error Report

```
P00>>> set d_report summary  # Command to set the type of error
                             # report to summary (default).
P00>>> test kzmsa            # Command to test the kzmsa adapter
                             # and its devices.
Configuring kzmsa...
Testing kzmsa
Type Ctrl/C to abort

Initializing KZMSA
kzmsa self-test passed
Starting device exerciser on duc1.0.0.12.0 (id #20) ❶
Starting device exerciser on duc2.0.0.12.0 (id #21)
Starting device exerciser on duc4.0.0.12.0 (id #22)

Bad MSCP status (4|0) received
Failed to send Read to duc4.0.0.12.0

*** Hard Error - Error #1 on FRU: duc4.0.0.12.0  ❷
Error in read of 2097152 bytes at location 07E00000 from duc4.0.0.12.0.
      ID  Program          Device     Pass Hard/Soft Test     Time
--------❸--------❹---------------❺--------❻---❼---❽---❾---------❿
     481   dsk_ex   duc4.0.0.12.0       18     1    0    1 12:07:01

***End of Error***
```

In Example 6-20:

❶ Testing begins on each disk.

❷ A hard error, error #1, is reported on FRU duc4.0.0.12.0, a disk associated with the kzmsa adapter. The three types of errors reported are hard, soft, and fatal. The error number, in this case error #1, corresponds to the location of the actual error report call within the source code for the failing diagnostic. The FRU, or field-replaceable unit, is duc4.0.0.12.0.

❸ The process identification number (ID) is 481. This is the process ID of the failing diagnostic.

❹ The program running when the error occurred is dsk_ex, or, the disk exerciser.

❺ The target device selected for testing at the time of the error. The device name in this field may or may not match the device mnemonic displayed in the FRU field (❷).

❻ The current pass count, 18, is the number of passes executed when the error was detected.

❼ The current hard error count is 1. The hard and soft (❽) error counts are the number of errors detected and reported by the failing diagnostic since the testing started.

❾ In this example, the failing test number is 1.

❿ The time stamp shows when the error occurred.

Example 6-21 Sample Full Error Report

```
P00>>> set d_report full  # Command to set the type of
                          # error report to full.
P00>>> test demna0        # Command to test the DEMNA
                          # adapter with the mnemonic
                          # demna0.

Console in diagnostic mode
Network adapter test selected for runtime of 120 seconds
Initializing demna0
Waiting for self-test to complete . . .
demna0 self-test passed
Starting network exerciser on exa0.0.0.1.0 (id #183)
***Hard Error - Error #15 on FRU: exa0.0.0.1.0
External loopback error, no packet received

      ID  Program          Device     Pass Hard/Soft Test     Time
-------- -------- --------------- -------- --------- ---- --------
     183   net_ex    exa0.0.0.1.0        5    1    0    1 12:31:01


Address   Expected  Received   ❶
--------  --------  --------
00240000  AAAAAAAA  AAAAAAAA
00240004  55555555  45554555   ❷
00240008  CCCCCCCC  CCCCCCCC
0024000C  33333333  33333333
00240010  88888888  88888888
00240014  77777777  77777777
00240018  FFFFFFFF  FFFFFFFF
*** End of Error ***
```

NOTE: Except for the extended error information shown in Example 6-21, the information shown in a full and a summary error report is the same. See Example 6-20 for a description of the common error report entries.

In Example 6-21:

❶ This full error report shows a list of addresses and the expected and received data values for each address.

❷ The expected value at address 00240004 was 55555555. The received value was 45554555.

Chapter 7

SRM Console Commands

This chapter describes the SRM console program's command language, console special characters, console environment variables, and console commands.

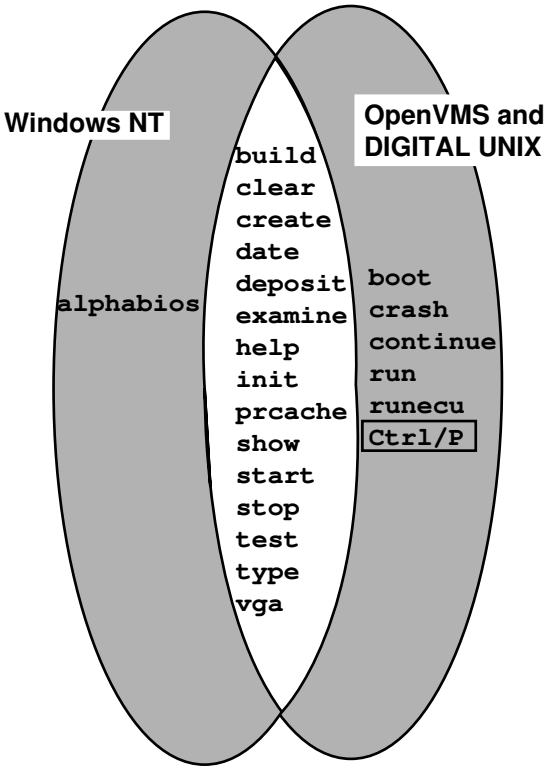
Sections in this chapter include:

- Overview
- SRM Command Syntax
- SRM Console Special Characters
- SRM Console Environment Variables
- SRM Console Commands

7.1 Overview

The SRM console firmware has evolved to provide capabilities for the three operating systems offered with DIGITAL AlphaServer 8200 and 8400 systems.

Figure 7-1 SRM Console Command and Feature Breakdown



BX-0659-98

For OpenVMS and DIGITAL UNIX systems, SRM console commands allow you to boot the OpenVMS and DIGITAL operating systems, display the configuration, and verify the system. The control character Ctrl/P allows you to halt the operating system and return to SRM console mode; the SRM command **continue** will then restart the operating system where it left off.

For Windows NT systems, SRM commands allow you to enter the AlphaBIOS firmware used for booting Windows NT, displaying the configuration, and verifying the system. The configuration nomenclature displayed and with which you verify the system is different from that used by the AlphaBIOS firmware, which is in turn different from the nomenclature used by the Windows NT operating system.

For example, a disk identified by the name dka100 with the SRM console command **show device** may be identified as the **c:** disk on your Windows NT system. Furthermore, Ctrl/P and **continue** do not work for Windows NT systems. See Section 8.2 for information on transferring between the Windows NT operating system, **AlphaBIOS Setup**, and the SRM console mode.

When a command, control character, or environment variable is specific to an operating system, that fact is so indicated in the headings and/or text of this chapter. Otherwise, they apply to all operating systems.

In any case, when the system is in SRM console mode, the system is halted and the SRM console firmware is executing. The operator communicates with the firmware through the console terminal, which displays the following prompt:

```
Pnn>>>
```

where *nn* is 00 to 13 (for AlphaServer 8400 systems) or 08 to 13 (for AlphaServer 8200 systems), depending on which TLSB slot the processor module is in and which processor on the module is the primary processor.

7.2 SRM Command Syntax

The SRM console command language has syntax rules for forming commands. Commands can contain up to 80 characters on a single line, can be abbreviated, and accept options. Tabs and spaces are compressed.

Table 7-1 SRM Console Command Language Syntax

Command Parameter	Attribute or Action
Length	256 characters maximum, including the terminating carriage return.
Case	Upper- or lowercase characters are accepted.
Abbreviation	Varies with the command; usually the shortest unique combination of letters
Options	Can appear after the command keyword or after any symbol or number in the command. Begin with a hyphen (-) and must be preceded by at least one space.
Numbers	Most numbers are decimal. Addresses and numbers used with the **deposit** command are hexadecimal.
No characters	Null command; no action taken
Multiple adjacent spaces and tabs	Compressed to a single space.

Length: The SRM console program accepts commands of up to 255 characters. This does not include the terminating carriage return or any characters deleted as the command is entered. A command longer than 80 characters, without the backslash character (see Section 7.2) causes the display of an error message.

Case: Upper- or lowercase characters can be used for input. Characters are displayed in the case they are entered.

Abbreviation: Commands and options can be abbreviated by dropping characters from the end of words. You must enter the minimum number of characters to identify the keyword unambiguously. All characters specified must match a keyword to be accepted. For example, although **E** uniquely identifies the **examine** command, **Exmn** is not a valid abbreviation. In the command reference sections that follow, characters that can be omitted appear in square brackets ([ ]). Abbreviation of environment variables (see Section 7.3) is allowed with the **show** command.

Options: You can use command options, to define or modify the environment, after the command keyword or after any symbol or number in the command. See individual keyword descriptions for examples.

Numbers: Most numbers in console commands are in decimal notation except for addresses and those used in the **deposit** command which are in hexadecimal. The default radix can be overridden by preceding decimal numbers with %d, binary with %b, hexadecimal with %x, and octal with %o. Refer to the individual command descriptions. Register names (R0, R1, and so on) 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. The console prompt returns. The console supports command line recall and editing.

Spaces: Multiple adjacent spaces and tabs are compressed and treated as a single space. The console program ignores leading and trailing spaces.

7.3 SRM Console Special Characters

The console program supports control characters, entered by holding down the Control (Ctrl) key and pressing the desired key, and other special characters.

Table 7-2 SRM Console Special Characters

Character	Function
Return	Carriage return; ends a command line.
Backslash	Line continuation.
<X]	Delete key; deletes previously typed character.
Help	By itself, displays first-level help. When pressed after part of a command, displays options available.
Ctrl/A, F14	Toggles between insertion/overstrike mode.
Ctrl/B ^ (up-arrow)	Recall previous command(s).
Ctrl/C	Terminate running process.
Ctrl/D, < (left-arrow)	Move cursor left one position.
Ctrl/E	Move cursor to end of line.
Ctrl/F, > (right-arrow)	Move cursor right one position.
Ctrl/H, BS, F12	Move cursor to beginning of line.
Ctrl/J	Delete word.
Ctrl/O	Stop output to console terminal for current command. Toggles between enable/disable.
Ctrl/P	In SRM console mode, acts like Ctrl/C. In program mode, on the OpenVMS or DIGITAL UNIX operating systems, causes the boot processor to halt and begin running the SRM console program. (To enter the SRM console from Windows NT program mode, see Section 8.2.)

Table 7–2 SRM Console Special Characters (Continued)

Character	Function
Ctrl/Q	Resume output to console terminal.
Ctrl/R	Redisplay the current line.
Ctrl/S	Stop output to console terminal.
Ctrl/U	Delete entire line.
*	Wildcarding for certain commands.
" "	Quotes for **set** environment variable name.
#	Comment specifier.

Return terminates command line input. No action is taken on a command line until it is terminated by a carriage return. If no characters are entered and the Return key is pressed, it is treated as a null command. No action is taken, and the console prompts for input. Carriage return is echoed as carriage return, line feed.

Backslash (\) allows continuation across lines from the terminal; must be the last character on the line to be continued.

When the **Delete** key is pressed, the console deletes the character previously typed.

Help provides additional information on console commands.

Ctrl/A or **F14** toggles between insertion mode and overstrike mode for command line editing. The default mode is overstrike.

Ctrl/B or **up-arrow/down-arrow** recall the previous command(s). The last 16 commands are stored in the recall buffer.

Ctrl/C terminates the current command. Echoed as ^C, Ctrl/C clears Ctrl/S and also resumes output that was suspended using Ctrl/O. When Ctrl/C is entered as part of a command line, the line is deleted as if you entered Ctrl/U. Ctrl/C has no effect as part of a binary data stream.

Ctrl/D or **left-arrow** moves the cursor one position to the left.

Ctrl/E moves the cursor to the end of the line.

Ctrl/F or **right-arrow** moves the cursor right one position.

Ctrl/H, **Backspace**, or **F12** moves the cursor to the beginning of the line.

Ctrl/J deletes previously typed word.

Ctrl/O stops output to the console terminal until Ctrl/O is entered again. Ctrl/O is echoed as ^O followed by a carriage return and is not echoed when output is reenabled.

Output is also reenabled when the console prompts for a command, issues an error message, enters program mode, or when Ctrl/P is entered. It is not reenabled by displaying a **repeat** command.

Ctrl/P works like Ctrl/C and is echoed as ^C, if the console terminal is in SRM console mode. Under the OpenVMS and DIGITAL UNIX operating systems, if the console terminal is in program mode and is secured, Ctrl/P is not echoed, but is passed to the operating system for processing. If the console terminal is in program mode and is not secured, Ctrl/P halts the processor and begins the SRM console program. See the **continue** command for additional information. Ctrl/P is not available when the Windows NT operating system is running. See Section 8.2 for information on how to switch between Windows NT, **AlphaBIOS Setup**, and the SRM console.

Ctrl/Q resumes console output to the console terminal that was suspended with Ctrl/S. Additional Ctrl/Q strokes are ignored. Ctrl/Q is not echoed.

Ctrl/R is echoed as ^R, followed by a carriage return, line feed, and printing the current command 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. Ctrl/S is not echoed.

Ctrl/U discards all characters that you entered on the current line. It is echoed as ^U, followed by a carriage return, line feed, and a new prompt.

***** allows wildcarding with device names and environment variables. With the **build** command, wildcarding is permitted for a class of device; for example, KFTIA*. Wildcarding is allowed with the following commands:

- **build**
- **clear**
- **initialize**
- **set -d**
- **show**
- **show configuration**
- **show device**
- **show <envar>**
- **show network**
- **test**

Double quotes (**" "**) allow you to denote a string for environment variable assignment.

allows you to enter a comment. All characters following the **#** are recognized as a comment only. Exceptions include the above control characters.

7.4 SRM Console Environment Variables

Console environment variables allow the user to modify the way the console commands operate.

An environment variable is a name and value association maintained by the console program. The value associated with an environment variable is an ASCII string (up to 127 characters in length) or an integer. Some environment variables can be set to tailor the recovery behavior of the system on power-up and, for OpenVMS and DIGITAL UNIX systems, after system failures.

Volatile environment variables are initialized to their default by a system reset. Nonvolatile environment variables stay set across system failures.

Environment variables can be created, modified, displayed, and deleted using the SRM commands **create**, **set**, **show**, and **clear**. A default value is associated with any variable that is stored in the EEPROM area.

Table 7-3 lists the predefined console environment variables, their attributes, and their functions.

Table 7-3 Environment Variables

Variable	Attribute	Function
arc_enable	Nonvolatile	Enables you to issue to **run** and **runecu** commands for OpenVMS and DIGITAL UNIX systems. Default value is **off**.
auto_action	Nonvolatile	Specifies the action the console will take following an error halt or power-up. Values are: **restart** - For OpenVMS and DIGITAL UNIX systems, automatically restart. If restart fails, boot the operating system. Not applicable to Windows NT systems. **boot** - Automatically boot the operating system. OpenVMS and DIGITAL UNIX systems will use as the default device that defined by manufacturing (for factory-installed software), or a default boot device selected by setting the **bootdef_dev** environment variable. For Windows NT, the default boot path is that defined by manufacturing (for factory-installed software), or a default boot path defined when the system is installed, using the Windows NT system itself. **halt** (default) - Enter SRM console mode.
bootdef_dev	Nonvolatile	For OpenVMS or DIGITAL UNIX systems, the default device or device list from which booting is attempted when no device name is specified by the **boot** command. Not used with Windows NT.
boot_file	Nonvolatile	For OpenVMS or DIGITAL UNIX sysems, the default file name used for the primary bootstrap when no file name is specified by the **boot** command, if appropriate. Not used with Windows NT.
boot_osflags	Nonvolatile	For OpenVMS or DIGITAL UNIX systems, additional parameters to be passed to the system software during booting if none are specified by the **boot** command with the **-flags** specifier. Not used with Windows NT.

Table 7-3 Environment Variables (Continued)

Variable	Attribute	Function
boot_reset	Nonvolatile	For OpenVMS and DIGITAL UNIX systems, the default file name used for the primary bootstrap when no file name is specified in the **boot** command. Not used with Windows NT.
console	Nonvolatile	Defines the type of console device. **serial** A serial console terminal **graphics** A graphics console device **both** Both a serial and graphics console device
cpu	Nonvolatile	Selects the current boot processor.
cpu_enabled	Nonvolatile	A bitmask indicating which processors are enabled to run (leave console mode). Default is **0xffff**.
cpu_primary	Nonvolatile	A bitmask indicating which processors are enabled to become the next boot processor, following the next reset. Default is **0xffff**.
d_harderr	Volatile	Determines action taken following a hard error. Values are **halt** (default) and **continue.** Applies only when using **test**.
d_report	Volatile	Determines level of information provided by the diagnostic reports. Values are **summary** and **full** (default). Applies only when using **test**.
d_softerr	Volatile	Determines action taken following a soft error. Values are **continue** (default) and **halt**. Applies only when using **test**.
d_trace	Nonvolatile	Specifies whether or not to display **test** trace messages. **off** (default) - Disables trace messages **on** - Enables trace messages
dump_dev	Nonvolatile	Device to which dump file is written if the system crashes, if supported by the operating system.

(Continued on next page)

Table 7-3 Environment Variables (Continued)

Variable	Attribute	Function
enable_audit	Nonvolatile	If set to **on** (default), enables the generation of audit trail messages. If set to **off**, audit trail messages are suppressed, Console initialization sets this to **on**.
ew*0_loop_count	Nonvolatile	Specifies the number of times each message is looped for a **test** command exercising a PCI network adapter.
ew*0_loop_inc	Nonvolatile	Specifies the amount the message size is increased from message to message.
ew*0_loop_patt	Nonvolatile	Specifies the type of data pattern to be used when doing loopback. Current patterns are accessed by: 0xffffffff All the patterns 1 All zeroes 2 All ones 3 All A's 4 Incrementing 5 Decrementing
ew*0_loop_size	Nonvolatile	Specifies the size of the loop data to be used.
ew*0_lp_msg_node	Nonvolatile	Specifies the number of messages originally sent to each node.
ew*0__mode	Nonvolatile	Specifies the value for the Ethernet port node when it is started. Allowed values are: Auto-Sensing AUI Twisted-Pair Full Duplex, Twisted Pair BNC FastFD (Full Duplex) Auto-Negotiate

Table 7-3 Environment Variables (Continued)

Variable	Attribute	Function
interleave	Nonvolatile	Specifies interleave arrangement for memory modules. Defined values are: DEFAULT The SRM console decides how to interleave memories. NONE Memory modules are not interleaved.
os_type	Nonvolatile	Used to store operating system type that user will be operating under. Values are **nt**, **vms**, **openvms**, **osf**, and **unix**.
simm_callout	See "Function"	Specifies that if any memory errors are detected during system self-test, the SRM console should identify the failing SIMM. Defined values are: OFF (default) ON Identify failing SIMM. The "ON" value is preserved across one system initialization only, then reset to OFF.
tta0_baud	Nonvolatile	Sets the serial console terminal port baud rate. Allowable values are 300, 600, 1200, 2400, 4800, and 9600. The default value is 9600.

7.5 SRM Console Commands

Console commands provide the capabilities to examine and modify system state. Additionally, they allow tests to be directed to functional components of the system.

7.5.1 AlphaBIOS (for Windows NT only)

The alphabios command is used to enter the AlphaBIOS firmware, either to boot the Windows NT operating system, or to use the AlphaBIOS Setup menus.

Example 7-1 AlphaBIOS Command

```
P08>>> alphabios
```

The alphabios command syntax is:

al[phabios]

Before you enter AlphaBIOS from the SRM console, make sure the **os_type** environment variable is set to **nt**, and that you have a graphics console monitor connected to a KFE72 adapter in your system. Use the **set console both** command to ensure that both a serial and a graphics monitor are usable.

You type **alphabios** to boot the Windows NT operating system, or to use the **AlphaBIOS Setup** menus. Once you type **alphabios**, initializing messages appear, and the displays end with the boot screen. To enter **AlphaBIOS Setup**, press the F2 key before the Windows NT system is booted. See Chapter 8 for a description of **AlphaBIOS Setup** menus.

*NOTE: The command **ar[c]** can also be used to enter AlphaBIOS firmware.*

7.5.2 Boot

The boot command boots the DIGITAL UNIX and OpenVMS operating systems, and the Loadable Firmware Update (LFU) utility. See Section 5.5 for information on booting Windows NT systems.

Example 7-2 Boot Command

```
P00>>> show boot*    ❶
boot_dev                dkc300.3.0.7.1
boot_file
boot_osflags            a
boot_reset              OFF
bootdef_dev             dkc300.3.0.7.1
booted_dev
booted_file
booted_osflags
P00>>> boot    ❷
 (boot dkc300.3.0.7.1 -flags a)
SRM boot identifier: scsi 1 7 0 3 300 ef00 81011
boot adapter: kzpsa0  rev 0 in bus slot 7 off of kftha0 in
TLSB slot 8
block 0 of dkc300.3.0.7.1 is a valid boot block
reading 16 blocks from dkc300.3.0.7.1
bootstrap code read in
Building FRU table.......
FRU table size = 0x2632
    .
    .
    .
[initialization and configuration information]
    .
    .
    .
Digital UNIX Version V4.0 (thenut.Eng.PKO.DEC.Com) console

login:    ❸
```

The boot command syntax is:

b[oot] [<device_name>] [<flags>]

Where **<device_name>** specifies a particular disk, and **flags** applicable to OpenVMS and DIGITAL UNIX are described in detail in Appendix A.

7.5.3 Building the EEPROM

The build -e command is used to initialize a module's EEPROM during installation or to restore a corrupted serial EEPROM image.

Example 7-3 Building the EEPROM

```
P00>>> build -e          # Initialize the EEPROM
                         # on kn7cf.
Build EEPROM on kn7cf-ab0? [Y/N]> Y
      EEPROM built on kn7cf-ab0
P00>>>
```

The **build** -**e** command syntax is:

bu[ild] -**e** <**device**>

where <**device**> is a processor module. If you are restoring a corrupted EEPROM, you will be prompted to supply the system serial number and module serial, part, and firmware revision numbers. Since some environment variables are volatile, before upgrading you should refer to Table 7-3 and use the **show** <**envar**> command (see Section 7.5.29) to display present environment variable values. After rebuilding, use the **set** <**envar**> command (see Section 7.5.21) to set the environment variables to their desired values.

7.5.4 Building the Nonvolatile RAM

The build -n command is used to initialize the CPU's nonvolatile RAM (NVR).

Example 7-4 Building the Nonvolatile RAM

```
P00>>> build -n

Build NVR on kn7cd-ab0? [Y/N]Y

NVR built on kn7cd-ab0
P00>>>
```

The **build** -**n** command syntax is:

bu[ild] -**n** **<device>**

where **<device>** is a processor module. The **build** -**n** command sets the NVR to its default values. The **build** -**e** command may be required during a console firmware upgrade.

7.5.5 Building the SEEPROM

The build -s command is used to restore a module's corrupted serial EEPROM.

Example 7-5 Building the SEEPROM

```
P00>>> build -s ms7cc0  # Initialize the serial
                        # EEPROM on ms7cc0.
Build serial EEPROM on ms7cc0? [Y/N]> Y
This program will take at most several minutes
      Serial EEPROM built on ms7cc0
P00>>>
```

The **build** -**s** command syntax is:

bu[ild] -**s** <**device**>

where <**device**> is MS7CC, MS7BB, KFTHA, KFTIA, DWLMA, DWLAA, or DWLPA/DWLPB. Wildcarding is allowed within a class of devices (for example, **build** -**s ms7cc***). The serial EEPROM in a device is used to log diagnostic symptom and test-directed data (DIAG_SDD, DIAG_TDD) and holds the module's serial number. It is also used by the operating system to log operating system symptom data. During manufacturing, all module serial EEPROMs are initialized (using **build** -**s** *). In the case of a module upgrade, the serial EEPROM should be initialized also.

7.5.6 Clear EEPROM

The clear eeprom command allows you to clear the selected EEPROM option.

Example 7-6 Clear EEPROM Command

```
P00>>> clear eeprom log      # Clears all failure
                             # information logged in
                             # EEPROM.
```

The **clear eeprom** command syntax is:

cl[ear] ee[prom] <option>

The **clear eeprom** command can be used to clear **diag_sdd**, **diag_tdd**, **halt**, **symptom**, or **log**.

7.5.7 Clear <envar>

Clear <envar> is used to remove an environment variable.

Example 7-7 Clear <envar>

```
P00>>> create fred     # Create fred with null value
fred set to
P00>>> set fred "this is a string in an environment
variable"
P00>>> show fred
fred        this is a string in an environment variable
P00>>> clear fred
P00>>> show fred
Environment variable not found
P00>>>
```

The **clear <envar>** removes an environment variable. However, some environment variables, such as **tta0_baud**, are permanent and cannot be removed.

The **clear <envar>** command syntax is:

cl[ear] <envar>

where **<envar>** is the name of an environment variable, for example, a boot specification to be cleared (see Table 7-3).

7.5.8 Clear Screen

The clear screen commands allows you to clear the terminal screen.

Example 7-8 Clear Screen Command

```
P00>>> clear screen          # Refresh the terminal
                             # screen.
```

The **clear screen** command syntax is:

cl[ear] sc[reen]

There are no parameters or options.

7.5.9 Continue (for OpenVMS or DIGITAL UNIX Only)

For OpenVMS or DIGITAL UNIX systems, the continue command resumes processing at the point where it was interrupted by a Ctrl/P. Programs continue executing at the address currently in the program counter of the processor.

Example 7-9 Continue Command

```
$ ^P        # Stop processing on boot processor;
            # processor enters console mode.
halted CPU 0
CPU 1 is not halted
halt code = 1
operator initiated halt # System responds with message;
PC = ffffffff80008c04   # system has halted with 80008c04
                        # in the program counter (PC).
P00>>>                  # Console session begins
               .        #
               .        #
               .        #
P00>>> continue      # Processor resumes at the address
                     # where processing was stopped by
                     # Ctrl/P.  Here processing continues
                     # at address 80008c04.
```

The **continue** command syntax is:

c[ontinue]

Continue causes the primary processor to resume program mode, executing at the address currently in the program counter (PC). This address is the address that was in the PC when the primary processor received a Ctrl/P command. The system displays the hexadecimal PC value.

When the boot processor receives a **continue** command, it does not perform processor initialization as it would for a boot procedure. The boot processor just returns to the program it was processing.

*NOTE: Following execution of the **continue** command, the console terminal enters program mode, and any ASCII characters entered on the console terminal are passed on to the operating system. In program mode, the console terminal acts like any other terminal on the system until a Ctrl/P is issued to return it to console mode.*

*NOTE: **^P** followed by **continue** should be used selectively since some console commands (for example, **deposit**, **set host**, **show device**, **show network**, and **test**) can corrupt the machine state so that the execution of the current program cannot resume successfully.*

7.5.10 Crash (for OpenVMS or DIGITAL UNIX Only)

For Open VMS or DIGITAL UNIX systems, the crash command causes the operating system to be restarted and generates a memory dump.

Example 7-10 Crash Command

```
P01>>> crash
       [operating system output appears]
```

The **crash** command causes the operating system to be restarted. This allows the user to ^P a hung system and generate a memory dump.

The **crash** command syntax is:

cra[sh]

There are no parameters or options.

7.5.11 Create

The create command allows you to create an environment variable.

Example 7-11 Create Command

```
1. P00>>> create fred    # Create a new environment
      fred set to        # variable fred with a value
   P00>>> show fred      # equal to null.
   fred

2. P00>>> create stuff 356     # Create a new environment
                               # variable stuff with a
                               # value equal to 356.
```

The **create** command syntax is:

cre[ate] <envar> [<value>]

where the **<envar>** is the environment variable name and **<value>** is the optional variable value. Created environment variables are volatile by default. **value** can be a quoted string for specifying boot specifications (see **boot** command description).

7.5.12 Date

The date command is used to display or set the system date and time.

Example 7-12 Date Command

```
P02>>> date
15:30:27 February 3, 1998      # System displays time
                               # and date.
```

The **date** command syntax is:

da[te] [<yyyymmddhhmm.ss>]

where you can set the date and time by using **yyyy** for the year, **mm** for the month, **dd** for the day, **hh** for the hour, **mm** for minutes, and **ss** for seconds.

7.5.13 Deposit

The deposit command stores data in a specified location.

Example 7-13 Deposit Command

```
1. P00>>> dep -b -n 1FF pmem:0 0 # Clear first 512 bytes
                                 # of physical memory.
2. P00>>> d -l -n 3 vmem:1234 5  # Deposit 5 into four
                                 # longwords starting at
                                 # virtual memory address
                                 # 1234.
3. P00>>> d -n 8 R0 FFFFFFFF     # Load GPRs R0 through R8
                                 # with -1.
4. P00>>> d -1 -n 10 -s 200 pmem:0 8   # Deposit 8 in the
                            # first longword of the first
                            # 17 pages in physical
                            # memory.
5. P00>>> d -1 pmem:0 0     # Deposit 0 to physical
                            # memory address 0.
6. P00>>> d + FF            # Deposit FF to physical
                            # memory address 4.
7. P00>>> d scbb 820000     # Deposit SCBB
                            # with 820000.
```

When using **deposit**, if no options are given in subsequent commands, the system uses the options from the preceding commands as the defaults for address or location referenced, data type (**-b**, **-l**, **-w**, and so forth), data size for increment (**-s**), and address space (**gpr**, **ipr**, **pmem**, and so forth).

The **deposit** command syntax is:

d[eposit] [-{b,w,l,q,o,h}] [-{n val, s val}] [space:]<address> <data>

where the options are values from Table 7-4, and <data> is the value to be stored. If the specified value is too large to fit in the data size to be deposited, the console ignores the command and issues an error response. For data lengths longer than a longword, each longword of data should be separated by a space. If the data is smaller than the data size to be deposited, the higher order bits are filled with zeros.

Table 7-4 Deposit Command Options

Option	Meaning
-b	Defines data size as a byte.
-h	Defines data size as a hexword.
-l	Defines data size as a longword; initial default.
-o	Defines data size as an octaword.
-q	Defines data size as a quadword.
-w	Defines data size as a word.
-n val	Number of consecutive locations to modify.
-s val	Specifies the address increment size. Default is data size.

space: is the optional device name (or address space) of the device to access (see Table 7-5), and address specifies the offset within a device to which data is deposited. Valid symbolic address forms include:

- fpr-name, a symbol representing a floating-point register.
- gpr-name, a symbol representing a general purpose register.
- ipr-name, a symbol representing the internal processor register.
- PC, the program counter. The address space is set to GPR.
- pt-name, a symbol representing a PAL temp register.
- +, the location immediately following the last location referenced in an examine or deposit 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 one.
- –, the location immediately preceding the last location referenced in an examine or deposit command. For physical and virtual memory, the referenced location is the last location minus 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 minus one.

- *, the last location referenced in an examine or deposit command.
- @, the location addressed by the last location referenced in an examine or deposit command.

NOTE: Since the console program actually resides in low memory when running, depositing to memory should be done with care.

Table 7-5 Device Name and Address Space Options

Option	Device Name and Address Space Meaning
<dev_ name>	Device name: pci0, xmi0, kn7cc1, demna0, and so forth.
fpr	Defines the address space as the floating-point register set, F0 through F31.
gpr	Defines the address space as the general register set, R0 through R31.
ipr	Defines the address space as the internal processor registers (IPRs).
pt	Defines the address space as the PAL temp register set, PT0 through PT23.
pmem	Defines the address space as physical memory; initial default.
vmem	Defines the address space as virtual memory. All access and protection checking occur.

For more information, see the *Alpha Architecture Reference Manual.*

7.5.14 Examine

The examine command displays the contents of a memory location, a register, or a device. The options are similar to the deposit command options.

Example 7-14 Examine Command

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

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

3. P00>>> e  -n 6 r4          # Examine register R4 and
                              # the next 6 registers -
   gpr: 20 (    R4) 0000000000005000
   gpr: 28 (    R5) 000000000FFFE000
   gpr: 30 (    R6) 00000003F8000C00
   gpr: 38 (    R7) 0000000053F761AE
   gpr: 40 (    R8) 0000010000000000
   gpr: 48 (    R9) 00000003F7800100
   gpr: 50 (   R10) 00000000000C7FFC

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

5. P00>>> examine demna0:0           # Examine demna0's
   demna0:    0 0000000108020C03     # Device Register.
```

The **examine** command syntax is:

e[xamine] [-{b,w,l,q,o,h,d}] [-{n val, s val}] [space:] <address>

where the options are values from Table 7-6, **space:** is the optional device name (or address space) of the device to access, and **address** is a longword that specifies the first location to be examined.

The display line consists of the device name, the hexadecimal address or offset within the device, and the examined data also in hexadecimal.

Table 7-6 Examine Command Options

Option	Meaning
-b	Defines data size as a byte.
-d	Disassembles instruction at current address.
-h	Defines data size as a hexword.
-l	Defines data size as a longword; initial default.
-o	Defines data size as an octaword.
-q	Defines data size as a quadword.
-w	Defines data size as a word.
-n val	Number of consecutive locations to examine.
-s val	Specifies the address increment size. Default is data size.

Examine uses most of the same options as **deposit**. Additionally, the **examine** command supports the **-d** option (instruction decode, which will disassemble the instructions at the current address). When using **examine**, if no options are given in subsequent commands, the system uses the options from the preceding commands as the defaults for address or location referenced, data type, including **-d**, (**-b**, **-l**, **-w**, and so forth), data size for increment (**-s**), and address space (**gpr**, **ipr**, **pmem**, and so forth).

After initialization, the default address space is physical memory, the default data size is a longword, the default address is zero, and the default address increment size is the data size. If conflicting address space or data sizes are specified, the console ignores the command and issues an error response.

Table 7-7 Device Name and Address Space Options

Option	Device Name and Address Space Meaning
<dev_name>	Device name: pci0, xmi0, kn7cc1, demna0, and so forth.
fpr	Defines the address space as the floating-point register set, F0 through F31.
gpr	Defines the address space as the general register set, R0 through R31. The data size is always a quadword.
ipr	Defines the address space as the internal processor registers (IPRs). The data size is always a longword.
pt	Defines the address space as the PAL temp register set, PT0 through PT23.
pmem	Defines the address space as physical memory.
vmem	Defines the address space as virtual memory. All access and protection checking occur.

7.5.15 Halt (OpenVMS or DIGITAL UNIX Only)

The halt command halts the specified processor or device. Equivalent to the stop command.

Example 7-15 Halt Command

```
P00>>> halt
P00>>>
```

The **halt** command syntax is:

halt [-drivers[<device_prefix>]][<processor_num>]

where **-drivers[device_prefix]** specifies the name of the device or device class to stop. If no device prefix is specified, then all drivers are stopped. The **<processor_num>** argument specifies the number of the processor to stop.

7.5.16 Help or Man

The help (or man) command provides basic information on the console commands.

Example 7-16 Help Command

```
1. P00>>> help create   # Display basic create command
   NAME                 # information.  Minimum
     create             # command input is highlighted.
   FUNCTION
     create environment variable
   SYNOPSIS
     create <envar> <value>
           [-nv] [integer] [-string]

2. P00>>> help h        # Display help information on
                        # all commands beginning with h.
  NAME
     halt
   FUNCTION
     Halt the specified processor or device.
   SYNOPSIS
     halt [-drivers [<device_prefix>]] [processor_num>]
  NAME
     help
   FUNCTION
     Display information about console commands.
   SYNOPSIS
      :
      :
```

Example 7-14 Help Command (Continued)

```
3. P02>>> help runecu
   NAME
            runecu
   FUNCTION
            Run the EISA Configuration Utility from floppy.
   SYNOPSIS
            runecu
   P02>>> help run
   NAME
            run
   FUNCTION
            Run an ARC utility program.
   SYNOPSIS
run <program> [-d <device>][-p <n>][-s <parameter string>]
 where:
<device> is the console device containing the program.
          Default is dva0.
<n> is the unit number of PCI to configure.(ie,0 for pci0)
<parameter string> Any optional parameters which must be
                   passed to the utility, must be enclosed
                   in quotes.
```

The **help** command syntax is:

he[lp] [<command>]

where **<command>** is one of the console commands. The **<helpkey>** can also be used after a partial command has been typed. For example, **set <helpkey>** will display the options supported by the **set** command.

7.5.17 Init

The init command performs a reset.

Example 7-17 Init Command

```
P00>>> init
```

The **init** command syntax is:

i[nit]

The **init** command is used to reset, or initialize, the entire system. The **init** command resets the machine and runs systemwide self-test. Self-test results are displayed after a system reset.

7.5.18 Prcache

The prcache command manipulates an optional NVRAM disk cache.

Example 7-18 Prcache Command

```
P00>>> prcache -f
PCI NVRAM Disk Cache:  passed
Size:   2MB
Base Address:  040000000
System ID:  21000035
State: valid
Battery Status:  good
Battery Disconnect Circuit Status: disabled
```

The **prchache** command syntax is:

prcache [-z] [-f] [-u] [nvram_bus]

where **-z** zeros the NVRAM disk cache

-f displays the NVRAM status

-u disables the NVRAM battery disconnect circuit

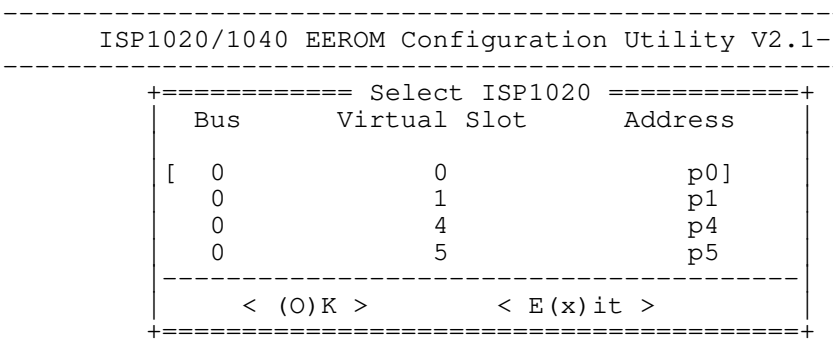
nvram_bus defines the action to be performed on the NVRAM card installed on the bus.

7.5.19 Run (for OpenVMS or DIGITAL UNIX Only)

For OpenVMS or DIGITAL UNIX systems, the run command is used to run an ARC utility program. Run has four options: rcu (the RAID Configuration Utility), swxcrfw, eeromcfg, and util_cli. The arc_enable environment variable must be set to on before this command can be used. See Appendix C for more information.

Example 7-19 Run Command

```
P00>>> init            # Initialize after setting
                       # arc_enable on.
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1  0  NODE #
                    A  P  P  M  A  P  P  P  P  TYP
                    o ++ ++  +  o ++ -+ ++ -+  ST1
                    . EE EE  .  . EE EE EE EB  BPD
                    o ++ ++  +  o ++ -+ ++ -+  ST2
                    . EE EE  .  . EE EE EE EB  BPD
                    + ++ ++  +  + ++ -+ ++ -+  ST3
                    . EE EE  .  . EE EE EE EB  BPD
   +  .  .  +  .  .  +  .  .  .  .  +  .  +     C0 XMI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C2
            .  +  .  +  +  .  .  .  .  .  .  +  C3 PCI +
                    .  .  .  .  .  .  +  .        EISA +
                       +  +  +  +  .  +  +  +  C11 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C12
                    .  .  .  A0 .  .  .  .  .  ILV
                    .  .  . 128 .  .  .  .  . 128MB
AlphaServer 8400 Console V5.1-30, 14-JAN-1998 SROM 3.1
P00>>> show dev isp3   # Use show dev to find CD drive.
polling for units on isp3, slot 5, bus 0, hose0...
dkd400.4.0.5.0     DKD400                    RRD46  0064
dkd500.5.0.5.0     DKD500                    RZ26L  440C
P00>>> run eeromcfg -d dkd400

Loading image...
Transferring control...
             # The screen is erased, followed by:
```


ISP1020/1040 EEROM Configuration Utility V2.1-102
Select ISP1020
Bus
Virtual Slot
Address
[0 0 p0]
0 1 p1
0 4 p4
0 5 p5
< (O)K >
< E(x)it >

The **run** command has four options:

1. **rcu** **-** Run the RAID Configuration Utility
2. **swxcrfw** - Update firmware on RAID controller.
3. **eeromcfg** - Run the ISP1020/1040 EEROM Configuration Utility.
4. **util_cli** - Run the KZPSA Configuration Utility

Options 1 and 2 (**rcu** and **swxcrfw**) are on floppy and require a floppy drive. Options 3 and 4 (**eeromcfg** and **util_cli**) are on CD.

The **run** command syntax is:

run <program> [-d <device>][-p <n>][-s <parameter string>]

where **<program>** is the command option selected, **<device>** is the console device containing the program (default is dva0), **<n>** is the unit number of the PCI to configure, and **<parameter string>** are any optional parameters that must be passed to the utility, which must be enclosed in quotes. The utility documentation specifies any required parameters.
To use this command, the **arc_enable** environment variable must be set to **on** and the system initialized. An example would be:

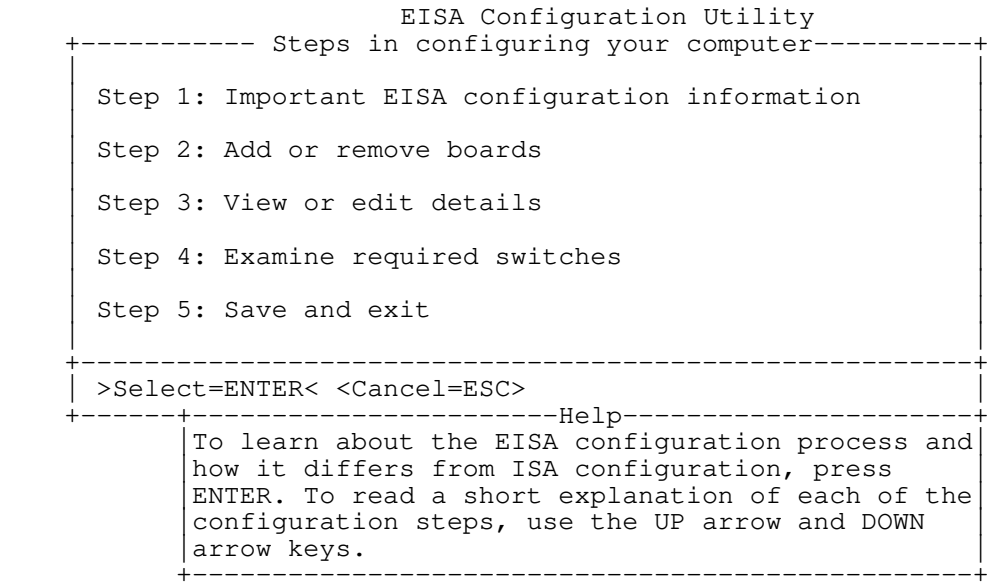
```
P08>>> run swxcrfw -p 1 n -s "swxcrfwp.215 -v 215"
```

7.5.20 Runecu (for OpenVMS and DIGITAL UNIX Only)

For OpenVMS and DIGITAL UNIX systems, the runecu command is used to invoke the EISA Configuration Utility (ECU). ECU comes on a floppy. The arc_enable environment variable must be set to on before this command can be used.

Example 7-20 Runecu Command

```
P00>>> runecu
Loading image...
Transferring Control... # After about 1 minute the
                  # following appears:

            EISA Configuration Utility    Help=F1
 +------------ EISA Configuration Utility -------------+
 |                                                      |
 | (C) Copyright 1989, 1994                             |
 | Micro Computer Systems, Inc.                         |
 | All Rights Reserved.                                 |
 |                                                      |
 | This program is provided to help you set your        |
 | computer's configuration. You should use this        |
 | program the first time you set up your computer      |
 | and whenever you add additional boards or options.   |
 |                                                      |
 | To continue, press ENTER.                            |
 +------------------------------------------------------+
 | >Ok=ENTER<                                           |
 +------------------------------------------------------+

After ENTER is pressed, the ECU will load the
configuration files from the floppy:

      EISA Configuration Utility    Help=F1
            +------------------------------+
            | Loading configuration files. |
            |                              |
            | Please wait ...              |
            |                              |
            +------------------------------+
After the files are verified, the ECU main menu displays:
```



The **runecu** command syntax is:

runecu

On OpenVMS and DIGITAL UNIX systems, the **runecu** command automatically boots the EISA Configuration Utility from the floppy disk. To use this command, the **arc_enable** environment variable must be set to **on** and the system initialized.

7.5.21 Set EEPROM

The set eeprom command allows you to set the selected EEPROM option.

Example 7-21 Set EEPROM Command

```
1. P00>>> set eeprom field
   LARS #> 09494820           # Enter labor activity
   Message> EEPROM update     # reporting system (LARS)
   P00>>>                     # number (8 digits) and
                              # message (up to 68
                              # characters).
2. P00>>> set eeprom man      # Enter module serial number
                              # and unified part number.

   Module Serial Number> SG226LFH01
   Module Unified 2-7-2.4 Part Number>   -E2040-AA. M06
   P00>>>
```

The **set eeprom** command syntax is:

se[t] ee[prom] <option>

where **option** is **field**, **halt**, **manufacturing**, **serial**, or **symptom**. You must use the **set cpu** command (see second example in Example 7-22) for the CPU you would like to act on.

7.5.22 Set <envar>

Set <envar> allows you to modify environment variables.

Example 7-22 Set <envar>

```
1. P00>>> set auto_action restart # On an error halt,
                                  # system will automatic-
                                  # ally restart. If restart
                                  # fails, boot the operat-
                                  # ing system.
2. P00>>> set cpu 1               # Designate CPU in slot
   cpu set to 1                   # 1 as the primary, or
   P01>>>                         # boot, processor.
3. P00>>> set d_harderr halt      # System will halt on hard
                                  # error.
4. P00>>> se class                # Set the value of
                                  # environment variable
                                  # class to null.
5. P00>>> show enable*     # Display the status
   enable_audit        OFF      # of the enable_audit
   P00>>> set enable_audit on  # environment variable. Set
                               # enable_audit on to enable
                               # generation of audit trail
                               # messages.
```

The **set <envar>** syntax is:

se[t] <envar> [value]

where **envar** (environment variable) and **value** are from Table 7-3, which also indicates which environment variables are volatile. Certain environment variables, such as boot specifications, must be defined using the **create** command. For additional information, see Section 7.5.11. Unambiguous abbreviations can be used for an environment variable name when using the **set** command. **Set -d envar** resets the value of **envar** to its default value. Wildcarding is also allowed with the **set** command. For example, **set -d *** resets all environment variables to their default values.

7.5.23 Set Host

The set host command allows you to connect to another console or service.

Example 7-23 Set Host Command

```
P00>>>  show configuration

    Name      Type    Rev   Mnemonic
TLSB
0++ KN7CC-AB  8014    0000  kn7cc-ab0
1+  MS7CC     5000    0000  ms7cc0
7+  MS7CC     5000    0000  ms7cc1
8+  KFTHA     2020    0000  kftha0
C0 XMI connected to kftha0  xmi0
8+  DWLMA     102A    0105   dwlma0
E+  DEMNA     0C03    0802   demna0
P00>>> set host demna0
Connecting to remote node, ^C to disconnect.
T/R               # To begin RBDs on DEMNA
RBDE>             # in Slot E.
```

The **set host** command syntax is:

se[t] h[ost] <device_adapter>

The **set host <device_adapter>** command is used to connect to a remote XMI adapter for running XMI module-resident ROM-based diagnostics, as shown in Example 7-23. Use Ctrl/C to terminate the command and return to the primary processor.

Set host can only be issued from the boot processor, and only one **set host** command is in effect at a time. Characters typed from the console terminal are passed through to the target node. All output from the target node is displayed on the console terminal.

7.5.24 Set Power

The set power command is used to configure the system power regulators for battery backup (AlphaServer 8400 with three-phase power only).

Example 7-24 Set Power Command

```
P00>>> set power -b 8 left
P00>>>
```

The **set power** command syntax is:

se[t] p[ower] -b <value> <option>

where **-b** allows you to configure the system with batteries**, <value>** is the number of batteries (**4** or **8**), and **<option>** is the cabinet containing the batteries (**main**, **left**, or **right**).

7.5.25 Set SEEPROM

The set seeprom command allows you to set the selected SEEPROM option.

Example 7-25 Set SEEPROM Command

```
1. P00>>> set seeprom field
   LARS #> 09494820           # Enter labor activity
   Message> SEEPROM update    # reporting system (LARS)
   P00>>>                     # number (8 digits) and
                              # message (up to 68
                              # characters).
2. P00>>> set seeprom man kftha0 # Enter module serial
                              # number, part number, and
                              # device type.
   Module Serial Number> SG226LFH01
   Module Unified 2-7-2.4 Part Number>   -E2040-AA. M06
   Module Device Type> 2000
   P00>>>
```

The **set seeprom** command syntax is:

se[t] see[prom] <option> <device>

where **option** is **field**, **manufacturing**, or **serial** and **device** is the device mnemonic; for example, **kftia0**, **kftha1**.

7.5.26 Show Configuration

The show configuration command displays the last configuration seen at system initialization.

Example 7-26 Show Configuration Command

```
P00>>> show configuration

      Name              Type    Rev    Mnemonic
  TLSB
  0++ KN7CC-AB          8014    0000   kn7cc-ab0
  6+  MS7CC             5000    0000   ms7cc0
  7+  KFTIA             2020    0000   kftia0
  8+  KFTHA             2000    0D02   kftha0

  C0 XMI connected to kftha0                xmi0
  1+  DEMNA        C03    0803   demna0
  3+  DEMFA        823    0514   demfa0
  8+  DWLMA       102A    020A   dwlma0
  E+  KZMSA        C22    2B01   kzmsa0

  C3 PCI connected to kftha0                pci0
  0+  SIO                4828086    0003   sio0
  7+  KZPSA                 8101    0000   kzpsa0
  8+  ISP1020               8101    0000   kzpsa1
  A+  DAC960               11069    0000   dac0

    Controllers on SIO                      sio0
  0+  DECchip 21040-AA       21011    0000  tulip0
  1+  FLOPPY                     2    0000  floppy0
  2+  KBD                        3    0000  kbd0
  3+  MOUSE                      4    0000  mouse0

    EISA connected to pci0 through sio0     eisa0
```

Example 7-23 Show Configuration Command (Continued)

```
C4 Internal PCI connected to kftia0      pci1
0+  ISP1020             10201077    0001   isp2
1+  ISP1020             10201077    0001   isp3
2+  DECchip 21040-AA       21011    0023   tulip1
4+  ISP1020             10201077    0001   isp4
5+  ISP1020             10201077    0001   isp5
6+  DECchip 21040-AA       21011    0023   tulip2
7+  PCI NVRAM              71011    0000   pci_nvram0

C5 FBUS connected to kftia0              fbus0
5+  DWLAA                   2003    0000   dwlaa0
P00>>>
```

The **show configuration** command syntax is:

sh[ow] c[onfiguration] [-v]

where **-v** indicates vector displayed.

The screen displays the system configuration, including the hardware device type, revision level, and mnemonic for each TLSB, XMI, PCI, and FBUS+ node.

7.5.27 Show CPU

The show CPU command displays information on CPUs in the system.

Example 7-27 Show CPU Command

```
P08>>> show cpu
Primary CPU:      08
Active CPUs:      08 09 10
Configured CPUs:  08 09 10 11
P08>>>
```

The **show cpu** command syntax is:

sh[ow] cpu

The console displays CPU information. **Primary CPU** is the current primary processor and 08 is its node number. **Active CPUs** are those CPUs actively running the console, and **configured CPUs** are those present in the system but not necessarily running the console (in Example 7-27, CPU 11 failed self-test).

7.5.28 Show Device

Displays device information for any disk/tape adapter or group of adapters.

Example 7-28 Show Device Command

```
P00>>> show device
polling for units on isp0, slot 0, bus 0, hose0...
dka200.2.0.0.0  DKA200   RZ26L  440C
dka200.4.0.0.0  DKA400   RZ26L  440C
polling for units on isp1, slot 1, bus 0, hose0...
dkb200.3.0.1.0  DKB300   RZ26L  440C
dkb200.5.0.1.0  DKB500   RZ26L  440C
dkb200.6.0.1.0  DKB600   RZ26L  440C
polling for units on isp2, slot 4, bus 0, hose0...
dkc200.1.0.4.0  DKC100   RZ26L  440C
dkc200.1.0.4.0  DKC200   RZ26L  440C
dkc200.1.0.4.0  DKC300   RZ26L  440C
polling for units on isp3, slot 5, bus 0, hose0...
dkd400.4.0.5.0  DKD400   RRD44  0064
dkd500.5.0.5.0  DKD500   RZ26L  440C
```

The **show device** command syntax is:

sh[ow] dev[ice] [<dev_name>]

Show device with no **<dev_name>** gives all devices in the system. **<dev_name>** can be any adapter name (wild-carding is allowed). For example, **show device cixcd*** will display information on all CIXCD devices in the system.

7.5.29 Show EEPROM

The show EEPROM command allows you to display selected EEPROM information.

Example 7-29 Show EEPROM Command

```
1. P00>>> show eeprom serial  # Display system serial
                              # number.
   System Serial Number = GAO1234567
2. P00>>> show eeprom manufacturing # Display manufactur-
                                    # ing information.
   Module Serial Number = SG226LFH01
   Module Unified 2-7-2.4 Part Number =   -E2040-AA. M06
```

The **show eeprom** command syntax is:

sh[ow] ee[prom] <option>

where **option** is **diag_sdd**, **diag_tdd**, **field**, **halt**, **manufacturing**, **serial**, or **symptom**. You must use the **set cpu** command (see second example in Example 7-22) for the CPU module you would like to act on.

7.5.30 Show <envar>

Show <envar> displays the current state of the specified environment variable.

Example 7-30 Show <envar> Command

```
1. P00>>> show auto_action
   auto_action        restart
   P00>>>
2. P00>>> show tta0_baud
   tta0_baud              9600
3. P00>>> show d_harderr
   d_harderr          halt
4. P00>>> show enable*              # Displays status
          enable_audit       OFF    # of enable_audit
5. P00>>> show interleave
   interleave         none
```

The **show envar** command syntax is:

sh[ow] <envar> or **sh[ow] ***

where **envar** is an environment variable name (see Table 7-3). Unambiguous abbreviations can be used for an environment variable name when using the **show <envar>** command. See the **set <envar>** command for related information.

7.5.31 Show Memory

The show memory command displays memory module information.

Example 7-31 Show Memory Command

```
P00>>> show memory

Set Node  Size    Base Address         Intlv   Position
--- ----  ----    ------------------   -----   --------
 A   7    256 Mb  00000000 000000000  2-Way       0
```

The **show memory** command syntax is:

sh[ow] m[emory]

In the above example, the memory module at node 7 is in an on-board two-way interleave indicated by the interleave set A. The total memory size is 256 Mbytes.

7.5.32 Show Network

The show network command displays the names and physical addresses of all known network devices in the system.

Example 7-32 Show Network Command

```
P00>>> show network
polling for units on demna0, slot 14, xmi0...
exa0.0.0.14.0: 08-00-2B-24-3F-E1
polling for units on demfa0, slot 14, xmi1...
exb0.0.0.14.2: 08-00-2B-0B-BB-FF
polling for units on tulip0, slot 2, bus 0, hose0...
ewa0.0.0.2.0: 08-00-2B-E2-B8-52
polling for units on tulip1, slot 6, bus 0, hose0...
ewb0.0.0.6.0: 08-00-2B-E2-B8-5F
```

The **show network** command syntax is:

sh[ow] ne[twork]

There are no options or qualifiers.

7.5.33 Show Power

The show power command gives the power status of the system (AlphaServer 8400 system with H7263 power regulators installed).

Example 7-33 Show Power Command

```
>P00>>> show power
Cabinet: Main              Regulator :          A       B       C
-------------------------------------------------------------------------
Primary Micro Firmware Rev :                    2.0     2.0     2.0
        Secondary Micro Firmware Rev :          2.0     2.0     2.0
                  Power Supply State :       NORMAL  NORMAL  NORMAL
             AC Line Voltage (V RMS) :       113.71  114.35  115.93
               DC Bulk Voltage (VDC) :       227.02  227.02  227.02
            48V DC Bus Voltage (VDC) :        47.57   47.57   47.57
            48V DC Bus Current (ADC) :        30.17   29.68   29.58
      48V Battery Pack Voltage (VDC) :        50.85   50.72   47.91
      24V Battery Pack Voltage (VDC) :        25.56   25.56   23.95
   Battery Pack Charge Current (IDC) :         2.91    2.90    0
      Ambient Temperature (Degree C) :        26.22   24.80   24.75
                Elapsed Time (Hours) :       290.00  290.00  290.00
Remaining Battery Capacity (Minutes) :         8.00    8.00    8.00
     Battery Cutoff Counter (Cycles) :         0       1.00    1.00
                Battery Configuration :  4 Batteries  4 Batteries  4 Batteries
                      Heatsink Status :       NORMAL  NORMAL  NORMAL
                  Battery Pack Status :       CHARGING CHARGING DISCHG'G
                 Last UPS Test Status :       PASSED  PASSED  TESTING
```

The **show power** command syntax is:

sh[ow] p[ower] [-{h,s}] [option]

where **-s** displays the current status (default) and **-h** the history status (value of each parameter at the last system shutdown) and **option** selects the cabinet (**main**, **right**, or **left**).

7.5.34 Show SEEPROM

The show SEEPROM command allows you to display selected SEEPROM information.

Example 7-34 Show SEEPROM Command

```
1. P00>>> show seeprom field kftha0 # Displays field
   LARS # = 0949820              # entered Labor Activity
   Message = EEPROM update       # Number and message.

2. P00>>> show seeprom manu kftha0 # Displays
                        # manufacturing information.
   Module Serial Number = SG226LFH01
   Module Unified 2-7-2.4 Part Number = -E2040-AA. M06
   Module Device Type = 2000
```

The **show seeprom** command syntax is:

sh[ow] see[prom] <option> <device>

where **option** is **diag_sdd**, **diag_tdd**, **symptom**, **field**, **manufacturing**, or **serial** and **device** is **kftha*n***, **kftia*n***.

7.5.35 Start (OpenVMS or DIGITAL UNIX Only)

The start command begins execution of an instruction at the address specified in the command string. The start command does not initialize the system.

Example 7-35 Start Command

```
P00>>> start 40000000    # Start processor at
                         # address 40000000.
```

The **start** command syntax is:

s[tart] address

where **address** is the address the program counter is set to start execution. The **start** command is equivalent to **continue**, except you can specify the address at which to begin executing.

*NOTE: The **start** command should be used selectively since some console commands (for example, **deposit**, **set host**, **show device**, **show network**, and **test**) may corrupt the machine state so that execution of the current program may not resume successfully.*

7.5.36 Stop

The stop command halts a specified processor or device. Equivalent to the halt command.

Example 7-36 Stop Command

```
P00>>> stop 1          # Stop CPU 1.
```

The **stop** command syntax is:

stop [-drivers[<device_prefix>]][<processor_num>]

where **-drivers[device_prefix]** specifies the name of the device or device class to stop. If no device prefix is specified, then all drivers are stopped. The **<processor_num>** argument specifies the number of the processor to stop.

7.5.37 Test

The test command allows you to test the entire system (focusing on memory and I/O), a portion of the system (subsystem), or a specific device. By default, the entire system is tested.

Example 7-37 Test Command

```
1. P00>>> test
   Complete Test Suite for runtime of 600 seconds
   Type ^C to stop testing
   Configuring system...
   polling for units on kzmsa0, slot 9, bus 0, xmi0...
   dkf0.0.0.9.8 DKF0    RZ26L 440C
   dkf200.2.0.9.8 DKF200      RZ26L 440C
        .
        .
        .
   polling for units on tulip1, slot 5, bus 0, hose0...
   ewb0.0.0.5.0: 08-00-2B-E2-11-0C
   Shutting down units on tulip1, slot 5, bus 0 hose 0...
   --------Testing done -----------
   Shutting down drivers...
   Shutting down units on demna0, slot 3, bus 0, hose 8...
   Shutting down units on kzmsa0, slot 9, bus 0, hose 8...
   Shutting down units on tulip2, slot 2, bus 0, hose 4...
   Shutting down units on tulip3, slot 6, bus 0, hose 4...
   P00>>>

2. P>>> test -q           # Runs a system test.  Since a test
                          # run time was not specified, the
                          # entire system will be tested
                          # for approximately 10 minutes.
                          # Status messages
                          # will not be displayed.

3. P00>>> test ms7cc*     # Tests all memory modules in the
                          # system.

4. P00>>> test pci0       # Tests the pci0 I/O subsystem.
```

The **test** command syntax is:

t[est][-write][-nowrite "list"][-t time][-q][dev_arg]

where **<dev_arg>** specifies the target device, group of devices, or subsystem to test. A list of available devices and subsystem mnemonics can be obtained by issuing a **show configuration**, **show device**, or **show network** command. You would then issue the **test dev_arg** command to test the desired device. Table 7-8 lists the command options.

If no parameter is specified, the entire system, except for memory, is tested. Note that system testing performed by the test command is very different from that performed during power-on or reset. To execute systemwide self-test, use the **inititalize** command. See Chapter 6 for additional examples of the **test** command.

Table 7-8 Test Command Options

Option	Meaning
write	Selects writes to media as well as reads (read only is the default). Only applicable to disk testing (ignored otherwise).
-nowrite "list"	Used with **-write** to prevent selected devices or groups of devices from being written to.
-t time	Run time in seconds, following system sizing and configuration; default for system test is 600 seconds (10 minutes).
-q	Disables the status messages displayed by default as exerciser processes are started and stopped during testing. **-q** sets **d_verbose** to zero.

7.5.38 Type

The type command displays the contents of a specified file, one screen at a time. One common use of type is to display the event log, as shown in Example 7-38. Type a space to see the next screen, or Enter to see the next line.

Example 7-38 Type Command

```
P00>>> type el
01:14.63
01:14.63 Starting console on CPU 0
01:14.63 initialized idle PCB
01:14.63 initializing semaphores
01:14.63 initializing heap
01:14.63 initial heap 100000
01:14.63 Twilight zone at 7e000, length 82000(532480)
01:14.63 Heap size effc0(982976)
01:14.63 memory low limit = 1f0000
01:14.63 heap = 100000, effc0
01:14.63 initializing driver structures
01:14.63 initializing overlays
01:14.63 initializing idle process PID
01:14.63 initializing flash index
01:14.63 hf_init
01:14.63 hf_init_platform: console_dlb_flag = 0
01:14.63 hf_init_platform: from_adr = 13d694, index_count
150
01:14.63 hf_setup_platform
01:14.63 load XDELTA
01:14.63 Transferring to xdelta
01:14.63 initializing file system
01:14.63 DDB Startup, phase = 0
[--More-- (SPACE - next page, ENTER - next line)
```

7.5.39 Vga

The vga command tests the graphics console monitor.

Example 7-39 Vga Command

```
P00>>> vga test 0
       .
       .
       .
[displays character set on graphics monitor]
```

The **vga** command syntax is:

vga [reinit] [test <test_nbr> [<n>]]

where **reinit** reinitializes the graphics monitor, and **test_nbr** can have three values:

test_nbr = 0 Display the entire character set on the graphics monitor **<n>** times.

1 Display color bars on the graphics monitor. You can then adjust the color values, if your monitor offers such adjustment.

2 Display the character E repeatedly on the screen. This allows you to determine whether the alignment of the screen is as it should be. You can then adjust the alignment, if your monitor offers such adjustment.

7.5.40 Comment (#)

A comment can be introduced using the # symbol. The entire comment is ignored.

Example 7-40 Comment (#) Command

```
1. P00>>> # This example illustrates the comment command.
      P00>>>

2. P00>>> exam pmem:0400EC  # Examine physical memory.
          pmem: 000400EC D0FFFFFD
      P00>>>
```

Chapter 8

AlphaBIOS Firmware

The AlphaBIOS firmware boots the Windows NT operating system, either automatically or when you request, depending on the setup factors described in Section 5.5. AlphaBIOS Setup, reached by pressing the F2 key before the operating system boots, 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 Setup, and it provides a reference for the **AlphaBIOS Setup** screens.

Sections in this chapter are:

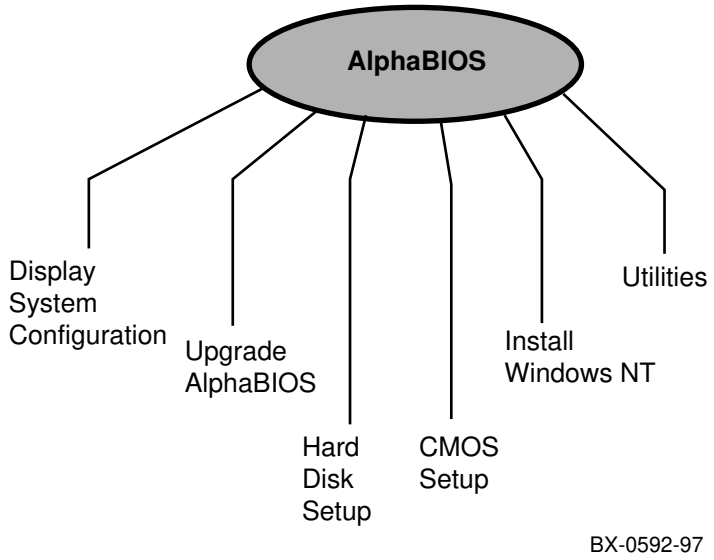
- Introduction
- Switching Between Windows NT, **AlphaBIOS Setup**, and the SRM Console
- Keyboard Conventions and Help
- Starting AlphaBIOS
- Displaying the System Configuration
- Updating Firmware
- Setting Up the Hard Disk
- Performing Setup Tasks
- Installing Windows NT
- Running Utility Programs
- 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.*

8.1 Introduction

AlphaBIOS boots the Windows NT operating system, as described in Section 5.5. AlphaBIOS Setup (reached by pressing F2 before the operating system boots) is the Windows NT system management tool. It allows you to display the system configuration, upgrade firmware using the LFU utility, perform hard disk setup, install Windows NT, and run utility programs.

Figure 8-1 AlphaBIOS Setup Functions



As shown in Figure 8-1, AlphaBIOS Setup has six major functions, provided through menus displayed on the graphics monitor:

Display System Configuration

The displays provided include:

- System board configuration
- Hard disk configuration
- PCI configuration
- EISA configuration
- SCSI configuration
- Memory configuration
- Integrated peripherals

Upgrade AlphaBIOS

This menu lets you upgrade firmware from the console CD. Although the menu item says "Upgrade AlphaBIOS," you can upgrade any firmware provided on the console CD. This menu item lets you boot the loadable firmware utility (LFU). LFU commands are described in Appendix B.

Hard Disk Setup

You can create and delete partitions, and format them as FAT file systems.

CMOS Setup

CMOS setup is used to configure system parameters such as graphics monitor screen colors, keyboard language, and date and time. Advanced CMOS setup lets you select AlphaBIOS password protection.

Install Windows NT

You can install Windows NT by selecting this option.

Utilities

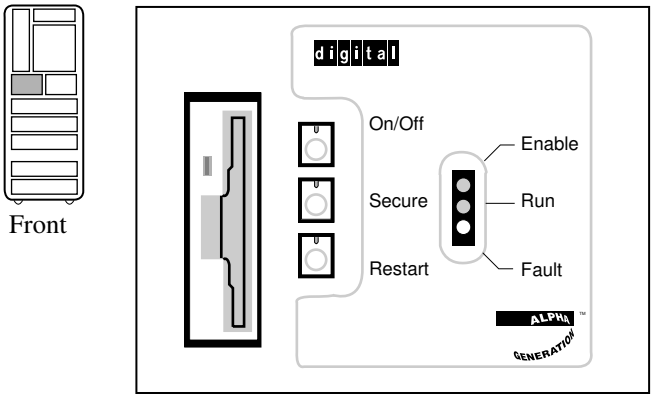
This menu lets you run ECU from floppy, define multiple Windows NT operating systems for selection to run, and run maintenance programs.

8.2 Switching Between Windows NT, AlphaBIOS Setup, and the SRM Console

To switch from Windows NT to the SRM console, to boot DIGITAL UNIX or OpenVMS, or to run firmware-based diagnostics, restart the system from the Windows NT taskbar or at the control panel on the front of the machine. If you have selected auto_action boot, then set the system to Secure at the control panel before AlphaBIOS is started. When the SRM console prompt appears, press Secure again (8200) or return the keyswitch to Enable (8400) to enter SRM commands.

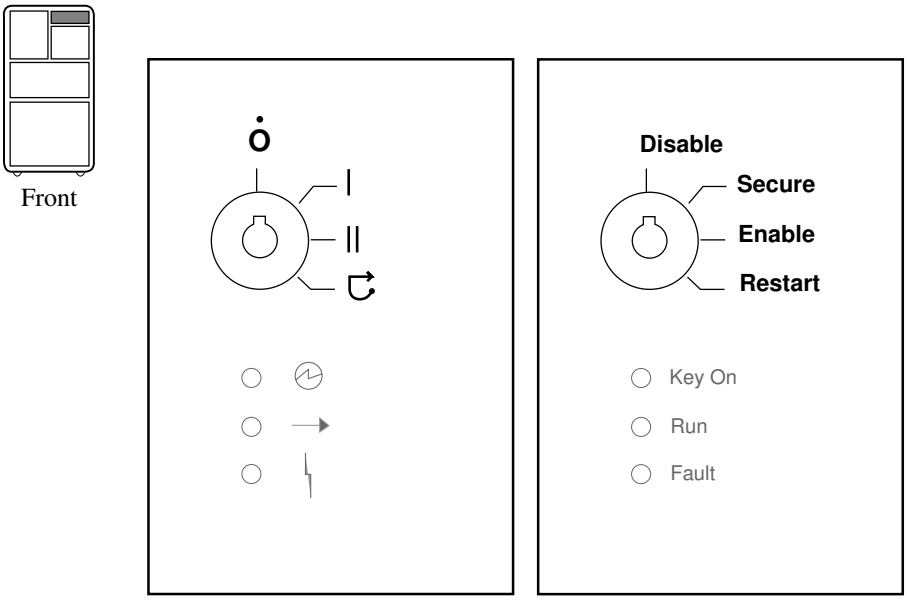
To return to AlphaBIOS, type the alphabios command at the SRM console prompt. Then, to enter AlphaBIOS Setup, press the F2 key before the Windows NT operating system boots.

Figure 8-2 8200 Control Panel



BX-0607-98

Figure 8-3 8400 Control Panel



Switch from Windows NT to the SRM Console

There are two ways to switch from Windows NT to the SRM console. Using the Windows NT taskbar to shut down the system ensures that files are closed properly before the system shuts down, and is preferred to switching to the SRM console from the control panel.

From the Windows NT Taskbar (Preferred Method):

1. Click on **Start** in the taskbar displayed at the bottom of the screen, and then click on Shut Down from the popup menu displayed.
2. A window displays with three options. Choose "yes" to "Shut down the computer" or "Restart the computer." (If you choose "Shut down the computer," another window will display telling you that it is safe to turn off the system, or you can Restart. Choose Restart.)
3. If the **auto_action** environment variable has been set to **halt**, the system will stop at the SRM console prompt and you can enter commands. Otherwise, proceed to Step 4.

4. If the **auto_action** environment variable has been set to **boot**, press the Secure pushbutton in (8200) or turn the keyswitch to Secure (8400) before AlphaBIOS starts automatically.
5. When the display stops at the SRM console prompt, return the Secure pushbutton to out (8200) or turn the keyswitch to Enable (8400). You can then enter SRM console commands.

From the System's Control Panel:

1. Press the Restart button (8200) or turn the keyswitch to Restart (8400). If the **auto_action** environment variable is set to **halt**, the system will halt at the SRM console prompt and you can enter commands. Otherwise, proceed to Step 2.
2. If the **auto_action** environment variable is set to **boot**, press the Secure pushbutton in (8200) or turn the keyswitch to Secure (8400) before AlphaBIOS starts automatically.
3. When the display stops with the SRM console prompt, return the Secure pushbutton to out (8200) or turn the keyswitch to Enable (8400). You can then enter SRM console commands.

Switch from Windows NT to AlphaBIOS Setup

Follow the instructions for switching from Windows NT to the SRM console. If the **auto_action** environment variable has been set to **boot,** do not use the Secure pushbutton or keyswitch. Simply let AlphaBIOS start automatically and press the F2 key at any time during its initialization process, but before the system boots.

If the **auto_action** environment variable has been set to **halt**, the system will halt with an SRM console prompt. See the following instructions for switching from the SRM console to **AlphaBIOS Setup**.

Switch from the SRM Console to AlphaBIOS Setup

To return to the AlphaBIOS firmware, issue the command **alphabios** at the SRM console prompt. Then, to enter **AlphaBIOS Setup**, press the F2 key before the operating system boots. If a **Setup Password** or **Startup Password** has been selected, you will have to enter the correct password to enter **AlphaBIOS Setup.**

Switch from the SRM Console to Windows NT

To boot the Windows NT operating system from the SRM console, type the **alphabios** command at the SRM console prompt. The AlphaBIOS firmware begins executing and terminates with the boot screen. See Section 5.5 for the effect of various environment variables and **AlphaBIOS Setup** options on how Windows NT is booted from this screen.

Switch from AlphaBIOS Setup to SRM Console

To return to the SRM console from **AlphaBIOS Setup**,

1. Press Ctrl/Alt/Delete.
2. If the **auto_action** environment variable is set to **halt**, the system will stop at the SRM console prompt and you can enter commands. Otherwise, proceed to Step 3.
3. If the **auto_action** environment variable has been set to **boot**, press the Secure pushbutton in (8200) or turn the keyswitch to Secure (8400) before AlphaBIOS starts automatically.
4. When the display stops at the SRM console prompt, return the Secure pushbutton to out (8200) or turn the keyswitch to Enable (8400). You can then enter SRM console commands.

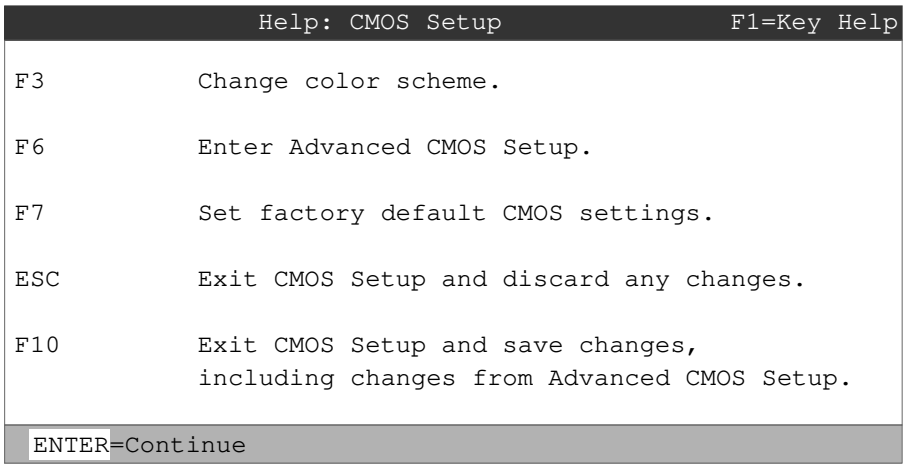
Switch from AlphaBIOS Setup to Windows NT

Press the Esc key, and AlphaBIOS returns to the boot screen. See Section 5.5 for the effect of **AlphaBIOS Setup** options on how Windows NT is booted from this screen.

8.3 Keyboard Conventions and Help

AlphaBIOS uses universally accepted keys and key combinations for navigating the interface and selecting items.

Figure 8-4 Typical First-Level Help Screen



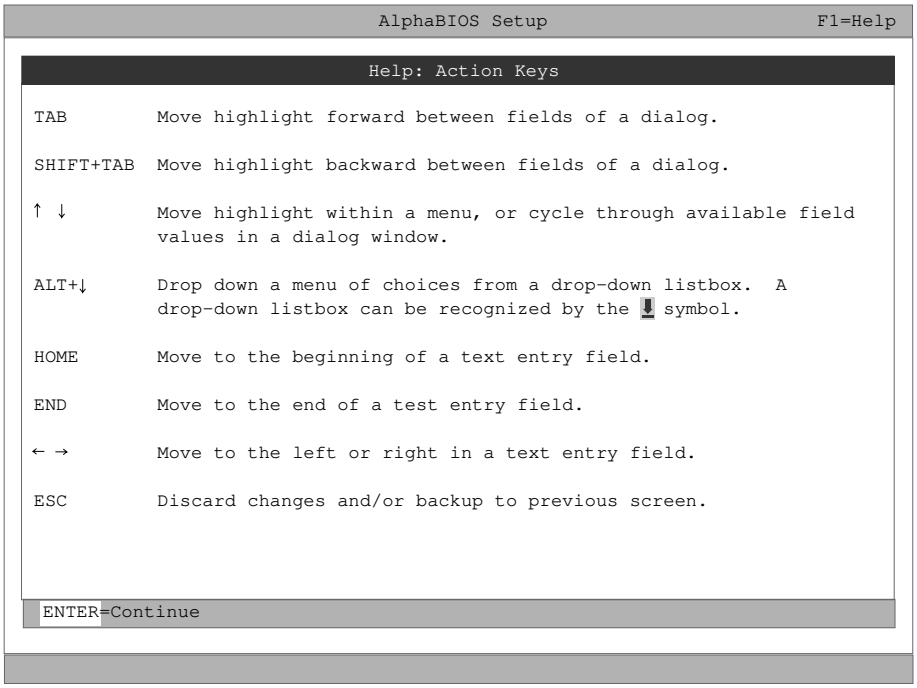
BX-0570-98

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 8-4.

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 8-5).

Figure 8-5 Second-Level Help Screen

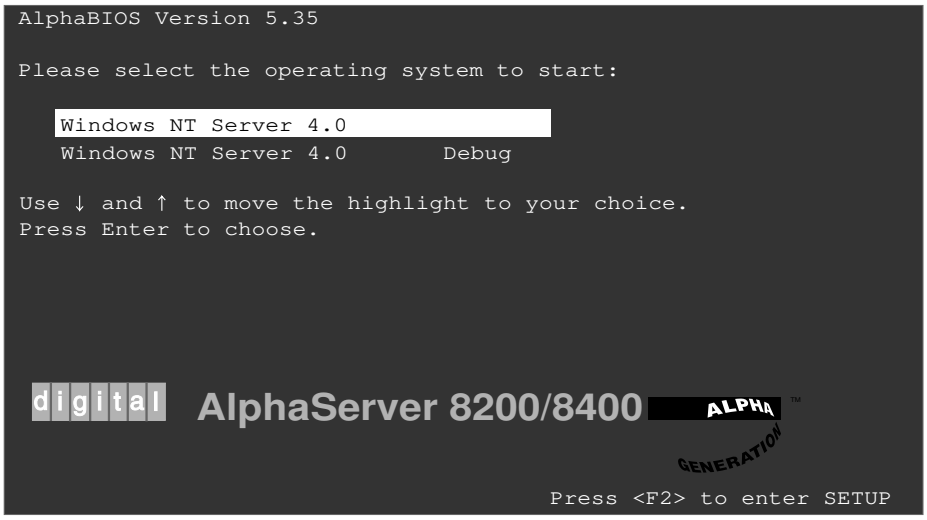


BX-0571-97

8.4 Starting AlphaBIOS

When NT is selected as the startup operating system, you can start AlphaBIOS Setup by pressing F2 from the Boot screen displayed at power-up or reset. You are allowed some time to select setup before the operating system is booted automatically. This time is displayed on the boot screen.

Figure 8–6 Boot Screen

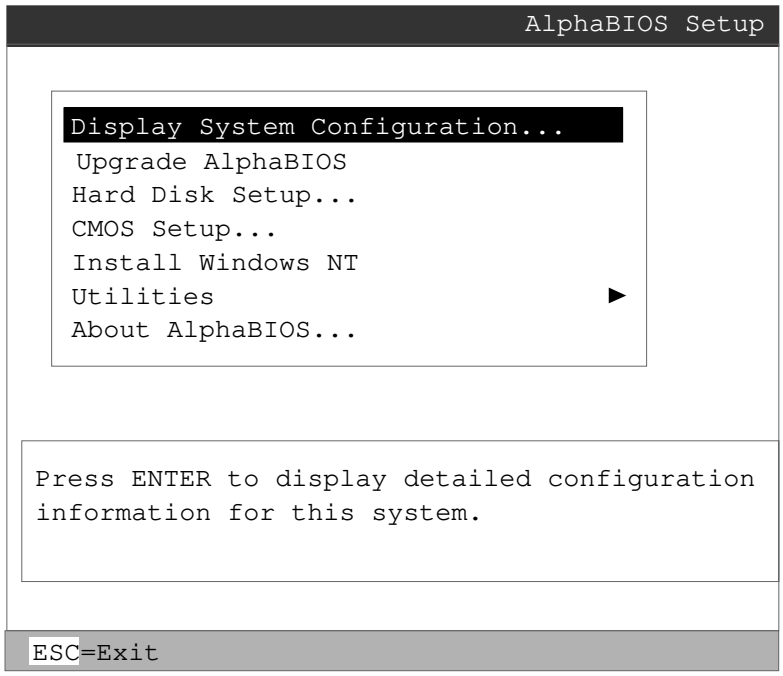


BX-0568-98

You define Windows NT as the operating system to be booted by setting the **os_type** environment variable to **nt** from the SRM console. When the system is powered up, reset, or initialized, the graphics monitor shows status and initialization messages, terminating in the **Boot** screen shown in Figure 8–6. At this point, you are allowed to select from multiple operating systems (if any) to boot. You can press F2 to enter setup at any time before the operating system is booted. **If you do not press F2 before the operating system is booted, you cannot enter AlphaBIOS Setup without resetting the system.**

Once you press F2 and the initialization finishes, the **AlphaBIOS Setup** screen (Figure 8-7) displays. From this screen you can select from the tasks described briefly in Section 8.1. Use the arrow keys to select the menu item you want and press Enter. Following sections describe the tasks you can perform from the **AlphaBIOS Setup** screen.

Figure 8-7 AlphaBIOS Setup Screen

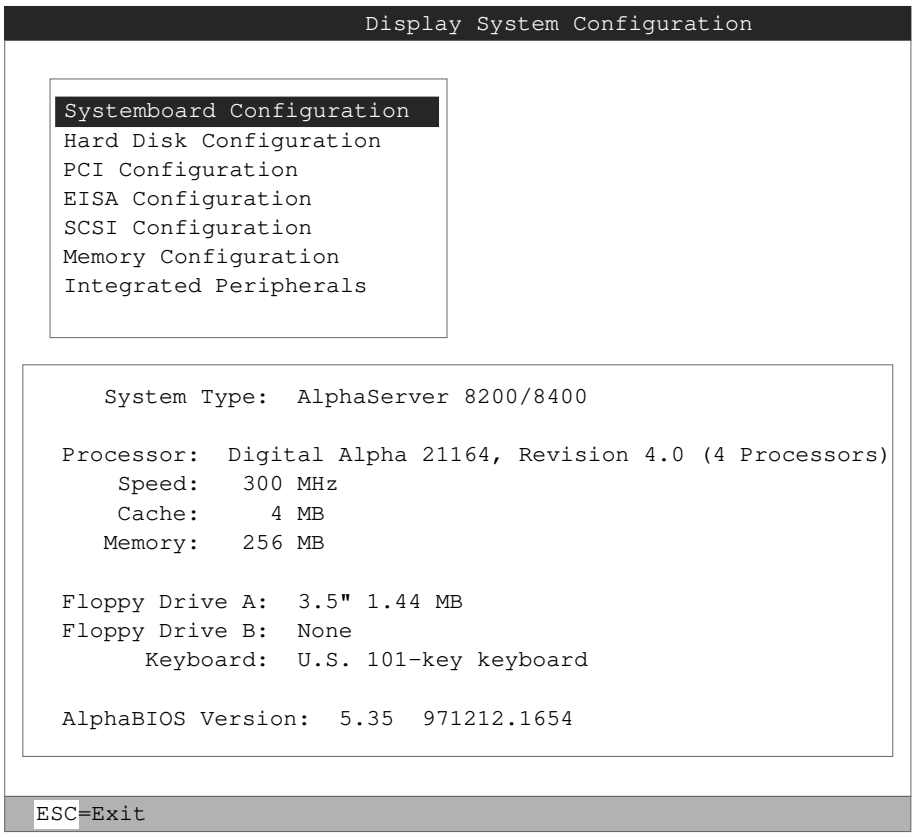


BX-0569-97

8.5 Displaying the System Configuration

The Display System Configuration screen provides information about the system's installed processor, memory, attached devices, and option boards.

Figure 8-8 Display System Configuration Screen



BX-0572-97

Use this procedure to display the system configuration.

1. Enter AlphaBIOS Setup, 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.

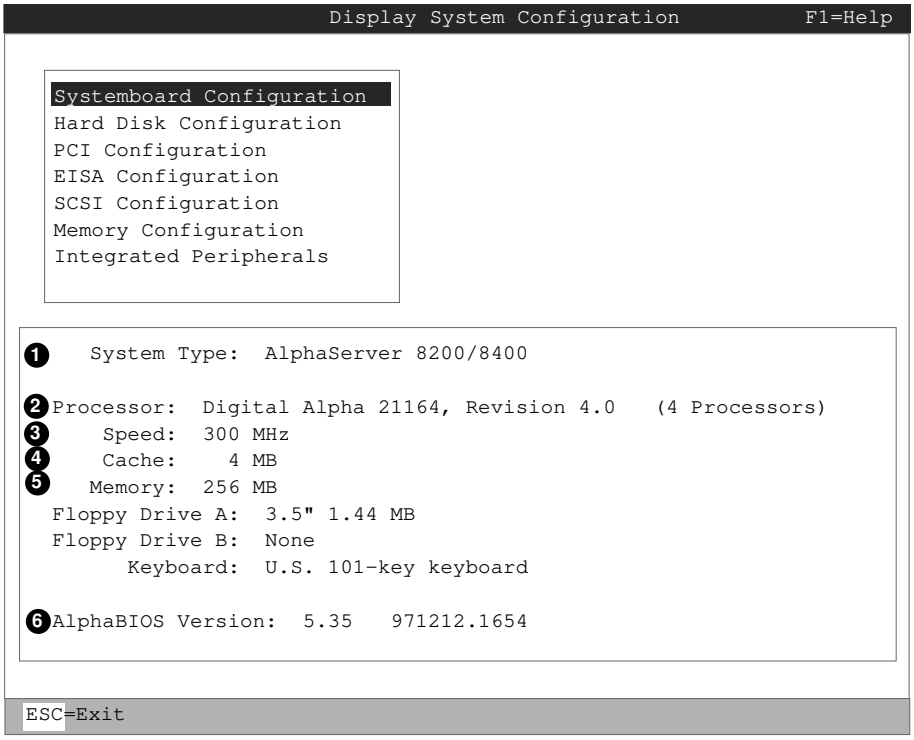
From this screen, you can view configuration information about these system components:

- System motherboard
- Hard disk
- PCI bus
- EISA bus
- SCSI devices
- Memory
- Integrated peripherals

The sections that follow explain the display for each component.

8.5.1 System Board Configuration

Figure 8-9 System Board Configuration



BX-0573-97

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.

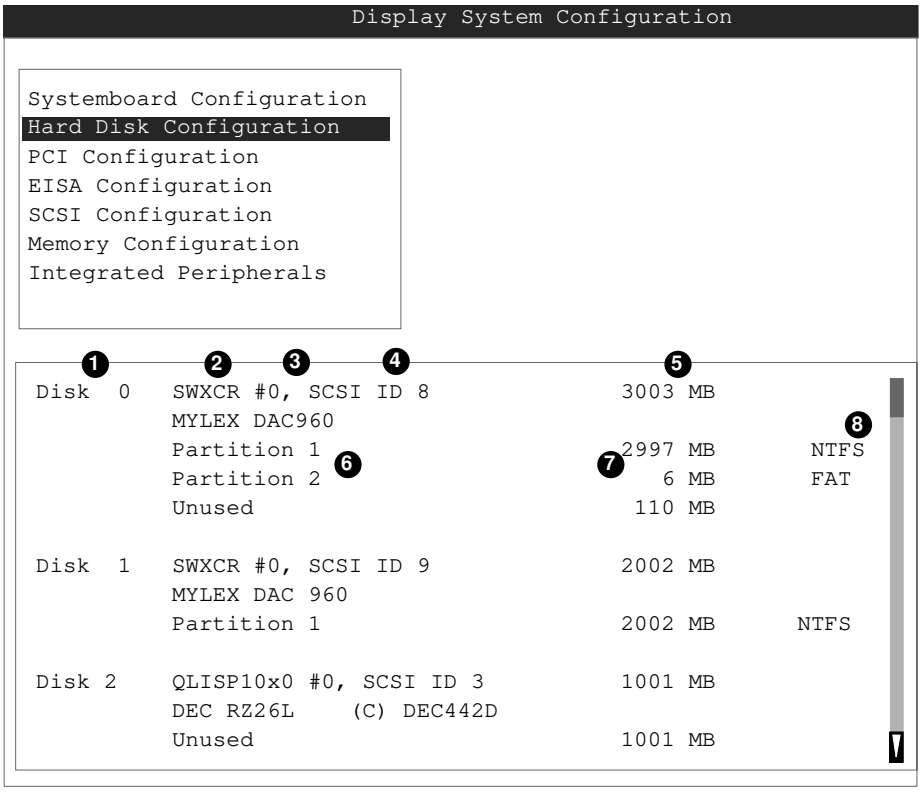
❹ Cache — The amount, in kilobytes, of static RAM cache memory installed.

❺ Memory — The amount, in megabytes, of main memory in the system.

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.

8.5.2 Hard Disk Configuration

Figure 8-10 Hard Disk Configuration



BX-0574-98

*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 8.7).*

Figure 8-10 shows a configuration with multiple hard disks, such that the bottom display shows an arrow to the left, indicating that the display continues. This can happen on several AlphaBIOS displays. To see the rest of the display in this case, press Enter to highlight the first disk in the bottom screen. Then you can use arrow keys to move through the disk display. To move back to the top screen, press ESC.

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.

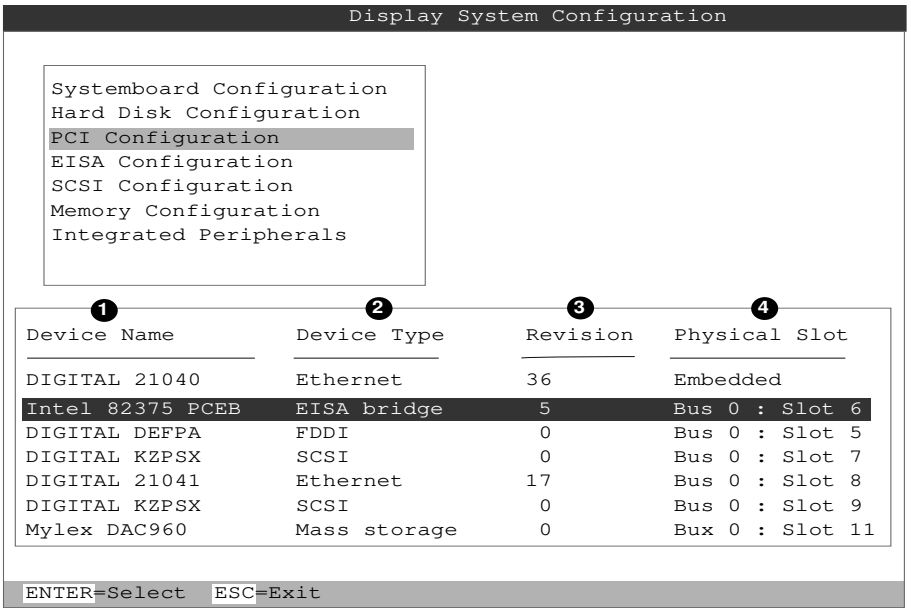
❻ Partition number — Within a single drive, partition numbers are assigned in sequential order: 1, 2, 3, and so on. The partitions populate the drive from the innermost cylinders to the outermost cylinders.

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 unrecognized (if the partition is unformatted).

8.5.3 PCI Configuration

Figure 8-11 PCI Configuration



BX-0575-98

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 Revision — The revision level of the device signifies the number of times it has been updated by the manufacturer.

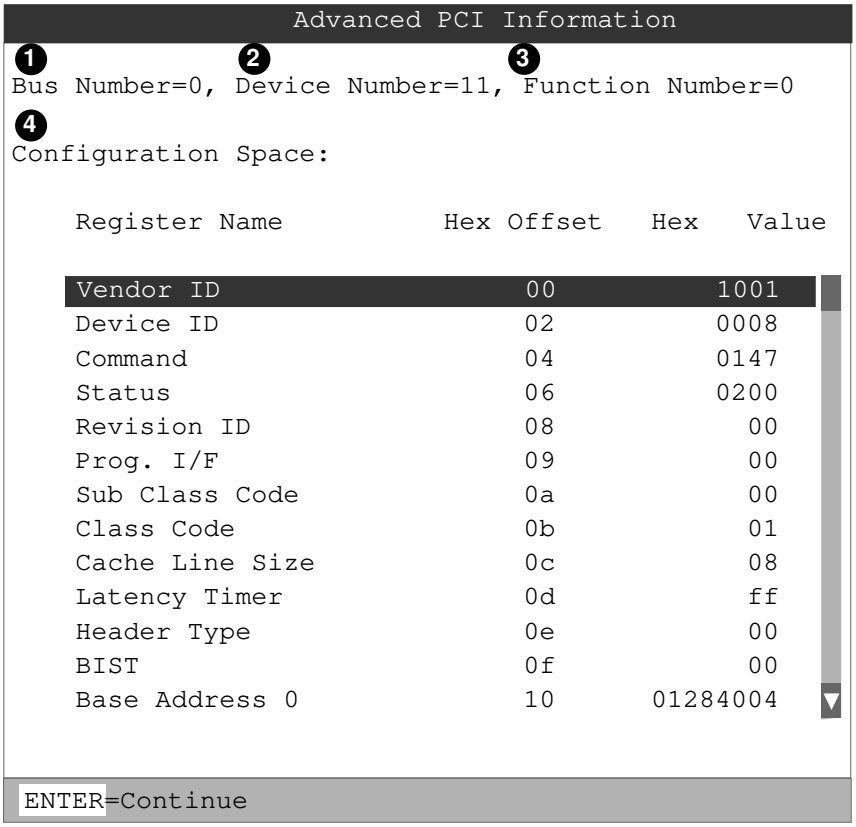
❹ Physical slot — Lists the actual PCI slot number to which the device is attached. Refer to your system manual for information on the physical location of PCI slot locations on your particular system board.

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 8-12 on the next page.

Figure 8-12 Advanced PCI Information



BX-0576-97

❶ Bus number — The virtual PCI bus number.

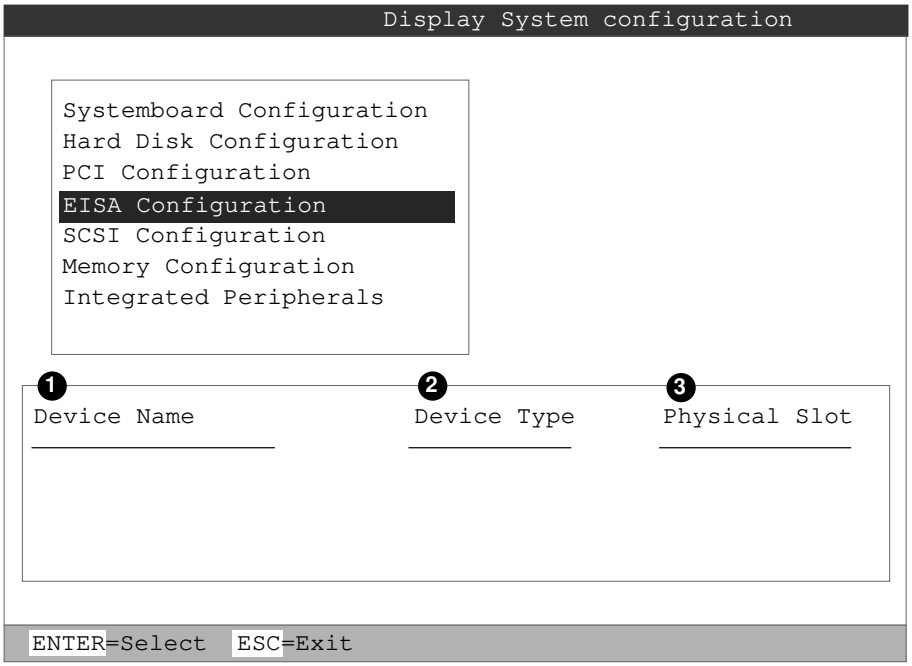
❷ Device number — The PCI bus device number.

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.

8.5.4 EISA Configuration

Figure 8-13 EISA Configuration



BX-0577-97

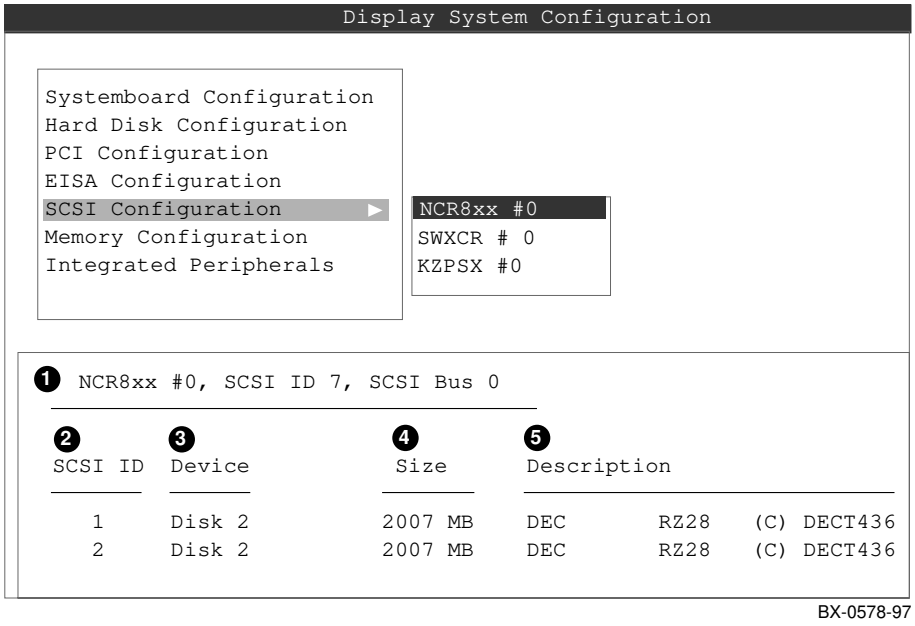
❶ Device name — Includes a three-character manufacturer code, followed by a three-digit board type, followed by a one-digit revision number.

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.

8.5.5 SCSI Configuration

Figure 8-14 SCSI Configuration



BX-0578-97

❶ SCSI controller information — Describes the physical characteristics 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.

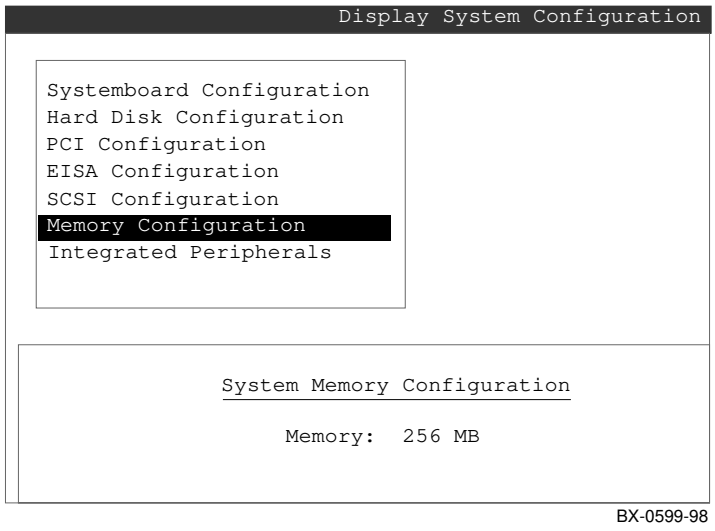
❸ Device type — Displays whether the device is a hard disk, CD-ROM, scanner, or other type of device.

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.

8.5.6 Memory Configuration

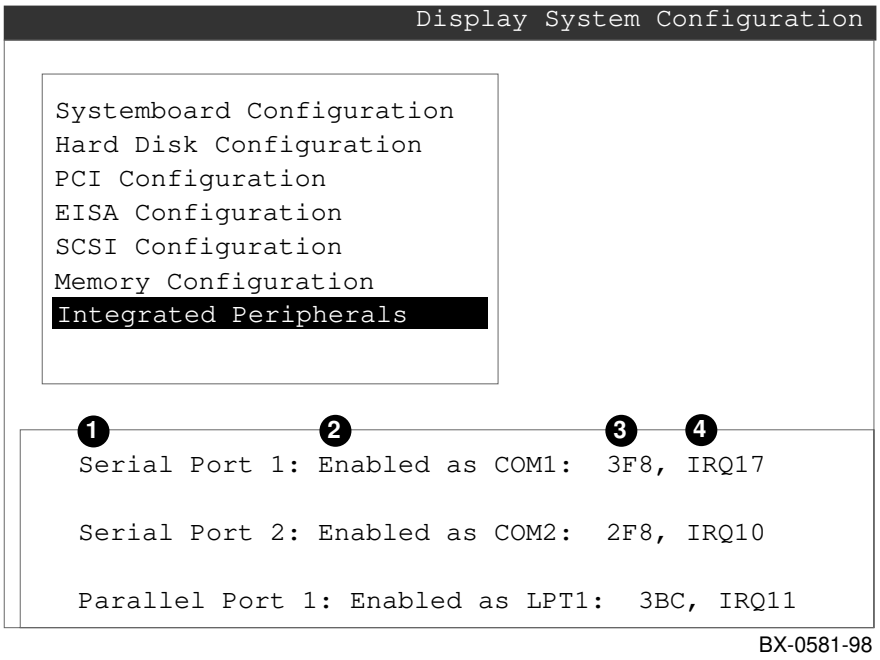
Figure 8-15 Memory Configuration



This screen shows the current memory configuration.

8.5.7 Integrated Peripherals

Figure 8-16 Integrated Peripherals



BX-0581-98

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.

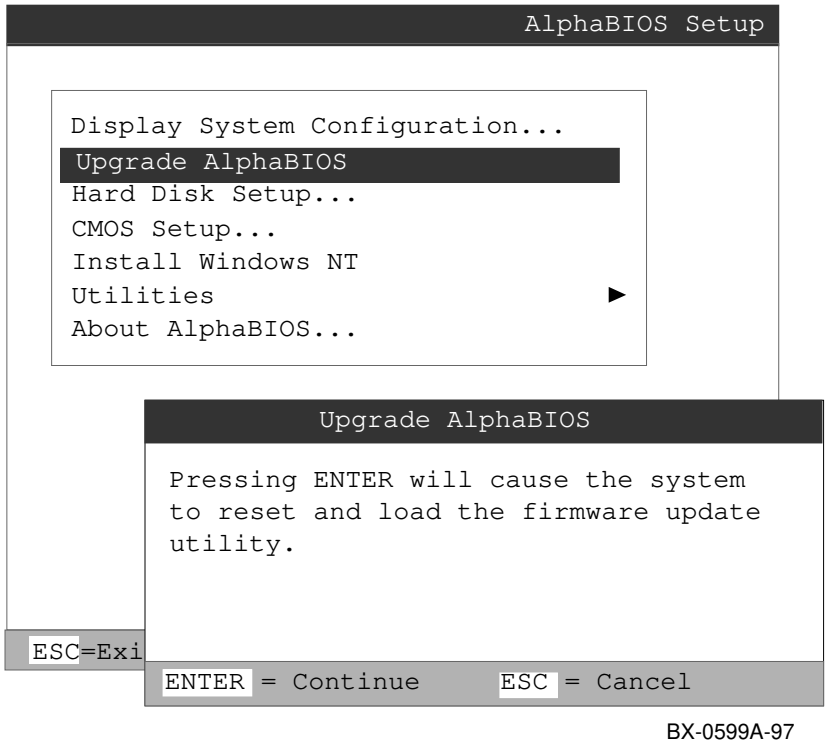
❸ Port address — The physical memory location from and to which data travels as it is received into the device, and sent from the device, respectively.

❹ Interrupt — The interrupt request line (IRQ) used by the device to get the CPU's attention.

8.6 Updating Firmware

Insert the CD-ROM 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 8-17 Updating Firmware



You can use this procedure to boot the Loadable Firmware Update (LFU) utility to upgrade firmware. (You can also use the SRM console command **boot** to run LFU, as described in Appendix B.)

1. Insert the upgrade CD-ROM.
2. Enter SRM console mode. Make sure that the **auto_action** environment variable is set to **boot**, since this setting will be used by the AlphaBIOS firmware to boot LFU. Make sure that the **console** environment variable is set to **serial**.
3. Enter **AlphaBIOS Setup** by restarting your system and pressing F2 before the operating system boots.
4. In the main **AlphaBIOS Setup** screen, select **Upgrade AlphaBIOS** and press Enter. The small window shown in Figure 8-17 displays. Press Enter.
5. The system is reset and initialization information (see ❶ in Figure 8-18) is shown. The LFU banner displays (see ❷). Select the firmware load device. In the example, ❸ shows the user selecting the CD-ROM device.
6. Enter the name of the firmware files list or use the default filename (see ❹). Press Enter.
7. Enter the name and use LFU to update firmware (LFU commands are described in detail in Appendix B). Note that LFU does not display the introductory information about the upgraded firmware contained on the new CD-ROM, as it does when you use the SRM console command **boot**, as described in Appendix B.
8. 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.

Figure 8-18 LFU Load Screen

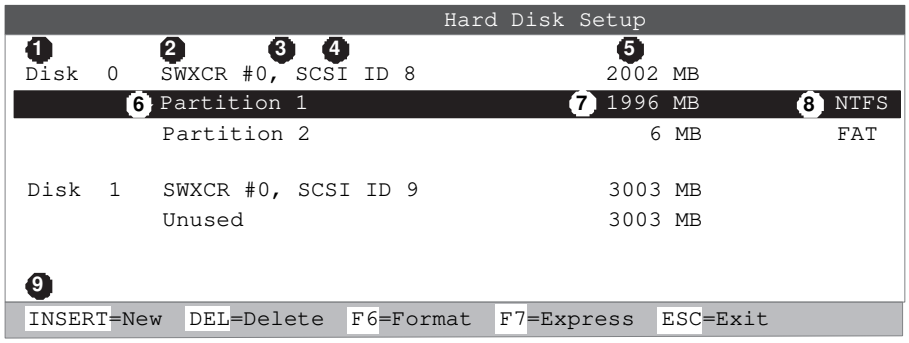
```
AlphaServer 8200 Console T5.1-30, 29-JAN-1998  14:01:45, SROM V3.1

Configuring I/O adapters ❶
tulip0, slot 2, bus 0, hose 1
          .
          .
          .

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

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

Please enter the name of the firmware files list, or
Press <return> to use the default filename [AS8X00FW]: ❹
```

BX-0599B-97

8.7 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 8.7.1 and 8.7.2.

Figure 8-19 Hard Disk Setup Screen



BX-0583-97

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, or unused. The partitions populate the drive from the innermost cylinders to the outermost.

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 unused (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 unused 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.

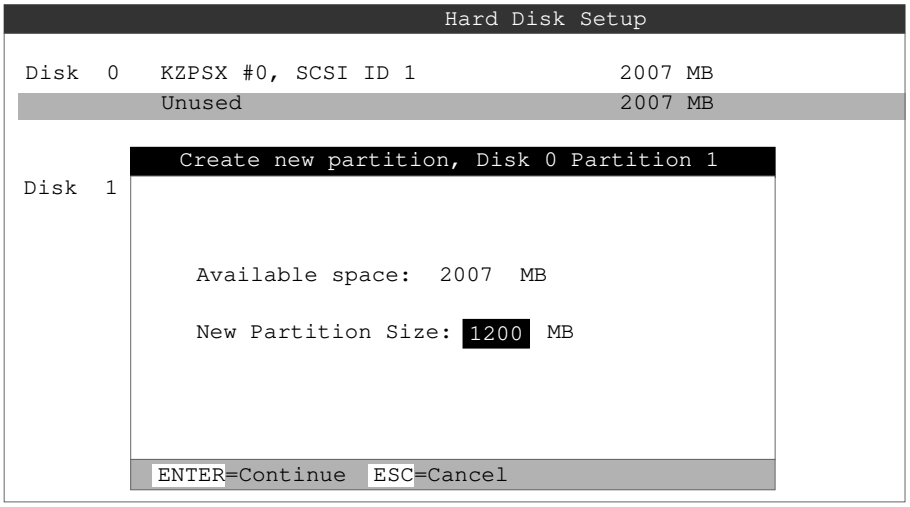
To perform an express hard disk setup

1. If you have any needed information on your disk, back it up before proceeding.
2. Enter **AlphaBIOS Setup**, select **Hard Disk Setup**, and press Enter.
3. Press F7 to enter **Express Setup**.
4. Press F10 to continue with the setup.

8.7.1 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 8-20 Create New Partition Dialog Box



BX-0584-98

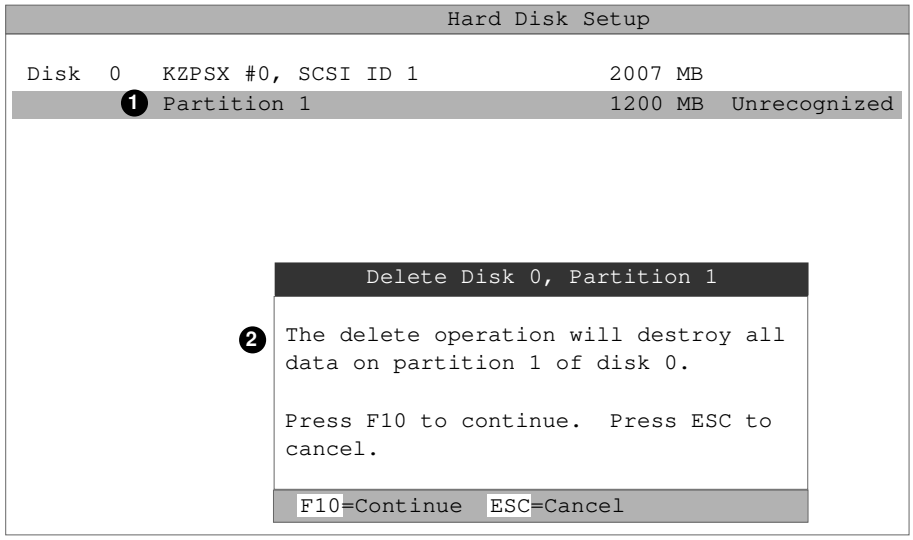
To create a partition

1. Enter **AlphaBIOS Setup** 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 8-20.
5. Type the size of the partition to create and press Enter.

To delete a partition

1. Enter **AlphaBIOS Setup** and select **Hard Disk Setup**. Press Enter.
2. Select the partition to be deleted (see ❶ in Figure 8-21).
3. Press Delete. A dialog box displays (see ❷).
4. Press F10 to confirm the deletion.

Figure 8-21 Delete Partition Dialog Box

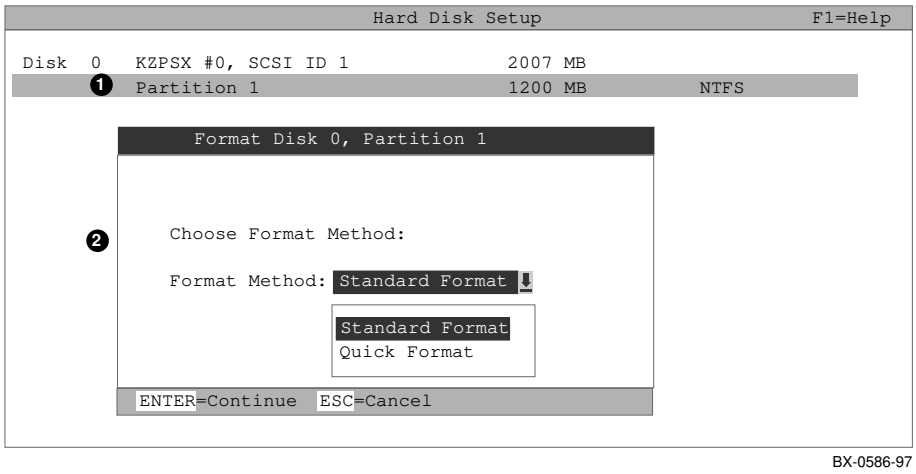


BX-0585-97

8.7.2 Formatting a FAT Partition

AlphaBIOS can format partitions with the FAT file system. Use Windows NT to format a partition using NTFS or to convert a FAT partition to NTFS.

Figure 8-22 Formatting a FAT Partition

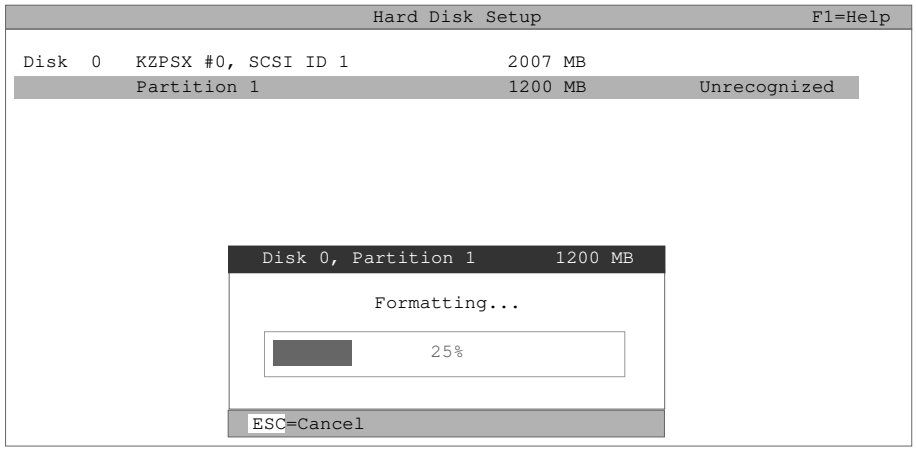


BX-0586-97

To format a FAT partition

1. Enter **AlphaBIOS Setup** and select **Hard Disk Setup**. Press Enter.
2. Select the partition to be formatted (see ❶ in Figure 8-22).
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 8-23 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 8-23 Standard Formatting

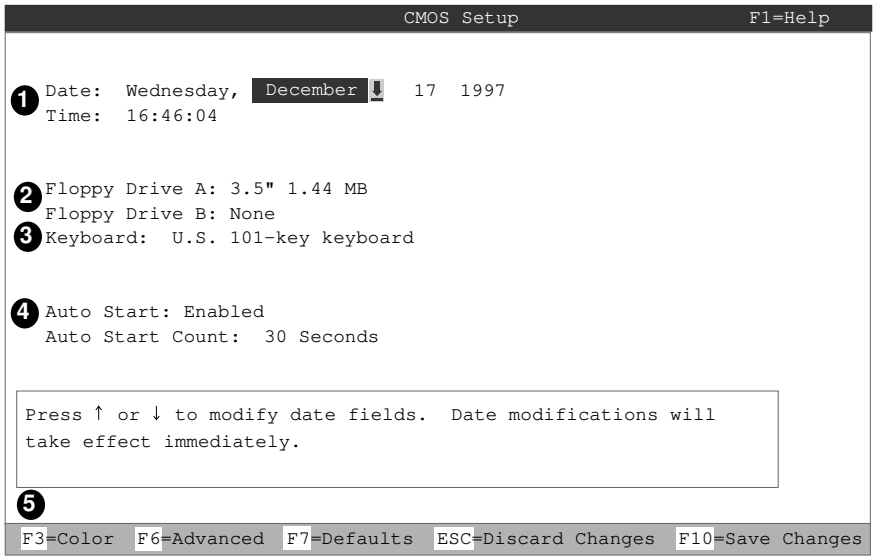


BX-0589-97

8.8 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 8-24 Standard CMOS Setup Screen



BX-0590-98

To enter Standard CMOS Setup

Enter **AlphaBIOS Setup**, 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.

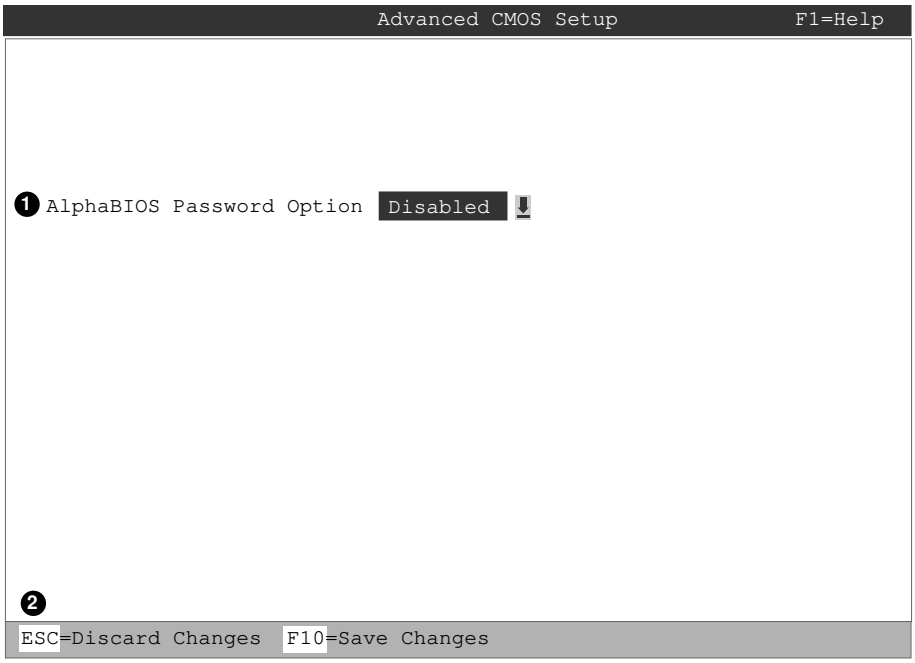
Defaults (F7 key) — Restores the default **Standard CMOS Setup** values without affecting the **Advanced CMOS Setup** values.

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 8-25 Advanced CMOS Setup Screen



BX-0591-97

To enter Advanced CMOS Setup

1. Enter **AlphaBIOS Setup**, select **CMOS Setup**, and press Enter.
2. In the **Standard CMOS Setup** screen, press F6.

❶ Password setup — Enables and disables settings for password setup.

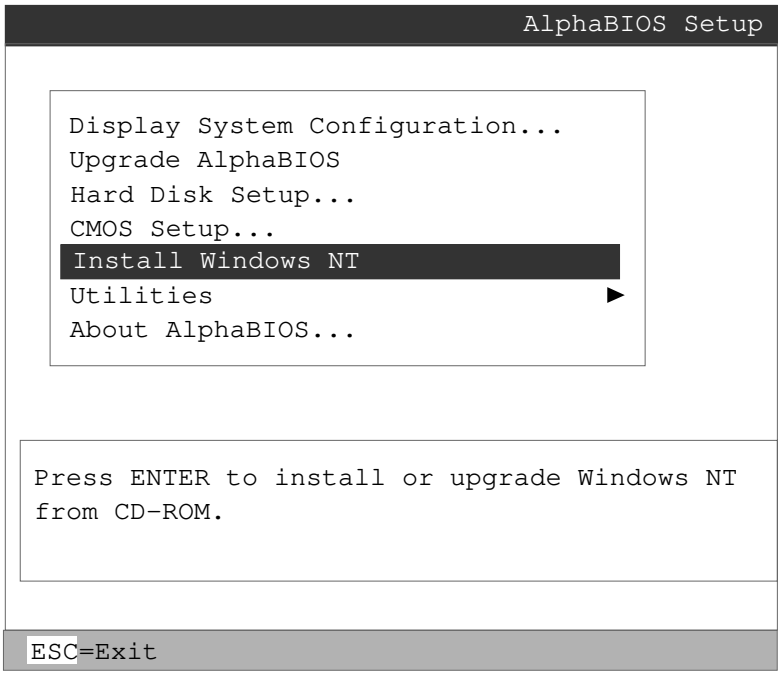
Password protection provides two levels of security for your system: setup, which requires a password to start **AlphaBIOS Setup**, and startup, which requires a password before the system initializes. Startup protection provides more comprehensive protection than setup protection because the system cannot be used at all until the correct password is entered.

❷ Use F10 to save changes, ESC to discard changes.

8.9 Installing Windows NT

Install Windows NT from CD-ROM. Multiple versions of the operating system can be resident in the system at one time.

Figure 8-26 Installing Windows NT



BX-0593-97

If Windows NT was installed at the factory, enter Windows NT setup as described in Section 8.4. If it was not installed, or if you are installing another version, you must have a CD-ROM drive attached to your system.

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. Enter **AlphaBIOS Setup** by pressing the F2 key before the operating system boots.
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. Select and format all disks. See Section 8.7 for information on formatting hard disks.
8. Continue the setup by pressing the F10 key.
9. Go to the procedure below.

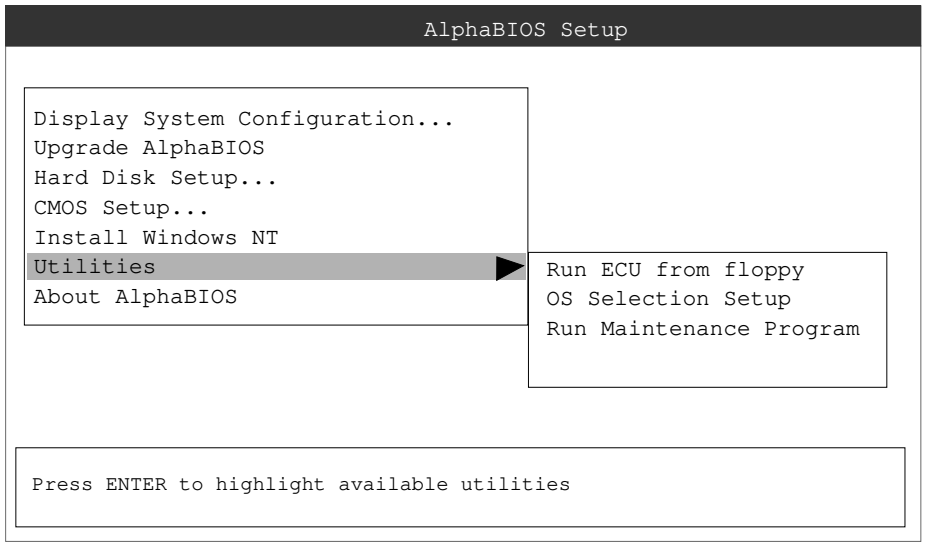
This procedure is for all Windows NT installations:

1. Put the Windows NT CD into the CD-ROM drive.
2. Enter **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.

8.10 Running Utility Programs

Maintenance programs such as RAID and EISA configuration utilities are run from the AlphaBIOS Utilities menu.

Figure 8-27 Utilities Selection



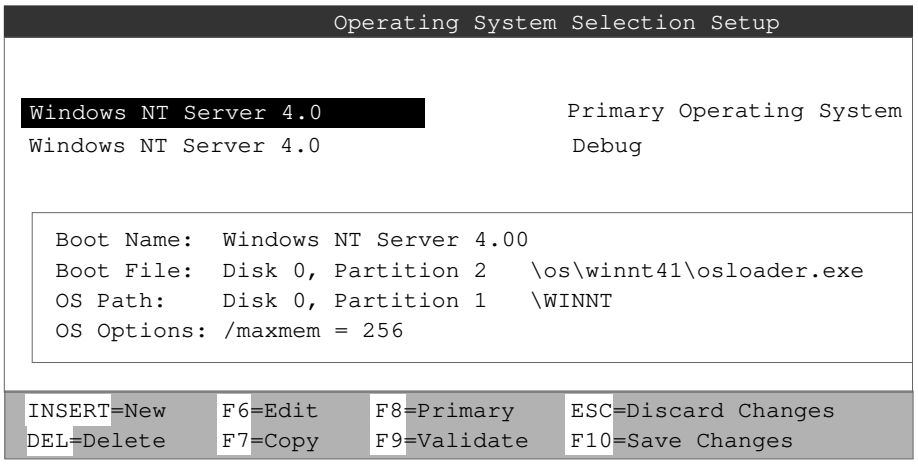
BX-0594-97

The **Utilities** selection from the **AlphaBIOS Setup** screen includes running the EISA configuration utility (ECU) from floppy, setting up a version or versions of the Windows NT operating system for selection, and running maintenance programs such as the RAID configuration and management utility (RA200RCU).

8.11 Selecting the Version of Windows NT

Multiple 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 8-28 Operating System Selections



BX-0595-97

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.

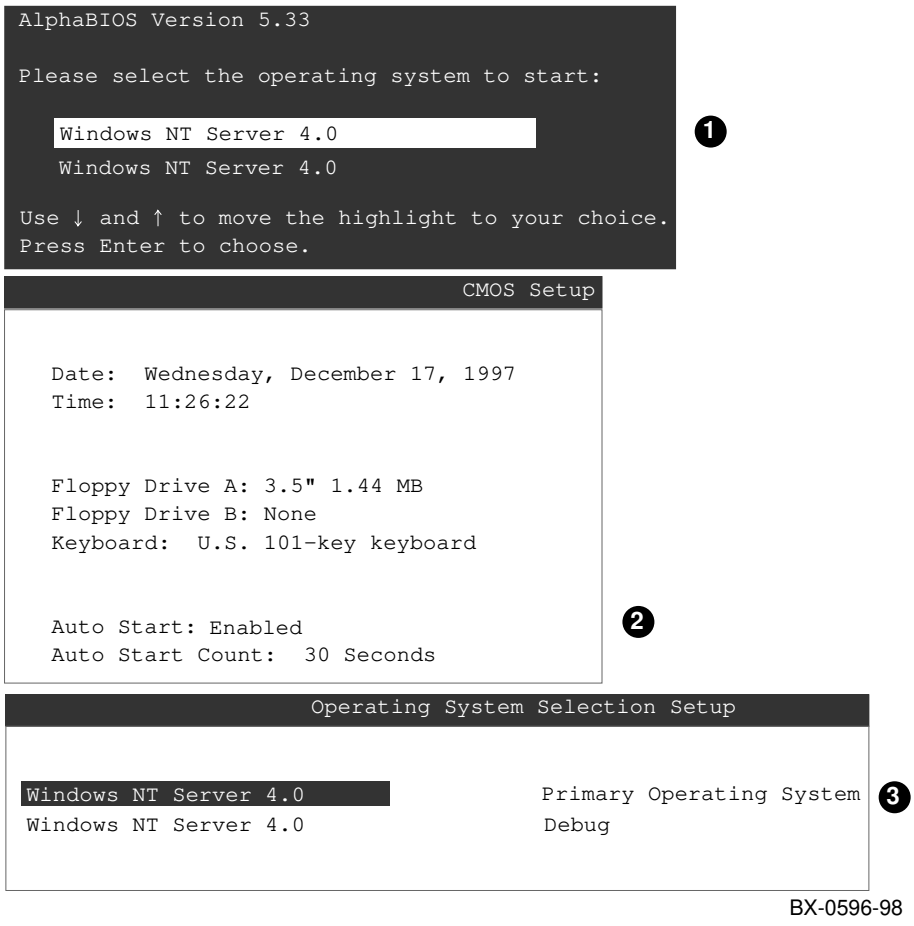
Each operating system selection is a set of information for a version of Windows NT. It describes the disk and partition containing the OSLOADER.EXE file associated with a particular operating system installation, as well as the path to the operating system itself. In addition, the operating system selection 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 8-28 shows how operating system selections appear in AlphaBIOS.

8.11.1 Designating a Primary Operating System

Figure 8-29 Primary Operating System



BX-0596-98

Up to three 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 8-29 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.

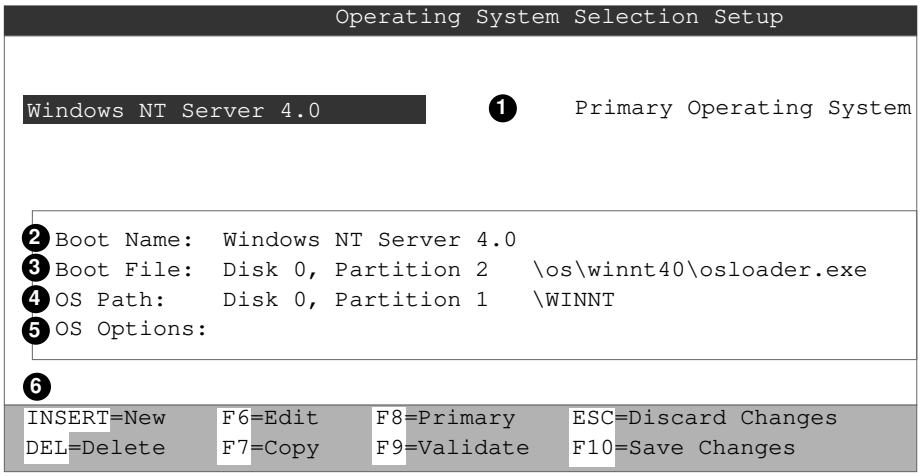
Follow this procedure to designate a new primary operating system:

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.

8.11.2 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 8-30 Operating System Selection Setup



BX-0597-97

AlphaBIOS boots Windows NT in two stages. The first stage involves reading an OS Loader program from a disk. 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.

Appendix A
OpenVMS and DIGITAL UNIX Boot Options

Table A-1 lists the DIGITAL UNIX options used with the **boot** command. Table A-2 lists the Alpha primary boot (APB) options used with the **boot** command for OpenVMS. These options allow you to control various phases of booting.

Table A-1 DIGITAL UNIX Boot Options

Option	Function
a	Boots the system disk to multiuser mode.
d	Do full dumps.
i	Boot to interactive mode plus options (prompt for system image to boot and boot options.
s	Boot to single-user mode.

Table A-2 OpenVMS Boot Options

Hexadecimal Value	Function
1	Allows a conversational boot.
2	Maps XDELTA to a running system
4	Stops the boot procedure at the initial system breakpoint.
8	Performs a diagnostic bootstrap.
10	Stops the boot procedure at the bootstrap breakpoints.
20	Omits the header from the secondary bootstrap image.
40	Inhibits memory testing.
80	Prompts for the name of the secondary bootstrap file.
100	Halts the system before the secondary bootstrap.
2000	Marks corrected read data error pages as bad.
10000	Enables debug messages in the APB.EXE, SYSBOOT.EXE, and EXEC_INIT.EXE files.
8200	Enables user messages in the APB.EXE, BOOT.EXE, and EXEC_INIT.EXE files.

The DIGITAL UNIX and OpenVMS options are used with the **set boot_osflags** command.

Appendix B

Updating Firmware

Use the Loadable Firmware Update (LFU) utility to update system firmware. LFU runs without any operating system and can update the firmware on any system module. LFU handles modules on the TLSB bus (for example, the CPU) as well as modules on the I/O buses (for example, a CI controller on the XMI bus). You are not required to specify any hardware path information, and the update process is highly automated.

Both the LFU program and the firmware microcode images it writes are supplied on a CD-ROM. From the SRM console, you start LFU with the **boot** command. (With Windows NT systems, you can also start LFU from the **Utilities** option in **AlphaBIOS Setup**, as described in Chapter 8.)

A typical update procedure is:

1. Verify the **console** environment variable setting (*must be **serial***).
2. Boot the LFU CD-ROM. (Use the **show config** command to find the device name of the CD-ROM device.)
3. Use the LFU **list** command to show the revisions of modules that LFU can update and the revisions of update firmware.
4. Use the LFU **update** command to write the new firmware.
5. Exit.

Sections in this appendix are:

- Booting LFU
- List
- Update
- Exit
- Display and Verify Commands
- How to Update Corrupted Firmware
- How to Modify Device Attributes

B.1 Booting LFU

LFU is supplied on the Alpha CD-ROM (Part Number AG–PTMW*–BE, where * is the letter that denotes the disk revision). Make sure this CD-ROM is mounted in the in-cabinet CD drive. Boot LFU from the CD-ROM.

Example B-1 Booting LFU from CD-ROM

```
P00>>> sho dev                          ❶
polling for units on isp0, slot 0, bus0, hose0...
dka400.4.0.0.0     DKA400                  RZ26L  440C
polling for units on isp1, slot 1, bus0, hose0...
polling for units on isp2, slot 4, bus0, hose0...
polling for units on isp3, slot 5, bus0, hose0...
dkd400.4.0.5.0     DKD400                  RRD46  0000
dkd500.5.0.5.0     DKD500                  RZ26L  440C
P00>>> boot dkd400              ❷
Building FRU table.............
(boot dkd400.4.0.5.0 -flags 0,a0)
SRM boot identifier: scsi 4 0 5 0 400 ef00 81011
boot adapter: isp3 rev 2 in bus slot 5 off of kftia0 in TLSB slot 8
block 0 of dkd400.4.0.5.0 is a valid boot block
reading 1150 blocks from dkd400.4.0.5.0
bootstrap code read in
Building FRU table…….
base = 200000, image_start = 0, image_bytes = 8fc00
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

The default bootfile for this platform is

        [AS8X00]SRM4920.EXE

Hit <RETURN> at the prompt to use the default bootfile.

Bootfile: ❸

Starting Firmware Update Utility
Unpacking firmware files
      .
      .
```

```
      .
              ***** Loadable Firmware Update Utility *****
--------------------------------------------------------------------------
Function  Description
--------------------------------------------------------------------------

Display     Displays the system's configuration table.
Exit        Return to loadable offline operating environment.
List        Lists the device types and firmware revisions supported by
            this revision of LFU.
Modify      Modifies port parameters and device attributes.
Show        Displays device mnemonic, hardware and firmware revisions.
Update      Replaces current firmware with loadable data image.
Verify      Compares loadable and device images.
? or Help   Scrolls the function table.
--------------------------------------------------------------------------

UPD>  ❹
```

❶ Use the **show device** command to find the name of the RRDCD drive.

❷ Enter the **boot** command to boot LFU from the RRDCD drive. This drive has the device name dkd400.

❸ Press Enter for the default bootfile, or enter the directory and file name of the utility.

LFU starts, displays a summary of its commands, and issues its prompt (UPD>).

❹ UPD> is the LFU prompt for command entry.

B.2 List

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

Example B-2 List Command

```
UPD> list

Device      Current Revision   Filename Update Revision
kn7cc-ab0_arc    V5.3-14        arcrom          V5.3-14
kn7cc-ab0        V4.9-20        kn7cc_fw        V4.9-20
kn7cc-ab1_arc    V5.3-14        arcrom          V5.3-14
kn7cc_ab1        V4.9-20        kn7cc_fw        V4.9-20
kn7cc-ab2_arc    V5.3-14        arcrom          V5.3-14
kn7cc-ab2        V4.9-20        kn7cc_fw        V4.9-20
kzpsa0          A10             kzpsa_fw       A10
kzpsa1          A10             kzpsa_fw       A10
pfi0              3.10          dfxaa_fw         3.10
                                cipca_fw       A315
                                cixcd_fw         5
                                defaa_fw         1.10
                                demfa_fw         2.1
                                demna_fw         8.3
                                kfmsb_fw         2.4
                                kzmsa_fw         5.6

UPD>
```

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 that is recommended for updating that firmware
- Update revision — The revision of the firmware update

B.3 Update

The update command writes new firmware from the CD-ROM 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 CD-ROM image.

Example B-3 Update Command

```
UPD> update kn7cc-ab0                                    ❶
WARNING:  updates may take several minutes to complete for each device.


Confirm update on:  kn7cc-ab0_arc   [Y/(N)] y ❷

                     DO NOT ABORT!
kn7cc-ab0_arc   Updating to V5.3-14… .Verifying V5.3-14…   Passed

Confirm update on:  kn7cc-ab0     [Y/(N)] y ❷

                     DO NOT ABORT!
kn7cc-ab0       Updating to V4.9-20... Verifying V4.9-20... Passed. ❸

UPD> update kzpsa0                                       ❹
WARNING:  updates may take several minutes to complete for each device.

Confirm update on:  kzpsa0  [Y/(N)] y

                     DO NOT ABORT!
kzpsa0          Updating to A10...  FAILED.  ❺

UPD> exit

Errors occurred during update with the following devices:
kzpsa0

Do you want to continue to exit?
Continue [Y/(N)]y
Initializing...
        [self-test display appears]
```

❶ This command requests a firmware update for a specific module. If you want 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.

❷ LFU requires you to confirm the update. For processors, the first update to confirm is the AlphaBIOS firmware; the second is the SRM console firmware. In either case, the default is no.

❸ Status message reports update and verification progress.

❹ This is a second example.

❺ The update failed. This could indicate a bad device.

Continued on next page

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

Example B–3 Update Command (Continued)

```
UPD> update                                                    ❻

confirm update on:                                             ❼
kzpsa0
kzpsa1
pfi0

[Y/(N)]n

UPD> update kzpsa0 -path cipca_fw                              ❽
WARNING: updates may take several minutes to complete for each device.


Confirm update on:  kzpsa0   [Y/(N)]y

                          DO NOT ABORT!

Kzpsa0            firmware filename 'kdm70_fw' is bad

UPD>
```

❻ When you do not specify a device name, LFU tries to update all devices.

❼ LFU lists the selected devices to update and prompts before devices are updated.

❽ In this next example, the **-path** option is used to update a device with different firmware from the LFU default. A network location for the firmware file can be specified with the **-path** option. In this example, the firmware filename is not a valid file for the device specified.

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

B.4 Exit

The exit command terminates the LFU program, causes system initialization and self-test, and returns the system to console mode.

Example B-4 Exit Command

```
UPD> exit                                                          ❶
Initializing...
       [self-test display appears]
P00>>>                                                             ❷

UPD> update kzpsa0
WARNING: updates may take several minutes to complete for each device.
Confirm update on:  kzpsa0  [Y/(N)]y

                          DO NOT ABORT!

kzpsa0          Updating to A10...  FAILED.

UPD> exit

Errors occurred during update with the following devices:          ❸
kzpsa0

Do you want to continue to exit?                                   ❹
Continue [Y/(N)]y                                                  ❺
Initializing...
       [self-test display appears]
P00>>>
```

❶ At the UPD> prompt, **exit** causes the system to be initialized.

❷ The console prompt appears.

❸ Errors occurred during an update.

❹ Because of the errors, confirmation of the exit is required.

❺ Typing **y** causes the system to be initialized and the console prompt to appear.

B.5 Display and Verify Commands

Display and verify commands are used in special situations. Display shows the physical configuration. Verify repeats the verification process performed by the update command.

Example B-5 Display and Verify Commands

```
UPD> display                    ❶

         Name                    Type   Rev  Mnemonic
   TLSB
   0++   KN7CC-AB                8014  0000  kn7cc-ab0
   3+    MS7CC                   5000  0000  ms7cc0
   5+    MS7CC                   5000  0000  ms7cc1
   7+    KFTHA                   2000  0036  kftha0
   8+    KFTIA                   2020  0000  kftia0

   C0 Internal PCI connected to kftia0     pci0
   0+    ISP1020            10201077  0001  isp0
   1+    ISP1020            10201077  0001  isp1
   2+    DECchip 21040-AA     21011  0023  tulip0
   4+    ISP1020            10201077  0001  isp2
   5+    ISP1020            10201077  0001  isp3
   6+    DECchip 21040-AA     21011  0023  tulip1
   7+    PCI NVRAM            71011  0000  pci_nvram0

   C4 XMI connected to kftha0              xmi0
   2+    DEMNA                  C03  0803  demna0
   4+    CIXCD                  C2F  0311  cixcd0
   8+    DWLMA                 102A  0105  dwlma0
   A+    KCM44                  C22  5E50  kcm440

   C7 PCI connected to kftha0              pci1
   6+    DECchip 21040-AA     21011  0023  tulip2
   A+    KZPSA                81011  0000  kzpsa0

UPD> verify kzpsa0              ❷
kzpsa0          Verifying A10...  PASSED.

UPD>
```

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

❷ **Verify** reads the firmware from the module into memory and compares it with the update firmware on the CD-ROM. If a module already verified successfully when you updated it, but later failed self-test, you can use **verify** to tell whether the firmware has become corrupted.

B.6 How to Update Corrupted Firmware

If LFU identifies a device as unknown, either the firmware on the module is corrupted or the console does not support or does not recognize the device.

Example B-6 Updating an "Unknown" Device

```
UPD> display                    ❶

        Name            Type   Rev  Mnemonic
  TLSB
  0+    KN7CC-AA        8014  0000  kn7cc-aa0
  4+    MS7CC           5000  0000  ms7cc0
  6+    MS7CC           5000  0000  ms7cc1
  8+    KFTHA           2000  0000  kftha0

  C0 XMI                              xmi0
  2+    ?????            810  A4A6  unknown0 ❷
  8+    DWLMA           102A  0003  dwlma0
  C+    KDM70            C22  2B01  kdm700
  E+    DEMNA            C03  0803  demna0

  C3 XMI                              xmi1
  1+    DEMFA            823  9120  demfa0
  2+    ?????            810  A4A6  unknown1 ❷
  8+    DWLMA           102A  0104  dwlma1
  A+    KCM44            C22  69F1  kcm440
  C-    ?????           FDFF  FFFF  unknown2 ❷
  E+    CIXCD            C2F  0211  cixcd0

UPD> update unknown*            ❸

Confirm update on:  unknown2   [Y/(N)]y
WARNING: updates may take several minutes to complete for each device.

                          DO NOT ABORT!
```

❶ Issue the **display** command. The display indicates an unknown device — LFU is unable to recognize the device type. You can identify the unknown device by looking at the physical configuration.

❷ **Display** shows that the unknown devices are in slot 2 of the first XMI bus and slots 2 and 12 of the second XMI.

❸ Issue the command **update unknown***.

Continued on next page

Example B–6 Updating an “Unknown” Device (Continued)

```
unknown2          Updating to 2...  Verifying 2...  PASSED.  ❹

UPD> exit         ❺
Initializing...

       [self-test display appears]
P00>>> sho config

        Name            Type   Rev  Mnemonic
  TLSB
  0+    KN7CC-AA        8014  0000  kn7cc-aa0
  4+    MS7CC           5000  0000  ms7cc0
  6+    MS7CC           5000  0000  ms7cc1
  8+    KFTHA           2000  0000  kftha0

  C0 XMI                              xmi0
  2+    ?????            810  A4A6  unknown0  ❻
  8+    DWLMA           102A  0003  dwlma0
  C+    KDM70            C22  2B01  kdm700
  E+    DEMNA            C03  0803  demna0

  C3 XMI                              xmi1
  1+    DEMFA            823  9120  demfa0
  2+    ?????            810  A4A6  unknown1  ❻
  8+    DWLMA           102A  0104  dwlma1
  A+    KCM44            C22  69F1  kcm440
  C+    CIXCD            C2F  0211  cixcd0    ❼
  E+    CIXCD            C2F  0211  cixcd1

UPD>
```

❹ Status message indicates that the update succeeded.

❺ To make the device known, initialize the system by exiting LFU.

❻ The modules in slot 2 of the first XMI and slot 2 of the second are still unknown. The console either does not support or does not recognize these devices. (The device may be new and not supported by the console installed. Check the current list of supported devices and upgrade the console revision if necessary. For information on finding the current list, see Appendix B.)

❼ Initialization has made the device known to the system. The previously unknown device is now assigned device mnemonic cixcd0 by the system. The previous cixcd0 is now cixcd1.

B.7 How to Modify Device Attributes

The modify command can change parameters stored in EEPROM on the following devices: KZMSA, DEC LANcontroller 400 (DEMNA), KCM44, and KFMSB. The attributes are specific to each device.

Example B-7 Modify Command

```
UPD> modify kzmsa0                                    ❶

kzmsa0

        Local Console:                  ENABLED

        Local Console:                  ENABLED
        Log Selftest Errors:            ENABLED
        Log NRC 53C710 RBD Errors:      ENABLED
        Log XMI RBD Errors:             ENABLED
        Log XZA RBD Errors:             ENABLED
        RBD Error Logging:              DISABLED
        RBD Error Frame Overflow:       ENABLED         Read Only
        Hard Error Frame Overflow:      ENABLED         Read Only
        Soft Error Frame Overflow:      DISABLED        Read Only
        FW Update Error Frame Overflow: DISABLED        Read Only
        Disable Reset Channel 0:        DISABLED
        Disable Reset Channel 1:        DISABLED
        Chnl 0 Fast SCSI:               DISABLED
        Chnl 1 Fast SCSI:               DISABLED

        Channel_0 ID:                   07
        Channel_1 ID:                   07

        Module Serial Number:           abcdefghi

Do you wish to modify any of these parameters? [y/(n)] y  ❷
```

❶ When you modify the KZMSA, LFU first displays all the parameters.

❷ LFU asks if you want to modify any parameter values. The default response is **no**.

Continued on next page

Example B–7 Modify Command (Continued)

```
    Local Console:                  ENABLED         Change? [y/(n)]
    Log Selftest Errors:            ENABLED         Change? [y/(n)]
    Log NRC 53C710 RBD Errors:      ENABLED         Change? [y/(n)] y ❸
    Log XMI RBD Errors:             ENABLED         Change? [y/(n)]
    Log XZA RBD Errors:             ENABLED         Change? [y/(n)]
    RBD Error Logging:              DISABLED        Change? [y/(n)]
    Disable Reset Channel 0:        DISABLED        Change? [y/(n)]
    Disable Reset Channel 1:        DISABLED        Change? [y/(n)]
    Chnl 0 Fast SCSI:               DISABLED        Change? [y/(n)]
    Chnl 1 Fast SCSI:               DISABLED        Change? [y/(n)]

    Channel_0 ID:                   07         Change? [y/(n)]
    Channel_1 ID:                   07         Change? [y/(n)]

    Module Serial Number:           abcdefghi   X   Change? [y/(n)]


    Local Console:                  ENABLED  ❹

    Local Console:                  ENABLED
    Log Selftest Errors:            ENABLED
    Log NRC 53C710 RBD Errors:      DISABLED
    Log XMI RBD Errors:             ENABLED
    Log XZA RBD Errors:             ENABLED
    RBD Error Logging:              DISABLED
    RBD Error Frame Overflow:       ENABLED         Read Only
    Hard Error Frame Overflow:      ENABLED         Read Only
    Soft Error Frame Overflow:      DISABLED        Read Only
    FW Update Error Frame Overflow: DISABLED        Read Only
    Disable Reset Channel 0:        DISABLED
    Disable Reset Channel 1:        DISABLED
    Chnl 0 Fast SCSI:               DISABLED
    Chnl 1 Fast SCSI:               DISABLED

    Channel_0 ID:                   07
    Channel_1 ID:                   07

    Module Serial Number:           abcdefghi

Modify kzmsa0 with these parameter values? [y/(n)] y ❺

UPD>
```

❸ LFU prompts for parameters to modify. This example modifies one parameter on the KZMSA0; logging of ROM-based diagnostic errors is now enabled.

❹ LFU displays the list of parameters with modifications.

❺ If these modified values are acceptable, type **y**.

Appendix C

Running Configuration Utilities from the SRM Console

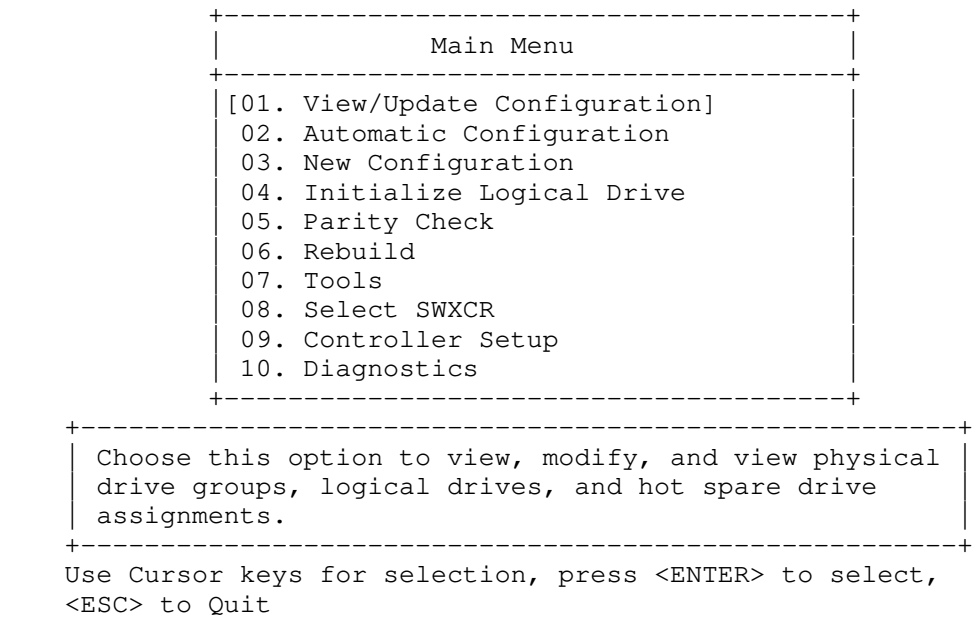
The **run** command has four options, as follows:

- **rcu** - Run the RAID Configuration Utility
- **swxcrfw** - Update the firmware on the RAID Controller
- **eeromcfg** - Run the ISP1020 EEPROM Configuration Utility
- **util_cli** - Run the KZPSA Configuration Utility

This appendix describes some of the utilities that are available when using the **run** command. **Rcu** and **swxcrfw** are on a floppy and require a floppy drive. **Eeromcfg** and **util_cli** are on a CD. Topics discussed here include running the RAID Configuration Utility (to automatically or interactively configure a RAID array) and the ISP1020 Configuration Utility. For further information on these utilities, and for information on the other two, refer to the user documentation that ships with each utility. For information on the EISA Configuration Utility (ECU), refer to Section 7.5.20.

C.1 Configuring a RAID Storage Array

Once you have selected the **run** command and the **rcu** option, the utility displays the main menu:



Using the Standalone Utility

Use the arrow keys to move through the menu and highlight your selection, or you can type the highlighted letter (usually the first letter of the option). Press the Enter key to select the option. You can use the Enter key or the Return key interchangeably.

As you move the cursor from one option to the next, the message window changes to display a message for the highlighted option.

Exiting the Utilities

Press the Esc key twice from the main menu to exit the standalone utility. The utility prompts you to confirm that you want to exit. Select *Yes* and press the Enter key to exit.

Understanding the Configuration Methods

You can configure your array in either of the following ways:

1. **Automatically**. You can configure automatically only if:
   a) You want to configure one RAID 5 logical RAID drive (requires between three and eight drives of same capacity)
   b) You want to configure up to 8 JBODs of any capacity
2. **Interactively**. Configure interactively if you want to:
   a) Create more than one drive group
   b) Create more than one logical RAID drive
   c) Specify RAID levels
   d) Specify caching policy
   e) Define a hot spare

Your RAID subsystem stores the configuration information on flash EEPROM/NVRAM on the RAID controller. If your RAID controller fails, you must restore your configuration from a backup copy on floppy disk, or create a new configuration to access the data on your subsystem.

Configuring Automatically

To configure automatically, you must do the following:

1. Select the Automatic Configuration option
2. Initialize the logical drives
3. Save your configuration to diskette

If you want to define a spare drive, you must configure interactively.

Selecting the Automatic Configuration Option

To configure automatically, follow these steps:

1. Select the *Automatic Configuration* option from the main menu and either one of the following happens depending upon whether a configuration currently exists. If a configuration currently exists, the utility displays a warning message saying that a valid configuration exists and if you proceed, you will destroy it. A confirmation window also appears. See Step 2. If no configuration exists, then the *Automatic Configuration* menu appears. Go to Step 3.
2. Select Yes to delete the current configuration (select No to quit without changing the configuration and to return to the main menu). The *Automatic Configuration* menu appears.
3. Use the arrow keys to highlight RAID 5 or JBOD and press the Enter key to select that configuration. The utility displays a window with this message: Do you want to have Write Cache enabled?
4. Press the Enter key to select No. This sets the cache policy to Write Through (the recommended policy). The utility displays a window with *Automatic Configuration* at the top of the screen and the message, `Saving configuration, please wait ...`, at the bottom of the screen. The utility then displays a screen with your system configuration information.
5. Press any key to return to the main menu.

```
--------------------------------------------------------------------

                 Automatic Configuration
    Number of Logical drives  = 1
    Raid Level                = 5
    Write Cache               = Enabled
    Number of Physical drives = 6
    Available Capacity        = 18245 MB
              Automatic configuration successfully done.
Make certain to INITIALIZE Logical drive 0 before exiting
this utility.
                  Press any key to return to Main Menu
--------------------------------------------------------------
```

See the *StorageWorks RAID 200 Subsystems Controller Installation and Standalone Configuration Utility User's Guide* for information on how to initialize a RAID array.

Configuring Interactively

To configure interactively, do the following:

1. Create one or more drive groups
2. Create one or more logical RAID drives
3. Create a logical RAID drive of RAID level 0, 0 + 1, or 1
4. Define a spare drive
5. Specify caching policy

To configure interactively, you must do the following:

1. Create one or more drive groups

   A drive group defines the drives that you want to work together as the available space for the logical RAID drives that you will create.

2. Create a logical RAID drive

   A logical RAID drive allows your system to see and respond to a drive group as a single drive and defines how your system will store data in that space, based on the RAID level selected.

3. Add a hot spare (optional)

   A hot spare drive is a drive available in your subsystem for the controller to automatically begin to use, in the event of a disk failure from a redundant logical RAID drive, to store the data of the failed drive.

4. Initialize the logical RAID drives

   Initialize a logical RAID drive before you use it to ensure consistent RAID parity information.

5. Save your configuration to diskette

   Your RAID subsystem stores the configuration information on flash EEPROM/NVRAM on the RAID controller. In the event that your RAID controller fails, you must restore your configuration from a backup copy on floppy disk, or create a new configuration to access the data on your subsystem.

Creating a Drive Group

A drive group is from one to eight drives that operate as a single drive. Determine how many drives to use for each drive group based on the following:

1. The RAID levels of the logical RAID drives you will create

   See Table C-1 to see the minimum and maximum number of drives allowed for each RAID level. (You can create multiple logical RAID drives of varying RAID levels on a drive group.)

2. The amount of disk space you need.

Table C-1 shows the amount of storage available for unique data for each of the RAID levels. Also, consider the capacities of each of the individual disks. The capacities of each of the individual disks are affected by the other disks in a group. Disk drives of varying capacities are limited to the capacity of the lowest capacity drive in the drive group. Table C- 2 provides examples of how the capacity of each individual drive affects the capacities of the other drives in the group. To maximize your disk space resources, use drives of the same capacity in a drive group.

If you plan to use more than one drive group, remember:

1. Each disk can belong to only one drive group at a time.
2. You can create a maximum of eight drive groups.
3. Also, after you create drive groups you must arrange them in the order in which you want to use them. (You must use all the available space on a drive group before you can use another one.) So, consider what data you want to put on which drive group before arranging your drives, because you can only delete drive groups in the reverse order in which they were arranged. For example if you create drive groups a, b, and c, and arrange them in that order, then you must delete drive groups c and b before you can delete drive group a.

Creating a drive group requires the following procedures:

1. Defining a drive group
2. Arranging a drive group

First determine how you want to allocate your drive resources and then configure the drives in your array.

Table C-1 Number of Drives You Can Use in a Drive Group for Each RAID Level

RAID Level	Number of Drives in Drive Group	Amount of Storage Available for Unique Data (percent)	Data Redundancy
0	2 to 8	100	No
1	2	50	Yes
0 + 1	3 to 8	50	Yes
5	3 tp 8	66 to 87	Yes
JBOD	1	100	No

*NOTE: The following equation determines the amount of storage available for unique information in a RAID 5logical RAID drive: Usable Space = Total Space * ((N-1_/N where N is the number of drives in the group.*

Table C-2 How the Capacity of Each Drive Affects the Capacity of the Drive Group

Drive Group	Drives	Drive's Capacity	Drive's Usable Capacity in This Group	Drive Group's Total Capacity
A	0	1 gigabyte	1 gigabyte	
	1	2 gigabytes	1 gigabyte	
	2	3 gigabytes	1 gigabyte	3 gigabytes (due to drive 0's capacity being 1 gigabyte)
B	0	2 gigabytes	2 gigabytes	
	1	2 gigabytes	2 gigabytes	
	2	2 gigabytes	2 gigabytes	6 gigabytes

Defining a Drive Group

Use this option to bind drives together into drive groups. To maximize your disk space resources, use drives of the same capacity in a drive group. To define a drive group, follow these steps:

1. Select the *New Configuration* option from the main menu and either one of the following things happens depending upon whether a configuration currently exists. If a configuration currently exists, the utility displays a warning message saying that a valid configuration exists and if you proceed, you will destroy it. A confirmation window also appears. Go to Step 2. If no configuration currently exists, then the utility displays the *New Configuration* menu. The *New Configuration* menu appears with the *Define Drive Group* option highlighted. Go to Step 3.
2. Select Yes to delete the current configuration (select No to quit without changing the configuration and to return to the main menu). The utility displays the *New Configuration* menu. The *New Configuration* menu appears with the *Define Drive Group* option highlighted.
3. Press the Enter key to select the Define Drive Group option. The utility displays a drive matrix that shows the drives connected to the adapter and the status of each. The Create Group function appears highlighted. If you are uncertain about the size of a particular drive, select the Device Information option on the Define Drive Group screen to check the size before you create a drive group.
4. Press the Enter key to select the *Create Group* option. The cursor moves to the drive matrix.
5. Position the cursor on each drive with a status of RDY (ready) that you want to add to the drive group, one at a time, and press the Enter key. The utility adds that drive to the drive group by changing its status from RDY to OPT (optimal); and assigning a group letter and a sequence number to the drive. To maximize the I/O performance of your multichannel RAID subsystem, locate each member of a drive group on a separate SCSI channel. This allows the RAID controller concurrent access to the disk drives.
6. Press the Esc key after you add all the drives that you want for this drive group.
7. Repeat Steps 3 through 6 of this procedure to create additional drive groups. If you plan to define a hot spare, leave at least one drive with RDY status. This drive must be of equal or greater capacity to the drives you assigned to drive groups. If you assign all the physical drives to drive groups, the system automatically highlights the *Arrange Group* option. Otherwise, press the Esc key and the cursor returns to the *Group Definition* menu.

NOTE: If you select a drive for the drive group by mistake or you want to redefine a drive group, highlight the Cancel Group option and press the Enter key. The cursor appears on the first drive in a group on the matrix. Move the cursor to the drive group you want to cancel and press the Enter key. You can only delete drive groups in the reverse order in which you arranged them.

For more information on arranging a drive group, creating a logical RAID drive, defining a spare drive, and specifying the caching policy, refer to the *StorageWorks RAID Array 200 Subsystems Controller Installation and Standalone Configuration Utility User's Guide.*

C.2 ISP1020/1040 Configuration Utility

When the EEROMCFG configuration utility is selected, after the "transferring control" message, a screen such as that shown in Example C-1 is displayed. This shows the ISP1020/1040 devices on the PCI device unit number you selected (if you are running the EEROMCFG utility from the SRM console) or all ISP1020/1040 devices on your system (if you are running EEROMCFG from **AlphaBIOS Setup**). You use the arrow keys to move the brackets [ ] up and down to select a particular ISP1020/1040 device.

Example C-1 ISP1020/1040 Configuration Utility Opening Screen

```
------------------------------------------------------------
      ISP1020/1040 EEROM Configuration Utility V2.1-101
------------------------------------------------------------
          +========Select ISP1020==========+
          |Bus   Virtual Slot     Address  |
          |[0        0              p0]    |
          | 0        1              p1     |
          | 0        3              p3     |
          | 0        4              p4     |
          | 0        6              p6     |
          | 0        7              p7     |
          | 0        9              p9     |
          | 0       10              p10    |
          |--------------------------------|
          |    < (O)K >       <E(x)it >    |
          +================================+
```

Once you have chosen a specific ISP1020/1040, type “O” for OK and a screen like or similar to the one shown in Example C-2 is shown. Choose the option you wish. The H option allows you to modify host adapter parameters.

Example C-2 ISP1020/1040 Configuration Utility Second Screen

```
------------------------------------------------------------
      ISP1020/1040 EEROM Configuration Utility V2.1-101
------------------------------------------------------------

          +======== Configure EEROM Parameters ========+
          | [(L)oad Default EEROM Parameters        ]  |
          | Edit EEROM (H)ost Adapter Parameters       |
          | Edit EEROM (D)evice Adapter Parameters     |
          | Enable (U)ltrafast [FAST-20] on all devices|
          | Enable (F)AST-10 on all devices            |
          | E(x)it                                     |
          +============================================+
```

Example C-3 shows an example of host adapter parameters for a KZPBA adapter. Table C-3 describes the host adapter parameters. (You can also refer to your adapter Installation Guide for further information.) You use the arrow keys to select a parameter to modify from the default and type in the modification. When parameters are as you wish, type "O" for OK.

Example C-3 Edit Host Adapter Parameters Example

```
+========+Edit EEROM Host Adapter Parameters ============+
|                                                        |
|         Parameters      Default  New                   |
|Fifo Threshold               3   [3_   ]                |
|Host Adapter Enable          1   [1    ]                |
|Initiator SCSI ID            7   [7    ]                |
|Bus Reset Delay              1   [1    ]                |
|Retry Count                  0   [0    ]                |
|Retry Delay                  1   [1    ]                |
|Asynchronous Data Setup Time 9   [9    ]                |
|REQ/ACK Active Negation      1   [1    ]                |
|Data Line Active Negation    1   [1    ]                |
|Data DMA Burst Enable        1   [1    ]                |
|Command DMA Burst Enable     1   [1    ]                |
|Tag Aging                    8   [8    ]                |
|Low Termination Enable       1   [1    ]                |
|High Termination Enable      1   [1    ]                |
|Selection Timeout          250   [250  ]                |
|Maximum Queue Depth        256   [256  ]                |
|Single Ended or Differential 0   [0    ]                |
|                                                        |
|--------------------------------------------------------|
|     |     < (O)K>        < (C)ANCEL >                  |
+========================================================+
```

Table C-3 Host Adapter Parameters

Parameter	Default	Description	Reason to Change
FIFO Threshold	2 (KFTIA) 3 (KZPBA)	Sets the FIFO threshold point at which burst transfers are requested on the ISP1020/1040 adapter.	Optimize system performance. [1]
Host Adapter Enable	1	Determines whether the BIOS recognizes the ISP1020/1040 host adapter.	Disable the host adapter without physically removing it from the system.
Initiator SCSI ID	7	Sets the SCSI ID for the ISP1020/1040.	If a SCSI bus has multiple SCSI adapters.
Bus Reset Delay	1	Sets the delay (after resetting the SCSI bus) before the firmware initiates any SCSI activity.	Optimize system performance. [1]
Retry Count	0	Determines the number of times the firmware attempts to retry a selection timeout or a busy status.	Optimize system performance. [1]
Retry Delay	1	Sets the time (in 100 microsecond increments) that the firmware waits before reattempting an operation.	Optimize system performance. [1]
Asynchronous Data Setup Time	9	Sets the number of clock periods the ISP1020/1040 host adapter card waits after driving the SCSI data signals before asserting a SCSI bus acknowledge signal.	Optimize system performance. [1]

Table C-3 Host Adapter Parameters (Continued)

Parameter	Default	Description	Reason to Change
REQ/ACK Active Negation	1	Provide active pullup assist in single-ended mode. (the REQ and ACK signals are pulled up.) By enabling active negation, the ISP1020/1040 host adapter is less sensitive to an imperfect SCSI bus.	
Data Line Active Negation	1	Provide active pullup assist in single-ended mode. The SD17-0 and SDP1-0 signals are pulled up.) By enable active negation, the ISP1020 host adapter is less sensitive to an imperfect SCSI bus.	
Data DMA Burst Enable	1	When set to 1, performs bust transfers on the data DMA channel. When set to 0, data is transferred in nonburst mode with each cycle initiated by a new address phase.	Optimize system performance. [1]
Command DMA Bust Enable	1	When set to 1, performs burst transfers on the data DMA channel. When set to 0, data is transferred in nonburst mode with each cycle initiated by a new address phase.	Optimize system performance. [1]
Tag Aging	8	Ensures tagged commands are not lost in the target device. Tag aging is a backup to the timeout mechanism.	Optimize system performance. [1]

Table C-3 Host Adapter Parameters (Continued)

Parameter	Default	Description	Reason to Change
Low Term Enable	1		If termination not required on the board.
High Term Enable	1		If termination not required on high-order 8 bits, or not required on the board
Selection Timeout	250	Sets the selection phase timeout value (in micro-seconds)	Optimize system performance.[1]
Maximum Queue Depth	256	Specifies the maximum number of outstanding commands issued to each SCSI target. When the number is reached, new commands are returned with Queue Full Status.	Optimize system performance.[1]
Single-Ended or Differential	0		Not applicable for the ISP1020.

[1] The default setting provides the best performance optimization for most system configurations. We recommend that you change this default setting only if it does not provide optimal performance for your system configuration.

Glossary

AC input box

Receives single or three-phase AC power and outputs that to the power regulators. The system circuit breaker and a Dranetz port are on the AC input box. See also *Power.*

Address space

See *Physical address space* and *Virtual address space.*

AlphaBIOS

The firmware subsystem that provides facilities for setup and maintenance of Windows NT systems.

Alpha primary boot program

The Alpha primary boot program (APB.EXE) that boots OpenVMS. APB is the primary bootstrap program and is stored on the boot device.

Asymmetric multiprocessing

A multiprocessing configuration in which the processors are not equal in their ability to execute operating system code. In general, a single processor is designated as the primary, or master, processor; other processors are the slaves. The slave processors are limited to performing certain tasks, whereas the master processor can perform all system tasks. Contrast with *Symmetric multiprocessing.*

Bandwidth

The data transfer rate measured in information units transferred per unit of time (for example, Mbytes per second).

Battery backup

Provides power to the entire system cabinet (or to an expander cabinet) in the event of a power failure. An AlphaServer 8400 system with three-phase power can maintain power for approximately 11 minutes and an AlphaServer 8200 system for 3 to 4 minutes.

Boot device

A storage device that holds the software that carries out the system bootstrap process.

Boot processor

The CPU module that boots the operating system and communicates with the console; also known as the primary processor.

Bootblock

Block zero on the system disk; it contains the block number where the Alpha primary boot (APB), DIGITAL UNIX boot, NT boot, or virtual memory boot (VMB) program is located on the system disk and contains a program that, with the boot primitive, reads APB or VMB from the system load device into memory.

CD-ROM

Compact disk read-only memory. The SRM console, LFU utility, and AlphaBIOS firmware are distributed on CD-ROM. The SRM console provides commands to boot OpenVMS and DIGITAL UNIX systems, test the system and its I/O devices, display the configuration, and other features. The LFU utility is used in upgrading firmware and copies of the firmware for all modules in the system, including the console and diagnostic firmware. The AlphaBIOS firmware provides a graphic interface for the setup, maintenance, and booting of Windows NT systems.

Centerplane

The TLSB backplane (8400 only), located in the center of the card cage, which physically shortens the bus thus increasing bandwidth. Since the 8200 system uses only the "back half" of the card cage, there is no centerplane.

CCL module

Module (at the top right of the 8400 system cabinet and at the upper left in the 8200 cabinet) that provides the control panel interface and power sequencing.

CIPCA

PCI CI port interface; connects a system to a Star Coupler.

CIXCD

XMI CI port interface; connects a system to a Star Coupler.

Compact disk server

In-cabinet CD server that provides access to CD-ROMs for software installation, diagnostics, and on-line documentation.

Console language

Refers to the SRM console firmware. The language used by the system operator at the SRM console terminal to communicate with the primary processor; provides the interface to diagnostics. The console language uses options, environment variables, and arguments. *Options* modify the action of the command in some way, or give details of how the command is to operate; they appear in the form -xxx and are preceded by a space. *Environment variables* determine the environment; some are set in manufacturing and set up a default environment. Most environment variables are defined with the set command. See also *Environment variables.*

Console mode

A mode of operation where the processor is not running the operating system but allows an operator to issue SRM console commands.

Console program

The code that the boot processor executes during console mode. Each processor has a copy of this code in flash ROMs. After a boot processor has been determined, that processor begins console initialization. The console code is then loaded into memory.

CPU module

The KN7CC, KN7CD, KN7CE, or KN7CF processors are the CPU modules used in 8200/8400 systems. They cannot be intermixed.

DECchip 21164

The Alpha processor chip used in the AlphaServer 8200/8400 systems.

DIGITAL UNIX operating system

A general-purpose operating system based on Open Software Foundation OSF/1 technology. DIGITAL UNIX runs on the range of Alpha systems, from workstations to servers.

DEFAA

Futurebus+ adapter to the FDDI (Fiber Distributed Data Interface).

DEFPA

PCI adapter to the FDDI (Fiber Distributed Data Interface).

Device

From the SRM console perspective, “device” can be used to refer to a physical device, a block of memory, or a set of registers. The console commands operate on byte streams, so any of these can be expressed as a byte stream.

Each physical device in a system has a unique mnemonic assigned by the console program, which is displayed as part of the SRM console commands **show configuration, show device, shot network,** and **test**. Devices are named differently by the AlphaBIOS Setup firmware, and again differently by the Windows NT operating system.

DEMFA

XMI adapter to the FDDI (Fiber Distributed Data Interface).

DEMNA

XMI adapter; Ethernet port interface.

DE435

PCI adapter; Ethernet port interface.

Disk array

A set of disk drives and a specialized aarray controller, which keeps track of how data is distributed across the drives.

Drive group

A set of drives logically tied together and addressed as a single unit.

DSSI

DIGITAL Storage Systems Interconnect. A DIGITAL Storage Architecture interconnect used by the KFMSB adapter and RF and TF series integrated storage elements to transfer data and to communicate with each other.

DSSI PIU

Houses DSSI based disks inside the system and expander cabinets (BA654).

DWLMA

An XMI adapter that is the interface between the TLSB bus and the XMI bus; always node 8 of the XMI. The XMI plug-in unit includes the T2028-AA module and necessary cables and the XMI clock card. See also *XMI PIU* and *XMI clock.*

DWLPA/DWLPB

A PCI adapter, housed in a unit that provides a PCI card cage, that is the interface with the TLSB. Note that the KFE72 adapter, used with systems using a graphics console monitor, is only supported by the DWLPB.

ECU

EISA (system) configuration utility used to configure EISA option boards.

EISA bus

Extended Industry Standard Architecture bus. A 32-bit industry standard I/O bus used primarily in high-end PCs and servers.

Environment variables

UNIX-like options used with SRM console commands. An environment variable consists of an identifier (ID) and a byte stream value maintained by the console. See also *Console language.*

Expander cabinet

A cabinet to hold additional I/O adapters or storage units that can be installed on either side of the system cabinet.

FDDI

Fiber Distributed Data Interface. A high-performance fiber optic network that can be accessed by the XMI DEMFA, PCI DE435, FBUS+ DEFAA, or EISA DEFAA adapters.

Filler module

Required to fill unused TLSB slots when fewer than six CPU, memory, or I/O modules are installed.

Flash ROM

Flash-erasable programmable read-only memory, which can be bulk erased and reprogrammed. The processor module uses flash ROMs to hold the console and diagnostic firmware. In addition, one flash ROM holds initialization code that bootstraps the main console/diagnostic firmware. See also *SROM code.*

Gbus

The path between the processor and the console/diagnostic firmware and to two UART chips. The Gbus has two lines, one to the serial console terminal and one to the power supply.

Hardware restart parameter block (HWRPB)

A page-aligned data structure shared between the console and system software; a critical resource during bootstraps, recovery from power failures, and other restart situations.

Hose

An interconnect between the KFTHA or KFTIA module on the TLSB bus and the interface module on another bus, such as a PCI, XMI, or Futurebus+ for AlphaServer 8400 systems running OpenVMS or DIGITAL UNIX systems, or a PCI bus for AlphaServer 8200 systems or 8400 systems running Windows NT. The KFTHA module contains four hoses. The KFTIA module contains one hose.

KFE70

An adapter providing support for a serial console terminal and to the console load devices.

KFE72

An adapter providing an I/O interface to the console load devices as well as support for a graphics monitor. It also provides the serial and parallel ports expected by the Windows NT operating system.

KFESA

EISA device; SHAC-based DSSI controller.

KFMSB

XMI adapter for RF disks and TF tapes used to enable connection to nodes on a DSSI bus. Each KFMSB adapter supports two DSSI buses.

KFTHA module

The TLSB module that provides the interface from the TLSB bus to I/O buses. The KFTHA module has four ports to support up to four I/O channels. Each channel is known as a "hose." A system can support up to three KFTHAs with the first one in node 8 of the TLSB. Node 8 is dedicated as both the highest and lowest arbitration level; the KFTHA usually arbitrates at the highest priority.

KFTIA module

The TLSB module that provides the interface from the TLSB bus to I/O buses. The KFTIA module has one port, known as a "hose," to provide access to a single PCI, XMI (8400 only), or Futurebus+ (8400 only) card cage. A system can support up to three KFTIAs with the first one in node 8 of the TLSB.

KN7CC CPU module

The 300 MHz CPU module that uses one or two DECchip 21164 chips with a superscalar pipelined design. Each CPU chip has two 8-Kbyte caches and a 4-Mbyte backup cache implemented in RAMs. The KN7CC processor supports writeback caching.

KN7CD CPU module

The 350 MHz CPU module that uses one or two DECchip 21164 chips with a superscalar pipelined design. Each CPU chip has two 8-Kbyte caches and a 4-Mbyte backup cache implemented in RAMs. The KN7CD processor supports writeback caching.

KN7CE CPU module

The 440 MHz CPU module that uses two DECchip 21164 chips with a superscalar pipelined design. Each CPU chip has two 8-Kbyte caches and a 4-Mbyte backup cache implemented in RAMs. The KN7CE processor supports writeback caching.

KN7CF CPU module

The 625 MHz CPU module that uses two DECchip 21164 chips with a superscalar pipelined design. Each CPU chip has two 8-Kbyte caches and a 4-Mbyte backup cache implemented in RAMs. The KN7CF processor supports writeback caching.

KZASA

Futurebus+ adapter; FWD SCSI adapter.

KZMSA

XMI adapter to the SCSI (Small Computer System Interface), enabling connections to nodes on a SCSI bus.

KZPSA

PCI adapter; fast wide SCSI controller.

ISE (integrated storage elment)

All DSSI storage devices, such as RF disks and TF tapes, are ISEs.

LCD (local disk converter)

Converts 48 VDC to +5V and +12V needed by the storage arrays in the DSSI PIU (BA654).

LFU (Loadable Firmware Update) Utility

Used to update firmware in EEPROM on modules. Firmware can include the SRM console, AlphaBIOS, and firmware on TLSB and I/O device modules.

Mailbox

A software-created data structure in memory used to read and write to I/O device registers on XMI and FBUS+ controllers.

Memory

Systems use the MS7CC memory, available with 128, 256, and 512 Mbytes and 1, 2, and 4 Gbytes of memory. The AlphaServer 8400 supports up to 28 Gbytes of memory, and the 8200 supports up to 12 Gbytes of memory.

MEMORY CHANNEL

A bus that, under the OpenVMS and DIGITAL UNIX operating systems, permits data written to one computer's memory to be shared by other computers on the bus.

Memory interleaving

Method to optimize memory access time; the SRM console program automatically interleaves the memories in the system unless a command is used to set a specific interleave or no interleave (which would result in serial access to each memory module). Interleaving allows a number of memories to operate in parallel.

More protocol

A protocol used on the XMI bus that allows XMI nodes to make noninterlocked memory reads and writes. Using the More protocol increases I/O performance.

Node, TLSB

AlphaServer 8400 TLSB nodes are numbered from 0 to 8 and correspond to specific slots in the TLSB card cage. The AlphaServer 8200 TLSB has five nodes or slots, numbered from 4 to 8.

NVRAM option

PCI controller containing nonvolatile memory that supports Prestoserve applications running under DIGITAL UNIX.

OpenVMS operating system

The OpenVMS operating system is a general-purpose multi-user operating system that supports industry standards and runs on Alpha machines.

PCI (Peripheral Component Interconnect) shelf

The unit containing the PCI adapter module, PCI card cage, and power.

Physical address space

The 1 terabyte of physical address space that the TLSB bus can access using a 40-bit memory address space.

Plug-in units (PIUs)

Self-contained assemblies that are easily installed in the 8400 system cabinet or expander cabinet. There are PIUs for the XMI bus, the Futurebus+ bus, PCI/EISA buses, as well as for StorageWorks disks and batteries.

Power regulators

The AlphaServer 8400 system can have H7264 single-phase power regulators, H7263-AA/AB three-phase power regulators, or H7263-AC/AD three-phase power regulators, which allow battery backup. The H7263 is required to support N+1 redundancy. Plug-in units are powered by the system, but in addition, bus PIUs have their own power regulators. Batteries supply power to the 8400 system cabinet for approximately 11 minutes (3 to 4 minutes for the AlphaServer 8200 system). Expander cabinets have their own supply.

RAID (Redundant Array of Independent Disks)

A set of storage techniques devised to increase the availability and performance of a storage subsystem.

Reset sequence

A process leading to the execution of a copy of the console firmware from memory. Purpose is to test the machine, establish the console environment, indicate that the console is executing, and dispatch to entry handling procedures. Power-up testing executes from the backup cache, a boot processor is determined, which creates the HWRPB and copies the console image from ROM/cache to memory. Registers are set to their default values, and internal console data structures are initialized. After power-up the secondary processors also use the console code from memory.

SCSI (Small Computer System Interface)

An industry-standard interface for connecting disks and other peripheral devices to computer systems.

SCSI PIU

Houses SCSI based disks and tapes inside 8400 system and expander cabinets (BA655).

SRM (serial ROM) code

The initialization code that bootstraps the main console/diagnostic firmware.

StorageWorks

DIGITAL's family of module data storage products that allows customers to design and configure their own storage subsystems. Components include power, packaging, cabling, devices, controllers, and software. Devices and array controllers can be integrated in StorageWorks enclosures to form storage subsystems.

Symmetric multiprocessing

A multiprocessing system configuration in which all processors have equal access to operating system code residing in shared memory and can perform all, or almost all, system tasks.

Transactions

TLSB transactions consist of a command and two data cycles, which follow some fixed time after an arbitration cycle. All TLSB modules monitor the request lines to see which of them wins the bus. Up to three transactions can be in progress at one time. Because arbitration occurs on a dedicated set of control lines, it can overlap with data transfer.

TLSB bus

The 256-bit data and 40-bit address buses comprising the TLSB. The TLSB bus is implemented in a 9-slot card cage in the 8400 system with its "backplane" in the center of the cage, so that modules are installed from the front and the back. Module placement is indicated by a node number 0 through 8. Node 8 is reserved for the TLSB I/O port (KFTHA or KFTIA) module. The AlphaServer 8200 system has a 5-slot card cage (numbered 4 through 8) with node 8 reserved for an I/O module.

Virtual address space

Memory space available to a user program. The operating system dynamically maps a given virtual address to a physical address.

Windows NT

A general-purpose operating system requiring a graphic interface for users.

XMI clock

Implemented on the T2030-YA module which must be node 7 of the XMI.

XMI PIU

A plug-in unit consisting of an XMI card cage and two power regulators which occupy two quadrants. Each XMI requires a DWLMA module in slot 8 and a clock module at slot 7. The first I/O option must be installed in slot 1 or 14 to terminate the XMI bus.

XMI-to-TLSB interface

On the XMI the interface is the DWLMA module which must be node 8 of the XMI. On the TLSB bus the interface is the KFTHA or KFTIA module, which must be node 8 of the TLSB if there is only one system I/O module. The connection between the two modules is known as a "hose." See also KFTHA module and KFTIA module.

Index

A

AC input box, 3-13
AC power cord, 3-9
AC power indicators, 3-27
Advanced CMOS setup screen, AlphaBIOS, 8-36
AlphaBIOS
  Auto Start Count option, 5-20
  Auto Start Option, 5-20
  Password option, 5-22
AlphaBIOS command, 7-14
AlphaBIOS firmware, 1-5, 8-1
  boot screen, 8-10
  displaying system configuration, 8-12
  finding help, 8-8
  first-level help screen, 8-8
  key conventions, 8-8
  second-level help screen, 8-9
  setting date and time, 8-35
  setting keyboard language, 8-35
  setting password, 8-37
  setup screen, 8-11
  starting, 8-10
  switching to SRM, 8-4
Arc command, 7-14
ARC utility program, 7-38
Arc_enable environment variable, 7-10, 7-40
Architecture, system, 1-6
Attributes, setting device, B-18
Auto_action environment variable, 5-4, 7-10
  effect on booting Windows NT, 5-19

B

Battery backup, 2-17, 2-23, 3-13
Battery packs, 2-17, 3-13
Battery PIU, 3-5, 3-7, 3-9, 3-21
Baud rate, 7-13
Blower, 3-9
Blower (8200), 2-11
Blower (8400), 3-18
Boot command, 7-15
Boot device, 5-6
  default with factory-installed software, 5-7
  setting default, 5-7
Boot processor, 1-7, 6-25, 7-23
Boot screen, AlphaBIOS, 8-10
Boot_file environment variable, 7-10
Boot_osflags environment variable, 7-10
Boot_reset environment variable, 7-11
Bootdef_dev environment variable, 5-7, 5-11, 7-10
Booting
  auto_action environment variable, 5-4
  boot device, 5-6
  boot path for Windows NT, 5-12
  console environment variable, 5-3
  device name for OpenVMS and DIGITAL UNIX, 5-8
  DIGITAL UNIX, 5-16
  OpenVMS, 5-14
  os_type environment variable, 5-2
  preparation, 5-2
  Windows NT, 5-18
Booting DIGITAL UNIX, 5-1
Booting LFU, B-2
Booting OpenVMS, 5-1
Booting Windows NT, 5-1
Build -e command, 7-16
Build -n command, 7-17
Build -s command, 7-18

C

Cabinet control logic panel (CCL), 2-12
CCL module, 3-13, 3-23
CD-ROM drive, 2-15, 3-7
Channel lines, 6-26
CI disk, 5-7
Circuit breaker, 2-17, 3-9, 3-27
  lockout, 3-27
  main input, 3-13
  trip indicator, 3-13, 3-27
Clear <envar> command, 7-20
Clear EEPROM command, 7-19
Clear Screen command, 7-21
CMOS setup screen, AlphaBIOS, 8-34, 8-36
Comment (#) command, 7-65
Configuring parameters, AlphaBIOS firmware, 8-34
Console
  switching from AlphaBIOS to SRM, 8-4
Console CD-ROM, B-2
Console device, 2-5, 3-5, 3-16
Console environment variable, 5-3, 7-11
Console load device, 2-5, 3-5, 3-10
Continue command, 7-3, 7-22
Control panel, 2-18, 3-7, 3-16, 3-22
Cooling system, 3-7
  airflow, 3-19
  blower, 3-19
CPU environment variable, 7-11
Cpu_enabled environment variable, 7-11
Cpu_primary environment variable, 7-11
Crash command, 7-24
Create command, 7-25
Ctrl/P, 7-22

D

D_harderr environment variable, 7-11
D_report environment variable, 7-11
D_softerr environment variable, 7-11
D_trace environment variable, 7-11
Date command, 7-26
Date, setting in AlphaBIOS firmware, 8-35
DC distribution box, 3-9, 3-13
Delete key, 7-7
Deposit command, 7-27
Device name format
  SRM console, 5-9
DIGITAL UNIX
  booting, 5-1, 5-16
Display command, LFU, B-12
Dump_dev environment variable, 7-11
DWLPA/DWLPB adapter, 4-16

E

EEROMCFG, 1-5, 7-40, C-1
EISA configuration screen, AlphaBIOS, 8-21
EISA Configuration Utility (ECU), 1-5, 4-20, 7-41, 7-42
EISA support, 4-19, 4-21
Electrical characteristics
  8200, 2-3
  8400, 3-3
Enable_audit environment variable, 7-12
Environment variable, 7-9
  arc_enable, 7-10
  auto_action, 7-10
  boot_file, 7-10
  boot_osflags, 7-10
  boot_reset, 7-11
  bootdef_dev, 7-10
  console, 7-11
  cpu, 7-11
  cpu_enabled, 7-11
  cpu_primary, 7-11
  d_harderr, 7-11
  d_report, 7-11
  d_softerr, 7-11
  d_trace, 7-11
  dump_dev, 7-11
  enable_audit, 7-12

interleave, 7-13
os_type, 7-13
removal, 7-20
simm_callout, 7-13
tta0_baud, 7-13
Environment variables
affecting test command, 6-39
ew* (network), 7-12
Environmental characteristics
8200, 2-3
8400, 3-3
Error reports
full, 6-50
summary, 6-50
Examine command, 7-31
Exit command, LFU, B-10
Expander cabinet, 2-5, 3-4
power connections, 3-17

F

Factory-installed software, 5-2
Fault light, 3-25, 6-17
Filler modules, 3-19
Finding help
AlphaBIOS firmware, 8-8
Firmware
AlphaBIOS, 8-1
corrupted, B-14
updating, B-1
updating from AlphaBIOS, 8-26
Firmware overview, 1-4
Floating-point register set, 7-29, 7-32
Floppy drive, 3-7
Formatting hard disk, AlphaBIOS, 8-28, 8-30, 8-32
Futurebus+ PIU, 3-21

G

General register set, 7-29, 7-32

H

Halt command, 7-33
Hard disk configuration screen, AlphaBIOS, 8-16
Hard disk, AlphaBIOS
creating and deleting partitions, 8-30
formatting a FAT partition, 8-32
setting up, 8-28
Help
AlphaBIOS firmware, 8-8
AlphaBIOS first-level help screen, 8-8
AlphaBIOS second-level help screen, 8-9
Help command, 7-34
Helpkey, 7-35
Hose numbering scheme, 4-7

I

I/O bulkhead, 3-9
I/O bulkhead (XMI), 3-17
I/O connections, 3-16
I/O PIU, 3-5, 3-7
I/O port module, 1-7, 4-2
I/O subsystems, 4-1
Initialize command, 6-9, 7-36
Insert mode, 7-7
Installing Windows NT, 8-38
Instruction decode, 7-32
Integrated peripherals screen, AlphaBIOS, 8-25
Interleave environment variable, 7-13
Internal processor registers, 7-29, 7-32
IOP module, 3-9

K

Key conventions
AlphaBIOS firmware, 8-8
Key On light, 3-25
Keyboard
setting language (AlphaBIOS), 8-35
KFE70 adapter, 4-17, 4-18, 4-19
KFE72 adapter, 4-17, 4-18, 4-19
KFTHA, 1-7
KFTHA module, 4-5
KFTIA, 1-7
KFTIA module, 2-11, 4-5
KZPSA configuration utility, 7-40

L

LEDs
  self-test, 6-17
LFU, B-1
  booting, B-2
  display command, B-12
  exit command, B-10
  list command, B-4
  modify command, B-18
  modifying device attributes, B-18
  update command, B-6
  updating corrupted firmware, B-14
  verify command, B-12
List command, LFU, B-4
Loadable Firmware Update (LFU) utility, 1-5, B-1

M

Maintenance programs, AlphaBIOS, 8-40
Memory
  interleaving, 6-29
  size, 6-29
Memory configuration screen, AlphaBIOS, 8-24
Memory dump, 7-24
Memory node, 1-7
Modify command, LFU, B-18
Multiprocessing, 1-7

N

Network environment variables, 7-12
Nonvolatile RAM (NVR), 7-17
Null command, 7-5

O

OpenVMS
  booting, 5-1, 5-14
Operating system selection setup screen, AlphaBIOS, 8-46
Operating system selection, defined, 8-43
Operating system selections screen, AlphaBIOS, 8-42
Options, 1-3, 3-20
Os_type environment variable, 5-2, 7-13
Overstrike mode, 7-7

P

PAL temp register set, 7-29, 7-32
Parameters, configuring in AlphaBIOS firmware, 8-34
Password, setting in AlphaBIOS firmware, 8-37
PCI
  finding device information, AlphaBIOS, 8-20
PCI adapter, 2-23, 4-16
PCI configuration
  no KFE70 or KFE72, 4-17
  with KFE70, 4-17
  with KFE72, 4-17
PCI configuration screen, AlphaBIOS, 8-18
PCI PIU, 3-21
PCI shelves, 2-5, 2-23
PCI/EISA slot configuration
  with KFE70 adapter, 4-16
Physical memory, 2-1
Plug-in units (PIUs), 3-6, 3-17
Power regulators, 3-6, 3-21
Power system, 3-7
Powerstorm 3D30 graphics module, 4-19
Power-up, 6-2
Prcache command, 7-37
Primary operating system, AlphaBIOS, 8-44
Processor node, 1-7
Processor system unit (PSU), 1-3, 2-10
Program counter, 7-22, 7-23

R

RAID Configuration Utility (RCU), 1-5, 7-38, 7-40, C-1
Recall buffer, 7-7
Run light, 3-25
Runecu command, 7-41

Running maintenance programs, AlphaBIOS, 8-40
Running utility programs, 8-40

S

SCSI configuration screen, AlphaBIOS, 8-22
SCSI configuration utilities, 1-5
Secondary processor, 1-7
Self-test
  explanation of sample configuration, 6-21
  firmware version, 6-30
  line
    ILV, 6-28
    Mb, 6-28
    NODE#, 6-22
    TYP, 6-22
  lines
    BPD, 6-24
    C0, C1, C2, and C3, 6-26
    ST1, ST2, and ST3, 6-24
  module types, 6-23
  node numbers, 6-23
  order of testing, 6-19
  SROM version, 6-30
Serial console terminal
  baud rate, 7-13
Set <envar> command, 7-44
Set EEPROM command, 7-43
Set host command, 7-45
Set power command, 7-47
Set seeprom command, 7-48
Setup screen, AlphaBIOS, 8-11
Show <envar> command, 7-54
Show config command, 5-8
Show configuration command, 6-32, 7-49
Show CPU command, 7-51
Show device command, 5-8, 6-36, 7-52
Show EEPROM command, 7-53
Show memory command, 6-29, 7-55
Show network command, 6-34, 7-56
Show power command, 6-15, 7-57
Show SEEPROM command, 7-58
Simm_callout environment variable, 7-13
SRM console
  switching from AlphaBIOS, 8-4
SRM console command
  overview, 7-3
  syntax, 7-4
SRM console firmware, 1-5
SRM console prompt, 7-3
Standard CMOS setup screen, AlphaBIOS, 8-34
Start command, 7-59
Starting AlphaBIOS, 8-10
Status indicator lights, 3-24
Stop command, 7-60
StorageWorks shelves, 2-5, 2-23
SWXCRFW, 1-5, 7-40, C-1
System
  footprint, 3-2
System architecture, 1-6
System board configuration screen, AlphaBIOS, 8-14
System configuration
  displaying in AlphaBIOS firmware, 8-12
System overview, 1-1

T

Test command, 6-38, 7-61
  environment variables, 6-39
  I/O adapter, 6-44
  I/O devices, 6-46
  I/O subsystem, 6-42
  memory, 6-45
  network adapters, 6-48
  options, 6-39
  whole system (except memory), 6-40
Time, setting in AlphaBIOS firmware, 8-35
TLSB bus, 1-7, 2-1, 3-14
TLSB card cage, 3-5, 3-7, 3-9
Troubleshooting, 6-1
Tta0_baud environment variable, 7-13

U

Uniprocessor
    in self-test display, 6-25
Update command, LFU, B-6
Updating firmware
    AlphaBIOS, 8-26
UTIL_CLI, 1-5, 7-40, C-1
Utilities overview, 1-4
Utility programs, 8-40

V

Verify command, LFU, B-12
Version selection, Windows NT, 8-42

W

Web (WWW)
    AlphaServer options, 1-3
Wildcarding, 7-18, 7-44
Windows NT
    auto start, 8-46
    booting, 5-1, 5-18
    designating primary operating system, 8-44
    installing, 8-38
    selecting version, 8-42
    starting, 8-46
    starting automatically, 8-35

X

XMI PIU, 3-5, 3-21