AlphaServer 8200/8400 Operations Manual

Order Number EK–T8030–OP. A01

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

First Printing, May 1995

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.

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 computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense may be required to take measures to correct the interference.

Contents

Chapter 4 I/O Subsystems

Chapter 5 Booting

Chapter 6 System Troubleshooting

Chapter 7 Console Commands

Appendix A Boot Options

Appendix B Updating Firmware

Appendix C Configuration Utilities

Glossary

Examples

Figures

Tables

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.

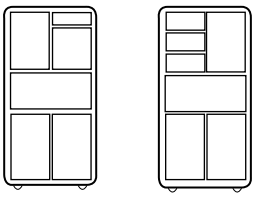
Document Structure

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

- **Chapter 1, Introduction,** provides a brief overview of the AlphaServer 8200 and AlphaServer 8400 systems.
- **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,** explains how you turn on the system and get it running.
- **Chapter 6, System Troubleshooting,** provides basic troubleshooting procedures.
- **Chapter 7, Console Commands,** lists the console commands with an example of each command.
- **Appendix A, Boot Options,** lists the options used with the boot command to control various phases of booting.
- **Appendix B, Updating Firmware,** explains how to run the Loadable Firmware Update (LFU) Utility.
- **Appendix C, Configuration Utilities,** explains how to run the configuration utilities required when installing some options.
- A **Glossary** and **Index** provide additional reference support.

Conventions Used in This Document

Icons. Icons similar to those shown below are used in illustrations for designating part placement in the system described. A shaded area in the icon shows the location of the component or part being discussed.



Documentation Titles

Table 1 lists the books in the AlphaServer 8200 and 8400 documentation set. Table 2 lists other documents that you may find useful.

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
Service Information Kit	QZ–00RAC–GC
Service Manual (hard copy)	EK–T8030–SV
Service Manual (diskette)	AK–QKNFA–CA

Table 1 AlphaServer 8200 and 8400 Documentation (Continued)

Title	Order Number
Reference Manuals	
System Technical Manual	EK–T8030–TM
DWLPA PCI Adapter Technical Manual	EK–DWLPA–TM
Upgrade Manuals for Both Systems	
KN7CC CPU Module Installation Card	EK–KN7CC–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: 8200 System Only	
DWLPA PCI Shelf Installation Guide	EK–DWL82–IN
H7266 Power Regulator Installation Card	EK–H7266–IN
H7267 Battery Backup Installation Card	EK–H7267–IN
Upgrade Manuals: 8400 System Only	
AlphaServer 8400 Upgrade Manual	EK–T8430–IN
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 PCI PIU Installation Guide	EK–DWL84–IN
H7237 Battery PIU Installation Guide	EK–H7237–IN
H7263 Power Regulator Installation Card	EK–H7263–IN
H9F00 Power Upgrade Manual	EK–H8463–UI
KFMSB Adapter Installation Guide	EK–KFMSB–IN
KZMSA Adapter Installation Guide	EK–KXMSX–IN
RRDCD Installation Guide	EK–RRDRX–IN

Table 2 Related Documents

Title	Order Number
General Site Preparation	
Site Environmental Preparation Guide	EK–CSEPG-MA
System I/O Options	
BA350 Modular Storage Shelf Subsystem Configuration Guide	EK–BA350–CG
BA350 Modular Storage Shelf Subsystem User's Guide	EK–BA350–UG
BA350-LA Modular Storage Shelf User's Guide	EK–350LA–UG
CIXCD Interface User Guide	EK–CIXCD–UG
DEC FDDIcontroller 400 Installation/Problem Solving	EK–DEMFA–IP
DEC FDDIcontroller/Futurebus+ Installation Guide	EK–DEFAA–IN
DEC FDDIcontroller/PCI User Information	EK–DEFPA–IN
DEC LANcontroller 400 Installation Guide	EK–DEMNA–IN
DSSI VAXcluster Installation/Troubleshooting Manual	EK–410AA–MG
EtherWORKS Turbo PCI User Information	EK–DE435–OM
KZPSA PCI to SCSI User's Guide	EK–KZPSA–UG
RF Series Integrated Storage Element User Guide	EK–RF72D–UG
StorageWorks RAID Array 200 Subsystem Family: Installation and Configuration Guide	EK–SWRA2–IG
StorageWorks RAID Array 200 Subsystem Family Software User's Guide for OpenVMS Alpha	AA–Q6WVA–TE
StorageWorks RAID Array 200 Subsystem Family Software User's Guide for Digital UNIX	AA–Q6TGA–TE
Operating System Manuals	
Alpha Architecture Reference Manual	EY–L520E–DP
DEC OSF/1 Guide to System Administration	AA–PJU7A–TE
Guide to Installing DEC OSF/1	AA–PS2DE–TE
OpenVMS Alpha Version 6.2 Upgrade and Installation Manual	AA–PV6XC–TE

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 12 high-performance Alpha CPUs, and many other features normally associated with mainframe systems.

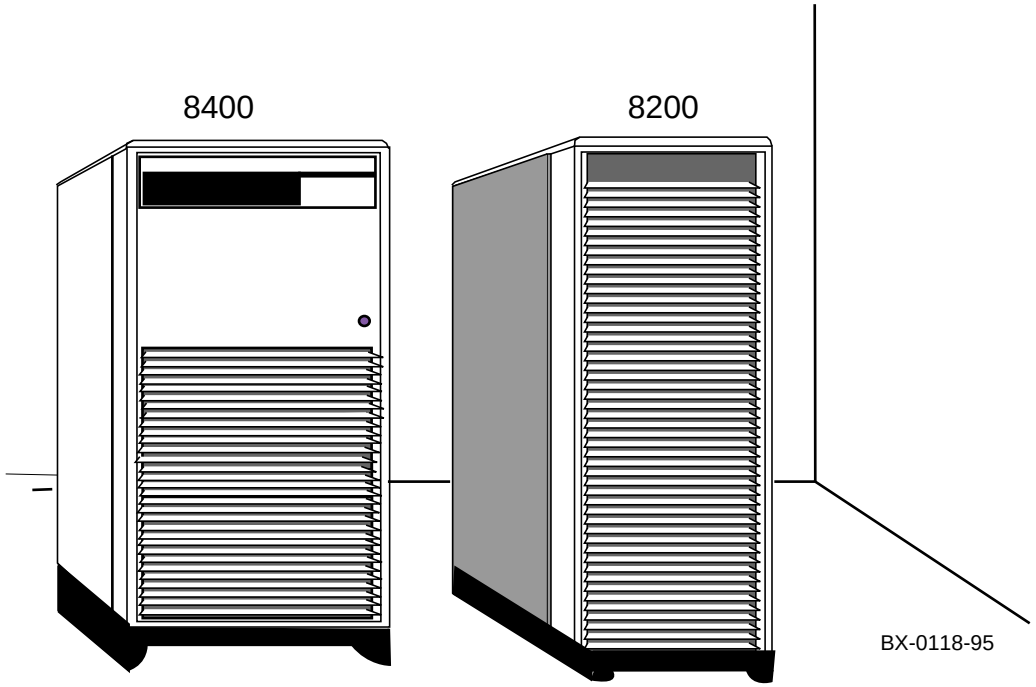
This chapter introduces the AlphaServer 8200 and AlphaServer 8400 systems. There are two sections:

- AlphaServer 8200/8400 System 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. Chapter 2 covers the AlphaServer 8200 system.

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 is discussed in Chapter 5, basic troubleshooting in Chapter 6, and console commands in Chapter 7.

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

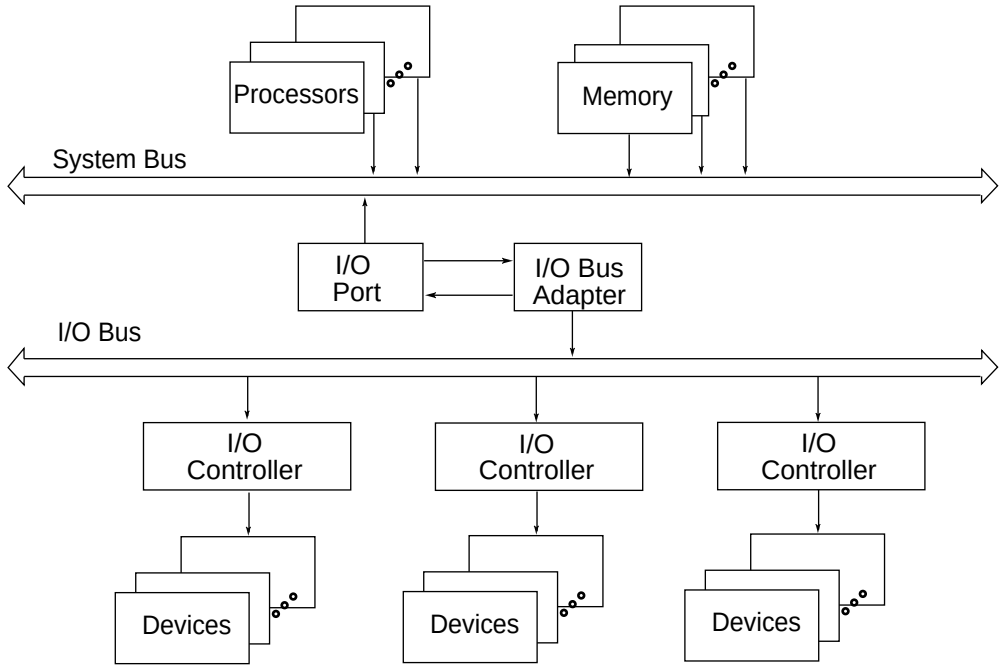
Using a Worldwide Web browser (such as Mosaic or Netscape), follow links from URL:

http://www.service.digital.com/alpha/server/8400.html

1.2 System Architecture

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

Figure 1-2 Sample System Architecture



BX0501-94

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. Multiple I/O port modules are supported.

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 (KN7CC), the TLSB bus interface, 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 six processor modules for a total of 12 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 or 2 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 optional I/O subsystem. In an 8400, the KFTHA provides connections for up to four optional PCI, Futurebus+, or XMI buses, using a cable called a hose. The KFTIA provides a connection to one I/O subsystem (a PCI in an 8200).

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

The PCI I/O bus adapter module connects to various interconnects such as SCSI, FDDI, Ethernet, NVRAM, and 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 Alpha and Digital UNIX 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 6 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

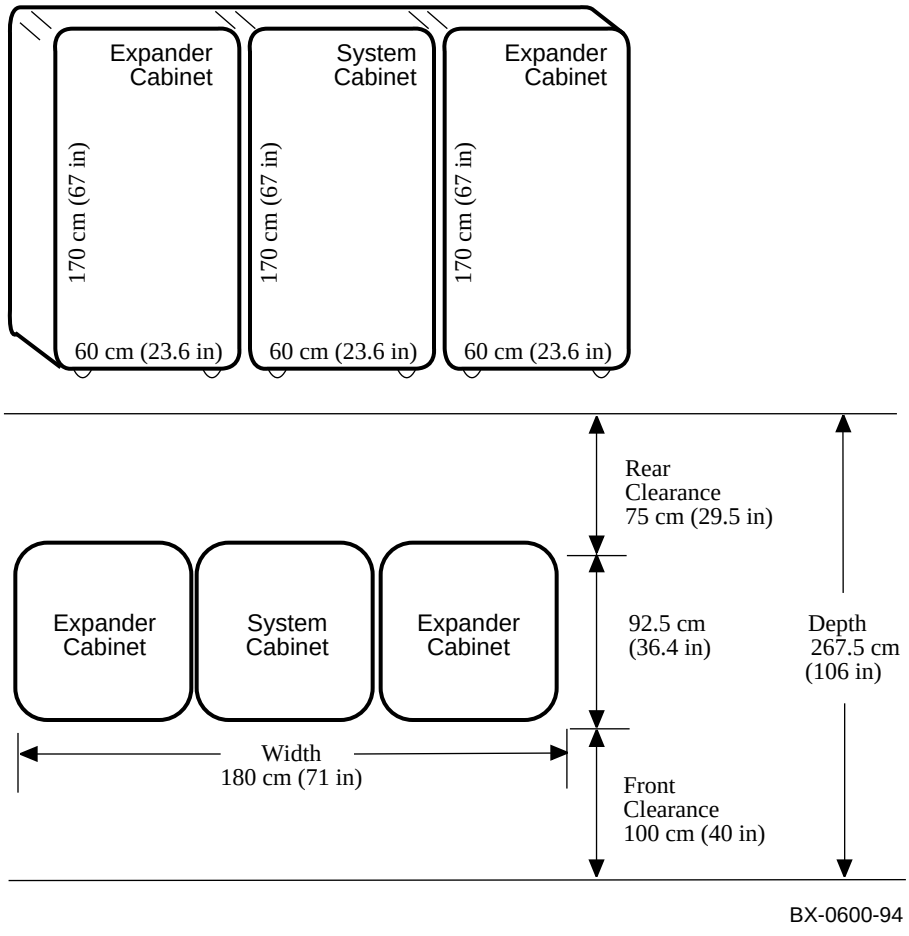
This chapter describes the system package and the location of components in the cabinet. Sections include:

- 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/APA 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/APA 30 A - Japan
AC power consumption (maximum)	2.6 KW

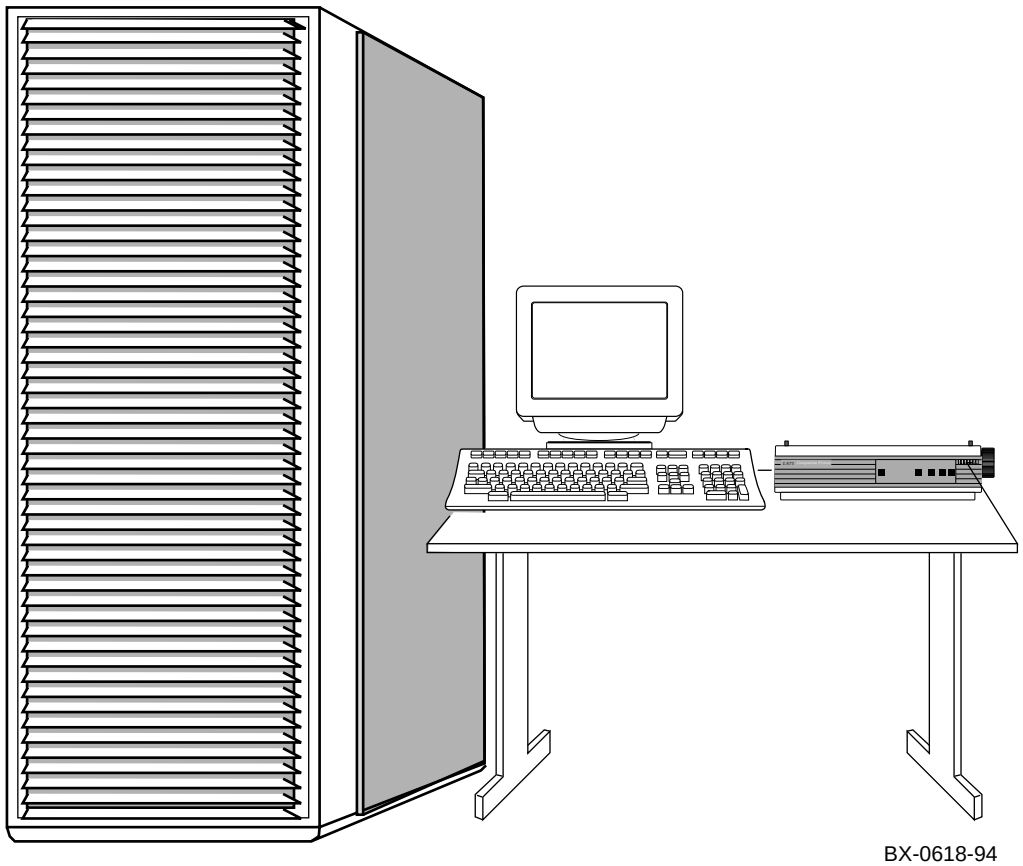
Table 2-2 Environmental Characteristics

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, an in-cabinet CD drive, disk drives, and optional battery backup.

Figure 2-2 Sample 8200 System



BX-0618-94

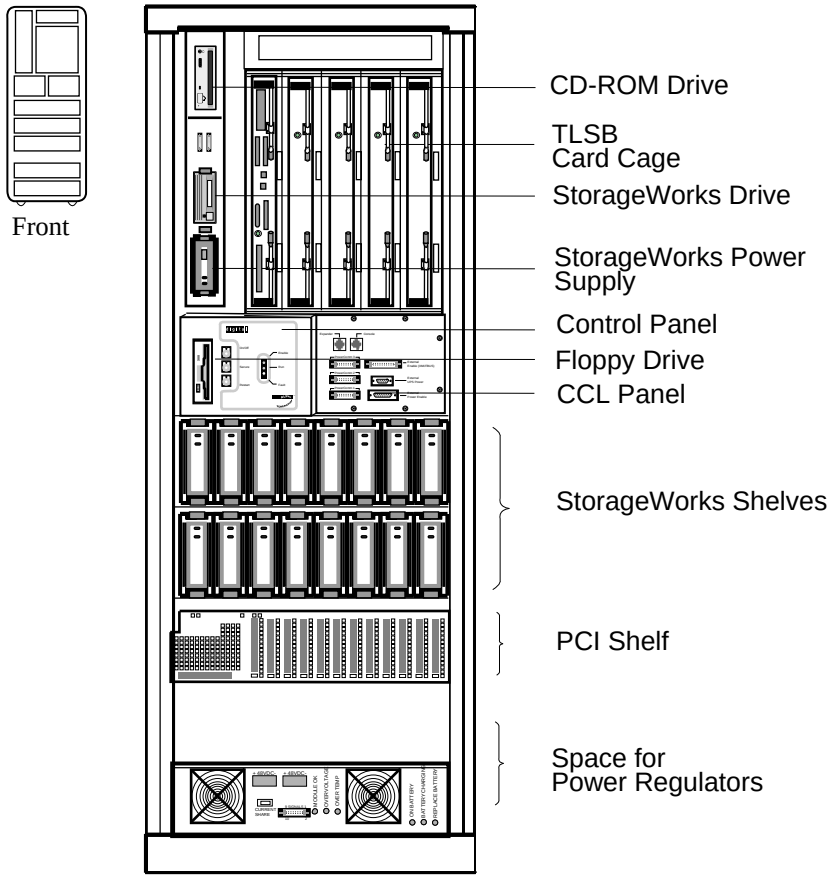
Your Digital customer service engineer has installed your system and verified that it is running properly. Before you turn on the system, familiarize yourself with its components:

- 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 an in-cabinet CD-ROM. Optional hardware includes StorageWorks shelves and PCI shelves.
- The **console load device** is used for installing operating systems and software.
- The **console terminal** is used for booting and for system management operations.
- 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-0604-94

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

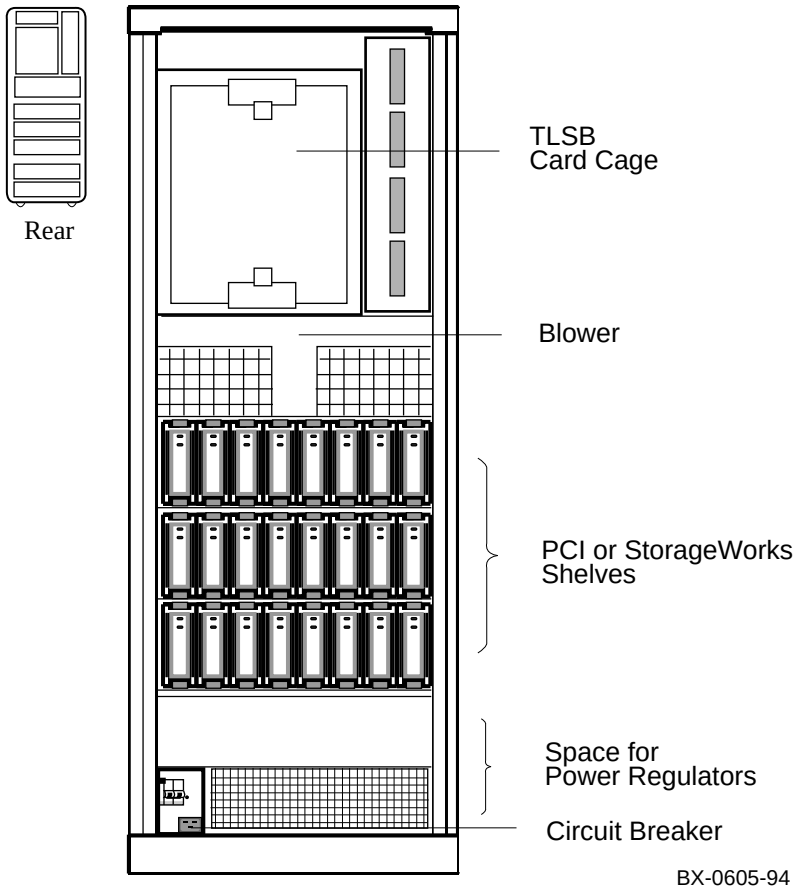
- TLSB card cage
- CD-ROM drive
- Floppy drive
- Control panel
- Cabinet control logic (CCL) panel
- PCI or StorageWorks shelves
- Power regulators

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

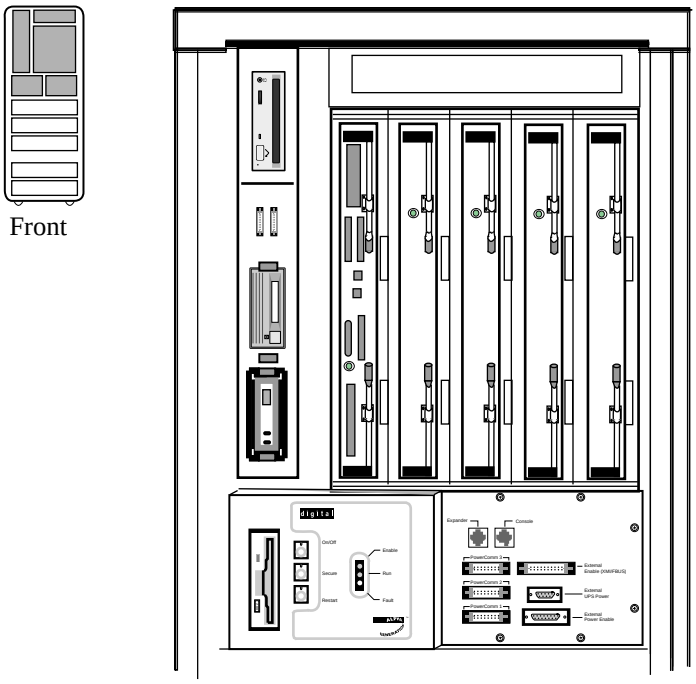
Optional components visible from the inside rear (and front) of the cabinet include PCI shelves, StorageWorks shelves, and an additional power regulator.

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



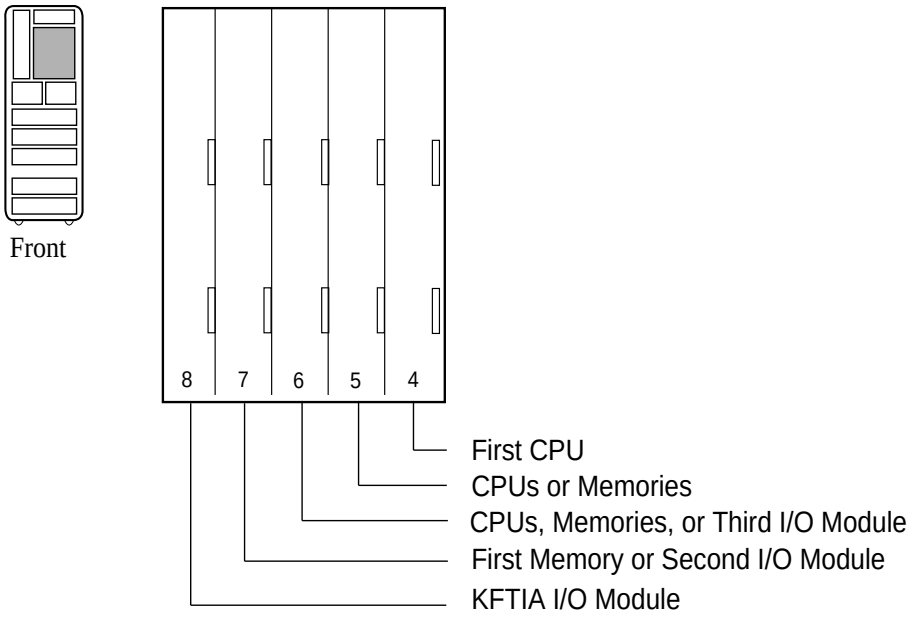
BX-0606-94

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 module, one CPU module, and one memory module as a minimum configuration. A KFTIA module is always installed in slot 8.

The blower cools the card cage.

Figure 2-6 TLSB Card Cage

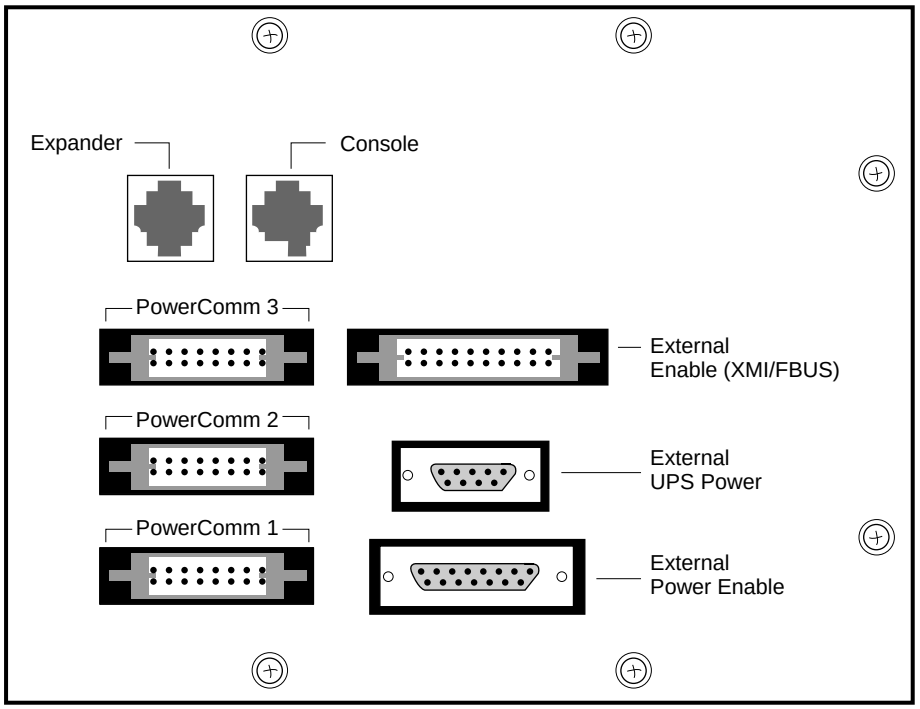


BX-0609-94

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

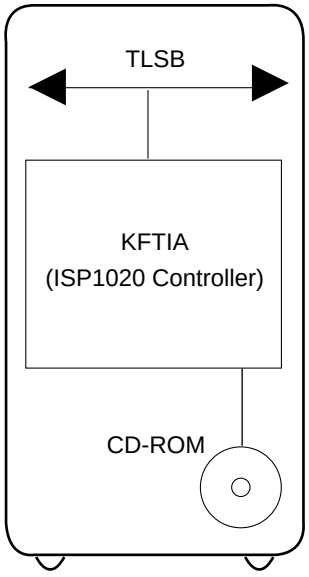
Table 2-3 Control/Status and I/O Connections

Connector Name	Function
Console	Console terminal connection.
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-0601-94

The console load device is used for:

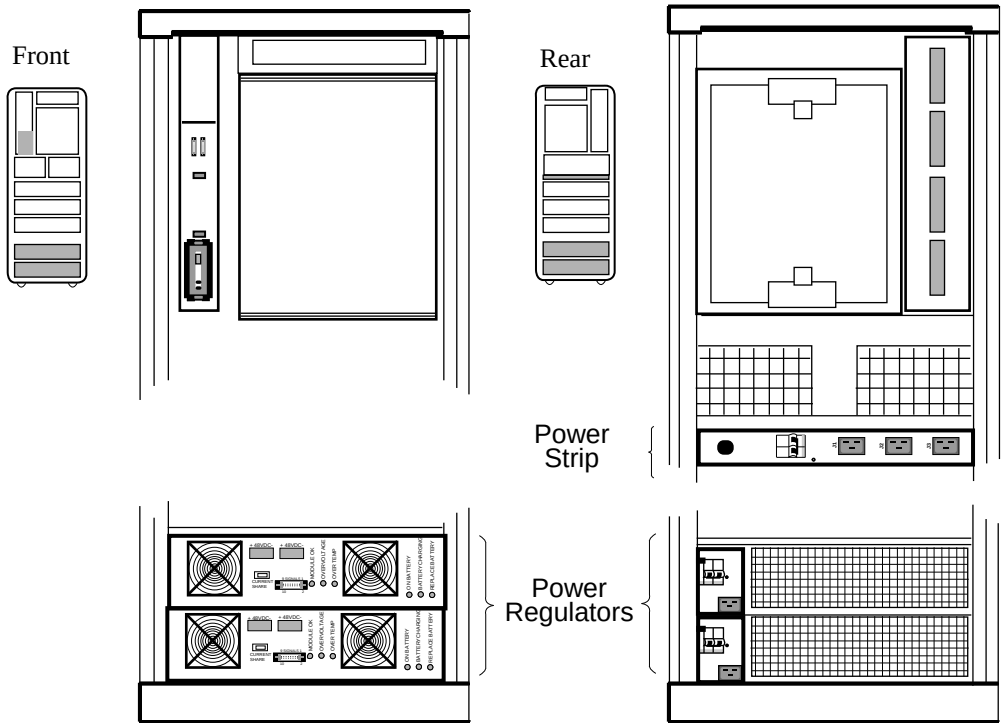
- Installing or updating software
- Loading a backup utility program
- Interchanging user data
- Updating module firmware

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. The KFTIA module, installed in the TLSB card cage, provides access to the CD-ROM.

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.

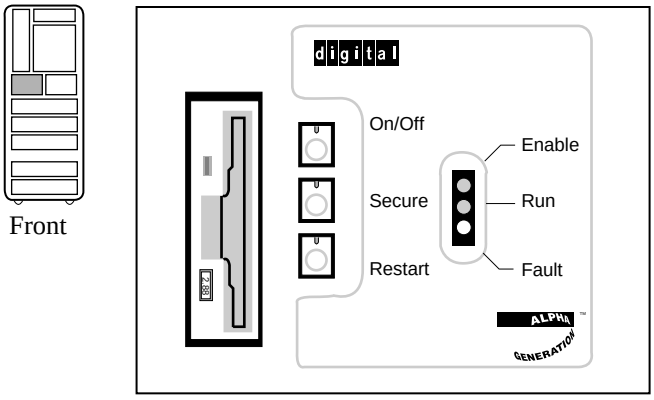
Each power regulator has an AC input assembly, a 48 VDC power regulator, two cooling fans, indicator lights, and optional battery backup (charger module and battery packs for UPS operation).

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-0607-94

Table 2-4 Control Panel Pushbuttons

Pushbutton	Position	Effect
On/Off	In	Applies power to the PSU and allows entry into console mode; position used while machine executes programs.
	Out	Removes 48 VDC power from the system.
Secure	In	Prevents entry into console mode; position used while machine executes programs.
	Out	Allows entry into console mode.
Restart	In	Momentary switch used to reinitialize the system. Causes self-test to run.

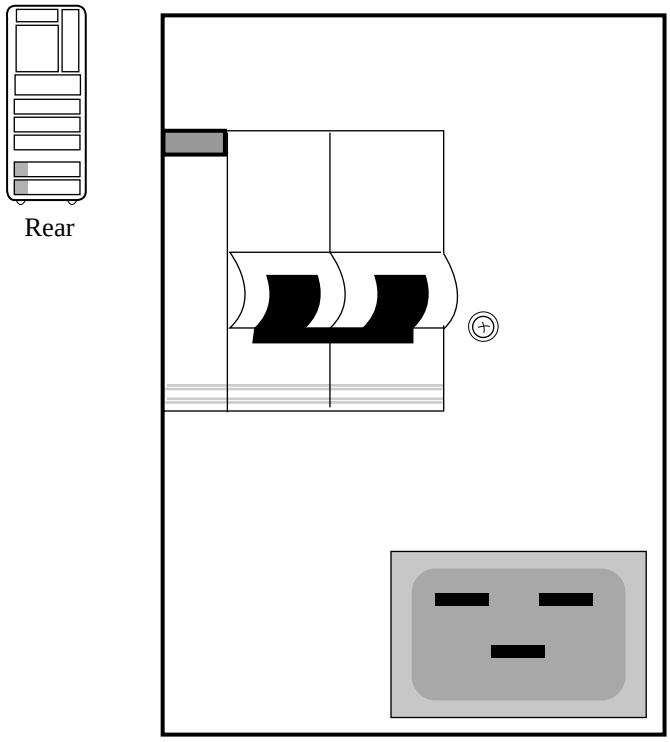
Table 2-5 Control Panel Indicator Lights

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.

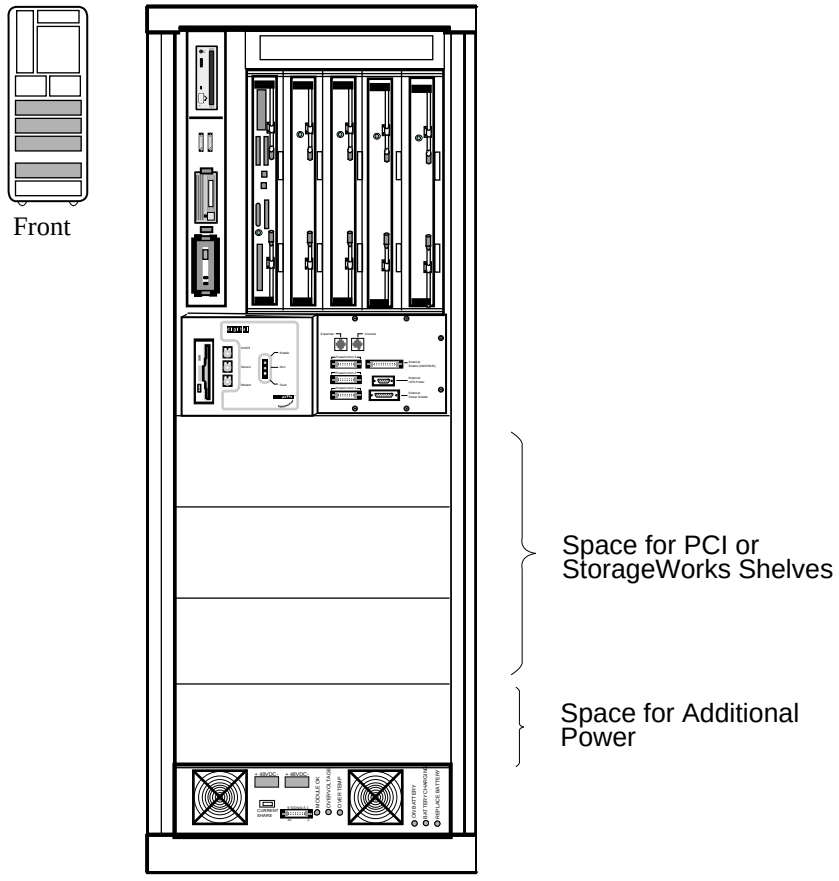
The circuit breaker controls power to the entire system.

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. A bridge module is available for access to EISA I/O. 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 are 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 processor modules for a total of 12 CPUs, up to seven memory modules for a total of 14 Gbytes of memory, a single-phase or three-phase power option with two optional battery cabinets, as well as XMI, FBUS+, PCI, SCSI, and battery PIUs.

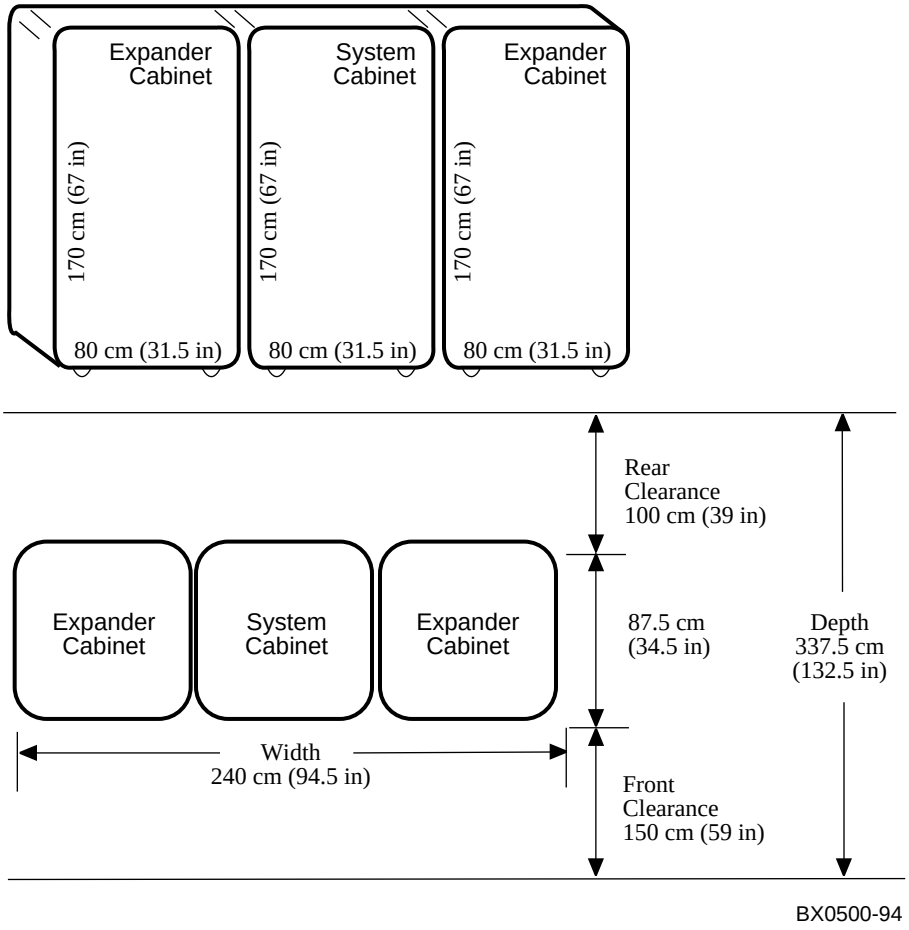
This chapter describes the AlphaServer 8400 system package, introduces the location of components in the cabinet—both front and rear views; and describes the system controls and indicators. Sections include:

- 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
Single-phase AC input voltage (current, maximum)	202–240 V (30 A) - North America 202–240 V (32 A) - Europe/APA 202–240 V (30 A) - Japan
3-phase AC input voltage (current, maximum)	120/208 V Wye (30 A) - North America 380–415 V Wye (30 A) - Europe/APA 202 V Delta (16 A) - Japan
Nominal frequency	50–60 Hz
AC power consumption (maximum)	4.6 KW

Table 3-2 Environmental Characteristics

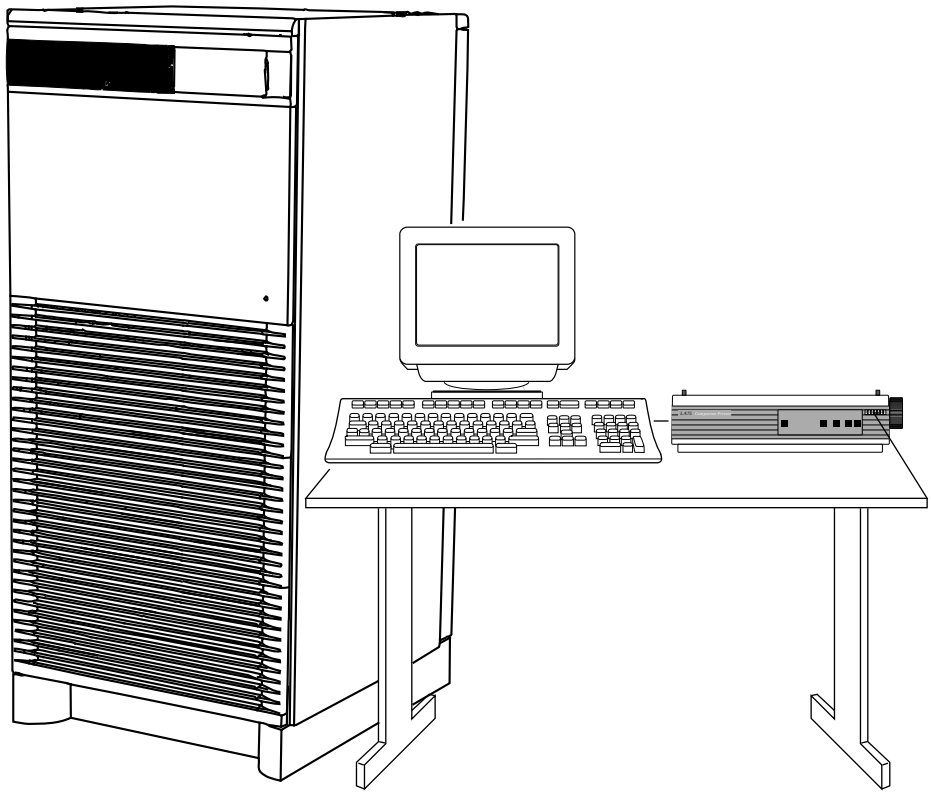
Environmental	Operating	Storage
Heat dissipation	15,700 Btu/hr	–
Temperature[1]	15°–28° C (59°–82° F)	-40°–66° C (-40°–151° F)
Relative humidity[1]	20–80%	10–95%
Altitude	0–2.4 km (0–8200 ft)	0–9.1 km (0–30,000 ft)

[1]Recommended operating temperature is 18°–24° C (65°–75° 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 terminal and printer, an accessories kit, and a documentation set. The system can have disk drives, a battery plug-in unit (PIU), up to two optional expander cabinets, and two optional battery cabinets.

Figure 3-2 Sample System



BX0502-94

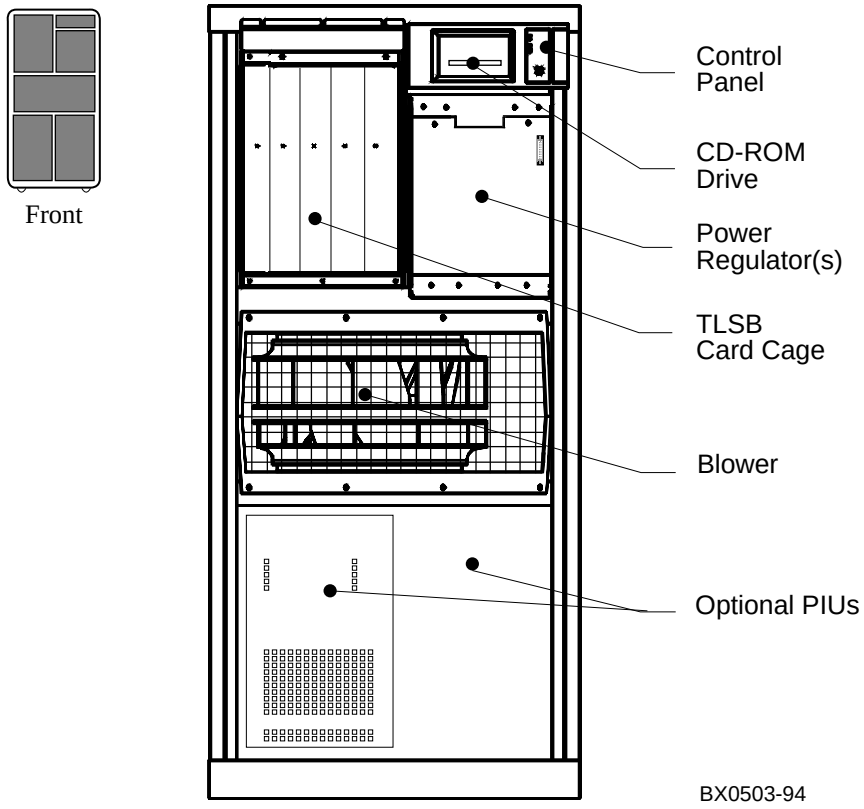
Your Digital customer service engineer has installed your system and verified that it is running properly. Before you turn on the system, familiarize yourself with its components:

- 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 terminal** is used for booting and for system management operations.
- The **console printer** provides a hardcopy record of system operations.
- **Optional PIUs** include the XMI PIU, Futurebus+ PIU, PCI PIU, BA655 SCSI PIU, and the battery PIU. 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 I/O devices, disk drives, and batteries to back up the power system.
- A **system documentation kit.**

3.3 System Front View

The control panel, plug-in unit panels, and CD-ROM 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
- Power regulator(s) (48 VDC)
- TLSB card cage (holds CPU/memory; slots 0–3)
- Cooling system (blower)

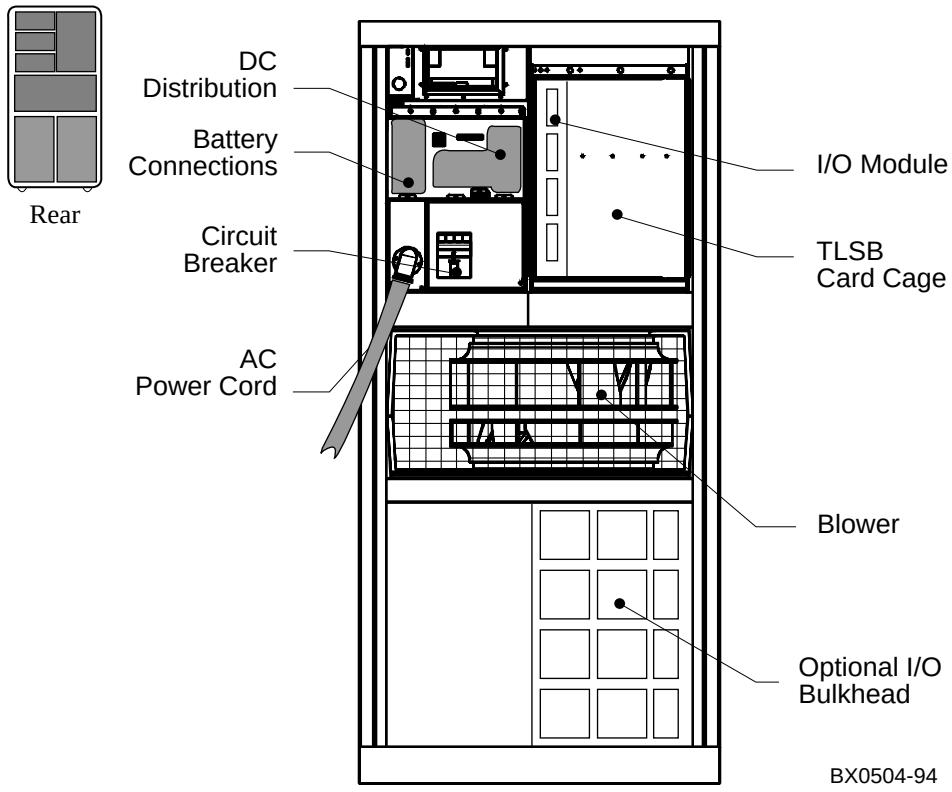
Optional components visible from the inside front include:

- 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:

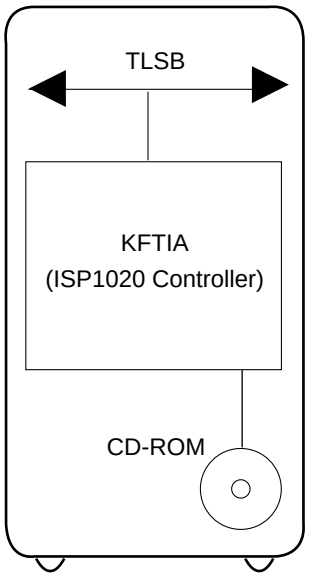
- I/O PIUs
- Battery PIU

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-0601-94

The console load device is used for:

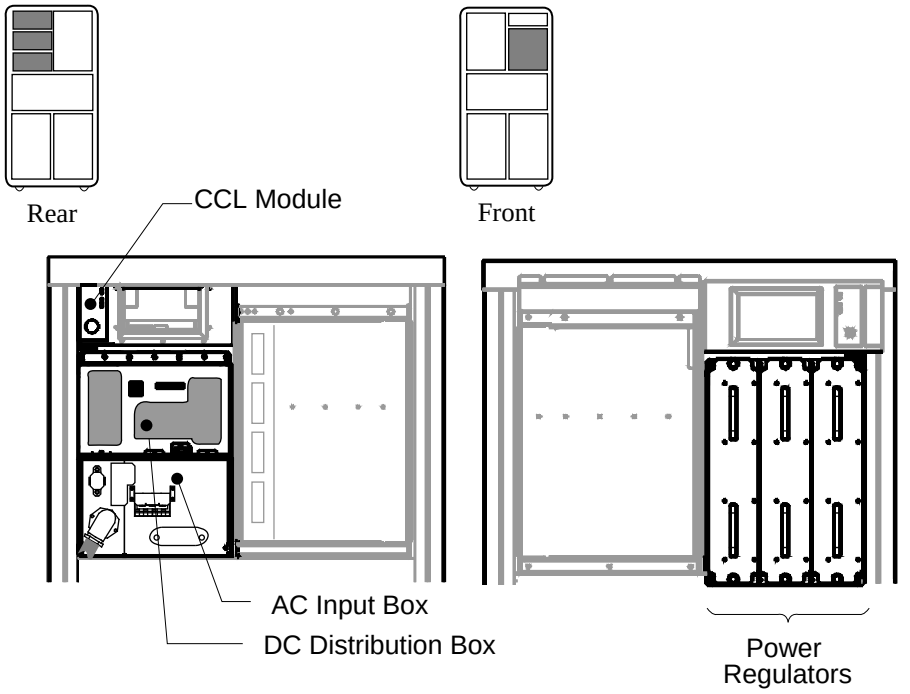
- Installing or updating software
- Loading a backup utility program
- Interchanging user data
- Updating module firmware

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. The KFTIA module is the interface between the CD-ROM drive and the TLSB.

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 (three-phase power only).

Figure 3-6 Power System



There are two variants to the 8400 power system. One variant consists of the AC input box and the H7264-AA regulator (see Figure 3-4) that uses single-phase AC. The other power system uses H7263 power regulators (see Figure 3-6), requiring three-phase AC. These regulators have the ability to use and charge optional battery packs and to switch 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.

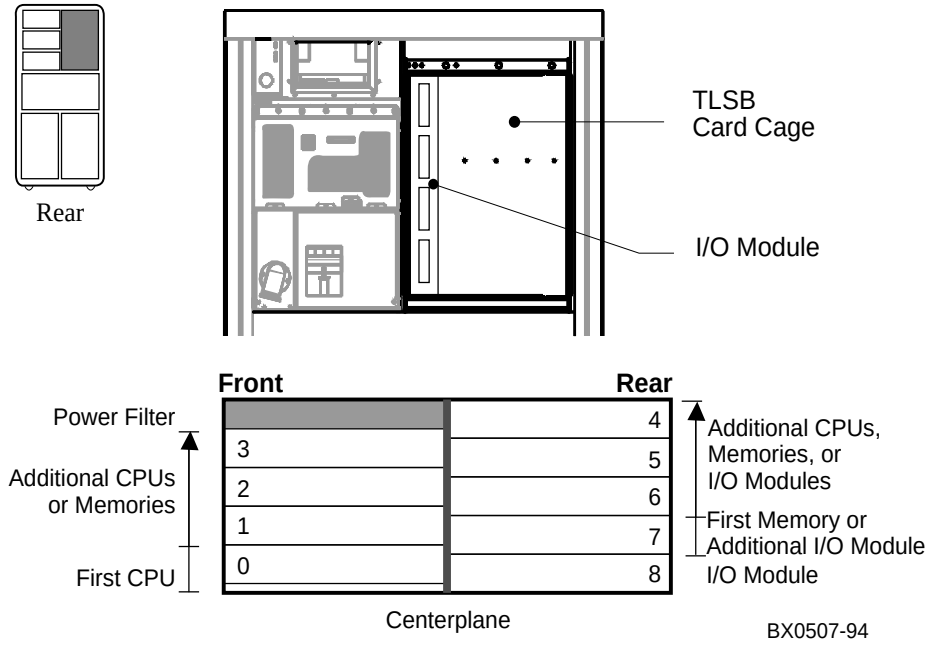
The cabinet control logic (CCL) module has a yellow power LED. When this LED is on, the CCL module is receiving 48 VDC power from the power regulators.

Battery backup capability can be provided in systems with three-phase power 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 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



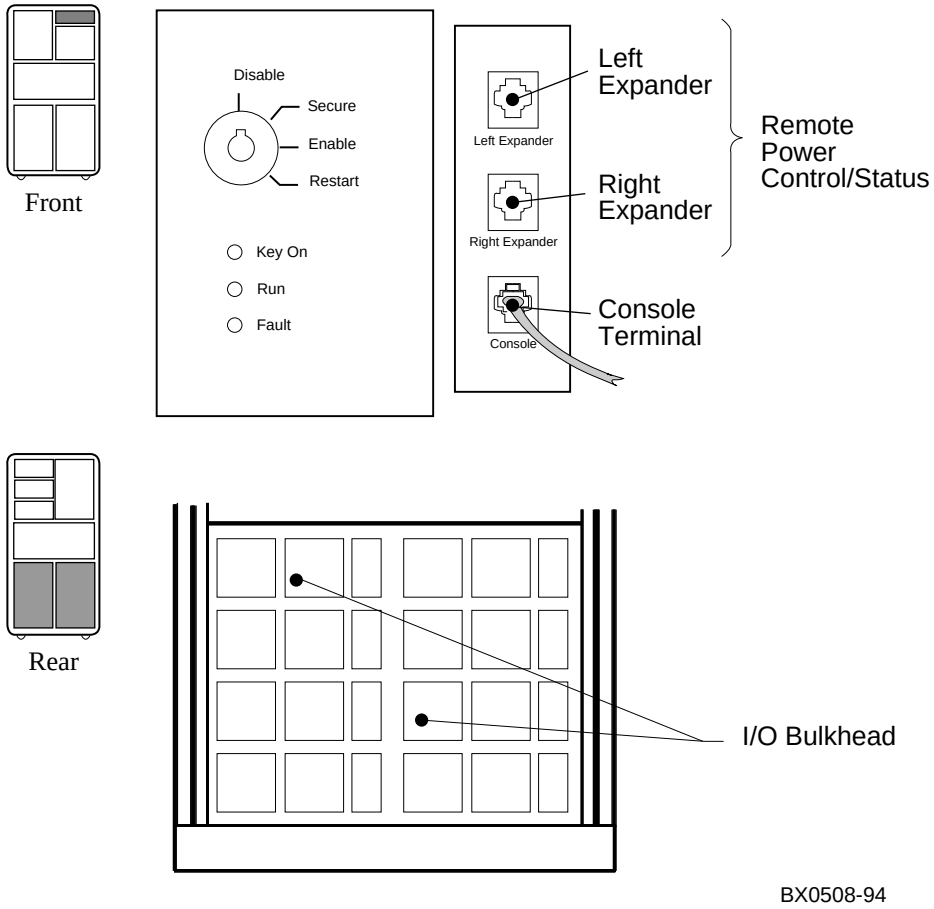
The 9-slot TLSB card cage is located in the upper left (front and rear) of the system cabinet, as viewed from the front. The TLSB card cage must contain one I/O port module, which is always installed in slot 8. The other eight slots contain a combination of KFTHA/KFTIA I/O modules (rear only), memory,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

Console terminal I/O 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 on the I/O modules, on the I/O bulkhead in the lower rear of the cabinet for the XMI PIU, and off the FBUS+ and PCI PIUs.

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



Console terminal I/O and expander cabinet remote power control/status connections are located to the right of the control panel. These three modular jacks allow power control/status connections to the left expander cabinet, right expander cabinet, and I/O connections to the console terminal. The console terminal modified modular jack is keyed so that an expander cabinet connector cannot be plugged into its jack.

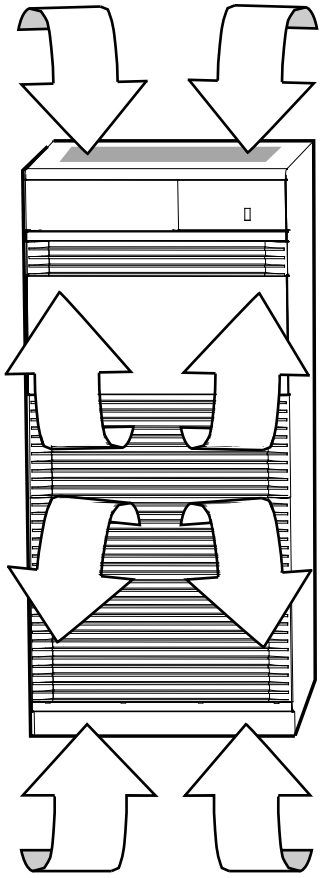
The system cabinet has four quadrants in the bottom of the cabinet. A Futurebus+ PIU, a PCI PIU, or XMI PIU can be installed in one or two quadrants and the other quadrants are used for expansion bays. These expansion bays can contain another Futurebus+ PIU, a PCI PIU, or XMI PIU, a BA655 PIU (containing SCSI drives), or a battery PIU. The number of expansion bays used by each PIU varies depending on the type of PIU.

The XMI PIU contains an I/O bulkhead, which houses the I/O connections for the devices in the PIU (such as adapter modules and disk drives). These I/O connections are located on a panel that is installed on the I/O bulkhead. The I/O bulkhead has single, dual, quad, and octal panels.

3.5.5 Cooling System

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

Figure 3-9 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 3-1) to maximize airflow.

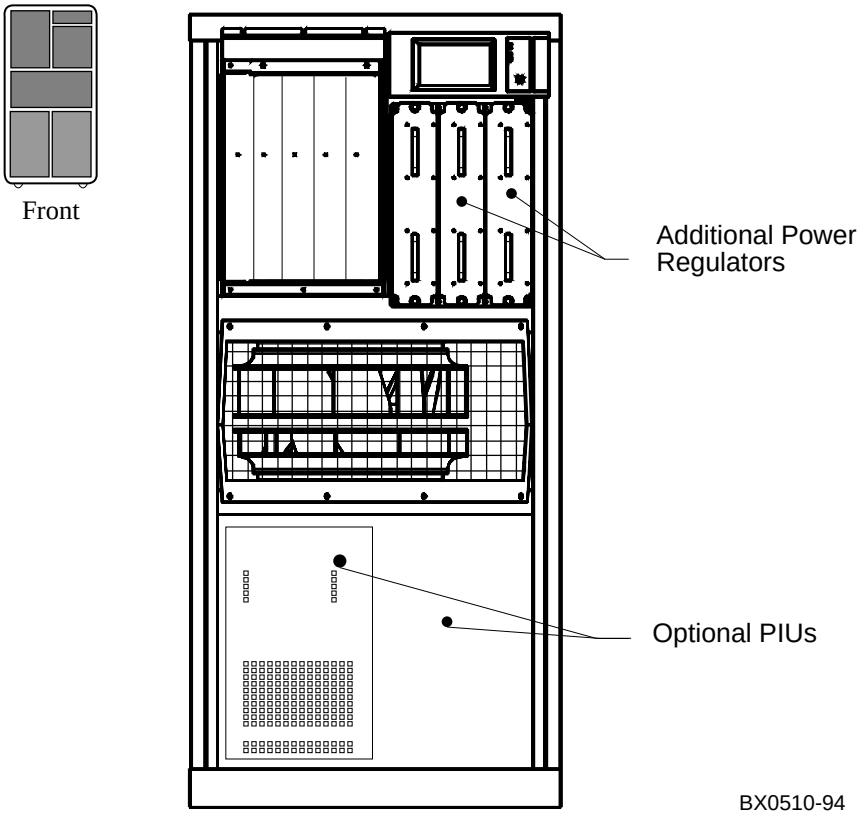
The blower, located in the center of the cabinet, draws air downward through the power regulators and TLSB card cage. It draws air upward through the PIUs. Filler modules, located in the TLSB card cage, help to direct airflow. Air is exhausted at the middle of the cabinet front and rear. The blower speed 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 BA655 PIUs can be installed in the system cabinet or six in an expander cabinet. The BA655 PIU occupies one quadrant and may be installed in any of the four quadrants in the main or expander cabinet or in Q5 or Q6 of an expander cabinet. The BA655 PIU can contain up to seven 3.5-inch or two 5.25-inch disks. Some of the supported devices include SCSI disk drives, CD-ROM, and tape drives.

XMI PIUs

A maximum of two XMI PIUs can be installed in the system or expander cabinet. 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. The Futurebus+ PIU occupies one quadrant and has 10 slots. The DWLAA module is standard and is installed in slot 5.

PCI PIUs

A maximum of two PCI PIUs can be installed in the system or expander cabinet. Each PCI PIU occupies one quadrant, either Q2 or Q4. A PCI PIU can contain up to two PCI shelves, each containing a 12-slot card cage, configured as three 4-slot PCI buses, or one PCI and one SCSI shelf. With a standard I/O module and connector module, access is provided to EISA options (see Chapter 4).

Battery PIUs

The system can be equipped with an optional battery PIU to provide uninterrupted power in case of a power failure. Each regulator requires a battery pack that is mounted in the bottom of the system cabinet. The battery PIUs provide a minimum of 8 minutes of full system operation when fully charged and nominally provide 11 minutes of full system operation.

Additional Power Regulators

The AlphaServer 8400 system 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.

If the AlphaServer 8400 system uses the H7263 power regulator, which requires 3-phase input, an optional second or third power regulator can be installed as a backup should a regulator fail or if the battery backup option is installed.

3.6 Controls and Indicators

This section introduces the system controls and indicators.

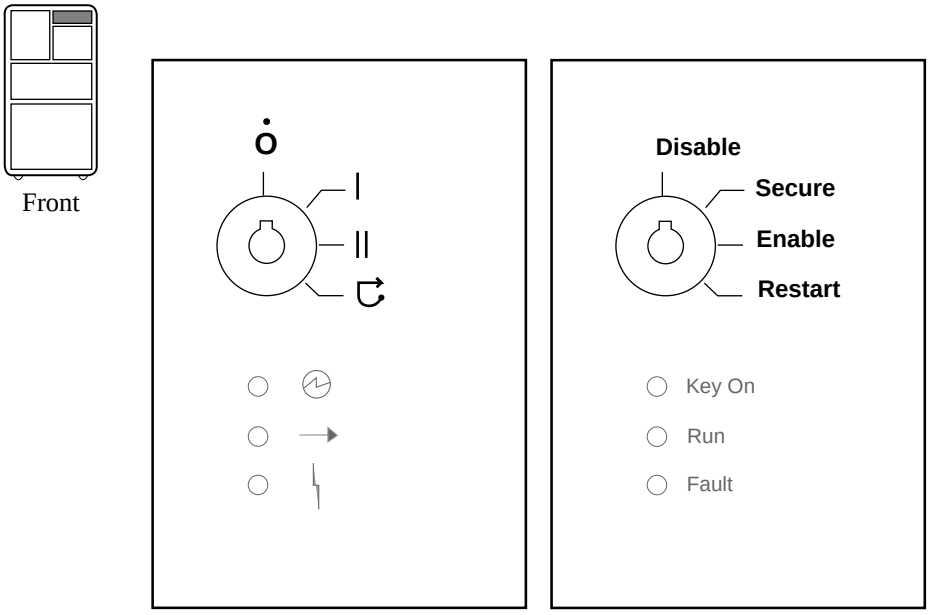
Subsections include:

- Control Panel Keyswitch
- Control Panel Indicator Lights
- Circuit Breaker and AC Power Indicators

3.6.1 Control Panel Keyswitch

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, determines the use of the console terminal, and controls system operation.

Figure 3-11 Control Panel Keyswitch



BX0511-94

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

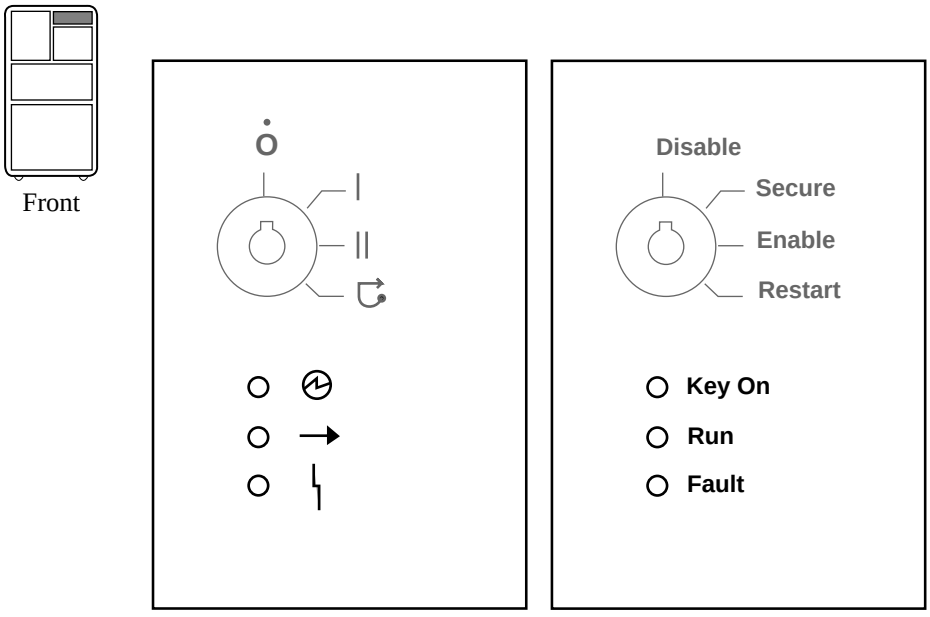
Table 3-3 Keyswitch Positions

Position	Effect
Disable	Removes 48 VDC power from the system. Power is still supplied to the CCL module.
Secure	Prevents entry into console mode; position used while machine executes programs.
Enable	Allows entry into console mode; position used while machine executes programs.
Restart	A momentary switch position, used to reinitialize the system; causes self-test to start running.

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.

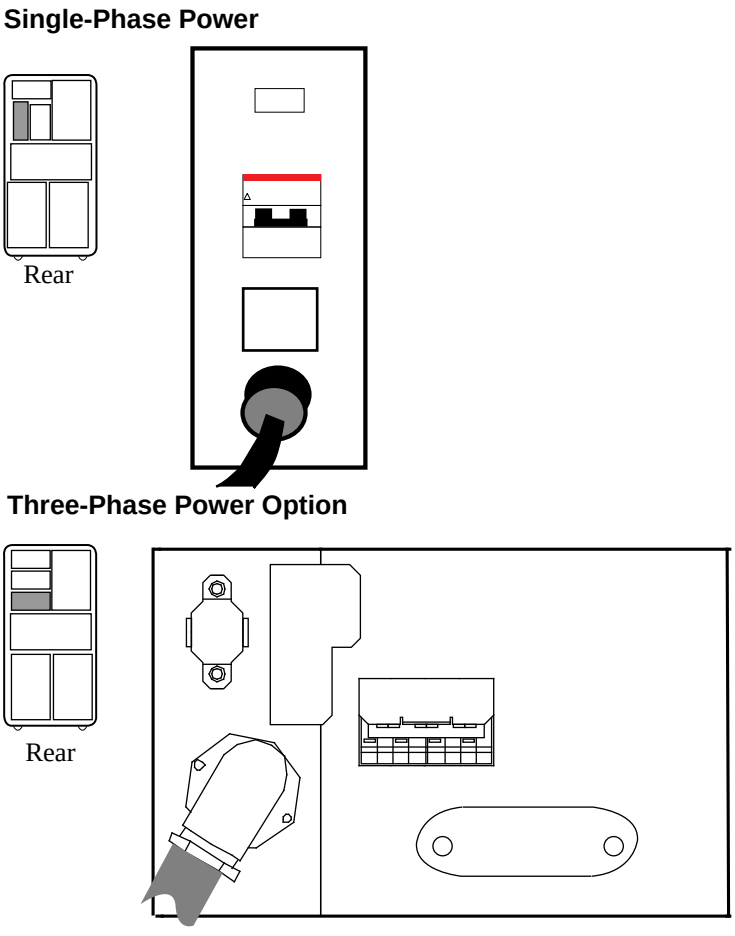
Table 3-4 Control Panel Indicator Lights

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. Ctrl/P halts the execution of operating system programs when the keyswitch is in the Enable position.
		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

There are no power indicators with single-phase power. 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

There is no circuit breaker lockout with single-phase power. 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.

Chapter 4

I/O Subsystems

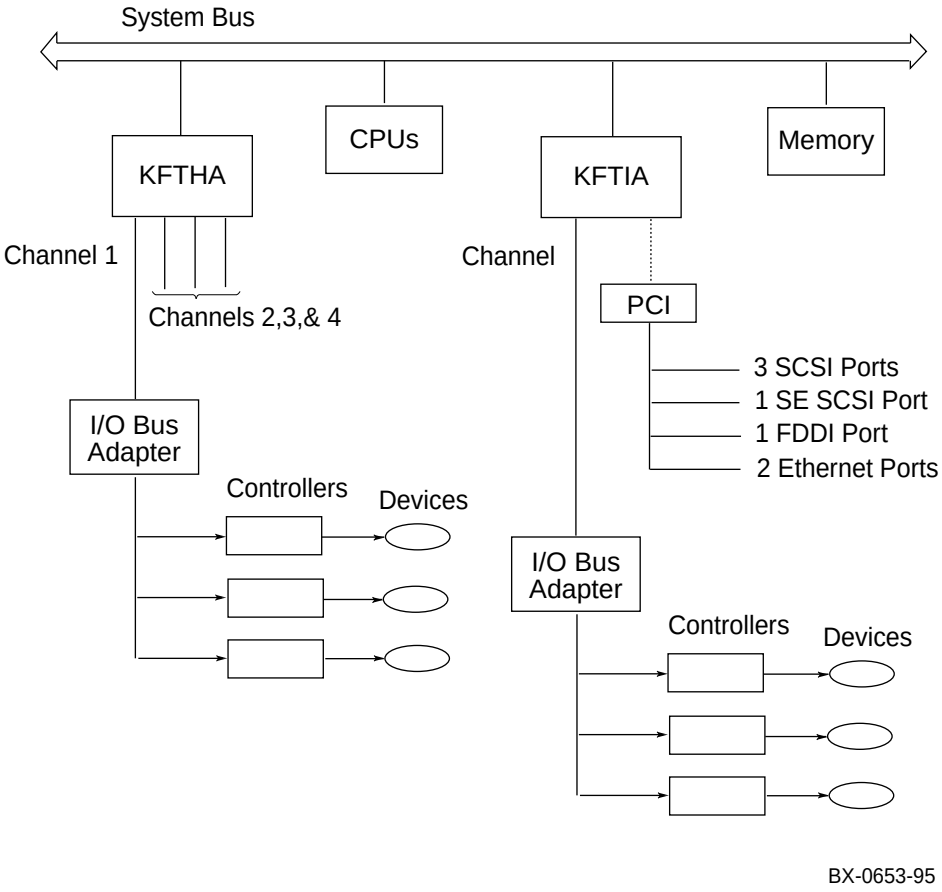
This chapter describes the AlphaServer 8200 and AlphaServer 8400 I/O subsystems. Sections include:

- I/O Subsystem Overview
- I/O Port Modules
- System Configuration Information
- PCI Adapter
- Standard I/O and Connector Modules
- 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

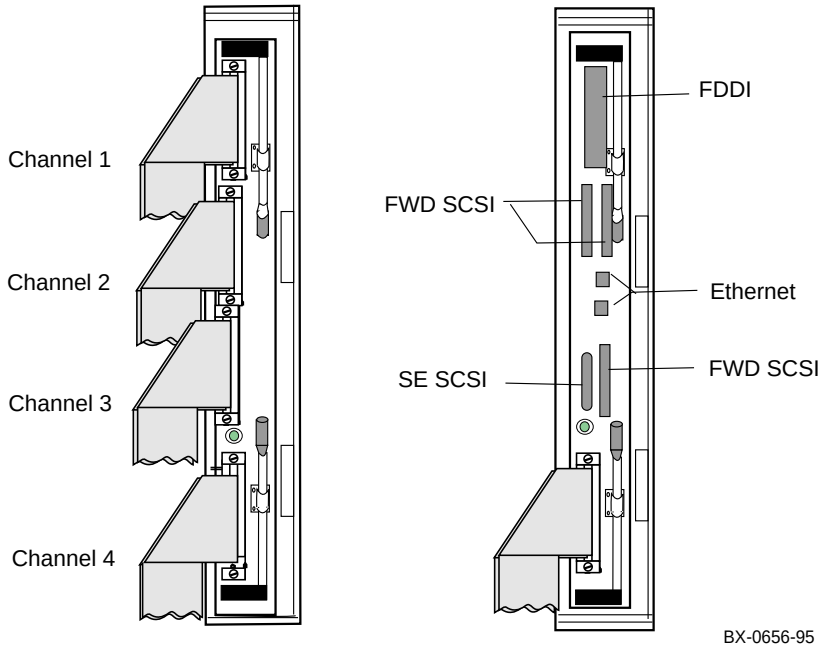


The interface from the AlphaServer 8200 and 8400 system bus to I/O is provided by two types of I/O adapter modules, the KFTHA and the KFTIA. The KFTHA has four channels, or hoses, to connect to I/O devices (see Figure 4-1). The KFTIA has one channel to connect to external I/O and one internal (integrated) PCI bus. Section 4.2 describes the KFTHA and KFTIA modules. System configuration information is given in Section 4.3.

4.2 I/O Port Modules

The KFTHA module has four channels, or "hoses," to external I/O subsystems such as PCI, Futurebus+, and XMI buses. The KFTIA module has one channel, or "hose," to external I/O such as a PCI (8200) or PCI, Futurebus+, or XMI bus (8400). 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 ("external hose" connections) for cables to go from the I/O port (see Figure 4-2) to I/O subsystems: a PCI, Futurebus+, or XMI PIU.

KFTIA

The KFTIA module has an internal peripheral component interconnect (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
- Three FWD (fast wide differential) SCSI connectors
- One single-ended (SE) SCSI connector
- Two Ethernet connectors

The 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.

The FWD SCSI connectors can be combined with Digital StorageWorks RAID controllers, storage cabinets and devices to access large amounts of SCSI disk storage.

The SE SCSI connector can be used to connect to a CD-ROM drive.

The KFTIA module also has one channel ("external hose" connection) for a cable to go from the I/O port to various shelves (8200) or plug-in units (8400). The plug-in units (PIUs) are self-contained assemblies that are easily installed in the system cabinet and expander cabinets. There are PIUs for the PCI, XMI, Futurebus+, disks, and batteries.

4.3 System Configuration Information

Basic information on the system and I/O subsystem configuration is displayed on power-up. Use the show configuration and show device commands for more information on the I/O subsystem options installed. Example 4-1 shows an AlphaServer 8400 system console self-test display.

Example 4-1 Console 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
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5 ❻
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6 ❼
          .  +  .  +  +  .  .  .  .  .  .  +  C7 PCI+ ❽
                    .  .  .  .  +  .  .  .        EISA+

                    .  .  .  .  .  .  . A0  .  ILV
                    .  .  .  .  .  .  .256  .  256MB
AlphaServer 8400 Console V1.0-0, SROM V1.0, Apr 2 1995
P00>>>
```

On power-up, the console displays the self-test results. Chapter 6 describes the console display 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 channel (C1) is connected to a Futurebus+ PIU, which has a DWLAA in slot 5.

As shown in Example 4-1 and Example 4-2, the KFTHA adapter has two of its four connectors used: the first channel, C4 ❺, is connected to an XMI PIU. The second and third channels, C5 ❻ and C6 ❼, are not used. The fourth channel, 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).

Use the **show configuration** (see Example 4-2) and **show device** (see Example 4-3) commands to obtain additional information about the system configuration. Chapter 7 provides additional information on these commands.

Example 4-2 Show Configuration Command

```
P00>>> show config                    ❸   ❹      ❺
   ❶     Name   ❷                    Type Rev   Mnemonic
  TLSB
  0++    KN7CC-AB                    8014 0000  kn7cc-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
```

The **show configuration** command (Example 4-2) displays system configuration information 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 Show Configuration Command (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+     ISP1020           10201077  0001   isp4
8+     ISP1020           10201077  0001   isp5
A+     DAC960               11069  0000   dac0

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

  EISA connected to pci0 through sio0     eisa0   ⓯
4+     KFESB              2EA310  0000   kfesb0
```

The KFTHA module in TLSB slot 8 has an XMI PIU attached to its first channel (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 channels (C5 and C6) are not used, so they do not appear on the show configuration display. The fourth channel, C7, is connected to a PCI adapter ❾. There are four modules in the PCI adapter:

- Slot 0 - SIO (standard I/O module)
- Slot 7 - ISP1020 – KZPSA (PCI SCSI)
- Slot 8 - ISP1020 – KZPSA (PCI SCSI)
- Slot A - DAC960 – KZPSC (SCSI RAID)

The standard I/O (PCI-to-EISA bridge) module ❿, located in slot 0 of the PCI adapter, along with the connector module (see Section 4.5), provides:

- Tulip Ethernet port ⓫
- Floppy controller ⓬
- Keyboard port ⓭
- Mouse port ⓮

Information for the KFESB (EISA to DSSI) is displayed. ⓯

Refer to Appendix A for additional information on device names and mnemonics.

Example 4-3 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 kfesb0, slot 4, bus 1, hose3...
duc5.5.0.1004.3    RF3111$DIA5              RF31
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...
polling for units on isp2, slot 0, bus 0, hose4...
polling for units on isp3, slot 1, bus 0, hose4...
polling for units on isp4, slot 4, bus 0, hose4...
polling for units on isp5, slot 5, bus 0, hose4...
P00>>>
```

The **show device** command provides the following I/O adapter device information:

- Device mnemonic ❶
- Slot number ❷
- Bus number ❸
- Hose number ❹

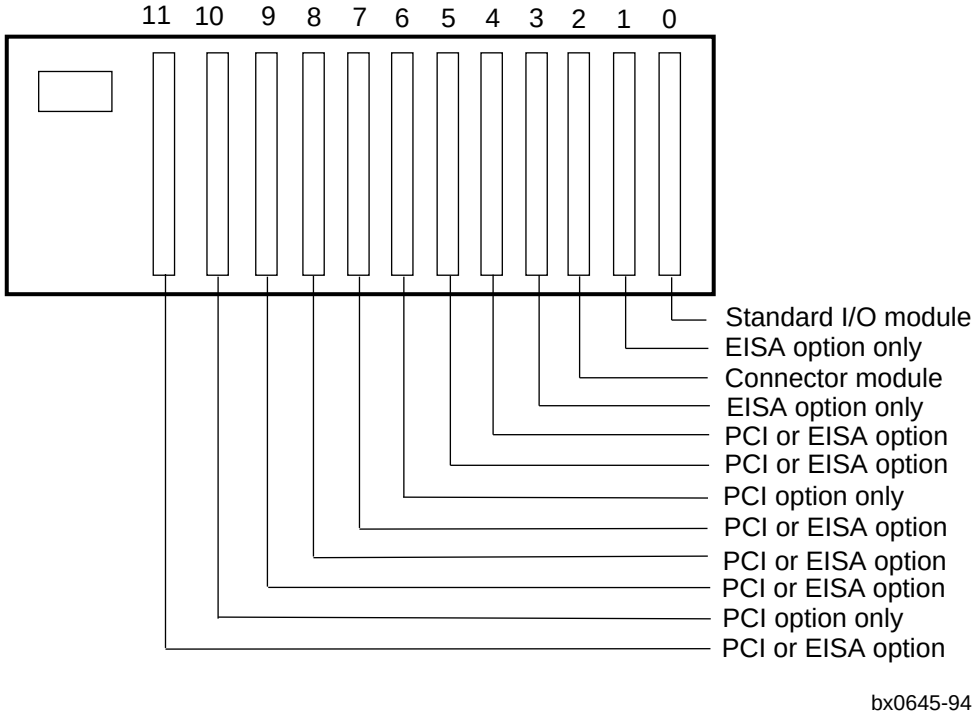
The KZMSA, in slot E (14) of the XMI PIU, is a SCSI controller with three RZ26L SCSI disk drives.

Hose 3 of the KFTHA connects to the PCI PIU, which has devices in slots 0, 7, 8, and A (10).

4.4 PCI Adapter

The PCI adapter, DWLPA, provides a complete PCI and EISA bus subsystem for use with the AlphaServer 8200 and 8400 systems.

Figure 4-3 PCI/EISA Slot Configuration



The PCI adapter has 12 slots for option modules, numbered right-to-left from 0 to 11. The shelf can be populated with PCI options or EISA options.

PCI Configuration Rules:

- PCI options only - With no EISA options, all twelve slots are available for PCI options.
- EISA options only - With no PCI options, eight slots are available for EISA options, since slots 0, 2, 6, and 10 cannot be used.
- PCI and EISA options - 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).

The standard I/O module, if present, is always installed in slot 0, with the connector module (see Section 4.5) installed in slot 2. Table 4-1 lists the rules for each PCI slot in a PCI/EISA configuration.

Table 4-1 PCI/EISA Configuration Rules

Slot	Usage
0	Standard I/O module (PCI-to-EISA bridge)
1	EISA option only
2	Connector module (floppy, keyboard, and mouse)
3	EISA option only
4	PCI or EISA option
5	PCI or EISA option
6	PCI option only
7	PCI or EISA option
8	PCI or EISA option
9	PCI or EISA option
10	PCI option only
11	PCI or EISA option

4.5 Standard I/O and Connector Modules

The standard I/O (SIO) module allows access to EISA options (see Figure 4-4). The connector module is used with the SIO module (see Figure 4-5).

Figure 4-4 EISA Support

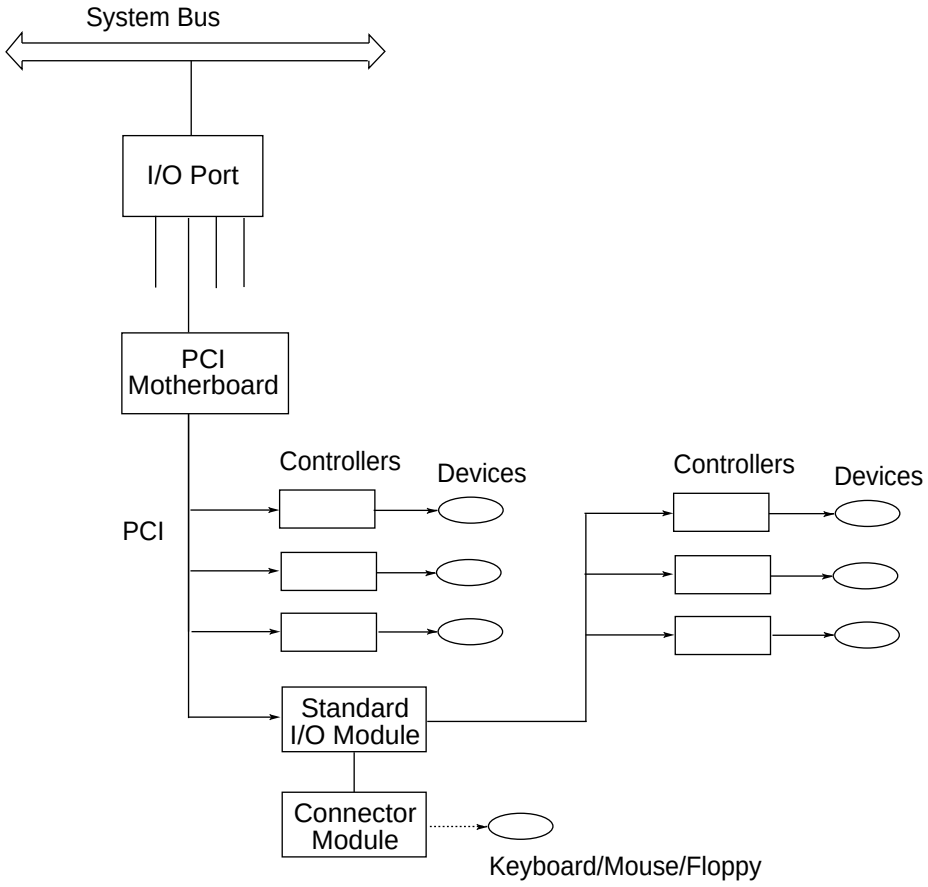
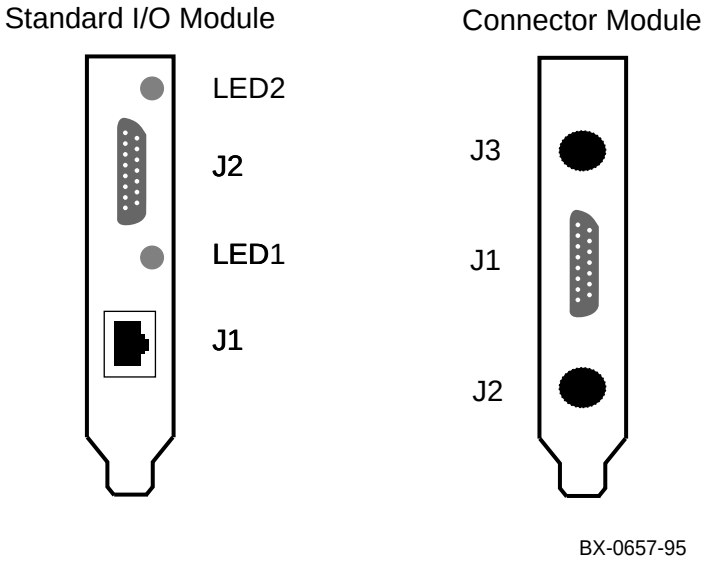


Figure 4-5 SIO and Connector Modules



When using EISA options, the PCI-to-EISA bridge (standard I/O module) and the connector module (see Figure 4-5) are both required. The SIO module is installed in slot 0 of the PCI adapter. The SIO module makes the slots indicated in Table 4-1 available for EISA modules. Example 4-2 illustrates how the **show configuration** command reports EISA controllers.

The standard I/O module has two connectors:

- J1 - ThinWire, twisted-pair Ethernet
- J2 - 15-pin thick wire Ethernet

The connector module (for floppy, keyboard, and mouse) is installed in slot 2 and cabled to the SIO module.

The connector module has three connectors:

- J1 - Floppy I/O (to run the ECU utility)
- J2 - Mouse I/O
- J3 - Keyboard I/O

With these modules installed, the EISA Configuration Utility (ECU) can be run (see Section 4.6). Chapter 7 gives additional information on the **run** and **runecu** commands.

4.6 EISA Configuration Utility

Run the EISA Configuration Utility (ECU) whenever you add, remove, or move an EISA board in your system.

Table 4-2 EISA Bus Configuration Procedure Summary

Step	Explanation
Install EISA option	Use the instructions supplied with the EISA option.
Power up and run ECU	If the 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.
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).
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-XA EISA bridge option. Use ECU to:

- Automatically configure EISA boards installed in your system.
- Obtain online help to guide you through the configuration process.

You need to run ECU whenever you add, remove, or move an EISA board in your system. The 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. The system configuration diskette must have the volume label SYSTEMCFG. Refer to Chapter 7 for a description of the **runecu** command used to invoke the ECU.

Chapter 5

Booting

This chapter describes how to boot AlphaServer 8200/8400 systems. The **boot** command is described in Chapter 7, and Appendix A provides information on device names, device mnemonics, and operating system boot flags. Sections include:

- Boot Devices
  - — Locating the Boot Device and Booting
  - — Show Configuration Command
- Booting OpenVMS Alpha
- Booting Digital UNIX

5.1 Boot Devices

The operating system 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 adapter, or by Ethernet from a remote disk on another system.

Figure 5-1 Boot Devices

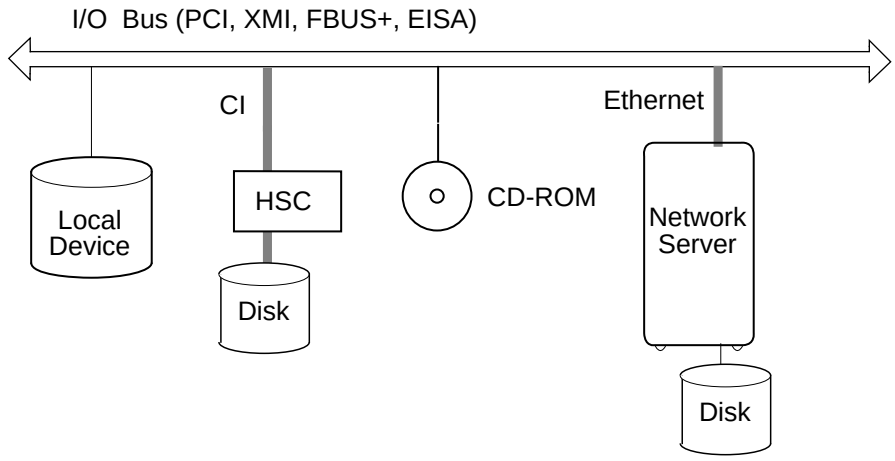


Table 5-1 Boot Devices

Device	Location
CD-ROM	In-cabinet compact disk drive, used for booting the Loadable Firmware Update (LFU) Utility. See Appendix B.
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.
Remote disk	Disk connected to another system on the Ethernet, through the Ethernet port interface or the adapter.

5.1.1 Locating the Boot Device and Booting

Use the show device command to locate the device from which to boot the operating system. Boot from the device.

Example 5-1 Show Device and Boot Commands

```
P00>>> show device          ❶
polling for units on kzmsa0, slot 14, bus 0, xmi0...
dkb100.1.1.14.0        DKB100          RZ26L    440C   ❷
dkb200.2.1.14.0        DKB200          RZ26L    440C
dkb300.3.1.14.0        DKB300          RZ26L    440C
polling for units on floppy0, slot 0, bus 1, hose3...
dva0.0.0.1100.3        DVA0            RX26
polling for units on kfesb0, slot 4, bus 1, hose3...
duc0.0.0.1004.3        R2WUIC$DIA0     RF72
duc5.5.0.1004.3        RF3111$DIA5     RF31
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...
polling for units on isp2, slot 0, bus 0, hose4...
polling for units on isp3, slot 1, bus 0, hose4...
polling for units on isp4, slot 4, bus 0, hose4...
polling for units on isp5, slot 5, bus 0, hose4...
P00>>> b dkb200                                    ❸
Building FRU table.............................
(boot dkb200.2.1.14.0 -flags 0)
SRM boot identifier: scsi 0 14 1 2 200 102a 0c36
boot adapter:kzmsa0 rev 5256 in bus slot 14
                          off of kftha0 in TLSB slot 8
block 0 of dkb200.2.1.14.0 is a valid boot block
reading 1018 blocks from dkb200.2.1.14.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7f400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
```

Example 5-1 Show Device and Boot Commands (Continued)

```
setting affinity to the primary CPU
jumping to bootstrap code

OpenVMS Alpha Operating System, Version V6.2 ❹
```

❶ **Show device** displays information about each I/O device. Polling checks the I/O bus for devices connected to <device>. The <device> is the name the console assigns to an I/O adapter or device (such as kzmsa0) in the system (see Example 5-2). A list of installed adapters is given in the **show configuratio**n command display.

❷ The next line contains four columns. The first column contains the device type and unit number, node number, device channel number, I/O node number, and I/O channel number, separated by periods. The second column displays the name of the device given by the device controller. The third column shows the device type. The fourth column reports the firmware revision on the device (not all report this).

❸ In the **boot** command, **dk** is the device code of the boot device, **b** is the boot device controller designation, and **200** specifies the unit number of the boot device.

❹ The operating system banner appears.

5.1.2 Show Configuration Command

Use the show configuration command to list the installed adapters.

Example 5-2 Show Configuration Command

```
P00>>> show conf
        Name              Type  Rev    Mnemonic
 TLSB
 0++ KN7CC-AB             8014  0000   kn7cc-ab0
 3+  MS7CC                5000  0000   ms7cc0
 7+  KFTHA                2000  0D02   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
  C1 FBUS connected to kftia0          fbus0
 5+  DWLAA                2003  0000   dwlaa0
```

Example 5-2 Show Configuration Command (Continued)

```
C4 XMI connected to kftha0                xmi0
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
0+  SIO                  4828086  0003    sio0
7+  ISP1020             10201077  0001    isp4
8+  ISP1020             10201077  0001    isp5
A+  DAC960                 11069  0000    dac0

  Controllers on SIO                      sio0
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 pci0 through sio0 eisa0
4+ KFESB                 2EA310  0000    kfesb0
```

The **show configuration** command lists the installed modules and adapters. The module/adapter name, type, revision, and mnemonic are listed in columns 2 through 5 in the display. Column 1 lists the module/adapter node number and self-test status (see Chapter 6).

5.2 Booting OpenVMS Alpha

Use the show device command to locate the device from which to boot the operating system. Boot from the device.

Example 5-3 FWD OpenVMS Alpha Boot

```
P00>>> show dev isp0     ❶
polling for units on isp0, slot 0, bus 0, hose0...
dka0.0.0.0.0     DKA0              RZ26L 440C      ❷
dka100.1.0.0.0   DKA100             RZ26 T384
dka200.2.0.0.0   DKA200             RZ28 D41C
dka300.3.0.0.0   DKA300             RZ26 T384
dka400.4.0.0.0   DKA400            RZ26L 440C
dka500.5.0.0.0   DKA500             RZ28 T436
dka600.6.0.0.0   DKA600            RZ26L 440C
P00>>> boot dka200.2.0.0.0     ❸
Building FRU table...................
(boot dka200.2.0.0.0 -flags 0,0)
SRM boot identifier: scsi 0 0 0 2 200 ef00 10201077
boot adapter: isp0 rev 1 in bus slot 0 off of
                  kftia0 in TLSB slot 8
block 0 of dka200.2.0.0.0 is a valid boot block
reading 1018 blocks from dka200.2.0.0.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7f400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

     OpenVMS Alpha Operating System, Version V6.2 ❹
```

❶ **Show device isp0** displays device information for each isp (device mnemonic is obtained from the **show configuration** command) device.

❷ Polling checks the I/O bus for device configurations. The next line contains four columns. The first column contains the device type and unit number, node number, device channel number, I/O node number, and I/O channel number, separated by periods. The second column displays the name of the device given by the device controller. The third column shows the device type. The fourth column reports the firmware revision on the device (not all report this).

❸ In the **boot** command, **dk** is the device code of the boot device, **d** is the boot device controller designation, and **200** specifies the unit number of the boot device.

❹ The operating system banner appears.

For more information:
OpenVMS Alpha Version 6.2 Upgrade and Installation Manual

Example 5-4 Single-Ended OpenVMS Alpha Boot

```
P00>>> show device isp3                          ❶
polling for units on isp3, slot 5, bus 0, hose0...❷
dkd100.1.0.5.0    DKD100            RZ28        D41C
dkd200.2.0.5.0    DKD200            RZ26L       440C
dkd300.3.0.5.0    DKD300            RZ26L       440C
dkd400.4.0.5.0    DKD400            RZ28        T436
P00>>> boot dkd100 -fl 0,0       ❸
Building FRU table..................
(boot dkd100.1.0.5.0 -flags 0,0)
SRM boot identifier: scsi 0 5 0 1 100 ef00 10201077
boot adapter: isp3 rev 1 in bus slot 5 off of
                       kftia0 in TLSB slot 8
block 0 of dkd100.1.0.5.0 is a valid boot block
reading 1018 blocks from dkd100.1.0.5.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7f400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

 OpenVMS Alpha Operating System, Version 6.2 ❹
```

❶ **Show device isp3** displays device information for each isp device.

❷ Polling checks the I/O bus for device configurations. The next line contains four columns. The first column contains the device type and unit number, node number, device channel number, I/O node number, and I/O channel number, separated by periods. The second column displays the name of the device given by the device controller. The third column shows the device type. The fourth column reports the firmware revision on the device (not all report this).

❸ In the **boot** command, **dk** is the device code of the boot device, **d** is the boot device controller designation, **100** specifies the unit number of the boot device, **-fl** indicates additional command options follow, **0** is the system root of the boot device, and **0** is the bootstrap option.

❹ The operating system banner appears.

Example 5-5 OpenVMS Alpha Boot from a KZPSA Device

```
P08>>> show dev kzpsa0                                    ❶
polling for units on kzpsa0, slot 3, bus 0, hose1... ❷
kzpsa0.7.0.3.1 dkc TPwr 1 Fast 1 Bus ID 7 E01 T-A04_11
dkc100.1.0.3.1        DKC100     RZ28  D41C
dkc200.2.0.3.1        DKC200     RZ28  D41C
dkc300.3.0.3.1        DKC300    RZ26L  440C
dkc400.4.0.3.1        DKC400    RZ26L  440C
dkc500.5.0.3.1        DKC500    RZ26L  440C
dkc600.6.0.3.1        DKC600    RZ26L  440C
P08>>> boot dkc100                          ❸
Building FRU table.........
(boot dkc100.1.0.3.1 -flags 0)
SRM boot identifier: scsi 1 3 0 1 100 ef00 81011
boot adapter: kzpsa0 rev 0 in bus slot 3
                    off of kftha0 in TLSB slot 8
block 0 of dkc100.1.0.3.1 is a valid boot block
reading 1018 blocks from dkc100.1.0.3.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7f400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

 OpenVMS Alpha Operating System, Version 6.2  ❹
```

❶ **Show device kzpsa** displays device information for each kzpsa device.

❷ Polling checks the I/O bus for device configurations. The next line contains four columns. The first column contains the device type and unit number, node number, device channel number, I/O node number, and I/O channel number, separated by periods. The second column displays the name of the device given by the device controller. The third column shows the device type. The fourth column reports the firmware revision on the device (not all report this).

❸ In the **boot** command, **dk** is the device code of the boot device, **c** is the boot device controller designation, and **100** specifies the unit number of the boot device. The system boots.

❹ The operating system banner appears.

Example 5-6 OpenVMS Alpha Boot from a CI Device

```
P00>>> sho dev cixcd0❶
polling for units on cixcd0, slot 4, bus 0, xmi0...
New node HSC011 HSC V835 HS70 cixcd_a.11.0.4.7 ❷
dua1.11.0.4.7      $1$DUA1 ()         RA92
dua2.11.0.4.7      $1$DUA2 ()         RA92
dua3.11.0.4.7      $1$DUA3 ()         RA92
dua4.11.0.4.7      $1$DUA4 ()         RA92
dua111.11.0.4.7    $1$DUA111 ()       RA70
dua112.11.0.4.7    $1$DUA112 ()       RA70
dua113.11.0.4.7    $1$DUA113 ()       RA70
dua114.11.0.4.7    $1$DUA114 ()       RA70
P00>>> b dua1.11.0.4.7 ❸
Building FRU table.........................
(boot dua1.11.0.4.7 -flags 0)
block 0 of dua1.11.0.4.7 is a valid boot block
reading 1018 blocks from dua1.11.0.4.7
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7f400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

     OpenVMS Alpha Operating System, Version 6.2 ❹
```

❶ **Show device cixcd0** displays device information for device cixcd0.

❷ Polling checks the I/O bus for device configurations. The next line contains three columns. The first column contains the device type and unit number, node number, device channel number, I/O node number, and I/O channel number, separated by periods. The second column displays the name of the device given by the device controller. The third column shows the device type.

❸ In the **boot** command, **dua1.11.0.4.7** is the device code of the boot device. The system boots.

❹ The operating system banner appears.

Example 5-7 Network Boot of LFU

```
P00>>> sho net demna0 ❶
polling for units on demna0, slot 3, bus 0, xmi0...
exa0.0.0.3.7: 08-00-2B-2A-76-44
P00>>> b -fi t40_lfu exa0.0.0.3.7 ❷
Building FRU table.........................
(boot exa0.0.0.3.7 -file t40_lfu -flags 0)

Trying MOP boot.
.....................................
Network load complete.
Host name: ACHE
Host address: aa-00-04-00-22-74
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 1eca00
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

          ***** Loadable Firmware Update Utility *****
------------------------------------------------------------------
Function     Description
------------------------------------------------------------------

Display   Displays the system's configuration table.
Exit      Done exit LFU (reset).
List      Lists the device, revision, firmware name, and update
                revision.
Modify    Modifies port parameters and device attributes.
Update    Replaces current firmware with loadable data image.
Verify    Compares loadable and hardware images.
? or Help Scrolls this function table.
------------------------------------------------------------------
UPD>      ❸
```

❶ **Show net demna0** displays information about the demna0 Ethernet controller. It includes information such as the console device name of the network device with path information (exa0.0.0.3.7) and the Ethernet controller's hardware address in hex (08-00-2B-2A-76-44).

❷ Boot LFU across the Ethernet by specifying command parameters **-fi**, the file name **t40_LFU**, and the console device name of the network device **exa0.0.0.3.7**.

❸ LFU starts, displays a summary of its commands, and issues its prompt (UPD>).

5.3 Booting Digital UNIX

Use the show device command to locate the device from which to boot the operating system. Boot from the device.

Example 5-8 Digital UNIX Single-Ended Boot

```
P08>>> sho dev                                  ❶
polling for units on isp0, slot 0, bus 0, hose0...
dka200.2.0.0.0     DKA200             RZ26L      440C
dka400.4.0.0.0     DKA400             RZ26L      440C
polling for units on isp1, slot 1, bus 0, hose0...
dkb300.3.0.1.0     DKB300             RZ26L      440C
dkb500.5.0.1.0     DKB500             RZ26L      440C
dkb600.6.0.1.0     DKB600             RZ26L      440C
polling for units on isp2, slot 4, bus 0, hose0...
dkc200.2.0.4.0     DKC200             RZ26L      440C
dkc400.4.0.4.0     DKC400             RZ26L      440C
polling for units on isp3, slot 5, bus 0, hose0...
dkd0.0.0.5.0       DKD0               RZ26L      440C
dkd400.4.0.5.0     DKD400             RRD43      0064
dkd500.5.0.5.0     DKD500             RZ26L      440C
polling for units on isp4, slot 0, bus 0, hose1...
dke0.0.0.0.1       DKE0               RZ26L      440C
dke200.2.0.0.1     DKE200             RZ26L      440C
polling for units on isp5, slot 5, bus 0, hose1...
dkf100.1.0.5.1     DKF100             RZ26L      440C
polling for units on kzpsa0, slot 8, bus 0, hose1...
kzpsa0.7.0.8.1 dkg TPwr 1 Fast 1 Bus ID 7 C01 A02
dkg300.3.0.8.1     DKG300             RZ26L      440C
dkg500.5.0.8.1     DKG500             RZ26L      440C
dkg600.6.0.8.1     DKG600             RZ26L      440C
P08>>> sho boot*                                 ❷
boot_dev              dkd0.0.0.5.0
boot_file
boot_osflags          a
boot_reset            OFF
bootdef_dev           dkd0.0.0.5.0
booted_dev
booted_file
booted_osflags
```

Example 5-6 Digital UNIX Single-Ended Boot (Continued)

```
P08>>> b                                          ❸
Building FRU table.................
(boot dkd0.0.0.5.0 -flags a)
SRM boot identifier: scsi 0 5 0 0 0 ef00 10201077
boot adapter: isp3 rev 1 in bus slot 5
              off of kftia0 in TLSB slot 8
block 0 of dkd0.0.0.5.0 is a valid boot block
reading 16 blocks from dkd0.0.0.5.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
               .
               .
               .
   Digital UNIX Version 3.2B       ❹
```

❶ **Show device** displays information about each I/O device. Polling checks the I/O bus for devices connected to <device>. The <device> is the name the console assigns to an I/O adapter or device in the system (such as kzmsa0). A list of installed adapters is given in the **show configuratio**n command display. Refer to Section 7.4.23 for details on the **show configuration** command.

❷ The **show boot*** command lists the boot environment variables and their values (see Section 7.3 for a list of environment variables and their definitions). The default boot device is **dkd0** as identified in **bootdef_dev**.

❸ Booting can now be done from the device by typing **boot** or just **b**.

❹ The operating system banner appears.

Example 5-9 Digital UNIX FWD SCSI Boot

```
P10>>> sho dev                  ❶
polling for units on isp0, slot 0, bus 0, hose0...   ❷
dka100.1.0.0.0           DKA100         RZ28     D41C
dka200.2.0.0.0           DKA200         RZ26L    440C
dka300.3.0.0.0           DKA300         RZ26L    440C
dka400.4.0.0.0           DKA400         RZ26L    440C
polling for units on isp1, slot 1, bus 0, hose0...
dkb100.1.0.1.0           DKB100         RZ26L    440C
dkb200.2.0.1.0           DKB200         RZ26L    440C
dkb300.3.0.1.0           DKB300         RZ26L    440C
dkb400.4.0.1.0           DKB400         RZ26L    440C
dkb500.5.0.1.0           DKB500         RZ26L    440C
dkb600.6.0.1.0           DKB600         RZ26L    440C
polling for units on isp2, slot 4, bus 0, hose0...
dkc100.1.0.4.0           DKC100         RZ26L    440C
dkc200.2.0.4.0           DKC200         RZ26L    440C
dkc300.3.0.4.0           DKC300         RZ26L    440C
dkc400.4.0.4.0           DKC400         RZ26L    440C
dkc500.5.0.4.0           DKC500         RZ26L    440C
dkc600.6.0.4.0           DKC600         RZ26L    440C
polling for units on isp3, slot 5, bus 0, hose0...
dkd0.0.0.5.0             DKD0           RZ28     D41C
dkd300.3.0.5.0           DKD300         RZ26L    440C
dkd400.4.0.5.0           DKD400         RRD44    0064
dkd500.5.0.5.0           DKD500         RZ26     T392
```

For more information:

Digital UNIX Guide to System Administration

Example 5-7 Digital UNIX FWD SCSI Boot (Continued)

```
P10>>> b dka100 -fl a ❸
(boot dka100.1.0.0.0 -flags a)
SRM boot identifier: scsi 0 0 0 1 100 ef00 10201077
boot adapter: isp0 rev 1 in bus slot 0
               off of kftia0 in TLSB slot 8
block 0 of dka100.1.0.0.0 is a valid boot block
reading 16 blocks from dka100.1.0.0.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
               .
               .
               .
   Digital UNIX Version 3.2B       ❹
```

❶ **Show device** displays information about each I/O device. Polling checks the I/O bus for devices connected to <device>. The <device> is the name the console assigns to an I/O adapter or device in the system (such as kzmsa0). A list of installed adapters is given in the **show configuratio**n command display. Refer to Section 7.5.23 for details on the **show configuration** command.

❷ Polling checks the I/O bus for device configurations. The next line contains four columns. The first column contains the device type and unit number, node number, device channel number, I/O node number, and I/O channel number, separated by periods. The second column displays the name of the device given by the device controller. The third column shows the device type. The fourth column reports the firmware revision on the device (not all report this).

❸ In the **boot** command, **dk** is the device code of the boot device, **a** is the boot device controller designation, **100** specifies the unit number of the boot device, **-fl** indicates additional command options follow, and **a** indicates the system disk is booted to multiuser mode.

❹ The operating system banner appears.

Chapter 6

System Troubleshooting

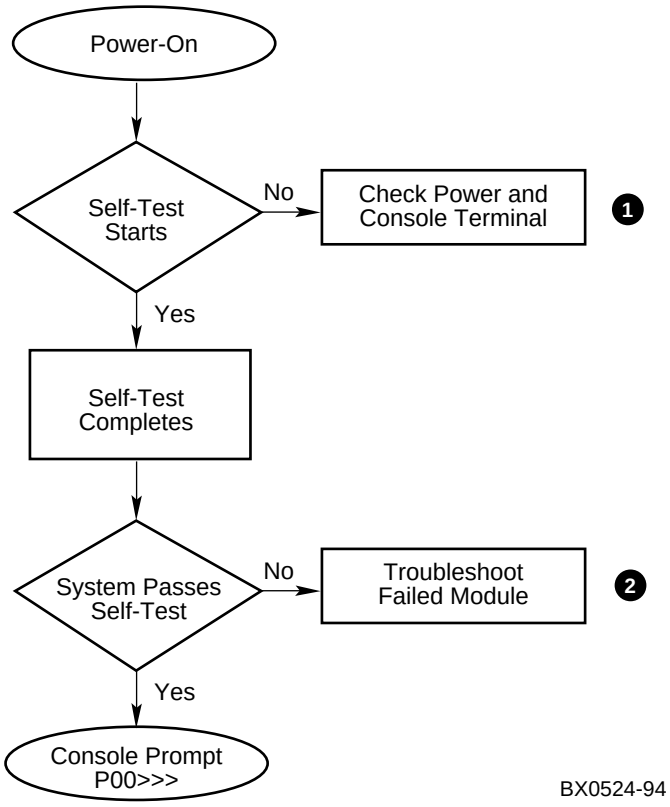
This chapter describes system problems and their symptoms during power-up, booting, and normal operation. Advanced troubleshooting is discussed in the *Service Manual*. Topics discussed include:

- 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 Lines ILV and MB
- Self-Test Identification Line
- Show Commands
- Test Command
- Error Reports

6.1 Troubleshooting During Power-Up

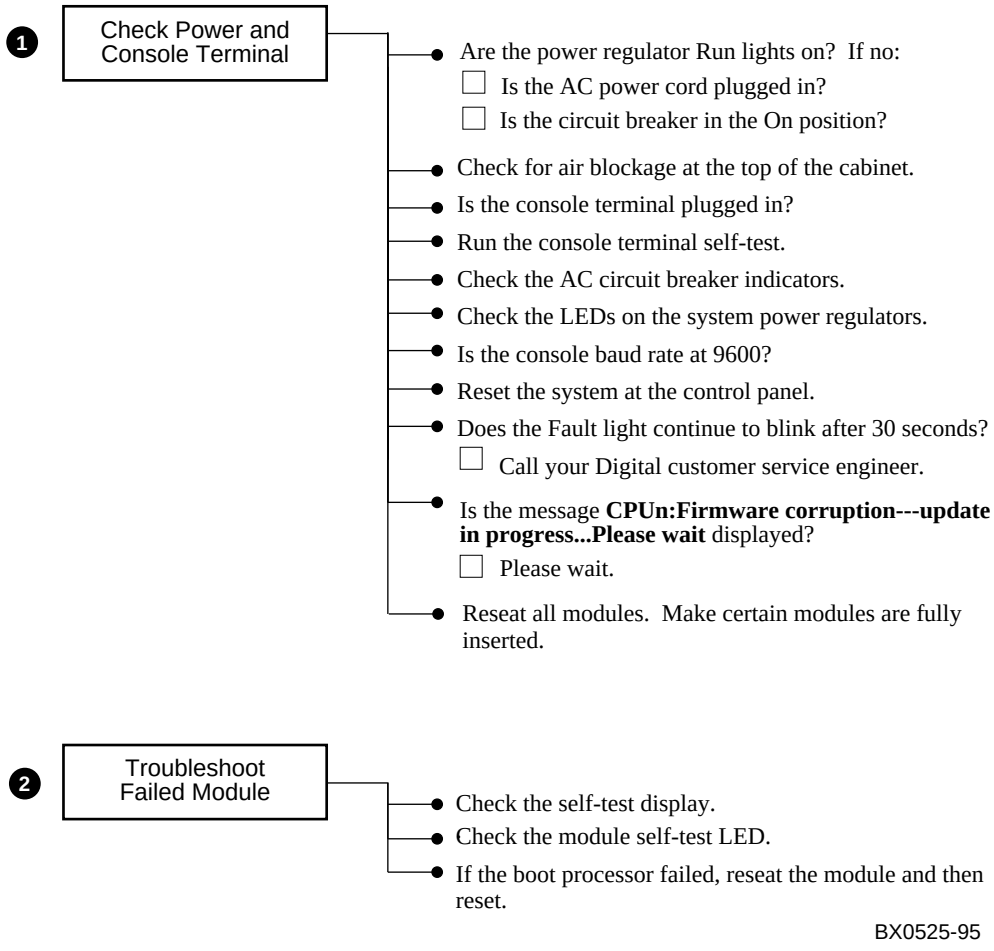
Figure 6-1 shows the power-up sequence and steps to take if a problem occurs.

Figure 6-1 Power-Up Troubleshooting Flowchart



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

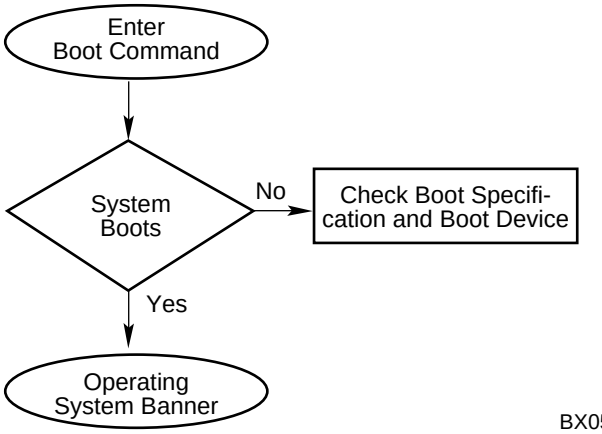
Figure 6-2 Power-Up Troubleshooting Steps



6.2 Troubleshooting During Booting

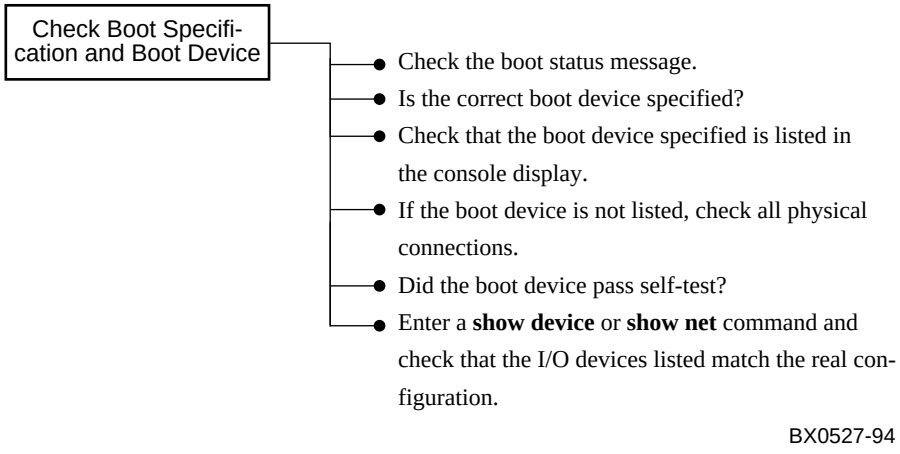
When booting fails, you can check several parameters. Figure 6-3 shows the boot sequence. Figure 6-4 shows the steps to take if a problem occurs during booting. If you are unable to correct the problem, call your Digital customer service engineer.

Figure 6-3 Booting Troubleshooting Flowchart



BX0526-94

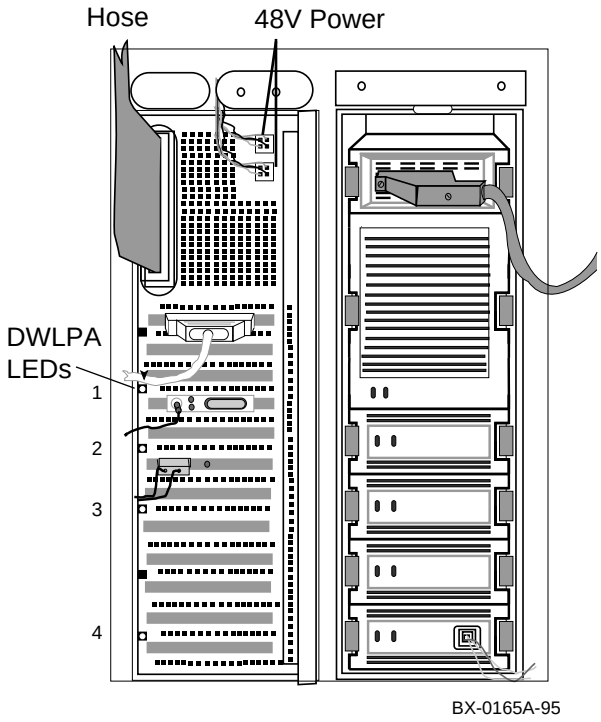
Figure 6-4 Troubleshooting Steps During Booting



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

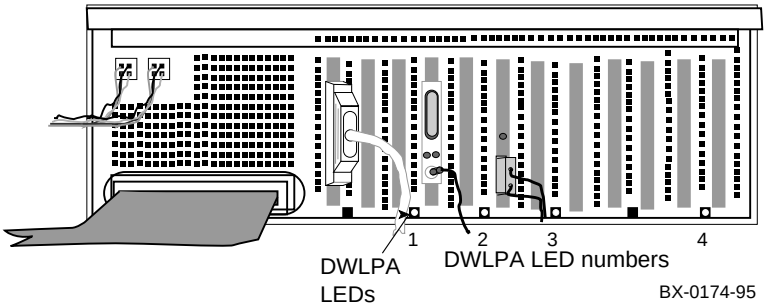
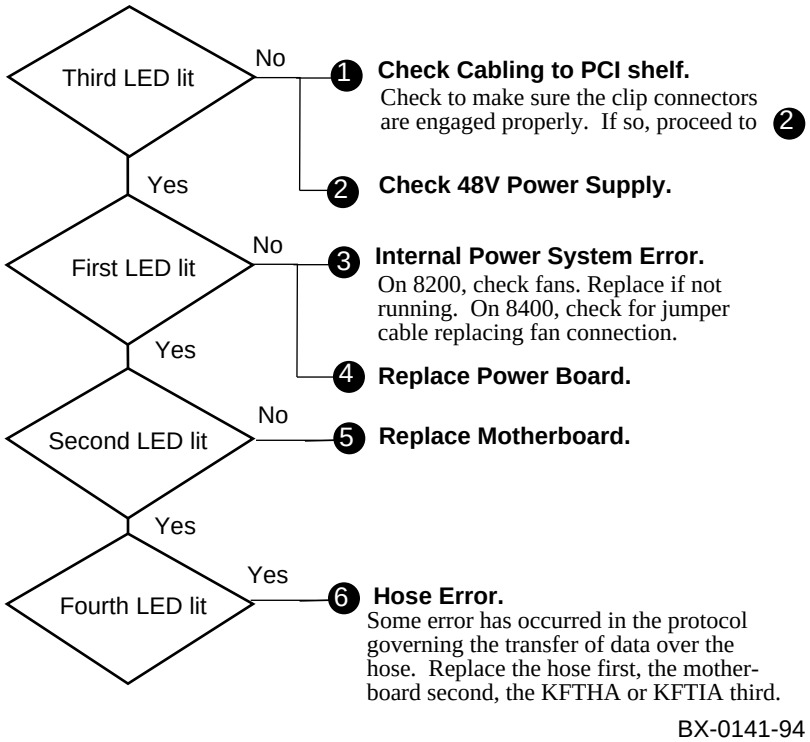


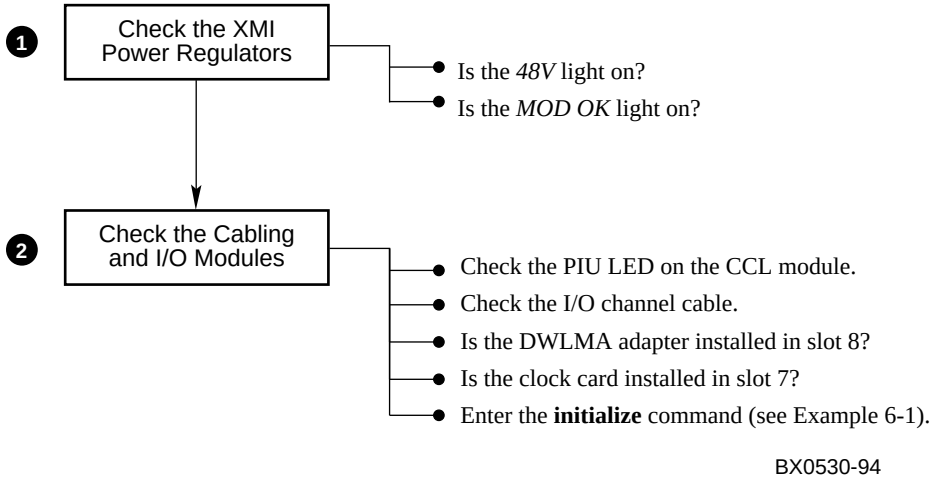
Figure 6-7 Troubleshooting Steps for Power in PCI Shelf



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
                      +  +  .  .  .  .  .  + ++  ST3
                      .  .  .  .  .  .  .  . EB  BPD

    .   .   .   .   .  .  .  .  .  .  .  .  .  .     C0 XMI- (2)
.   .   .   .   .   .  .  .  .  .  .  .  .  .  .  .  C1
.   .   .   .   .   .  .  .  .  .  .  .  .  .  .  .  C2
                .   .  .  .  .  .  .  .  .  .  .  +  C3 PCI +
                      .  .  .  .  .  .  +  +     EISA +

                      . B0  .  .  .  .  . A0  .  ILV
                      .128  .  .  .  .  .256  .  384MB
AlphaServer 8400 Console V1.0, SROM V1.0, Apr 2 1995 00:00
P00>>>
```

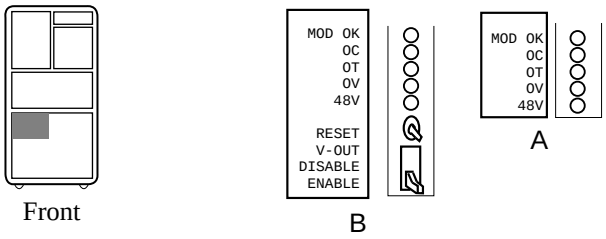
(1) In Example 6-1 the KFTHA module passes self-test, as indicated by the plus sign **(+)** at node 8 on self-test line ST3.

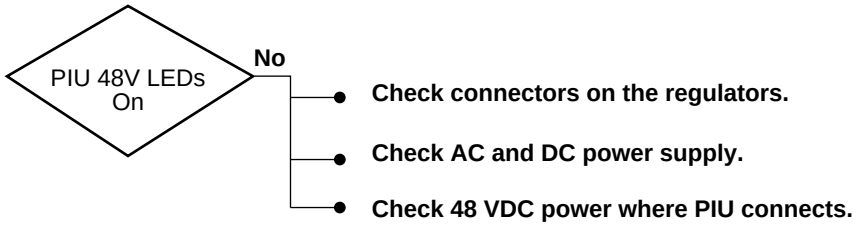
(2) The DWLMA adapter fails self-test, as indicated by the minus sign **(−)** displayed on the **C0 XMI** line. When a DWLMA fails self-test, the failure is reported, and the results of the I/O adapter self-tests are not displayed.

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.

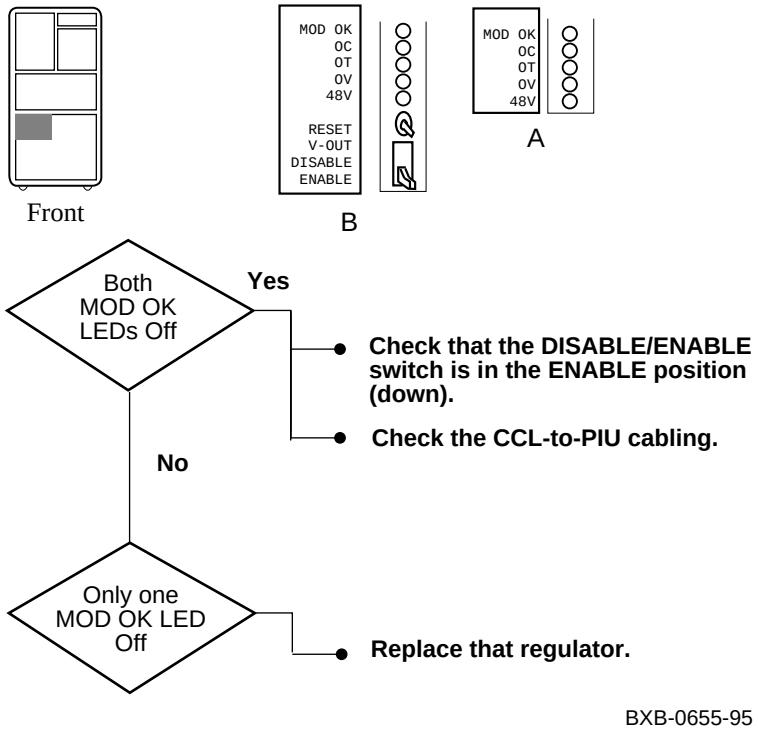
Figure 6-9 FBUS+ PIU Troubleshooting - 48V LED Off





BXB-0654-95

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

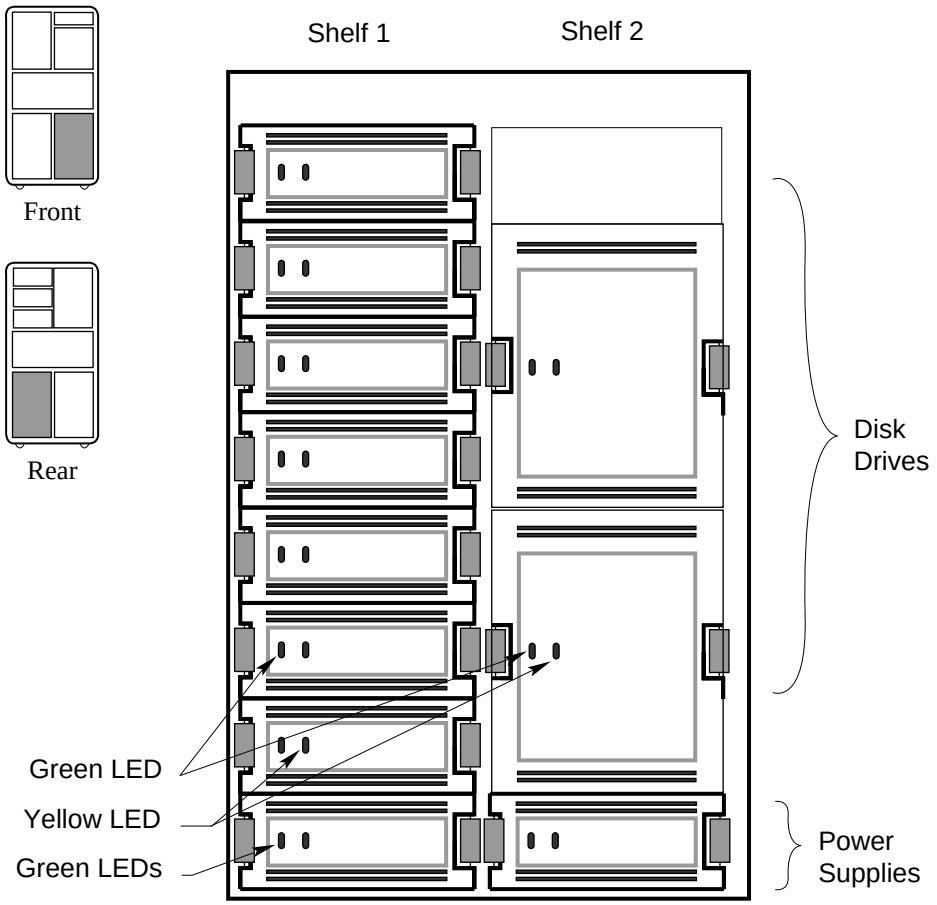


Regulator A and B in the FBUS+ PIU are dependent on each other. For the MOD OK LEDs to be on, the following sequence of power applies: 3 volts from A, followed by 5 volts from B, followed by 2 volts from A. Therefore, if any one of the voltages fails, both regulators will turn the MOD OK LED off.

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



BX0529-94

Table 6-1 SCSI Disk Drive LEDs

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

Table 6-2 SCSI Power Supply LEDs

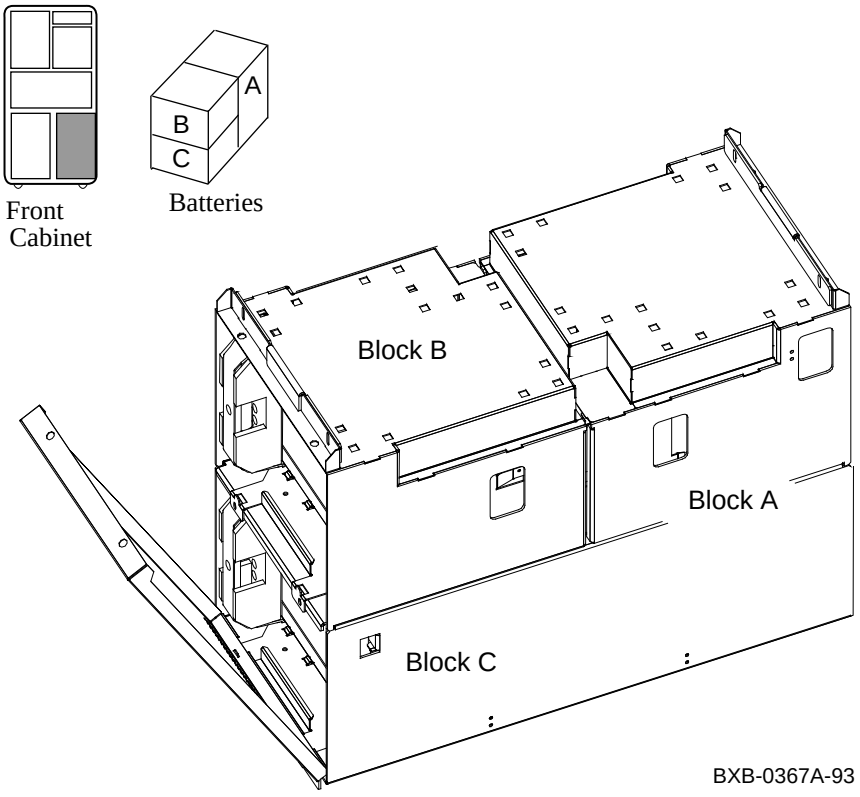
Indicator LED	LED State	Meaning
Green (left)	Off	Shelf fault
	On	Shelf OK
Green (right)	Off	Power fault
	On	Power OK

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 power regulator. The battery PIU is installed in the bottom of the system or expander cabinet (8400 with three-phase power 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
LDC POWER Status     : OK  ❷
PIU Primary Status   : OK
PIU Secondary Status : OK
```

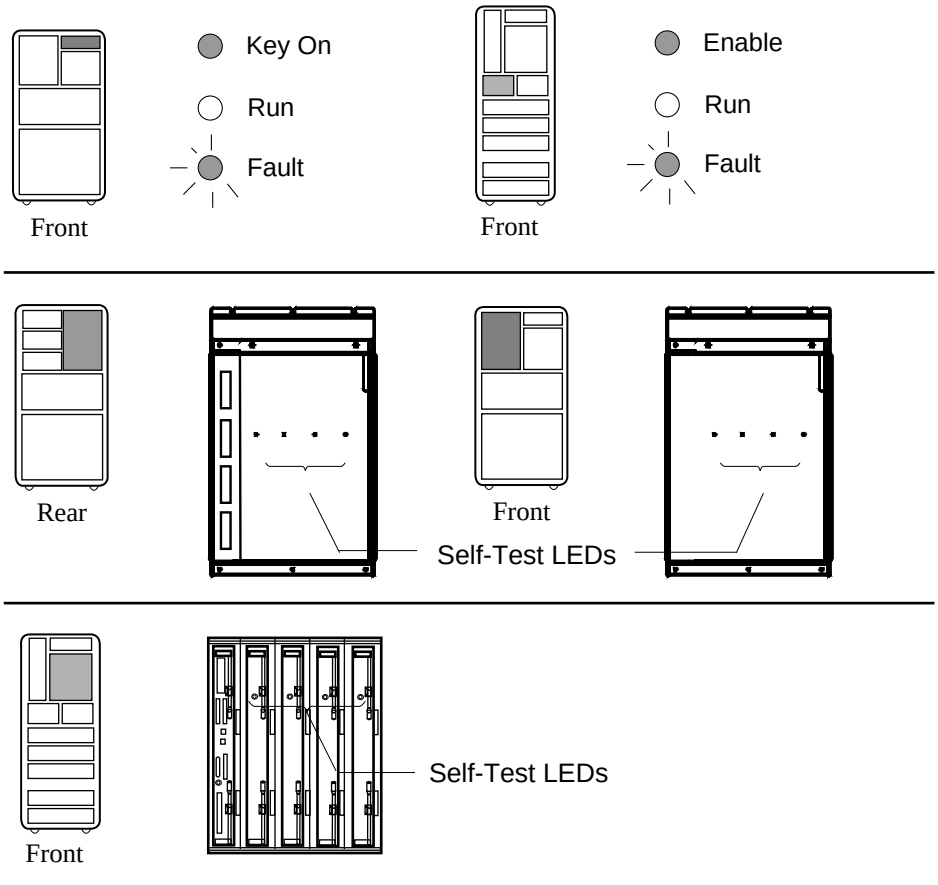
❶ The user enters a **show power** command (Example 6-2). The main cabinet has three power regulators.

❷ The bottom three lines of the output, showing PIU power status, are printed for the main cabinet only.

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 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
                   .  .  .  .  .  .  .  .E  EB  BPD (1)
                   o  +  .  .  .  .  +  .+  +-  ST2 (2)
                   .  .  .  .  .  .  .  .E  BE  BPD
                   +  +  .  .  .  .  +  .+  +-  ST3
                   .  .  .  .  .  .  .  .E  BE BPD  (3)
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .      C0 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .      C1
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .      C2
            .  .  .  .  .  .  .  .  .  .  .  .  +   C3 PCI +
                   .  .  .  .  .  .  .  +  +         EISA +
                   . B0  .  .  .  . A0  .  .   ILV
                   .128  .  .  .  .256  .  .   384MB
AlphaServer 8400 Console V1.0,SROM V1.0, Apr 2 1995 00:00
-------------------------------------------------------
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 V1.0,SROM V1.0, Apr 2 1995 00: (4)
```

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 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. 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 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
                   .  .  .  .  .  .  .  .E  EB BPD    ❹
                   o  o  +  .  .  .  +  .+  +- ST2    ❺
                   .  .  .  .  .  .  .  .E  BE BPD    ❻
                   +  +  +  .  .  .  +  .+  +- ST3    ❼
                   .  .  .  .  .  .  .  .E  BE BPD    ❽
                      +  +  +  +  .  +  +  +  C0 PCI +❾
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-❿
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -⓫
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5      ⓬
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6      ⓬
            .  .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +⓭
                   .  .  .  .  .  .  .  +  +        EISA+⓮

                   .  . B0  .  .  . A0  .  .  ILV    ⓯
                   .  .128  .  .  .256  .  .  384MB  ⓰
AlphaServer 8400 Console V1.0,SROM V1.0, Apr 2 1995 00:00

      ⓱                          ⓲                    ⓳

P01>>>  ⓴
```

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
KN7CC-AB	0	Dual processor; CPU0 fails ST2 and ST3 testing. CPU1 becomes boot processor.
KN7CC-AA	1	Single 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.
DEFEA	1	EISA to FDDI option; passes self-test.
KFESB	2	EISA to DSSI option; 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
                   .  .  .  .  .  .  .  E  EB BPD
                   o  +  +  .  .  .  +  +  +- ST2
                   .  .  .  .  .  .  .  E  BE BPD
                   +  +  +  .  .  .  +  +  +- ST3
                   .  .  .  .  .  .  .  E  BE BPD

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

                   .  . B0  .  .  . A0  .  .  ILV
                   .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V1.0,SROM V1.0, Apr 2 1995 00:00
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)

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
                      .  .  .  .  .  .  .  E  EB BPD   ❹
                      o  o  +  .  .  .  +  +  +- ST2   ❺
                      .  .  .  .  .  .  .  E  BE BPD   ❻
                      +  +  +  .  .  .  +  +  +- ST3   ❼
                      .  .  .  .  .  .  .  E  BE BPD   ❽
                         +  +  +  +  .  +  +  +  C0 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6
            .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +
                      .  .  .  .  .  .  +  +        EISA+
                      .  . B0  .  .  . A0  .  .  ILV
                      .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V1.0,SROM V1.0, Apr 2 1995 00:00
P01>>>
```

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

- .+ (uniprocessor)
- + (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 to become 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 multiprocessor 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
                    .  .  .  .  .  .  .  E  EB BPD
                    o  +  +  .  .  .  +  +  +- ST2
                    .  .  .  .  .  .  .  E  BE BPD
                    +  +  +  .  .  .  +  +  +- ST3
                    .  .  .  .  .  .  .  E  BE BPD
                       +  +  +  +  .  +  +  +  C0 PCI +(9)
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-(10)
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -(11)
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5       (12)
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6       (12)
            .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +(13)
                    .  .  .  .  .  .  +  +        EISA+(14)
                    .  . B0  .  .  . A0  .  .  ILV
                    .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V1.0,SROM V1.0, Apr 2 1995 00:00
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 C4) 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−).

⓫ The DWLMA adapter accessed through C4 failed self-test (C0 XMI −).

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

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

⓮ 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 7.4.23).

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
                   .  .  .  .  .  .  .  E EB  BPD
                   o  +  +  .  .  .  +  + +-  ST2
                   .  .  .  .  .  .  .  E BE  BPD
                   +  +  +  .  .  .  +  + +-  ST3
                   .  .  .  .  .  .  .  E BE  BPD
                      +  +  +  +  .  +  +  +  C0 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6
            .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +
                   .  .  .  .  .  .  +  +        EISA+
                   . B0  .  .  .  . A0  .  .  ILV   (15)
                   .128  .  .  .  .256  .  .  384MB (16)
AlphaServer 8400 Console V1.0,SROM V1.0, Apr 2 1995 00:00
P01>>>
```

⓯ 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.

⓰ 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.

Figure 6-14 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
                   .  .  .  .  .  .  .  E EB  BPD
                   o  o  +  .  .  .  +  + +-  ST2
                   .  .  .  .  .  .  .  E BE  BPD
                   +  +  +  .  .  .  +  + +-  ST3
                   .  .  .  .  .  .  .  E BE  BPD

                      +  +  +  +  .  +  +  +  C0 PCI +
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C1 FBUS-

   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .     C4 XMI -
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C5
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  C6
            .  .  .  .  .  .  .  .  .  .  .  .  +  C7 PCI +
                   .  .  .  .  .  .  +  +        EISA+

                   .  . B0  .  .  . A0  .  .  ILV
                   .  .128  .  .  .256  .  .  384MB
AlphaServer 8400 Console V1.0,SROM V1.0, Apr 2 1995 00:00
       ❶❼                              ❶❽                ❶❾

P01>>> ❷⓿
```

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

⓲ 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).

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

⓴ The console prompt indicates the number of the primary processor. In this example used to describe the self-test results, there are three CPUs, two on the module in node 0, and one on the module in node 1. Therefore, possible console prompts that could appear are **P00>>>**, **P01>>>**, and **P02>>>**. 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, enter a show configuration, show network, or show device command. 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 6-9 Sample System Hardware Configuration

```
P00>>> show configuration ❶
      Name                Type      Rev     Mnemonic ❷
  TLSB
  0++ KN7CC-AB            8014      0000    kn7cc-ab0
  1+  KN7CC-AA            8011      0000    kn7cc-aa0
  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+  ISP1020          10201077     0001    isp4
8+  ISP1020          10201077     0001    isp5
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
4+ KFESB               2E7310     0000    kfesb0
```

❶ In Example 6-9 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 one PCI; in the mnemonic column they are identified as **xmi0** and **pci0**. To test the xmi0 subsystem, you would issue the command **test xmi0** (see Section 6.17.2).

❹ To test an I/O adapter, such as the Ethernet adapter, you would enter the command **test tulip1** (see Section 6.17.3).

6.16.2 Show Network

Enter a show network command to display network devices.

Example 6-10 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 6-10 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 6-11 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 6-11 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 6-12 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 is
                           # 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, enter the test command.

Example 6-13 Sample Test Command, System Test

```
P00>>> test -t 180 ❶
Configuring system...
Testing system...
Type Ctrl/C to abort ❷
Starting memory exerciser, running on kn7cc0 (id #59)❸
Starting memory exerciser, running on kn7cc1 (id #60)
Starting network exerciser on exa0.0.0.4.0 in external 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 6-13:

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 6-14 Sample Test Command, I/O Subsystem Test

```
P00>>> test xmi0 ❶
Configuring xmi0 subsystem...
Testing xmi0
Type Ctrl/C to abort
kn7cc0 running module tests on DWLMA0
DWLMA0 module tests passed 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 6-15 Sample Test Command, I/O Adapter Test

```
P00>>> test demna0      ❶
Configuring demna0...
Testing demna0
Type Ctrl/C to abort

Initializing DEMNA0 ❷
DEMNA0 self-test passed
Starting network exerciser on exa0.0.0.4.0 in internal mode (id #30) ❸

Stopping network exerciser on exa0.0.0.4.0 (id #30)

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 6-16 Sample Test Command, Memory Module Test

```
P00>>> test ms7cc0 ❶
Testing ms7cc0
Type Ctrl/C to abort
Starting memory exerciser, running on kn7cc0 (id #77) ❷
Stopping memory exerciser on kn7cc0 (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 kn7cc0 processor.

Example 6-17 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)
```

Example 6-18 Sample Test Command, Testing Devices (Continued)

```
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.

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 6-19 and Example 6-20.

Example 6-18 Sample Summary Error Report

```
P00>>> set d_report summary  # Command to set the type of error
                             # report to summary (default value).
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 6-19:

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 6-19 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.

Configuring demna0...
Testing demna0
Type Ctrl/C to abort

Initializing demna0
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 6-19, the information shown in a full and a summary error report is the same. See Example 6-18 for a description of the common error report entries.

In Example 6-19:

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

Console Commands

This chapter describes the console program's command language, console special characters, console environment variables, and console commands. Console commands allow you to boot the operating system, display the configuration, and verify the system.

When the system is in console mode, the system is halted and the 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 11 (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.

Sections in this chapter include:

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

7.1 Command Syntax

The 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 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 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 5.2) 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.

7.2 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 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 console mode, acts like Ctrl/C. In program mode, causes the boot processor to halt and begin running the console program.
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 console mode. 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 console program. See the **continue** command for additional information.

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.

7.3 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. Certain environment variables are typically modified by the user to tailor the recovery behavior of the system on power-up and after system failures. Volatile environment variables are initialized by a system reset; others are nonvolatile across system failures.

Environment variables can be created, modified, displayed, and deleted using the **create**, **set**, **show**, and **clear** commands. 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	Non-volatile	Enables the console ARC interface, allowing booting of the ECU and other ARC utilities. Default value is **off**.
auto_action	Non-volatile	Specifies the action the console will take following an error halt or power-up. Values are: **restart** - Automatically restart. If restart fails, boot the operating system. **boot** - Automatically boot the operating system. **halt** (default) - Enter console mode.
bootdef_dev	Non-volatile	The default device or device list from which booting is attempted when no device name is specified by the **boot** command.

Table 7-3 Environment Variables (Continued)

Variable	Attribute	Function
boot_file	Non-volatile	The default file name used for the primary bootstrap when no file name is specified by the **boot** command, if appropriate.
boot_osflags	Non-volatile	Additional parameters to be passed to the system software during booting if none are specified by the **boot** command with the **-flags** qualifier.
boot_reset	Non-volatile	Resets system and displays self-test results during booting. Default value is **off**.
cpu	Volatile	Selects the current boot processor.
cpu_enabled	Non-volatile	A bitmask indicating which processors are enabled to run (leave console mode). Default is **0xffff**.
cpu_primary	Non-volatile	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 the **test** command.
d_report	Volatile	Determines level of information provided by the diagnostic reports. Values are **summary** and **full** (default). Applies only when using the **test** command.
d_softerr	Volatile	Determines action taken following a soft error. Values are **continue** (default) and **halt**. Applies only when using the **test** command.

Table 7-3 Environment Variables [Continued)

Variable	Attribute	Function
enable_audit	Non-volatile	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**.
tta0_baud	Non-volatile	Sets the console terminal port baud rate. Allowable values are 300, 600, 1200, 2400, 4800, and 9600. The default value is 9600.

7.4 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.

The following console commands are described:

- boot
- build (-e, -n, -s)
- clear (eeprom, <envar>, screen)
- continue
- crash
- create
- date
- deposit
- examine
- help or man
- initialize
- run
- runecu
- set (eeprom, <envar>, host, power, seeprom)
- show (configuration, cpu, device, eeprom, <envar>, memory, network, power, seeprom)
- start
- stop
- test
- comment (#)

7.4.1 Boot

The boot command boots the operating system.

Example 7-1 Boot Command

```
P08>>>               # Boot from a PCI device.
P08>>> sh dev        # Display I/O device information.
polling for units on isp0, slot 0, bus 0, hose0...
polling for units on isp1, slot 1, bus 0, hose0...
polling for units on isp2, slot 4, bus 0, hose0...
dkd400.4.0.5.0        DKD400             RRD44  0064
dkd500.5.0.5.0        DKD500             RZ26L  440C

P08>>> b -fl -i dkd500 # -fl[ags] indicates additional
                       # command options follow.
                       # -i = flag for booting OSF.
                       # dk = device code.
                       # d = controller designation.
                       # 500 = disk device unit
                       # number.
```

The **boot** command syntax is:

b[oot] [-flags M, PPPP] [-file <filename>] <device_name>

where the **-flags** parameter allows additional **boot** command parameters **M** and **PPPP**. Specifying **-fl[ags]** overrides the **boot_osflags** environment variable (see Section 7.3). The **M** flag, dependent on the system configuration, specifies the system root to be booted from the system disk. The **PPPP** flags are for the operating system bootstrap loader options. The **-file** parameter indicates booting from the file **<filename>**. Specifying **-file** overrides the **boot_file** environment variable (see Section 7.3). Device names can be found by using the **show device** and **show network** commands.

Boot command flags can be shortened, since values such as zero or commas (which can be used as placeholders) do not have to be specified. These parameters are read from right to left (**PPPP**, **M**). For example, **boot -fl 0,100** or **boot -fl ,100** are the same as **boot -fl 100**, where **100** is the value of the **PPPP** option.

7.4.2 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 7-2 Building the EEPROM

```
P00>>> build -e              # Initialize the EEPROM
                             # on kn7cc.
Build EEPROM on kn7cc-ab0? [Y/N]> Y
          EEPROM built on kn7cc-ab0
P00>>>
```

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

bu[ild] -e <device>

where **<device>** is KN7CC-AB or KN7CC-AA. 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 (see Section 7.3), before upgrading you should refer to Table 7-3 and use the **show <envar>** command (see Section 7.4.25) to display present environment variable values. After rebuilding, use the **set <envar>** command (see Section 7.4.18) to set the environment variables to their desired values.

7.4.3 Building the Nonvolatile RAM

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

Example 7-3 Building the Nonvolatile RAM

```
P00>>> build -n

Build non-volatile RAM on kn7cc-ab0? [Y/N]Y

Creating new non-volatile RAM image
P00>>>
```

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

bu[ild] -n <device>

where **<device>** is KN7CC-AA or KN7CC-AB. The **build -n** command sets the NVR to its default values. The **build -e** command may be required during a console firmware upgrade.

7.4.4 Building the SEEPROM

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

Example 7-4 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. 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 installation, all module serial EEPROMs should be initialized (using **build -s ***). In the case of a module upgrade, the serial EEPROM should be initialized also.

7.4.5 Clear EEPROM

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

Example 7-5 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**, **symptom**, or **log**.

7.4.6 Clear <envar>

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

Example 7-6 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).

7.4.7 Clear Screen

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

Example 7-7 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.

7.4.8 Continue

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 7-8 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.

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.*

7.4.9 Crash

The crash command causes the operating system to be restarted and generates a memory dump.

Example 7-9 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.

7.4.10 Create

The create command allows you to create an environment variable.

Example 7-10 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).

7.4.11 Date

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

Example 7-11 Date Command

```
P02>>> date
12:00:01 PM March 1, 1995    # System displays time
                             # and date.
P02>>> date 199503051200      # Sets date and time to
                             # 12:00 PM March 5, 1995.
```

The **date** command syntax is:

da[te] [<yyyymmddhhmm.ss>]

where **yyyy** is for the year, **mm** for the month, **dd** for the day, **hh** for the hour, **mm** for minutes, and **ss** for seconds.

7.4.12 Deposit

The deposit command stores data in a specified location.

Example 7-12 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:

Alpha Architecture Reference Manual

7.4.13 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 7-13 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.

7.4.14 Help or man

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

Example 7-14 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 run
   run: No such command     # The arc_enable environment
   P02>>> set arc_enable on # variable must be set to on
   P02>>> init              # and the system initialized
   Initializing...          # before the run and runecu
   (self-test display)      # commands are recognized.
            :
            :
   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.

7.4.15 Initialize

The initialize command performs a reset.

Example 7-15 Initialize Command

```
P00>>> initialize
```

The **initialize** command syntax is:

i[nitialize]

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

7.4.16 Run

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 7-16 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 V0.1,SROM 3.0, Apr 2,1995

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                    RRD44  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 EEROM Configuration Utility v1.1
------------------------------------------------------------

         +============ Select ISP1020 ============+
         |  Bus      Virtual Slot      Address    |
         |                                        |
         |[  0             0           01003000]  |
         |   0             1           01002000   |
         |   0             4           01001000   |
         |   0             5           01000000   |
         |----------------------------------------|
         |     < (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 EEPROM 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"
```

7.4.17 Runecu

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.

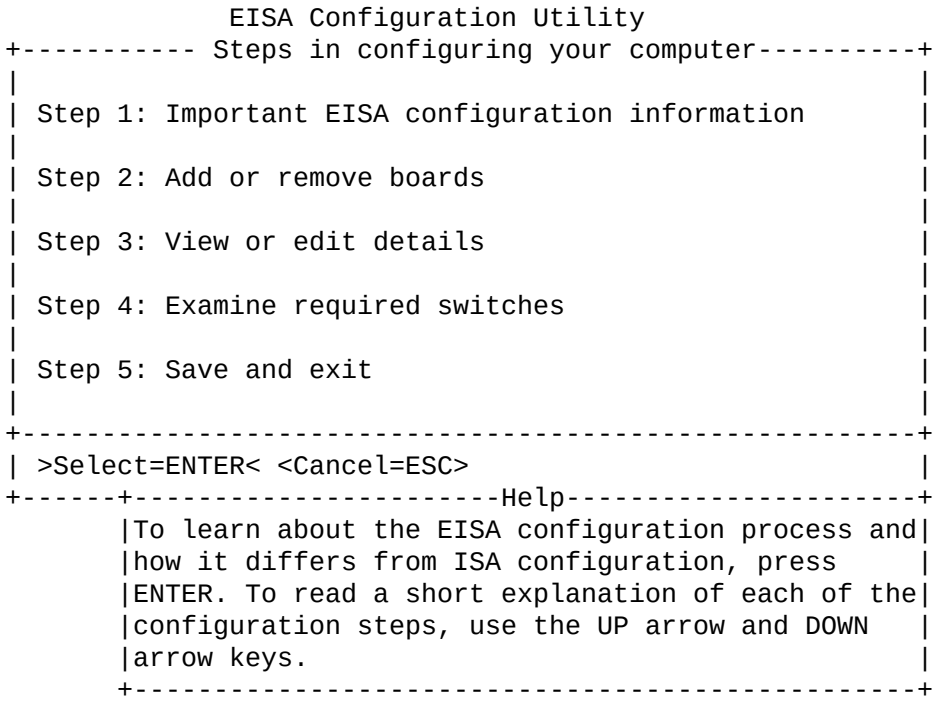
Example 7-17 Runecu Command

```
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

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.

7.4.18 Set EEPROM

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

Example 7-18 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 7-19) for the CPU you would like to act on.

7.4.19 Set <envar>

Set <envar> allows you to modify environment variables.

Example 7-19 Set <envar>

```
1. P00>>> set auto_action restart # On an error halt,
                                  # system will automatically
                                  # restart. If restart
                                  # fails, boot the operating
                                  # 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 7.4.10. 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.

7.4.20 Set Host

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

Example 7-20 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 the first example in Example 7-20. 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.

7.4.21 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 7-21 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**).

7.4.22 Set SEEPROM

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

Example 7-22 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**.

7.4.23 Show Configuration

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

Example 7-23 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+  ISP1020            10201077    0001   isp0
  8+  ISP1020            10201077    0001   isp1
  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
  4+  KFESB                 2EA310    0000  kfesb0
```

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]

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

7.4.24 Show CPU

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

Example 7-24 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 7-24, CPU 11 failed self-test).

7.4.25 Show Device

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

Example 7-25 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.

7.4.26 Show EEPROM

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

Example 7-26 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 7-19) for the CPU you would like to act on.

7.4.27 Show <envar>

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

Example 7-27 Show <envar>

```
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.

7.4.28 Show Memory

The show memory command displays memory module information.

Example 7-28 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.

7.4.29 Show Network

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

Example 7-29 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.

7.4.30 Show Power

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

Example 7-30 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
LDC POWER Status     : OK
PIU Primary Status   : OK
PIU Secondary Status : OK
```

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**).

7.4.31 Show SEEPROM

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

Example 7-31 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**, **kftia**.

7.4.32 Start

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 7-32 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.*

7.4.33 Stop

The stop command halts a specified processor.

Example 7-33 Stop Command

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

The **stop** command syntax is:

sto[p] <processor_number>

where **<processor_number>** is the logical CPU number displayed using the **show cpu** command. The console's method of numbering CPUs is identical to that of the operating system. The **stop** command does not control the running of diagnostics and does not apply to adapters or memories.

7.4.34 Test

The test command allows you to test the entire system, a portion of the system (subsystem), or a specific device. By default, the entire system is tested.

Example 7-34 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
                           # provided that testing does not
                           # exceed 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**"**][-omit** "**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 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.
-omit "**list**"	Specifies device not to test; takes a single device or device list as a qualifier.
-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.

7.4.35 Comment (#)

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

Example 7-35 Comment (#) Command

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

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

Appendix A

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-3 lists AlphaServer 8200/8400 system devices and how they are displayed by the console.

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 Alpha 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 OpenVMS Alpha options are used with the **set boot_osflags** command.

Table A-3 AlphaServer 8200/8400 Devices

In Console Display as	Device	Description
KN7CC-AA (kn7cc-aa*n)* KN7CC-AB (kn7cc-ab*n)*	KN7CC	CPU module with one DECchip 21164 chip CPU module with two DECchip 21164 chips
MS7CC (ms7cc*n*)	MS7CC	Memory module
KFTHA (kftha*n*)	KFTHA	I/O port module (supports four hoses)
KFTIA(kftia*n*) DECchip 21040-AA (tulip*n*) ISP1020 (isp*n*)	KFTIA	Integrated I/O port module (supports one hose)
DEC PCI FDDI (pfi*n*)	DEFPZ-AA	Optional FDDI card on KFTIA, single-attachment station
DEC PCI FDDI (pfi*n*)	DEFPZ-UA	Optional FDDI card on KFTIA, twisted-pair copper
SIO (sio*n*)	KFE70	EISA standard I/O module and connector module
KFESB (kfesb*n*)	KFESB-AA	EISA DSSI controller (requires KFE70)
DEMNA (demna*n*)	DEC LAN-controller 400	XMI to Ethernet controller

Table A-3 AlphaServer 8200/8400 Devices (Continued)

In Console Display as	Device	Description
DECchip 21040-AA (tulip*n*)	EtherWORKS *Turbo*PCI Ethernet Controller (DE435-AA)	PCI Ethernet controller
DEMFA (demfa*n*)	DEC FDDI-controller 400	XMI to FDDI controller
DEFAA (defaa*n*)	DEC FDDI-controller/ Futurebus+	Futurebus+ FDDI controller
DEC PCI FDDI (pfi*n*)	DEFPZ-AA DEFPZ-UA DEFPA-AA DEFPA-DA DEFPA-UA	Optional FDDI card on KFTIA Optional FDDI card on KFTIA PCI FDDI controller (single attach.) PCI FDDI controller (dual attach.) PCI FDDI controller (UTP, single att.)
KFMSB (kfmsb*n*)	KFMSB-AA	XMI DSSI controller
KFESB (kfesb*n*)	KFESB-AA	EISA DSSI controller
DEFEA (defea*n*)	DEFEA	EISA FDDI controller
ISP1020 (isp*n*)		One SE SCSI and three FWD SCSI (using KFTIA)
KZMSA (kzmsa*n*)	KZMSA-AB	XMI SCSI controller

Table A-3 AlphaServer 8200/8400 Devices (Continued)

In Console Display as	Device	Description
KZPAA (kzpaa*n*)	KZPAA	PCI SCSI, single-ended
ISP1020 (isp*n*)	KZPBA-BB	PCI SCSI, FWD
KZPSA (kzpsa*n*)	KZPSA-BB	PCI SCSI, FWD
DAC960 (dac*n*)	KZPSC-AA	Mylex fast SCSI RAID controller (1 port)
DAC960 (dac*n*)	KZPSC-BA	Mylex fast SCSI RAID controller (3 ports)
PCI NVRAM (pci_nvram*n*)	DJ-ML300-BA	KFTIA NVRAM daughter card (4 Mbyte)
PCI NVRAM (pci_nvram*n*)	DJ-ML200-CA	PCI NVRAM option (8 Mbyte)
CIXCD (cixcd*n*)	CIXCD-AC	XMI to CI controller

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. You start LFU on AlphaServer 8400 and 8200 systems by booting the CD-ROM.

A typical update procedure is:

1. Boot the LFU CD-ROM.
2. Use the LFU **show** command to indicate modules whose firmware needs to be updated.
3. Use the LFU **list** command if you want to check the firmware version numbers on the CD-ROM.
4. Use the LFU **update** command to write the new firmware.
5. Exit.

B.1 Booting LFU from Local CD-ROM Drive

LFU is supplied on the Alpha Systems Firmware Update 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 CD drive. Boot LFU.

Example B-1 CD-ROM LFU Booting

```
P00>>> show device                          ❶
polling for units on cixcd0, slot 2, xmi0...
polling for units on kzmsa0, slot 4, xmi0...
dka500.5.0.4.0      DKA500          RRD44  1084
P00>>> boot dka500 -fl 0,A0          ❷
(boot dka500.5.0.4.0 -flags 0,a0)
block 0 of dka500.5.0.4.0 is a valid boot block
reading 1018 blocks from dka500.5.0.4.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7f400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
Bootfile:[ALPHA8400]AS8000_V01.exe      ❸

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

------------------------------------------------------------------
 Function   Description
------------------------------------------------------------------
 Display    Displays the system's configuration table.
 Exit       Done exit LFU (reset).
 List       Lists the device, revision, firmware name and update rev.
 Modify     Modifies port parameters and device attributes.
 Update     Replaces current firmware with loadable data image.
 Verify     Compares loadable and hardware images.
 ? or Help Scrolls the function table.
------------------------------------------------------------------

UPD>          ❹
```

❶ Use the **show device** command to find the name of the CD drive.

❷ Enter the **boot** command to boot from the RRD44. The RRD44 has a device name of dka500.

❸ Enter the file name of the firmware update utility, [ALPHA8200]AS8000_Vnn.exe for an AlphaServer 8200 or [ALPHA8400]AS8000_Vnn.exe for an AlphaServer 8400. Vnn is the LFU version number (such as V01). The latest Firmware Release Notes, released with this particular version of the CD-ROM, give the correct file name.

❹ LFU starts, displays a summary of its commands, and issues its prompt (UPD>).

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

```
UDP> list

Device          Current Revision    Filename        Update Revision

cixcd0               3              cixcd_fw              3

demna0               8.3            demna_fw              8.3

kcm440               3.0            kcm44_fw              3.1

kncc-ab0             2.0-1545       kn7cc_fw              2.0-1543

kzpsa0           A02_1              kzpsa_fw            A02_1

UPD>
```

The **list** command shows the current revision of firmware, the update revision of firmware, and the file that is recommended for updating the firmware, for each device.

B.3 Update

The update command writes new firmware from the CD-ROM to the module. Then LFU automatically verifies the update.

Example B-3 Update Command

```
UPD>  update kn7c*                          ❶

Confirm update on:                          ❷
kn7cc-ab0
[Y/(N)]y
WARNING: updates may take several minutes to complete for each device.
                         DO NOT ABORT!
kn7cc-ab0   update rev 2.0-1543 is less than current rev 2.0-1545.
Continue [Y/(N)] y

kn7cc-ab0   Updating to 2.0-1543...  Verifying 2.0-1543... PASSED.❸

UPD> update                                 ❹

Confirm update on:                          ❺
cixcd0
demna0
unknown2
[Y/(N)]

UPD> u cixcd0 -path demna_fw                ❻

Confirm update on:
cixcd0
[Y/(N)]y
WARNING: updates may take several minutes to complete for each device.

                         DO NOT ABORT!
cixcd0            firmware filename 'demna_fw' is bad

UDP>
```

❶ This example requests a firmware update for the kn7cc-ab0 module. A wildcard can be used to specify two or more different devices (for example, k*, for kcm440 and kzpsa0 modules); a list cannot be specified.

❷ LFU requires you to confirm each update. The default is no. This example shows an update with an older revision which requires a second confirmation.

❸ Status message reports update and verification progress.

❹ This is a second example. When you do not specify a device name, LFU tries to update all devices.

❺ LFU lists all the devices to update and asks for confirmation.

❻ In this second 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.

CAUTION: Never abort an update operation; you will corrupt the firmware on the module.

B.4 Exit

The exit command terminates the LFU program, causes system initialization and self-test, and returns to the system console prompt.

Example B-4 Exit Command

```
UPD> exit                    ❶
Initializing...

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 XV1.072,SROM V1.0,Feb 1 1995 03:28:33
P08>>>                       ❷
```

❶ At the UPD> prompt, **exit** causes the system to be initialized.

❷ The console prompt appears.

Example B-4 Exit Command (Continued)

```
UPD> update kzpsa0

Confirm update on:
kzpsa0
[Y/(N)]y
WARNING: updates may take several minutes to complete for each device.
                        DO NOT ABORT!

kzpsa0             Updating to A02_1...  FAILED.

UPD> exit

Errors occurred during update with the following devices:    ❸
kzpsa0

Do you want to exit?                                         ❹
Continue [Y/(N)]y                                            ❺
Initializing...

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

P08>>>
```

❸ Errors occurred during an update.

❹ Because of 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> disp      ❶
      Name             Type     Rev    Mnemonic
TLSB
4++  KN7CC-AB          8014     0000   kn7cc-ab0
5+   MS7CC             5000     0000   ms7cc1
8+   KFTIA             2020     0000   kftia0

C0 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

C1 PCI connected to kftia0
6+   DECchip 21040-AA 21011     0023   tulip2
A+   KZPSA            81011     0000   kzpsa0

UPD> verify kzpsa0    ❷
kzpsa0    Verifying A02_1...  PASSED.
UPD>
```

❶ **Display** shows the system physical configuration. **Display** is equivalent to issuing the console command **show configuration**. Because it shows the TLSB 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. In the case of corrupted firmware, the update command allows you to specify the correct device type so that new firmware can be written to the module.

Example B-6 Updating an "Unknown" Device

```
UPD> disp      ❶
     Name              Type     Rev    Mnemonic
TLSB
2++  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+    ?????            810     A4A6   unknown0   ❷
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+   ?????             810     A4A6   unknown1   ❷
3+   DEMNA             C03     0803   demna0
4+   CIXCD             C2F     0311   cixcd0
8+   ?????             810     A4A6   unknown2   ❷
A+   KCM44             C22     5E50   kcm440

C7 PCI connected to kftha0
6+   DECchip 21040-AA 21011     0023   tulip2
A+   KZPSA            81011     0000   kzpsa0

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!

unknown2  Updating to 2...  PASSED.  ❹
```

❶ Issue the **display** command. The display indicates several unknown devices. LFU is unable to recognize the device type. Because it shows the slot for each module, **display** can help you identify the location of a device.

❷ **Display** shows that the unknown devices are located in node 1 of the internal PCI bus and slots 2 and 8 of the XMI bus.

❸ Issue the command **update unknown***.

❹ Status message indicates that the update succeeded.

Continued on next page

Example B-6 Updating an "Unknown" Device (continued)

```
UPD> exit                    ❺
Initializing...

F  E  D  C  B  A  9  8  7  6  5  4  3  2  1  0  NODE #
[self-test display appears]
                     . A0  .  .  .  .  .  .  .  ILV
                     .128  .  .  .  .  .  .  .  128MB
AlphaServer 8200 Console V1.0,SROM V1.0,Apr 2 1995 03:28:33
P00>>> sho config
      Name            Type      Rev    Mnemonic
TLSB
2++  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+    ?????             810     A4A6   unknown0    ❻
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+   ?????             810     A4A6   unknown1    ❻
3+   DEMNA             C03     0803   demna0
4+   CIXCD             C2F     0311   cixcd0
8+   CIXCD             C2F     0211   cixcd1      ❼
A+   KCM44             C22     5E50   kcm440

C7 PCI connected to kftha0
6+   DECchip 21040-AA 21011     0023   tulip2
A+   KZPSA            81011     0000   kzpsa0
UPD>
```

❺ To make the device known, initialize the system by exiting LFU.

❻ The modules in node 2 of the internal PCI and slot 2 of the XMI are still unknown. The console does not recognize these devices.

❼ Initialization has made the device known to the system. The previously unknown device is now assigned device mnemonic cixcd1 by the system.

B.7 How to Modify Device Attributes

The following command can change parameters stored in EEPROM on the following devices: KZMSA, DEC LANcontroller 400 (DEMNA), KCM44, KFMSB, and KZPSA. The attributes are specific to each device.

Example B-7 Modify Command

```
UPD> modify kzmsa0
kzmsa0                                                    ❶
    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: 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 ❺

UPD>
```

❸ LFU prompts for parameters to modify. This example modifies one parameter on the KZMSA0; logging of ROM-based diagnostics is now enabled.

❹ LFU displays the list of parameters with modifications.

❺ If these modified values are acceptable, type **n**.

Appendix C

Configuration Utilities

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 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 7.4.17.

For more information:
StorageWorks RAID Array 200 Subsystems Controller Installation and Standalone Configuration Utility User's Guide
KZPSA Adapter User's Guide

C.1 Configuring a RAID Storage Array

Once you have selected the **run** command and the **rcu** option, the utility displays the main menu:

```
          +--------------------------------------+
          |             Main Menu                |
          +--------------------------------------+
          |[01. View/Update Configuration]       |
          | 02. Automatic Configuration          |
          | 03. New Configuration                |
          | 04. Initialize Logical Drive         |
          | 05. Parity Check                     |
          | 06. Rebuild                          |
          | 07. Tools                            |
          | 08. Select SWXCR                     |
          | 09. Controller Setup                 |
          | 10. Diagnostics                      |
          +--------------------------------------+
+-----------------------------------------------------+
| Choose this option to view, modify, and view physical |
| drive groups, logical drives, and hot spare drive     |
| assignments.                                          |
+-----------------------------------------------------+
Use Cursor keys for selection, press <ENTER> to select,
<ESC> to Quit
```

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 to 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 5 logical RAID drive: Usable Space = Total Space * ((N-1)/N) where N is the number of drives in the drive 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 *Ar-*

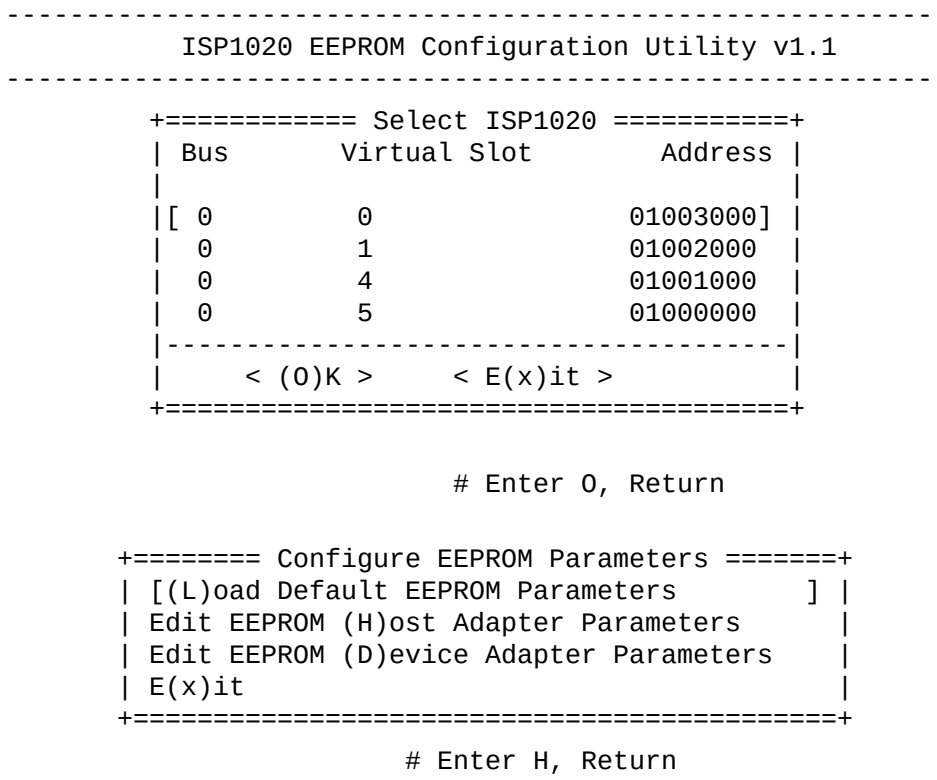
range Group option. Otherwise, press the Esc key and the cursor returns to the *Group Definition* menu.

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 Configuration Utility

When the EEROMCFG configuration utility is selected, after the "transferring control" message, the following screen is displayed:



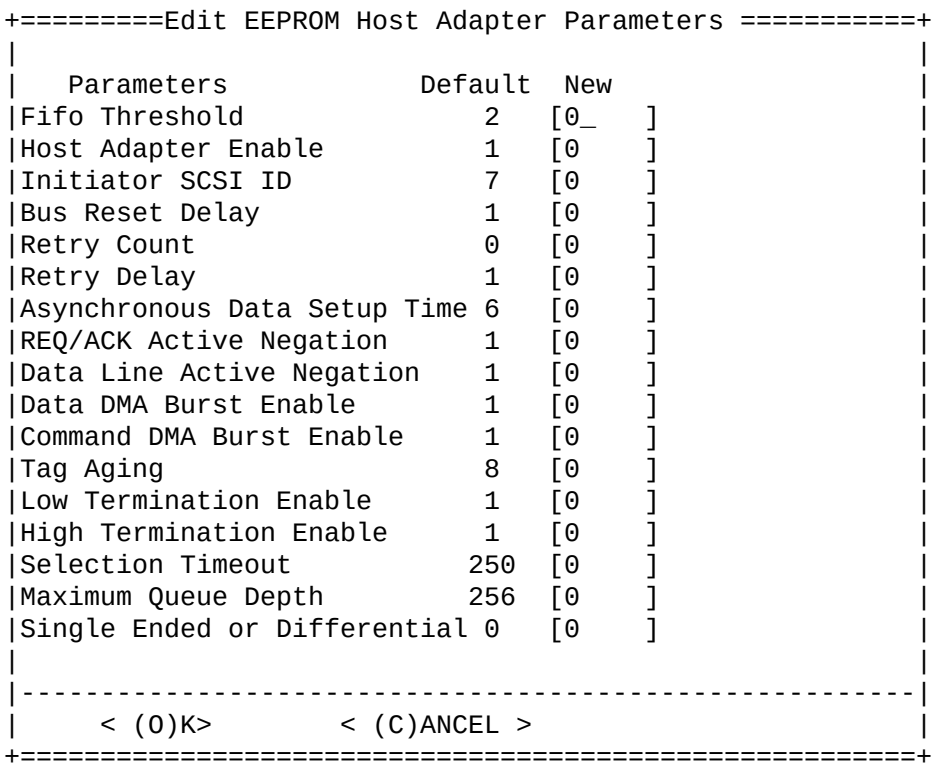


Table C-1 describes the ISP1020 host adapter parameters.

Table C-3 Host Adapter Parameters

Parameter	Default	Description	Reason to Change
FIFO Threshold	2	Sets the FIFO threshold point at which burst transfers are requested on the ISP1020 host adapter.	Optimize system performance.*
Host Adapter Enable	1	Determines whether the BIOS recognizes the ISP1020 host adapter.	Disable the host adapter without physically removing it from the system.
Initiator SCSI ID	7	Sets the SCSI ID for the ISP1020.	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.*
Retry Count	0	Determines the number of times the firmware attempts to retry a selection timeout or a busy status.	Optimize system performance.*
Retry Delay	1	Sets the time (in 100 microsecond increments) that the firmware waits before reattempting an operation.	Optimize system performance.*
Asynchronous Data Setup Time	6	Sets the number of clock periods the ISP1020 host adapter card waits after driving the SCSI data signals before asserting a SCSI bus acknowledge signal.	Optimize system performance.*

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 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 enabling active negation, the ISP1020 host adapter is less sensitive to an imperfect SCSI bus.	
Data DMA Burst 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.*
Command DMA Burst 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.*
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.*

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 microseconds).	Optimize system performance.*
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.*
Single Ended or Differential	0		Not applicable for the ISP1020.

* 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.*

Alpha primary boot program

The Alpha primary boot program (APB.EXE) that boots OpenVMS Alpha. 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), UNIX 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 containing the LFU utility used in upgrading firmware and copies of the firmware for all modules in the system, including the console and diagnostic firmware.

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.

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
Used by the system operator at the 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 manu-

facturing 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 a console terminal operator to communicate with nodes on the TLSB bus and I/O bus adapters and devices.

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 processor is the CPU module used in the 8200/8400 systems.

DECchip 21164

The Alpha processor chip.

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 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 **show configuration, show device**, and **show network** commands.

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 array 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 adapter
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*.

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 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 KN7CC processor 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 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
The interconnect between the KFTHA module on the TLSB bus and the interface module on another bus, such as the DWLMA module on the XMI bus. The KFTIA module contains one hose, so that a single XMI, Futurebus+, or PCI card cage may be added to the 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 XMI (8400 only), Futurebus+ (8400 only), or PCI card cage. A system can support up to three KFTIAs with the first one in node 8 of the TLSB. Node 8 is dedicated as both the highest and lowest arbitration level; the KFTIA usually arbitrates at the highest priority.

KN7CC CPU module
The CPU module that uses the DECchip 21164 with a superscalar superpipelined design. Each CPU chip has two 8-Kbyte caches and a 4-Mbyte backup cache implemented in RAMs. The KN7CC 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 differential SCSI controller.

ISE (integrated storage element)
All DSSI storage devices, such as RF disks and TF tapes, are ISEs.

LDC (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 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, or 512 Mbytes and 1 Gbyte and 2 Gbytes of memory. The AlphaServer 8200 system supports up to 6 Gbytes of memory, and the AlphaServer 8400 up to 14 Gbytes of memory.

Memory interleaving
Method to optimize memory access time; the 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 Alpha operating system
The OpenVMS operating system is a general-purpose multi-user operating system that supports industry standards and runs on Digital Equipment Corporation's 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, disks, PCI, and batteries.

Power regulators
The AlphaServer 8400 system can have the H7264 single-phase power regulator or the H7263 three-phase power regulator. The H7263 is required to support battery backup or 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).

SROM (serial ROM) code
The initialization code that bootstraps the main console/diagnostic firmware.

StorageWorks

Digital's family of modular 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.

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

P

R

S

T

U

V

W

X